Constructing Indoor Region-Based Radio Map Without Location Labels

Zheng Xing, *Graduate Student Member, IEEE* and Junting Chen, *Member, IEEE*

***Abstract*—Radio map construction requires a large amount of radio measurement data with location labels, which imposes a high deployment cost. This paper develops a region-based radio map from received signal strength (RSS) measurements without location labels. The construction is based on a set of blindly collected RSS measurement data from a device that visits each region in an indoor area exactly once, where the footprints and timestamps are not recorded. The main challenge is to cluster the RSS data and match clusters with the physical regions. Classical clustering algorithms fail to work as the RSS data naturally appears as non-clustered due to multipaths and noise. In this paper, a signal subspace model with a sequential prior is constructed for the RSS data, and an integrated segmentation and clustering algorithm is developed, which is shown to find the globally optimal solution in a special case. Furthermore, the clustered data is matched with the physical regions using a graph-based approach. Based on real measurements from an office space, the proposed scheme reduces the region localization error by roughly 50% compared to a weighted centroid localization (WCL) baseline, and it even outperforms some supervised localization schemes, including k-nearest neighbor (KNN), support vector machine (SVM), and deep neural network (DNN), which require labeled data for training.**

***Index Terms*—Localization, blind calibration, radio map, subspace clustering, segmentation.**

I. Introduction

LOCATION-BASED services have gained significant attention in the industry and research community due to the proliferation of mobile devices [1], [2]. While several advanced triangulation-based approaches, such as time-of-arrival (TOA) [3], time difference of arrival (TDOA) [4], or angle of arrival (AoA) [5], can achieve sub-meter level localization performance under line-of-sight (LOS) conditions, they require specialized and complex hardware to enable. In many indoor applications, rough accuracy is acceptable, but hardware cost is a primary concern. For instance, monitoring the locations of numerous equipment in a factory necessitates a large number of low-cost, battery-powered devices, while, in this application, meter-level accuracy suffices, *e.g.*, it suffices to determine whether the equipment is in room A or B. Consequently, received signal strength (RSS)-based localization may be found as the most cost-effective solution for indoor localization in these scenarios, as it does not require complicated hardware or a sophisticated localization protocol.

Traditional RSS-based indoor localization algorithms can be roughly categorized into model-based, model-free, and data-driven approaches. Model-based approaches [6], [7] first estimate a path loss model to describe how the RSS varies with propagation distance, and then estimate the target location by measuring the propagation distance from sensors with known locations based on the measured RSS. However, these approaches require calibration for the path loss model and are also highly sensitive to signal blockage. Model-free approaches employ an empirical formula to estimate the target location. For example, weighted centroid localization (WCL) approaches [6], [8] estimate the target location as the weighted average of sensor locations using an empirical formula, where the RSS values can be used as weights. However, the choice of the empirical formula significantly affects the localization accuracy of WCL. Data-driven approaches mostly require an offline measurement campaign to collect *location-labeled* RSS measurements at numerous spots in the target area to build a *fingerprint* database [9], [10], [11]. The target is localized by comparing the RSS measurements with the fingerprints in the database. However, such fingerprinting approaches not only require extensive labor to collect a large amount of RSS measurements tagged with location labels, but also require significant calibration effort after the system is deployed, because the RSS signature may vary due to changes in the environment, such as of the change of furniture. An outdated fingerprint database may degrade the localization performance. Therefore, reducing the construction and calibration costs is a critical issue for RSS-based localization.

There are some works on reducing the construction and calibration effort required for fingerprint localization. For example,

Manuscript received 27 June 2023; revised 27 December 2023; accepted 5 February 2024. Date of publication 22 February 2024; date of current version 28 May 2024. This work was supported in part by the National Science Foundation of China (NSFC) under Grant 62171398, by the Basic Research Project HZQB-KCZYZ-2021067 of Hetao Shenzhen-HK S&T Cooperation Zone, by NSFC under Grant 62293482, by the Shenzhen Science and Technology Program under Grant JCYJ20210324134612033, Grant JCYJ20220530143804010, and under Grant KQTD20200909114730003, by Guangdong Research Projects 2019QN01X895, 2017ZT07X152, and 2019CX01X104, by the Shenzhen Outstanding Talents Training Fund 202002, by the Guangdong Provincial Key Laboratory of Future Networks of Intelligence under Grant 2022B1212010001, by the National Key R&D Program of China with Grant 2018YFB1800800, and by the Key Area R&D Program of Guangdong Province with Grant 2018B030338001. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Kunal Narayan Chaudhury. *(Corresponding author: Junting Chen.)*

The authors are with the School of Science and Engineering, and the Future Network of Intelligence Institute (FNii), The Chinese University of Hong Kong, Shenzhen 518172, China (e-mail: zhengxing@link.cuhk.edu.cn; juntingc@cuhk.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TSP.2024.3368769, provided by the authors.

Digital Object Identifier 10.1109/TSP.2024.3368769

the work [12] employed generative adversarial networks (GAN) to augment the fingerprint training dataset. Moreover, the work [13] proposed a meta-learning approach to train the fingerprint with a few labeled samples. In addition, some works utilize interpolation, such as Kriging spatial interpolation [14], to recover a significant amount of unlabeled data based on a small amount of labeled data. Furthermore, the work [15] proposed a spatial channel state information (CSI) process (channel chart) based on CSI data without location labels, and calibrated the geographic coordinates of the channel chart based on a small amount of location-labeled CSI data. These approaches still require a certain amount of location-labeled RSS measurement data, resulting in non-negligible construction and calibration cost. Finally, there are some works on constructing fingerprint database using unsupervised approaches. For example, the work [16] recovered location labels by estimating the RSS measurement trajectory using a hidden Markov model (HMM)-based method based on a number of revisiting trajectories. Other attempts [17], [18], [19] required mobility information obtained from additional sensors, including inertial sensors, accelerometers, gyroscopes, and magnetometers, etc.

This paper proposes a *region-based radio map* for coarse indoor localization, where the radio map is constructed via *unsupervised* learning from RSS measurements *without* location labels. For coarse localization, the indoor area is divided into several regions of interest, and the target is localized to one of these regions. It is important to note that such a localization problem arises in various application scenarios, such as tracking equipment in a factory and monitoring visitors in restricted areas, where having a coarse accuracy is sufficient, but hardware cost and calibration cost are the primary concerns. The region-based radio map consists of RSS features associated with each region. Therefore, the fundamental question is how to construct the radio map using unlabeled RSS data to reduce the calibration cost.

To tackle this problem, we construct and explore some characteristics for the RSS data. First, we assume that the data is mostly collected sequentially from each region. This corresponds to a scenario where a mobile device visits each region once, while the sensor network collects the RSS of the signal emitted from the mobile. It is important to note that the trajectory of the mobile and the timing the mobile entering or leaving the region are not necessarily known by the network. Therefore, such an assumption induces almost no calibration cost. Second, we assume a subspace model for the RSS data, where the RSS vector lies in a low-dimensional affine subspace that varies across different regions.

With these two assumptions, the construction of a region-based radio map can be formulated as a clustering problem using sequential data. However, classical subspace clustering algorithms [20], [21], [22] were not optimized for RSS sequential data. Some works in a broader domain investigated sequential data clustering for video segmentation applications. For example, the work [23] partitioned the entire sequence into parts and grouped segments together. The works [24], [25], [26] learned the similarity between samples with sequential priors and adopted graph-based clustering methods. Furthermore, the work [27] leveraged knowledge from relevant labeled source sequential data to enhance the similarity graph of unlabeled target sequential data. However, these methods cannot be extended to clustering RSS data. This is because the RSS data naturally appears as non-clustered due to multipaths and noise, and even adjacent RSS data collected in the same room can be divided into two clusters, despite the usage of the sequential prior. Moreover, it is challenging to associate clusters with physical regions using the irregular clustering results generated by these methods. Thus, we need to address the following two major challenges:

- **How to cluster the unlabeled RSS measurement data?** In practice, it is observed that the measured RSS fluctuates significantly even within a small area. Therefore, clustering them into groups using classical clustering approaches poses a challenge.
- **How to match the clustered, yet unlabeled data to the physical regions?** Since the sensor network cannot observe any location labels, extracting location information remains a challenge, even if the RSS data is perfectly clustered.

In this paper, we formulate a maximum-likelihood estimation problem with a sequential prior to cluster the RSS data. Consequently, the clustering problem is transformed into a sequence segmentation problem. Our preliminary work [28] attempted to solve the segmentation problem using a gradient-type algorithm, but the solution is prone to getting trapped at a poor local optimum. Here, we introduce a merge-and-split algorithm that is to converge to a globally optimal solution for a special case. Global convergence in a general case is also observed in our numerical experiments. To match the clusters with the physical regions, we construct a graph model for a set of possible routes that may form the RSS sequential data; such a model leads to a Viterbi algorithm for the region matching, which can achieve a matching error of less than 1%. To compare with some existing unsupervised learning approaches, such as HMM-based approaches [16], the proposed method does not require a parametric propagation model for the RSS, which is usually needed in a conventional HMM, and hence, the proposed work is algorithmically more stable as compared to HMM which needs to alternatively learn the model parameters and recovering the physical trajectories.

To summarize, we make the following contributions:

- We develop an unsupervised learning framework to construct a region-based radio map without location labels. The approach is based on solving a subspace clustering problem with sequential prior.
- We transform the clustering problem to a segmentation problem which is solved by a novel merge-and-split algorithm. We establish optimality guarantees for the algorithm under a special case.
- We conduct numerical experiments using real measurements from an office space and a larger area. It is found that the proposed unsupervised scheme even achieves a better localization performance than several supervised learning schemes which use location labels during the training, including k-nearest neighbor (KNN), support vector machine (SVM), and deep neural network (DNN).

The remaining part of the paper is organized as follows. Section II introduces a signal subspace model for the region-based radio map, a sequential data collection model, and a

probability model with a sequential prior. Section III develops the subspace feature solution, and the solution for matching clusters to physical regions. Section IV focuses on the development of the clustering algorithm and the potential optimality guarantees. Experimental results are reported in Section V and the paper is concluded in Section VI.

II. System Model

A. Signal Subspace Model for the Region-Based Radio Map

Suppose that there are D sensors with locations $\mathbf{z}_j \in \mathbb{R}^2$, $j = 1, 2, \ldots, D$, deployed in an indoor area. The sensors, such as WiFi sensors, are capable of measuring the RSS of the signal emitted by a wireless device, forming an RSS measurement vector $\mathbf{x} \in \mathbb{R}^D$, although they may not be able to decode the message of the device. Consider partitioning the indoor area into K non-overlapping regions. It is assumed that signals emitted from the same region to share common feature due to the proximity of the transmission location and the similarity of the propagation environment. In practice, a room or a semi-closed space separated by large furniture or walls can be naturally considered as a region, where the intuition is that walls and furniture may shape a common feature for signals emitted from a neighborhood surrounded by these objects. Given the region partition, this paper focuses on extracting the large-scale *feature* of each of the regions and building a region-based radio map from the RSS measurements $\{\mathbf{x}\}$ without location labels.

Assume that the RSS measurements $\mathbf{x}_i$ in decibel scale taken in region k satisfy

$$\mathbf{x}_i = \mathbf{U}_k \boldsymbol{\theta}_i + \boldsymbol{\mu}_k + \boldsymbol{\epsilon}_i, \quad \forall i \in \mathcal{C}_k \tag{1}$$

where $\mathbf{U}_k \in \mathbb{R}^{D \times d_k}$ is a semi-unitary matrix with $\mathbf{U}_k^{\mathrm{T}} \mathbf{U}_k = \mathbf{I}$, $\boldsymbol{\theta}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_k)$ is an independent variable that models the uncertainty due to the actual measurement location when taking the measurement sample $\mathbf{x}_i$ in region k, $\boldsymbol{\Sigma}_k$ is assumed as a full-rank diagonal matrix with non-negative diagonal elements, $\boldsymbol{\mu}_k \in \mathbb{R}^D$ captures the offset of the signal subspace, $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, s_k^2 \mathbf{I}_{D \times D})$ models the independent measurement noise, and $\mathcal{C}_k$ is the index set of the measurements $\mathbf{x}_i$ taken within the kth region. As such, the parameters $\{\mathbf{U}_k, \boldsymbol{\mu}_k\}$ specify an affine subspace with dimension d_k for the noisy measurement $\mathbf{x}_i$, and they are the *feature* to be extracted from the measurement data $\{\mathbf{x}_i\}$. Thus, region-based radio map is a database that maps the kth physical region to the signal subspace feature $\{\mathbf{U}_k, \boldsymbol{\mu}_k\}$.

Remark 1 (Interpretation of the Subspace Model): As the user has 2 spatial degrees of freedom to move around in the region, the RSS vector $\mathbf{x}$ may be modeled as a point moving on a two-dimensional hyper-surface $\mathcal{S}$ embedded in $\mathbb{R}^D$. Therefore, for a sufficiently small area, an affine subspace with dimension $d_k = 2$ or 3 can locally be a good approximation of $\mathcal{S}$.

B. Sequential Data Collection and the Graph Model

When the measurement location label sets are *not* available, it is very difficult to obtain the subspace feature $\{\mathbf{U}_k, \boldsymbol{\mu}_k\}$. While conventional subspace clustering algorithms, such as the expectation-maximization (EM) approach, are designed for recovering both the location label sets and the subspace feature $\{\mathbf{U}_k, \boldsymbol{\mu}_k\}$, they may not work for the large noise case, which is a typical scenario here as the RSS data has a large fluctuation due to the multipath effect. To tackle this challenge, we consider a type of measurement that provide some implicit structural information without substantially increasing the effort on data collection.

We assume that the sequence of measurements $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ are taken along an arbitrary route that visits all the K regions without repetition.[1] Recall that $\mathcal{C}_k$ is the collection of the measurements collected from the kth region along the route, and therefore, for any $i \in \mathcal{C}_k$ and $j \in \mathcal{C}_{k+1}$, we must have $i < j$. Note that the exact route, the locations of the measurements, the sojourn time that the mobile device spends in each region, and the association between the sequential measurement set $\mathcal{C}_k$ and the kth physical region are *unknown* to the system.

Nevertheless, we can model the eligibility of a specific route. A route $\boldsymbol{\pi}$ is modeled as a permutation sequence with the first K natural numbers, where the kth element $\pi(k)$ refers to the location label of the kth region along the route. Define a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where each node in $\mathcal{V} = \{1, 2, \ldots, K\}$ represents one of the K regions, and each edge $(k, j) \in \mathcal{E}$ represents that it is possible to directly travel from region k to region j without entering the other region. Therefore, a route $\boldsymbol{\pi}$ is eligible only if there is an edge between adjacent nodes along the route, *i.e.*, $(\pi(j), \pi(j+1)) \in \mathcal{E}$, for $\forall 1 \le j \le K-1$.

C. Probability Model With a Sequential Prior

From the signal subspace model (1), the conditional distribution of $\mathbf{x}_i$, given that it belongs to the kth region, is given by

$$p_k(\mathbf{x}; \boldsymbol{\Theta}) = \frac{1}{(2\pi)^{D/2} |\mathbf{C}_k|^{1/2}} \times \exp\left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^{\mathrm{T}} \mathbf{C}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \right) \tag{2}$$

where $\mathbf{C}_k = \mathbf{U}_k \boldsymbol{\Sigma}_k \mathbf{U}_k^{\mathrm{T}} + s_k^2 \mathbf{I}$ is the conditional covariance matrix for the kth cluster, and $\boldsymbol{\Theta} = \{\mathbf{U}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\mu}_k, s_k^2\}_{k=1}^K$ is a shorthand notation for the collection of parameters.

Let t_k be the last index of $\mathbf{x}_i$ before the device leaves the kth region and enters the $(k+1)$th region. Consequently, we have $t_0 = 0 < t_1 < t_2 < \cdots < t_{K-1} < t_K = N$, and for each $k = 1, 2, \ldots, K$, all elements $i \in \mathcal{C}_k$ satisfy $t_{k-1} < i \le t_k$. For a pair of parameters $a < b$, an indicator function is defined as

$$z_i(a, b) = \begin{cases} 1, & a < i \le b \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

As a result, the probability density function of measurement $\mathbf{x}_i$ can be given by

$$p(\mathbf{x}_i; \boldsymbol{\Theta}, \boldsymbol{t}) = \prod_{k=1}^{K} p_k(\mathbf{x}_i; \boldsymbol{\Theta})^{z_i(t_{k-1}, t_k)} \tag{4}$$

[1]Note that, our model and algorithm can accommodate some repetition, i.e., visiting a region multiple times, as long as the repetition is limited. This is accounted for by the noise term in the subspace model (1).

where $\boldsymbol{t} = (t_1, t_2, \ldots, t_{K-1})$ is a collection of the time indices of the segment boundaries. Note that, for each i, $\sum_k z_i(t_{k-1}, t_k) = 1$, and $z_i(t_{k-1}, t_k) = 1$ only under $t_{k-1} < i \leq t_k$. Therefore, given $i \in \mathcal{C}_k$, equation (4) reduces to $p(\mathbf{x}_i) = p_k(\mathbf{x}_i)$.

Consider a log-likelihood cost function $\log \prod_{i=1}^{N} p(\mathbf{x}_i; \boldsymbol{\Theta}, \boldsymbol{\tau})$ which can be equivalently written as

$$\mathcal{J}(\boldsymbol{\Theta}, \boldsymbol{\tau}) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} z_i(\tau_{k-1}, \tau_k) \log p_k(\mathbf{x}_i; \boldsymbol{\Theta}, \boldsymbol{\tau}) \tag{5}$$

where $\boldsymbol{\tau} = (\tau_1, \tau_2, \ldots, \tau_{K-1})$ denotes an estimator for $\boldsymbol{t}$. Throughout the paper, we implicitly define $\tau_0 = 0, \tau_K = N$ for mathematical convenience.

It follows that $z_i(\tau_{k-1}, \tau_k)$ represents a rectangle window that selects only the terms $\log p_k(\mathbf{x}_i; \boldsymbol{\Theta}, \boldsymbol{\tau})$ for $\tau_{k-1} < i \leq \tau_k$ and suppresses all the other terms. Thus, it acts as a *sequential prior* that selects a subset of $\{\mathbf{x}_i\}$ in a row for $\mathcal{C}_k$.

While we have made an assumption from model (1) that the measurements are statistically independent, in practice, the measurements $\mathbf{x}_i$ taken in the transient phase from one region to the other may substantially deviate from both subspaces $\{\mathbf{U}_{k-1}, \boldsymbol{\mu}_{k-1}\}$ and $\{\mathbf{U}_k, \boldsymbol{\mu}_k\}$, leading to large modeling noise $\boldsymbol{\epsilon}_i$. To down-weight the data possibly taken in the transient phase, we extend the rectangle window model $z_i(a, b)$ in (3) for the sequential prior to a smooth window

$$z_i(a, b) = \sigma_\beta(i - a) - \sigma_\beta(i - b) \tag{6}$$

where $\sigma_\beta(x)$ is a sigmoid function that maps $\mathbb{R}$ to $[0, 1]$ with a property that $\sigma_\beta(x) \to 0$ as $x \to -\infty$ and $\sigma_\beta(x) \to 1$ as $x \to +\infty$. The parameter β controls the slope of the transition from 0 to 1. A choice of a specific sigmoid function is $\sigma_\beta(x) = (1 + e^{-(x-1/2)/\beta})^{-1}$. For such a choice of $\sigma_\beta(x)$, the rectangle window (3) is a special case of (6) for $\beta \to 0$, where one can easily verify that, for all k, we have $z_i(t_{k-1}, t_k) \to 1$ for all $i \in \mathcal{C}_k$ and $z_i(t_{k-1}, t_k) \to 0$ for $i \notin \mathcal{C}_k$.

The subspace clustering problem with a sequential prior is formulated as follows

$$\underset{\boldsymbol{\Theta}, \boldsymbol{\tau}}{\text{maximize}} \quad \mathcal{J}(\boldsymbol{\Theta}, \boldsymbol{\tau}) \tag{7}$$

$$\text{subject to} \quad 0 < \tau_1 < \tau_2 < \ldots < \tau_{K-1} < N. \tag{8}$$

Note that an EM-type algorithm cannot solve (7) due to the sequential structure imposed by $z_i(\tau_{k-1}, \tau_k)$. Our prior work [28] investigated a gradient approach, but the iteration easily gets trapped at a bad local optimum even under the simplest form of the model at $d_k = 0$, as shown later in the experiment section. In the rest of the paper, we establish a new solution framework to solve (7) and derive the conditions for achieving the optimality.

III. Subspace Feature Extraction and Region Matching

In this section, we focus on the solution $\boldsymbol{\Theta}$ when the partition variable $\boldsymbol{\tau}$ in (7) is fixed, and develop a maximum likelihood estimator $\hat{\boldsymbol{\Theta}}(\boldsymbol{\tau})$ as a function of $\boldsymbol{\tau}$ and the data. Then, we develop a method to map the subspace feature $\{\mathbf{U}_k, \boldsymbol{\mu}_k\}$ to the physical region.

A. Subspace Feature via Maximum-Likelihood Principal Component Analysis (PCA)

The solution $\boldsymbol{\Theta}$ for a given $\boldsymbol{\tau}$ can be derived as follows. First, from the conditional probability model (2), the maximizer of $\boldsymbol{\mu}_k$ to $\mathcal{J}$ in (7) can be obtained by setting the derivative of $\mathcal{J}$ with respect to $\boldsymbol{\mu}_k$ to zero, leading to the unique solution

$$\hat{\boldsymbol{\mu}}_k = \frac{1}{\sum_{i=1}^{N} z_i(\tau_{k-1}, \tau_k)} \sum_{i=1}^{N} z_i(\tau_{k-1}, \tau_k) \mathbf{x}_i. \tag{9}$$

Then, denote $\mathbf{W}_k = \mathbf{U}_k \boldsymbol{\Sigma}_k^{1/2}$ for notational convenience. The critical point of $\mathcal{J}$ with respect to the variable $\mathbf{W}_k$ is obtained by setting the derivative

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}_k} = \mathbf{C}_k^{-1} \mathbf{S}_k \mathbf{C}_k^{-1} \mathbf{W}_k - \mathbf{C}_k^{-1} \mathbf{W}_k \tag{10}$$

to zero, leading to the equation

$$\mathbf{S}_k \mathbf{C}_k^{-1} \mathbf{W}_k = \mathbf{W}_k \tag{11}$$

where

$$\mathbf{S}_k = \frac{1}{\sum_{i=1}^{N} z_i(\tau_{k-1}, \tau_k)} \sum_{i=1}^{N} z_i(\tau_{k-1}, \tau_k)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^{\mathrm{T}} \tag{12}$$

is the sample covariance of the data $\{\mathbf{x}_i\}$ that are weighted by the kth sequential prior $z_i(\tau_{k-1}, \tau_k)$ for the kth subspace.

Recall that $\mathbf{C}_k = \mathbf{W}_k \mathbf{W}_k^{\mathrm{T}} + s_k^2 \mathbf{I}$, and $\mathbf{W}_k^{\mathrm{T}} \mathbf{W}_k = \boldsymbol{\Sigma}_k^{1/2} \mathbf{U}_k^{\mathrm{T}} \mathbf{U}_k \boldsymbol{\Sigma}_k^{1/2} = \boldsymbol{\Sigma}_k$, since $\mathbf{U}_k$ is semi-orthogonal and $\boldsymbol{\Sigma}_k$ is diagonal. We have the identity $\mathbf{C}_k^{-1} \mathbf{W}_k = \mathbf{W}_k (\boldsymbol{\Sigma}_k + s_k^2 \mathbf{I})^{-1}$, which can be easily verified by the relation $\mathbf{W}_k = \mathbf{U}_k \boldsymbol{\Sigma}_k^{1/2}$. Hence,

$$\mathbf{S}_k \mathbf{C}_k^{-1} \mathbf{W}_k = \mathbf{S}_k \mathbf{W}_k (\boldsymbol{\Sigma}_k + s_k^2 \mathbf{I})^{-1}. \tag{13}$$

Using (13) and $\mathbf{W}_k = \mathbf{U}_k \boldsymbol{\Sigma}_k^{1/2}$, equation (11) becomes $\mathbf{S}_k \mathbf{U}_k \boldsymbol{\Sigma}_k^{1/2} (\boldsymbol{\Sigma}_k + s_k^2 \mathbf{I})^{-1} = \mathbf{U}_k \boldsymbol{\Sigma}_k^{1/2}$, which can be simplified to

$$\mathbf{S}_k \mathbf{U}_k = \mathbf{U}_k (\boldsymbol{\Sigma}_k + s_k^2 \mathbf{I}). \tag{14}$$

It follows that (14) is an eigenvalue problem, which can be solved by constructing $\mathbf{U}_k$ as a collection of d_k eigenvectors of $\mathbf{S}_k$. In addition, let $\lambda_{k,j}^2$ be the corresponding eigenvalue of the jth selected eigenvector of$\mathbf{S}_k$ for the construction of $\mathbf{U}_k$. Then, the jth diagonal element of the diagonal matrix $\boldsymbol{\Sigma}_k$ can be set as $\sigma_{k,j}^2 = \lambda_{k,j}^2 - s_k^2$. It will become clear as explained below that the best choice for constructing $\mathbf{U}_k$ is to select the eigenvectors of $\mathbf{S}_k$ corresponding to the d_k-largest eigenvalues, and the best estimate of s_k^2 is given by

$$s_k^2 = \frac{\sum_{j=d_k+1}^{D} \lambda_{k,j}^2}{D - d_k}. \tag{15}$$

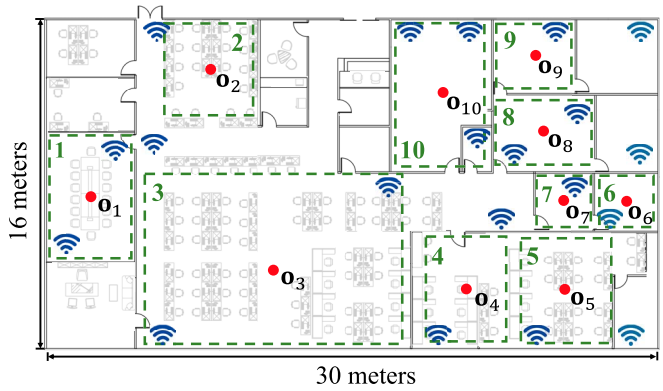

Fig. 1. A 30×16 m^2 indoor area deployed with 21 sensors (blue icons) partitioned into 10 non-overlapping regions (green dashed rectangles), where the red circles refers to the reference region center.

To see this, substituting $\hat{\boldsymbol{\mu}}_k$ from (9) and the solution $\mathbf{U}_k$ and $\boldsymbol{\Sigma}_k$ obtained from (14) to the log-likelihood function $\mathcal{J}(\boldsymbol{\Theta}, \boldsymbol{\tau})$ in (5), we obtain

$$\mathcal{J} = -\frac{1}{2N}\sum_{k=1}^{K}\sum_{i=1}^{N} z_i(\tau_{k-1},\tau_k)\Bigg\{D\log(2\pi) + \log\prod_{j=1}^{d_k}\lambda_{k,j}^2 + \log\left(s_k^{2(D-d_k)}\right) + \frac{1}{s_k^2}\sum_{j=d_k+1}^{D}\lambda_{k,j}^2 + d_k\Bigg\}. \tag{16}$$

It has been shown in [29] that the maximizer to (16) is obtained as the solution (15) with $\lambda_{k,j}$, $j = 1, 2, \ldots, d_k$, being chosen as the d_k-largest eigenvalues of $\mathbf{S}_k$. As such, we have obtained the solution $\boldsymbol{\Theta}$ from (9), (14)–(15).

B. Matching Clusters to Physical Regions

Denote $\mathcal{D}_k \subseteq \mathbb{R}^2$ as the area of the kth physical region. Our goal here is to map the subspace feature $\{\mathbf{U}_k, \boldsymbol{\mu}_k\}$ or the segment (τ_{k-1}, τ_k) for the kth cluster of $\{\mathbf{x}_i\}$ to the physical region $\mathcal{D}_k$. Mathematically, recall that the route $\boldsymbol{\pi} = (\pi(1), \pi(2), \ldots, \pi(K))$ is modeled as a permutation sequence of the first K natural numbers. As a result, $\boldsymbol{\pi}$ is a function that matches the kth subspace $\{\mathbf{U}_k, \boldsymbol{\mu}_k\}$ to the physical region $\mathcal{D}_{\pi(k)}$, for $1 \le k \le K$.

Define $\mathbf{o}_k$ as the reference center location of the kth region $\mathcal{D}_k$ as indicated by the red dots in Fig. 1. Thus, one essential idea is to link the clustered RSS measurements $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,D})$ with the location topology of the sensors. Specifically, we employ the WCL approach to compute a reference location $\hat{\mathbf{o}}_k$ for the measurements in the kth cluster $\mathcal{C}_k$:

$$\hat{\mathbf{o}}_k = \frac{1}{|\mathcal{C}_k|}\sum_{i\in\mathcal{C}_k}\frac{\sum_{j=1}^{D} w_{i,j}\mathbf{z}_j}{\sum_{j=1}^{D} w_{i,j}} \tag{17}$$

where $w_{i,j} = (10^{x_{i,j}/10})^{\alpha}$ is the weight on the location $\mathbf{z}_j$ of the jth sensor, $x_{i,j}$ is the RSS of the jth sensor in the ith measurement $\mathbf{x}_i$, and α is an empirical parameter typical chosen as [6], [8], [30], [31].

Then, we exploit the *consistency property* that adjacent features $\{\mathbf{U}_k, \boldsymbol{\mu}_k\}$ and $\{\mathbf{U}_{k+1}, \boldsymbol{\mu}_{k+1}\}$ should be mapped to physically adjacent regions $\mathcal{D}_{\pi(k)}$ and $\mathcal{D}_{\pi(k+1)}$, where $(\pi(k), \pi(k+1)) \in \mathcal{E}$. In Section II-B, we have modeled the adjacency of any two physical regions using the graph model $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ based on the layout of the target area. With such a graph model, a graph-constrained matching problem can be formulated as

$$\underset{\boldsymbol{\pi}}{\text{minimize}} \quad \sum_{k=1}^{K} c(\hat{\mathbf{o}}_{\pi(k)}, \mathbf{o}_k) \tag{18}$$

$$\text{subject to} \quad (\pi(j), \pi(j+1)) \in \mathcal{E},\ \forall j = 1, 2, \ldots, K-1 \tag{19}$$

where $c(\mathbf{p}_1, \mathbf{p}_2)$ is a cost function that quantifies the difference between the two locations $\mathbf{p}_1$ and $\mathbf{p}_2$, for example, $c(\mathbf{p}_1, \mathbf{p}_2) = \|\mathbf{p}_1 - \mathbf{p}_2\|_2$. The constraint is to ensure that an eligible route only travels along the edges of the graph $\mathcal{G}$ and the chosen region is non-repetitive.

Problem (18) can be solved by the dynamic programming-based path searching strategy. For example, the Viterbi algorithm can be applied to optimally solve (18) with a complexity of $\mathcal{O}(K^3N)$. To see this, any feasible route $\boldsymbol{\pi}$ consists of K moves. At each region, there are at most $K-1$ candidate regions for the next move, and therefore, there are at most $K(K-1)$ moves to evaluate in the Viterbi algorithm. To evaluate the utility of each move, one computes (18) with a complexity of $\mathcal{O}(KN)$. This leads to the overall complexity of $\mathcal{O}(K^3N)$ to solve (18) using the Viterbi algorithm.

IV. Clustering via Segmentation for Sequential Data

Based on the solution $\hat{\boldsymbol{\Theta}}(\boldsymbol{\tau})$ developed in Section III-A, it remains to maximize $\mathcal{J}(\hat{\boldsymbol{\Theta}}(\boldsymbol{\tau}), \boldsymbol{\tau})$ over $\boldsymbol{\tau}$ subject to (8), and the subspace clustering problem (7) becomes a segmentation problem. The main challenge is the existence of multiple local maxima of $\mathcal{J}(\hat{\boldsymbol{\Theta}}(\boldsymbol{\tau}), \boldsymbol{\tau})$.

In this section, we first establish the optimality for a special case of $d_k = 0$ and $s_k^2 = s^2$ for some $s > 0$, which corresponds to K-means clustering with a sequential prior. Based on this theoretical insight, we then develop a robust algorithm to solve for the general case of $d_k \ge 0$.

For the special case of $d_k = 0$ and $s_k^2 = s^2$, the subspace model (1) degenerates to $\mathbf{x}_i = \boldsymbol{\mu}_k + \boldsymbol{\epsilon}_i$, where $\boldsymbol{\mu}_k$ is the offset of the kth 0-dimensional subspace and $\mathbf{U}_k = 0$. The conditional probability (2) becomes $p_k(\mathbf{x}_i; \boldsymbol{\Theta}) = \frac{1}{(2\pi)^{D/2}s}\exp(-\frac{1}{2s^2}\|\mathbf{x}_i - \boldsymbol{\mu}_k\|_2^2)$. Then, problem (7) is equivalent to the following minimization problem

$$\underset{\boldsymbol{\Theta},\boldsymbol{\tau}}{\text{minimize}} \quad \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} z_i(\tau_{k-1},\tau_k)\|\mathbf{x}_i - \boldsymbol{\mu}_k\|_2^2$$
$$\text{subject to} \quad 0 < \tau_1 < \tau_2 < \ldots < \tau_{K-1} < N \tag{20}$$

where the variable $\boldsymbol{\Theta}$ degenerates to a matrix $\boldsymbol{\Theta} = [\boldsymbol{\mu}_1\ \boldsymbol{\mu}_2\ \cdots\ \boldsymbol{\mu}_K]$ that captures the centers of the clusters. Substituting the solution $\hat{\boldsymbol{\mu}}_k(\tau_{k-1}, \tau_k)$ in (9) to (20), the cost function (20) simplifies to

$$\tilde{f}(\boldsymbol{\tau}) \triangleq \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} z_i(\tau_{k-1},\tau_k)\left\|\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k(\tau_{k-1},\tau_k)\right\|_2^2$$

which is a function with respect to (w.r.t.) $\boldsymbol{\tau}$.

A. Asymptotic Property for Small β

For each $\boldsymbol{\tau}$, it is observed from the solution $\hat{\boldsymbol{\mu}}_k(\tau_{k-1},\tau_k)$ in (9) and the asymptotic property of the window function $z_i(\cdot)$ in (6) that, as $\beta \to 0$,

$$\hat{\boldsymbol{\mu}}_k(\tau_{k-1},\tau_k) \to \tilde{\boldsymbol{\mu}}(\tau_{k-1},\tau_k) \triangleq \frac{1}{\tau_k-\tau_{k-1}} \sum_{j=\tau_{k-1}+1}^{\tau_k} \mathbf{x}_j \quad (21)$$

uniformly for each τ_{k-1} and τ_k that satisfy $\tau_{k-1}+1 \le \tau_k$.

Based on $\tilde{\boldsymbol{\mu}}(\tau_{k-1},\tau_k)$ in (21), define

$$\begin{aligned} f_k(\tau_k;\boldsymbol{\tau}_{-k}) \triangleq \frac{1}{N} \sum_{i=\tau_{k-1}+1}^{\tau_{k+1}} \Big[\Big(1-\sigma_\beta(i-\tau_k)\Big) \\ \times \big\|\mathbf{x}_i - \tilde{\boldsymbol{\mu}}(\tau_{k-1},\tau_k)\big\|_2^2 \\ + \sigma_\beta(i-\tau_k)\big\|\mathbf{x}_i - \tilde{\boldsymbol{\mu}}(\tau_k,\tau_{k+1})\big\|_2^2 \Big] \end{aligned} \quad (22)$$

for $k=1,2,\ldots,K-1$, where $\boldsymbol{\tau}_{-k} \triangleq (\tau_{k-1},\tau_{k+1})$. In addition, define

$$f_0(\tau_0) = \frac{1}{N}\sum_{i=1}^{\tau_1} \sigma_\beta(i-\tau_0)\big\|\mathbf{x}_i - \tilde{\boldsymbol{\mu}}(\tau_0,\tau_1)\big\|_2^2 \quad (23)$$

$$\begin{aligned} f_K(\tau_K) = \frac{1}{N} \sum_{i=\tau_{K-1}+1}^{N} \big(1-\sigma_\beta(i-\tau_K)\big) \\ \times \big\|\mathbf{x}_i - \tilde{\boldsymbol{\mu}}(\tau_{K-1},\tau_K)\big\|_2^2 \end{aligned} \quad (24)$$

for mathematical convenience. The cost function $\tilde{f}(\boldsymbol{\tau})$ can be asymptotically approximated as $\frac{1}{2}\sum_{k=0}^{K} f_k(\tau_k;\boldsymbol{\tau}_{-k})$ as formally stated in the following result.

Proposition 1 (Asymptotic Hardening): As $\beta \to 0$, we have

$$\tilde{f}(\boldsymbol{\tau}) \to \frac{1}{2}\sum_{k=0}^{K} f_k(\tau_k;\boldsymbol{\tau}_{-k})$$

uniformly for every sequence $\boldsymbol{\tau}$ satisfying the constraint (8).

Proof: See Appendix A. □

Proposition 1 decomposes $\tilde{f}(\boldsymbol{\tau})$ into sub-functions $f_k(\tau_k;\boldsymbol{\tau}_{-k})$ that only depend on a subset of data $\mathbf{x}_i$ from $i=\tau_{k-1}+1$ to $i=\tau_{k+1}$.

B. Asymptotic Consistency for Large N

While the function $f_k(\cdot)$ in (22) has been much simplified from $\tilde{f}(\boldsymbol{\tau})$, it is a stochastic function affected by the measurement noise in (1). Consider a scenario of large N for asymptotically dense measurement in each region k. Specifically, the total number of the sequential measurements grows in such a way that the segment boundary indices $t_1,t_2,\ldots,t_{K-1}$ grow with a constant ratio $t_k/N=\bar{\gamma}_k$ with respect to N as N grows, and the measurements in each region are independent. In practice, this may correspond to a random walk in each of the region for a fix, but *unknown*, portion of time $\bar{\gamma}_k$ as N grows.

We consider a deterministic proxy for $f_k(\cdot)$ under large N defined as

$$F_k(\tau_k;\boldsymbol{\tau}_{-k}) = \mathbb{E}\{f_k(\tau_k;\boldsymbol{\tau}_{-k})\} \quad (25)$$

where the expectation is over the randomness of the measurement noise $\boldsymbol{\epsilon}_i$ in (1). As a result, $F_k(\cdot)$ represents the cost in the noiseless case. In the remaining part of the paper, we may omit the argument $\boldsymbol{\tau}_{-k}$ and write $f_k(\tau_k)$ and $F_k(\tau_k)$ for simplicity, as long as it is clear from the context.

To investigate the asymptotic property for large N, we replace τ_k with $\gamma_k N$ in (22) and (25). We define $\bar{f}_k(\gamma_k) = \lim_{\beta\to 0} f_k(\gamma_k N)$, and $\bar{F}_k(\gamma_k) = \lim_{\beta\to 0} F_k(\gamma_k N)$ for $\gamma_k \in \Gamma = \{i/N : i=1,2,...,N\}$. Denote γ_k^* as the minimizer of $\bar{F}_k(\gamma_k)$, and $\hat{\gamma}_k$ as the minimizer of $\bar{f}_k(\gamma_k)$. Then, we have the following result.

Proposition 2 (Asymptotic Consistency): Suppose that, for some k, there exists only one index $t_j \in \{t_1,t_2,\ldots,t_{K-1}\}$ within the interval (τ_{k-1},τ_{k+1}). Then, it holds that $\hat{\gamma}_k \to \gamma_k^*$ in probability as $N\to\infty$.

Proof: See Appendix B. □

Proposition 2 implies that if $\hat{\tau}_k$ minimizes $f_k(\tau_k)$, and τ_k^* minimizes $F_k(\tau_k)$ as $\beta\to 0$, then, we have $\hat{\tau}_k/N \to \tau_k^*/N$ as $N\to\infty$ and $\beta\to 0$. Thus, the estimator $\hat{\tau}_k$ obtained from $f_k(\cdot)$ and the solution τ_k^* obtained from $F_k(\cdot)$ are asymptotically consistent in a wide sense. Therefore, we use $F_k(\tau_k)$ as a deterministic proxy for the stochastic function $f_k(\tau_k;\boldsymbol{\tau}_{-k})$ and we thus study the property of $F_k(\tau_k)$.

C. The Property of the Deterministic Proxy $F_k(\tau_k)$

We find the following properties for $F_k(\tau_k)$.

Proposition 3 (Unimodality): Suppose that, for some k, there exists only one index $t_j \in \{t_1,t_2,\ldots,t_{K-1}\}$, within the interval (τ_{k-1},τ_{k+1}). Then, for any $\varepsilon>0$, there exists a small enough β, and some finite constants $C_1,C_2,C_1',C_2'>0$ independent of ε, such that $F_k(\tau)-F_k(\tau-1) < \varepsilon C_1 - C_2 < 0$, for $\tau_{k-1}<\tau\le t_j$, and $F_k(\tau)-F_k(\tau-1) > C_2' - \varepsilon C_1' > 0$, for $t_j<\tau<\tau_{k+1}$. In addition, t_j minimizes $F_k(\tau)$ in (τ_{k-1},τ_{k+1}).

Proof: See Appendix C-1. □

This result implies that, once the condition is satisfied, there exists a unique local minima t_j of $F_k(\tau)$ over (τ_{k-1},τ_{k+1}).

Proposition 4 (Flatness): Suppose that, for some k, there is no index $t_j \in \{t_1,t_2,\ldots,t_{K-1}\}$ in the interval (τ_{k-1},τ_{k+1}). Then, for any $\varepsilon>0$, there exists a small enough β and a finite constant $C_0>0$ independent of ε, such that $|F_k(\tau)-F_k(\tau-1)| < \varepsilon s^2 C_0$ for all $\tau\in(\tau_{k-1},\tau_{k+1})$.

Proof: See Appendix C-2. □

It follows that, when the interval (τ_{k-1},τ_{k+1}) is completely contained in (t_j,t_{j+1}) for some j, the function $F_k(\tau)$ appears as an almost flat function for $\tau\in(\tau_{k-1},\tau_{k+1})$ with only small fluctuation according to the noise variance s^2.

Proposition 5 (Monotonicity near Boundary): Suppose that, for some k, there are multiple partition indices $t_j,t_{j+1},\ldots,t_{j+J} \in \{t_1,t_2,\ldots,t_{K-1}\}$ within the interval (τ_{k-1},τ_{k+1}). In addition, assume that the vectors $\{\boldsymbol{\mu}_k\}$ are linearly independent. Then, for any $\varepsilon>0$, there exists a small enough β, and some finite constant $C_3,C_4,C_3',C_4'>0$, such that $F_k(\tau)-F_k(\tau-1) < \varepsilon C_3 - C_4 < 0$ for $\tau_{k-1}<\tau\le t_j$, and $F_k(\tau)-F_k(\tau-1) > C_4' - \varepsilon C_3' > 0$ for $t_{j+J}<\tau<\tau_{k+1}$.

Proof: See Appendix C-3. □

This result implies that the proxy cost function $F_k(\tau_k)$ monotonically decreases in $(\tau_{k-1}, t_j]$ and increases in $[t_{j+J}, \tau_{k+1})$.

Properties in Proposition 3–5 lead to a useful design intuition for the algorithm. First, if (τ_{k-1}, τ_{k+1}) contains only one index $t_j \in \{t_1, t_2, \ldots, t_{K-1}\}$, then t_j can be found by minimizing $F_k(\tau)$ in (τ_{k-1}, τ_{k+1}). Second, if (τ_{k-1}, τ_{k+1}) contains none index t_j, then there must be another interval $(\tau_{k'-1}, \tau_{k'+1})$ containing more than one indices $t_j, t_{j+1}, \ldots, t_{j+J}$, and as a result, by minimizing $F_{k'}(\tau)$ over $(\tau_{k'-1}, \tau_{k'+1})$, the solution satisfies $\tau_{k'} \in [t_j, t_{j+J}]$. This intuition leads to a successive merge-and-split algorithm derived as follows.

D. A Merge-and-Split Algorithm Under the Proxy Cost

Denote $\boldsymbol{\tau}^{(m)}$ as the segmentation variable from the mth iteration, and $\mathcal{C}_k^{(m)} = \{\tau_{k-1}^{(m)} + 1, \tau_{k-1}^{(m)} + 2, \ldots, \tau_k^{(m)}\}$ as the corresponding index set of the kth cluster based on the segmentation variable $\boldsymbol{\tau}^{(m)}$. For the $(m+1)$th iteration, the **merge** step picks a cluster $\mathcal{C}_k^{(m)}$, $k \in \{1, 2, ..., K-1\}$, and merges it with the adjacent cluster $\mathcal{C}_{k+1}^{(m)}$, forming in a new set of $K-1$ clusters $\tilde{\mathcal{C}}_1^{(m,k)}, \tilde{\mathcal{C}}_2^{(m,k)}, \ldots, \tilde{\mathcal{C}}_{K-1}^{(m,k)}$; algebraically, it is equivalent to removing the kth variable $\tau_k^{(m)}$, resulting in a set of $K-2$ segmentation variables, denoted as an $(K-2)$-tuple $\boldsymbol{\tau}^{(m,k)} = (\tau_1^{(m,k)}, \tau_2^{(m,k)}, \ldots, \tau_{K-2}^{(m,k)})$.

In the **split** step, a cluster $\tilde{\mathcal{C}}_j^{(m,k)}$ is selected and split it into two, resulting in a new set of K clusters $\tilde{\mathcal{C}}_1^{(m,k,j)}, \tilde{\mathcal{C}}_2^{(m,k,j)}, \ldots, \tilde{\mathcal{C}}_K^{(m,k,j)}$. The corresponding segmentation indices are denoted as an $(K-1)$-tuple $\boldsymbol{\tau}^{(m,k,j)} = (\tau_1^{(m,k,j)}, \tau_2^{(m,k,j)}, ..., \tau_{K-1}^{(m,k,j)})$. Then, we minimize $F_j(\tau; \boldsymbol{\tau}_{-j}^{(m,k,j)})$ subject to $\tau \in (\tau_{j-1}^{(m,k,j)}, \tau_{j+1}^{(m,k,j)})$, and denote the minimal value as $F_*^{(m,k,j)}$. In addition, denote the cost reduction as

$$\triangle F_*^{(m,k,j)} = F_j(\tau_{j-1}^{(m,k,j)}; \boldsymbol{\tau}_{-j}^{(m,k,j)}) - F_*^{(m,k,j)} \tag{26}$$

where $F_j(\tau_{j-1}; \boldsymbol{\tau}_{-j}^{(m,k,j)})$ equals to cost of not splitting $\tilde{\mathcal{C}}_j^{(m,k)}$.

As a result, $\triangle F_*^{(m,k,j)}$ corresponds to the cost reduction of merging clusters $\mathcal{C}_k^{(m)}$ and $\mathcal{C}_{k+1}^{(m)}$ followed by an optimal split of the jth cluster after the merge. Then, by evaluating the cost reduction for all $(K-1)^2$ possible combinations of the merge-and-split, one can find the best segmentation variable $\boldsymbol{\tau}^{(m+1)}$ that yields the least cost for the $(m+1)$th iteration. The overall procedure is summarized in Algorithm 1.

The complexity can be analyzed as follows. For each iteration, there are $(K-1)^2$ merge-and-split operations. An exhaustive approach to compute Step 2b) in Algorithm 1 requires $\mathcal{O}(N/K)$ steps to enumerate all possible integer values τ in the interval $(\tau_{j-1}^{(m,k,j)}, \tau_{j+1}^{(m,k,j)})$, while for each step, the function $F_j(\tau; \boldsymbol{\tau}_{-j}^{(m,k,j)})$ can be approximately computed by its stochastic approximation $f_j(\tau; \boldsymbol{\tau}_{-j}^{(m,k,j)})$ in (22), which requires a complexity of $\mathcal{O}(DN/K)$. As a result, for each iteration, it requires a complexity of $\mathcal{O}(DN^2)$. For a benchmark, an exhaustive approach to solve (20) requires $\mathcal{O}(N^{(K-1)})$ iterations and each iteration requires a complexity of $\mathcal{O}(NK)$ to evaluate (20). Thus, the proposed merge-and-split is efficient.

Algorithm 1 The merge-and-split algorithm

Initialize $\tau_k^{(0)} = \lfloor kN/K \rfloor$ for $k = 0, 1, 2, \ldots, K-1, K$.

Loop

- For each $k = 1, 2, \ldots, K-1$,
  1) Merge the adjacent clusters $\mathcal{C}_k^{(m)}$ and $\mathcal{C}_{k+1}^{(m)}$ to form a new set of clusters $\tilde{\mathcal{C}}_1^{(m,k)}, \tilde{\mathcal{C}}_2^{(m,k)}, \ldots, \tilde{\mathcal{C}}_{K-1}^{(m,k)}$.
  2) For each $j = 1, 2, \ldots, K-1$,
     a) Split the jth set $\tilde{\mathcal{C}}_j^{(m,k)}$ into two subsets to form the new clustering $\tilde{\mathcal{C}}_1^{(m,k,j)}, \tilde{\mathcal{C}}_2^{(m,k,j)}, \ldots, \tilde{\mathcal{C}}_K^{(m,k,j)}$ with the corresponding segmentation indices $\boldsymbol{\tau}^{(m,k,j)}$;
     b) Compute $F_*^{(m,k,j)} = \min\{F_j(\tau; \boldsymbol{\tau}_{-j}^{(m,k,j)}) : \tau_{j-1}^{(m,k,j)} < \tau < \tau_{j+1}^{(m,k,j)}\}$, and denote the minimizer as τ_j^*. The merge-and-split forms a new segmentation $\hat{\boldsymbol{\tau}}^{(m,k,j)} = (\tau_1^{(m,k,j)}, ..., \tau_{j-1}^{(m,k,j)}, \tau_j^*, \tau_{j+1}^{(m,k,j)}, ..., \tau_{K-1}^{(m,k,j)})$;
     c) Compute the cost reduction $\triangle F_*^{(m,k,j)}$ as in (26);
  3) Pick $j^*(k) \triangleq \arg\max_j \triangle F_*^{(m,k,j)}$.
- For all $\{\hat{\boldsymbol{\tau}}^{(m,k,j^*(k))}, k = 1, 2, \ldots, K-1\}$, pick the one that minimizes $\sum_{k=1}^{K-1} F_k(\hat{\tau}_k^{(m,k,j^*(k))}; \hat{\boldsymbol{\tau}}_{-k}^{(m,k,j^*(k))})$, $k \in \{1, 2, \ldots, K-1\}$, and assign it as $\boldsymbol{\tau}^{(m+1)}$.

Repeat until $\tilde{F}(\boldsymbol{\tau}^{(m+1)}) = \tilde{F}(\boldsymbol{\tau}^{(m)})$.

E. Convergence and Optimality

Algorithm 1 must converge due to the following two properties. First, the cost function is lower bounded by 0 since it is the sum of squares. Second, define $\tilde{F}(\boldsymbol{\tau}) \triangleq \sum_{k=1}^{K-1} F_k(\tau_k; \boldsymbol{\tau}_{-k})$, if $\tilde{F}(\boldsymbol{\tau}^{(m+1)}) \neq \tilde{F}(\boldsymbol{\tau}^{(m)})$, then the mth iteration must *strictly* decrease the cost. Specifically, for each k, Step 3) in Algorithm 1 guarantees $\triangle F_*^{(m,k,j^*(k))} \geq \triangle F_*^{(m,k,k)}$, implying that $\tilde{F}(\hat{\boldsymbol{\tau}}^{(m,k,j^*(k))}) \leq \tilde{F}(\boldsymbol{\tau}^{(m)})$ for all k; in addition, the output of the outer loop implies that $\tilde{F}(\boldsymbol{\tau}^{(m+1)}) \leq \tilde{F}(\hat{\boldsymbol{\tau}}^{(m,k,j^*(k))})$ for all k, and there must be at least one strict inequality for $\tilde{F}(\boldsymbol{\tau}^{(m+1)}) < \tilde{F}(\boldsymbol{\tau}^{(m)})$ if $\boldsymbol{\tau}^{(m)} \neq (t_1, t_2, ..., t_{K-1})$, which is proved in Appendix E.

To investigate the optimality of the converged solution from Algorithm 1, consider the clusters $\tilde{\mathcal{C}}_j^{(m,k)}$ constructed in Step 1) of Algorithm 1. Recall that the data $\{\mathbf{x}_i\}$ is clustered sequentially with segment boundaries $t_1, t_2, \ldots, t_{K-1}$. We have the following property on the cost reduction $\triangle F_*^{(m,k,j)}$ in (26).

Lemma 1 (Cost Reduction): Consider two distinct clusters $\tilde{\mathcal{C}}_j^{(m,k)}$ and $\tilde{\mathcal{C}}_{j'}^{(m,k)}$ constructed from the mth iteration and the kth loop of Step 1) in Algorithm 1. Suppose that there exists at least one index $t_{k'} \in \{t_1, t_2, \ldots, t_{K-1}\}$ in $\tilde{\mathcal{C}}_j^{(m,k)}$, and no such $t_{k'}$ in $\tilde{\mathcal{C}}_{j'}^{(m,k)}$. Then, for a sufficiently small β, $\triangle F_*^{(m,k,j)} > \triangle F_*^{(m,k,j')}$.

Proof: See Appendix D. □

Lemma 1 can be intuitively understood from Propositions 3 and 4, which suggest that $F_j(\tau; \boldsymbol{\tau}_{-j}^{(m,k,j)})$ is unimodal in $(\tau_{j-1}^{(m,k,j)}, \tau_{j+1}^{(m,k,j)})$, but $F_{j'}(\tau; \boldsymbol{\tau}_{-j'}^{(m,k,j')})$ is almost flat in

$(\boldsymbol{\tau}_{j'-1}^{(m,k,j')}, \boldsymbol{\tau}_{j'+1}^{(m,k,j')})$, and hence, the former one has a larger potential to reduce the total cost $\tilde{F}(\boldsymbol{\tau}) = \sum_{k=1}^{K-1} F_k(\tau_k; \boldsymbol{\tau}_{-k})$.

Theorem 1 (Optimality): Algorithm 1 terminates at $\boldsymbol{\tau}^* = (\tau_1^*, \tau_2^*, \ldots, \tau_{K-1}^*)$, with $\tau_k^* = t_k$, $k = 1, 2, \ldots, K-1$.

Proof: See Appendix E. □

As a result, Algorithm 1 can converge to the globally optimal solution t_k under the proxy cost $F_k(\cdot)$ for $d_k = 0$ and $s_k^2 = s^2$ despite the problem being non-convex.

F. Merge-and-Split Clustering for Sequential Data

We now extend Algorithm 1 to the general case for $d_k \geq 0$. Recall from Proposition 2 that the segment boundary estimator $\hat{\tau}_k^{(N)}$ obtained from minimizing $f_k(\tau_k; \boldsymbol{\tau}_{-k})$ is asymptotically consistent with the minimizer τ_k^* of the proxy cost $F_k(\tau_k; \boldsymbol{\tau}_{-k})$ for asymptotically small β and large N. Therefore, it is encouraged to extend Algorithm 1 for minimizing the actual cost $f_k(\tau; \boldsymbol{\tau}_{-k})$. It is clear that f_k, which can be computed directly from the data $\{\mathbf{x}_i\}$, is a stochastic approximation of the proxy function $F_k(\tau_k; \boldsymbol{\tau}_{-k})$.

Specifically, based on the probability model (2), for $d_k \geq 0$, we define

$$\begin{aligned}\mathcal{F}_k(\boldsymbol{\Theta}, \boldsymbol{\tau}) \triangleq \frac{1}{N} \sum_{i=\tau_{k-1}+1}^{\tau_{k+1}} \Big[&\big(1 - \sigma_\beta(i - \tau_k)\big) \Big(\ln|\mathbf{C}_k| \\ &+ (\mathbf{x}_i - \boldsymbol{\mu}_k)^{\mathrm{T}} \mathbf{C}_k^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k) \Big) \\ &+ \sigma_\beta(i - \tau_k) \Big(\ln|\mathbf{C}_{k+1}| \\ &+ (\mathbf{x}_i - \boldsymbol{\mu}_{k+1})^{\mathrm{T}} \mathbf{C}_{k+1}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_{k+1}) \Big) \Big] \end{aligned} \tag{27}$$

for $k = 1, 2, \ldots, K-1$, and

$$\begin{aligned}\mathcal{F}_0(\boldsymbol{\Theta}, \boldsymbol{\tau}) &= \frac{1}{N} \sum_{i=1}^{\tau_1} \sigma_\beta(i - \tau_0) \Big(\ln|\mathbf{C}_1| \\ &\quad + (\mathbf{x}_i - \boldsymbol{\mu}_1)^{\mathrm{T}} \mathbf{C}_1^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_1) \Big) \\ \mathcal{F}_K(\boldsymbol{\Theta}, \boldsymbol{\tau}) &= \frac{1}{N} \sum_{i=\tau_{K-1}+1}^{N} \big(1 - \sigma_\beta(i - \tau_K)\big) \Big(\ln|\mathbf{C}_K| \\ &\quad + (\mathbf{x}_i - \boldsymbol{\mu}_K)^{\mathrm{T}} \mathbf{C}_K^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_K) \Big) \end{aligned}$$

in the same way as (22)–(24). Following the same argument as in Proposition 1, it is observed that maximizing $\mathcal{J}(\boldsymbol{\Theta}, \boldsymbol{\tau})$ in (5) is asymptotically equivalent to minimizing $\frac{1}{2} \sum_{k=0}^{K} \mathcal{F}_k(\boldsymbol{\Theta}, \boldsymbol{\tau})$ for $\beta \to 0$.

In Section III-A, it has been shown that, for a given segmentation variable $\boldsymbol{\tau}$, the solution $\hat{\boldsymbol{\Theta}}(\boldsymbol{\tau})$ can be constructed from (9), (14)–(15). Therefore, the f_k function (22) studied in Section IV-A to Section IV-E can be obtained in a similar way as

$$f_k(\tau_k; \boldsymbol{\tau}_{-k}) = \mathcal{F}_k(\hat{\boldsymbol{\Theta}}(\boldsymbol{\tau}), \boldsymbol{\tau}). \tag{28}$$

As a result, a direct extension of Algorithm 1 to solve for the segmentation $\boldsymbol{\tau}$ is to replace the function $F_k(\tau_k; \boldsymbol{\tau}_{-k})$ in Algorithm 1 with $f_k(\tau_k; \boldsymbol{\tau}_{-k})$ defined in (28). In the following, we call the extended version Algorithm 1 for convenience.

Algorithm 2 Alternating optimization with merge-and-split

Initialize $\tau_k^{(0)}$ as the output of Algorithm 1 by assuming $d_k = 0$.
Loop for the $(m+1)$th iteration:

- Compute $\boldsymbol{\Theta}^{(m)} = \hat{\boldsymbol{\Theta}}(\boldsymbol{\tau}^{(m)})$ from (9), (14)–(15);
- Define the function $f_k(\tau_k; \boldsymbol{\tau}_{-k}) = \mathcal{F}_k(\boldsymbol{\Theta}^{(m)}, \boldsymbol{\tau})$ by fixing $\boldsymbol{\Theta}^{(m)}$;
- Implement one iteration in Algorithm 1 to obtain the update $\boldsymbol{\tau}^{(m+1)}$ by replacing the function $F_k(\tau_k; \boldsymbol{\tau}_{-k})$ in Algorithm 1 with $f_k(\tau_k; \boldsymbol{\tau}_{-k})$ defined in the previous step.

Repeat until $\sum_{k=1}^{K-1} \mathcal{F}_k(\boldsymbol{\Theta}^{(m)}, \boldsymbol{\tau}^{(m+1)}) = \sum_{k=1}^{K-1} \mathcal{F}_k(\boldsymbol{\Theta}^{(m)}, \boldsymbol{\tau}^{(m)})$.

Note that the complexity of evaluating (28) is $\mathcal{O}(D^2N + D^3)$ because computing (27) requires $\mathcal{O}(D^2N/K)$ and computing $\hat{\boldsymbol{\Theta}}(\boldsymbol{\tau})$ requires $\mathcal{O}(D^2N + D^3)$, leading to a total complexity of $\mathcal{O}(D^2N^2K)$ per iteration in Algorithm 1.

To reduce the complexity, one may consider to *alternatively* update $\boldsymbol{\Theta}$ and $\boldsymbol{\tau}$ in $\mathcal{F}_k(\boldsymbol{\Theta}, \boldsymbol{\tau})$. Such an alternating optimization approach can be developed from Algorithm 1 in a straightforward way and is summarized in Algorithm 2. As such, the per iteration complexity is reduced to $\mathcal{O}(D^2N + D^3 + N^2)$.

V. Numerical Experiments

Our experiments were conducted in an office space measuring 30 meters by 16 meters (480 square meters), which is divided into 10 non-overlapping regions as shown in Fig. 1. There are 21 sensors installed as indicated by the blue icons, which are ultrawideband (UWB) sensors operating at 6 GHz frequency band. Measurement data was collected in 3 different time periods spread across 3 different days to capture the time-variation of the environment. During the measurement process in each time period, a mobile device visited all the 10 regions once without repetition, and the RSS value measured by the receivers were recorded. The time interval of the collected data samples is 200 ms. The data collected on the first day was used for clustering or training (for the baseline schemes), and the data collected on the second and third days served as test datasets I and II, respectively. The training dataset comprises 1,455 records, while the two testing datasets contain 1,294 and 1,055 records, respectively.

We also expanded our experiments to a larger area ($53 \times 55\,\mathrm{m}^2$) with 24 regions of interest, including corridors and rooms with various layouts. A total of 50 sensors are installed in the whole area. The samples were collected every 200 ms, and the number of RSS data samples collected in each region ranges from 2,000 to 5,000. The data collected on the first day was used for clustering or training, and the data collected on the second day was used as test dataset III.

A. Verification of the Subspace Model From Real Data

We first justify the subspace model (1) from real data. Specifically, we need to verify that the covariance of the data $\mathbf{x}_i$ has a low rank structure, and moreover, the subspaces are non-identical across regions.

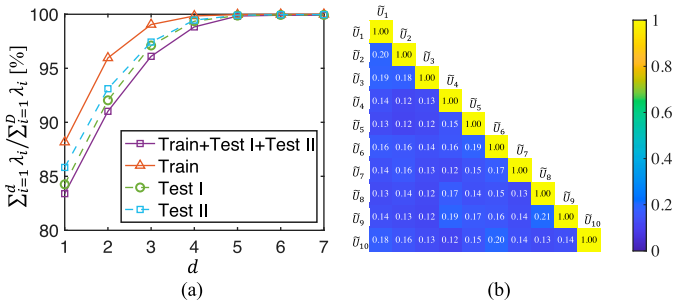

Fig. 2. (a) Percentage of the sum of the first d dominant eigenvalues of the covariance matrix of the data collected from Region 1. (b) Similarity scores $[0, 1]$ between subspaces, where 0 means the two subspaces are orthogonal and 1 means identical.

In Fig. 2(a), we extract the subset of data $\{\mathbf{x}_i, i \in \mathcal{C}_1\}$ collected from Region 1 illustrated in Fig. 1, and compute the corresponding covariance matrix $\hat{\mathbf{C}}_1$. Denote the eigenvalues of $\hat{\mathbf{C}}_1$ as $\lambda_1, \lambda_2, ..., \lambda_D$. Fig. 2(a) shows the percentage of the sum of the first d dominant eigenvalues, i.e., $\sum_{i=1}^{d} \lambda_i / \sum_{i=1}^{D} \lambda_i$. It is observed that, the top 3 principal components already contains up to 97.0% of energy. This indicates that the data collected from Region 1 exhibits low-rank characteristics. A similar observation can be made from the data collected from other regions. Hence, a subspace model can describe the data with a good accuracy.

Fig. 2(b) plots the similarity between affine subspaces. Specifically, the affine subspace model (1) can also be written as a linear subspace model form $\mathbf{x}_i = \tilde{\mathbf{U}}_k \left[\boldsymbol{\theta}_i, \|\tilde{\boldsymbol{\mu}}_k\|_2\right]^{\mathrm{T}} + \boldsymbol{\epsilon}_i$, where $\tilde{\mathbf{U}}_k = [\mathbf{U}_k, \tilde{\boldsymbol{\mu}}_k / \|\tilde{\boldsymbol{\mu}}_k\|_2]$ is the subspace basis of the linear subspace, and $\tilde{\boldsymbol{\mu}}_k = (\mathbf{I}_D - \mathbf{U}_k \mathbf{U}_k^{\mathrm{T}}) \boldsymbol{\mu}_k$ is the offset of the kth affine subspace satisfying orthogonality to the columns of $\mathbf{U}_k$. The similarity between the linear subspaces $\tilde{\mathbf{U}}_i$ and $\tilde{\mathbf{U}}_j$ is calculated by $\text{trace}\{\tilde{\mathbf{U}}_i \tilde{\mathbf{U}}_i^{\mathrm{T}} \tilde{\mathbf{U}}_j \tilde{\mathbf{U}}_j^{\mathrm{T}}\} / \min(d_i + 1, d_j + 1)$. It is observed that the subspaces are nearly orthogonal to each other.

B. Clustering Performance

Four clustering metrics are used: clustering accuracy (Acc), normalized mutual information (NMI), F-Score (F1), adjusted rand index (ARI), and precision (P) [32]. We compare our method with a classical subspace clustering method: EM-based subspace clustering (ESC) [20] that iterates between clustering to subspace and estimating the subspace model under a probabilistic PCA structure. We also compare our method to two recently developed clustering algorithms reported in the literature: self-constrained spectral clustering (SCS) [32], and local self-expression subspace clustering (LSE) [25]. SCS is an extension of spectral clustering that incorporates pairwise and label self-constrained terms into the objective function of spectral clustering to guide the clustering process using some prior information. LSE incorporates a variational autoencoder (VAE) neural network with a temporal convolution module to construct a self-expression affinity matrix with temporally consistent priors, followed by a conventional graph-based clustering algorithm. The local self-expression layer in the temporal convolution module only maintains representation relations with temporally adjacent data to implement the local validity of self-expression.

TABLE I
COMPARISON ON THE CLUSTERING ACCURACY

Methods	Acc	NMI	F1	ARI	P
ESC [20]	48.9	62.7	49.7	42.2	70.9
SCS [32]	68.0	81.7	68.4	64.5	82.5
LSE [25]	80.6	87.6	80.8	79.2	88.4
Algorithm 1	98.5	96.8	97.0	96.6	97.0
Algorithm 2	98.5	96.8	97.0	96.6	97.0

Table I summarizes the comparison of clustering performance. The classical subspace clustering algorithm ESC performs poorly due to significant fluctuations in RSS measurements. The proposed algorithms, and LSE both outperform SCS, highlighting the limitations of non-subspace clustering algorithms when applied to high-dimensional sequential data clustering problems. Although LSE outperforms SCS by employing high-dimensional feature extraction with temporal consistency priors, the proposed algorithms still performs better than LSE. This is because LSE focuses solely on extracting smooth data features through the construction of a self-expression matrix using a temporal convolution module, without considering temporal consistency in identifying cluster structures. The proposed schemes perform the best among all the schemes and achieve a 98.5% clustering accuracy. Note that in real datasets, there exists a transition phase when a device moves from one region to another. Data in this transition phase may not belong to any specific subspace, which could contribute to the 1.5% clustering error observed in the proposed algorithm.

C. Convergence and Verification of the Theoretical Results

In practice, the mobile device has non-negligible transition from one region to another, and therefore, the exact timing at the region boundary is not well-defined. We thus define the ε-tolerance error

$$E_\varepsilon = \frac{1}{N} \sum_{k=1}^{K-1} \max\{|\tau_k - t_k| - \varepsilon N, 0\}$$

to evaluate the convergence of the algorithms. Here, global optimality is claimed when $E_\varepsilon = 0$.

The proposed Algorithms 1 and 2 and their variants, marked as "R", are evaluated. The "R" version of the proposed algorithms are randomly initialized following a uniform distribution for $\tau_k^{(0)}$ in the first line of Algorithm 1. Algorithm 2 is initialized by running Algorithm 1 for 15 iterations, which have been counted in the total iterations of Algorithm 2 in Fig. 3. In Fig. 3(a), the cost value reduction of Algorithm 2 tends to saturate at the 15th iteration (initial phase of Algorithm 2), but the cost value continues to quickly decreases starting from the 16th iteration (main loop of Algorithm 2). The convergence is benchmarked with a gradient-based subspace clustering (GSC) method [28], which employs stochastic gradient descent to search for $\boldsymbol{\tau}$.

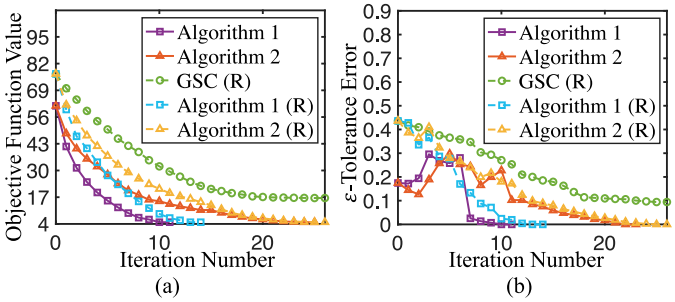

Fig. 3. Convergence of the proposed algorithms, where the curves marked with "R" represents the mean trajectories for 20 independent random initializations.

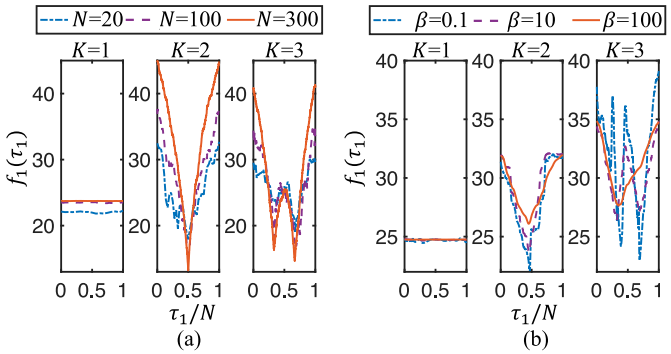

Fig. 4. Verification of the unimodality, the flatness, and the monotonicity near boundary of the cost function.

Fig. 3 shows the objective function value $\sum_{k=1}^{K-1} \mathcal{F}_k(\boldsymbol{\Theta}(\boldsymbol{\tau}), \boldsymbol{\tau})$ and the ε-tolerance error E_ε against the iteration number, where ε is chosen as $\varepsilon = 0.3\%$. Both Algorithms 1 and 2, as well as their variants with random initialization, converge to the globally optimal solution. By contrast, the GSC baseline is not guaranteed to converge to $E_\varepsilon = 0$, as it is easily trapped at poor local optimum. In addition, Algorithm 1 requires fewer iterations than Algorithm 2, but it has a higher computational complexity per iteration. Specifically, for the dataset we used, the total computational time of Algorithm 1 is 1033 seconds, whereas, that of Algorithm 2 is 245 seconds, 4X faster than Algorithm 1. Nevertheless, Algorithm 1 is guaranteed to globally converge in a special case as stated in Theorem 1.

Next, we verify the theoretical properties in Propositions 3–5 through two numerical examples. In Fig. 4(a), a simulated dataset is constructed based on the subspace model (1) with $D = 40$, $d_k = 0$, and K clusters. The ratio $\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_2^2/s^2$ of the squared-distance between the cluster centers over the noise variance s^2 is set as 2.5. The data is segmented into two parts by τ_1 and the cost function $f_1(\tau_1)$ is plotted. First, as the number of samples N increases, the cost function $f_1(\tau_1)$ eventually becomes the deterministic proxy $F_1(\tau_1) = \mathbb{E}\{f_1(\tau_1)\}$ as shown by the group of curves for $K = 2$ clusters. Second, $f_1(\tau_1)$ appears as unimodal for $K = 2$ clusters under large N, which agrees with the results in Proposition 3. Third, for $K = 1$ cluster, $f_1(\tau_1)$ is an almost flat function under large N as discussed in Proposition 4. For $K = 3$ clusters under large N, $f_1(\tau_1)$ appears as monotonic near the left and right boundaries as discussed in Proposition 5.

In Fig. 4(b), the experiment is extended to real data. We extract a subset of data samples belonging to $K = 1, 2, 3$ consecutive clusters from the measurement dataset, and plot the cost function $f_1(\tau_1)$ under parameter $d_k = 2$ and different β values. It is observed that the cost function appears as unimodal disturbed by noise for $K = 2$ clusters, which is consistent with the results in Proposition 3. This also provides a justification that the proposed subspace model is essentially accurate in real data. In addition, while a small β may help amplify the unimodality of the cost function as implied by Proposition 3, the cost function is prone to be disturbed by the modeling noise, resulting in multiple local minimizers. On the contrary, a medium to large β may help eliminate the noise for a unique local minimizer.

TABLE II
AVERAGE MATCHING ERROR (×100%) PERFORMANCE OF OUR SUBSPACE MATCHING

$\lvert\mathcal{E}\rvert$	18	19	20	21	22	23	24	25	26	27
$\alpha = 1$	0.3	0.5	1.0	1.3	1.4	1.5	2.0	1.9	2.8	1.9
$\alpha = 2$	0.3	0.5	1.0	2.5	3.0	3.3	3.8	3.0	3.3	3.8
$\alpha = 4$	0.5	1.0	0.8	2.5	4.0	3.8	4.0	4.5	4.0	3.5

D. Cluster to Physical Region Matching

The proposed cluster-to-region matching is based on a graph $\mathcal{G}$, which may be generated from the floor plan in practice. For evaluation purpose here, we generate a set of graphs by randomly assigning edges between regions. The probability of an edge joining two regions j and k is $q_{jk} = C_\mathrm{e} \cdot \exp(-\|\mathbf{o}_j - \mathbf{o}_k\|_2^2)$, *i.e.*, the smaller the distance, the higher the probability, where C_e is a normalizing factor for the expected number of edges in the graph, and $\mathbf{o}_j$ and $\mathbf{o}_k$ are the reference locations of the jth and kth region, respectively, as shown in Fig. 1. The matching error is computed as $E_\mathrm{m} = \frac{1}{K}\sum_k \mathbb{I}\{\pi^*(k) \neq \pi(k)\}$ where $\boldsymbol{\pi}^*$ is the desired matching.

Table II summarizes the matching error for different graphs. It is observed that the fewer the edges, the lower the matching error. This is because the constraint set in (19) is smaller for fewer edges. For the graphs with 18 edges, the average number of eligible routes satisfying constraint (19) is 287 in our region topology; for the graphs with 27 edges, that number is above 13,000. Nevertheless, the overall matching error is mostly below 3% under parameter $\alpha = 1$ and below 1% for $|\mathcal{E}| \leq 20$ under all parameter values $\alpha = 1, 2, 4$ when computing the reference centroid in (17). In addition, we compare the performance under different parameters α. It is shown that the performance is not sensitive with α, and the overall matching error is mostly below 4%.

E. Localization Performance

We evaluate the localization performance of the region-based radio map using test datasets I and II, which have not been

TABLE III
REGION LOCALIZATION ERROR [METERS] ON 10-REGION TEST DATASET I AND II

Method	Supervised			Unsupervised		
	KNN	SVM	DNN	MR	WCL	Proposed
Test Dataset I (Day 2)	0.81	0.81	0.79	1.08	1.67	**0.49**
Test Dataset II (Day 3)	0.92	0.91	0.88	1.40	1.90	**0.68**

used for the radio map construction. A maximum-likelihood approach is used based on the conditional probability function (2), and the estimated region $\hat{k}$ given the RSS measurement vector $\mathbf{x}$ is given by

$$\hat{k} = \underset{k\in\{1,2,\ldots,K\}}{\operatorname{argmax}} \; p_k(\mathbf{x};\boldsymbol{\Theta}). \tag{29}$$

We compare the localization performance with two unsupervised schemes, max-RSS (MR), which picks the location of the sensor that observes the largest RSS as the target location, and WCL [8], which estimates the location as (17). For performance benchmarking, we also evaluate three supervised localization approaches KNN [33], [34], SVM [35], and DNN [36], [37], which are trained using the training set with region labels that were not available to the proposed scheme. The parameters of baseline methods are determined and tuned using a ten-fold cross validation. For KNN, the optimal number of neighbors was found to be 8. A Gaussian kernel was used for SVM. For DNN, we adopt a three layer multilayer perceptron (MLP) neural network with 30 nodes in each layer to train the localization classifier. The parameter β in (7) is set to be 1. The subspace dimension d_k is chosen from 1 to 3 according to the number of sensors located in the region as shown in Fig. 1. The performance is evaluated using the mean *region localization error* defined as $\mathbb{E}\{\|\mathbf{o}_{\hat{k}} - \mathbf{o}_k\|\}$, where $\mathbf{o}_k$ is the reference location of the kth region.

1) Region Localization Performance on 10-Region Dataset: Although we assume that the mobile device visits each divided region only once, which imposes limitations on the applicability of our model in certain scenarios, this assumption ensures that we achieve a good localization performance. Table III summarizes the region localization error. Somewhat surprisingly, the proposed scheme, trained without labels, performs even better then the supervised methods; in fact, it performs the best among all the schemes tested. The major performance gain is contributed by the signal subspace model (1), which allows more fluctuation of the data, and tolerates the slight change of the environment. Moreover, SVM does not fit the features of RSS data but instead searches for the boundaries of RSS clusters. DNN attempts to fit the features of RSS data using a general non-linear function, allowing it to capture complex patterns and intricate relationships in the data. Consequently, DNN and SVM are highly sensitive to noise, leading to potential overfitting issues. SVM achieves optimal performance on test dataset I when the slack variable is configured to 0.0007, resulting in a localization error of 0.60 meters. DNN achieves its best performance on test dataset I when the regularization hyperparameter is configured to 1.67, resulting in a localization error of 0.54 meters. Despite alleviating overfitting by tuning overfitting parameters, finding the optimal parameter setting remains challenging. Therefore, although they are supervised methods with labels, the performance of SVM and DNN is slightly inferior to the proposed one.

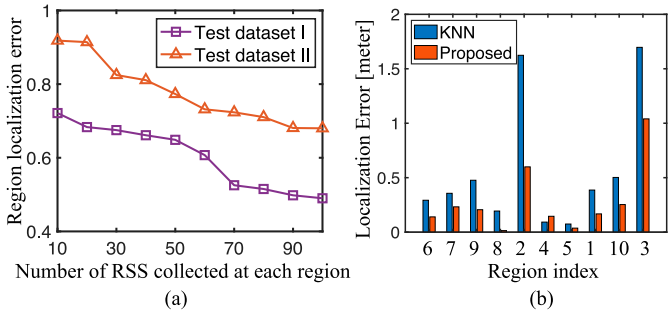

Fig. 5. (a) Region localization error of our method versus the time interval for collecting the training data. (b) Region localization error at each region, where the regions are arranged in an increasing order of their sizes.

The decrease in positioning accuracy from Day 2 to Day 3 by more than 10% is because we slightly change the environment to verify the robustness of the algorithm. Specifically, the environment undergoes changes as we gather data during regular working hours, simulating real-world positioning scenarios, where we vary the numbers and locations of personnel, and we vary the obstacles such as chairs and decorations on tables. Moreover, the pose of the mobile device also changes during measurements for test dataset I and II. For the training dataset and the test dataset I, the mobile device is held in the hand, and it swings with the arm. For the test dataset II, it is placed in a backpack. Therefore, the fluctuation in the height of the mobile device and variations in signal transmission strength result in a series of impacts. Nevertheless, the proposed method is the least affected.

2) Effect of Region Size and Measurement Quantity in Each Region: Note that the limitation of the proposed work is to provide only a region-based localization, and therefore, the localization accuracy depends on the size of the region and the amount of data collected from each region. We numerically evaluate the localization performance versus the amount of training data as shown in Fig. 5(a). We vary the number of RSS data points collected in each region within the training set. As the quantity of RSS data increases, there is a reduction in localization error as expected.

To evaluate the impact from the size of the area, we sorted 10 regions based on their sizes in increasing order and calculated the localization error for each region in test dataset I. As shown in Fig. 5(b), there is no clear trend of an increase in localization error with the increase in the area of the region. However, regions 2 and 3 exhibit relatively large localization errors for all methods, attributed to the fact that these areas lack distinct separation by large furniture or walls. Moreover, in our measurement, the mobile device traveled around a region, and also stayed for a while at a spot. The measurement was recorded for every 200 ms. Hence, at least, a subset of data is highly

TABLE IV
REGION LOCALIZATION ERROR [METERS] ON 24-REGION TEST DATASET III

Dataset III	Supervised			Unsupervised		
	KNN	SVM	DNN	MR	WCL	Proposed
Corridor	0.69	0.60	0.59	0.99	1.15	**0.36**
Room	0.62	0.58	0.55	0.95	1.08	**0.32**
All	0.65	0.59	0.58	0.97	1.12	**0.34**

correlated. From our results, we did not notice any significant performance impact from the data correlation so far.

3) Region Localization Performance on 24-Region Dataset: While the algorithm was tested in a relatively small area (but real office environment), the proposed method is scalable to a larger indoor area, such as shopping malls. From a computational complexity perspective, a large indoor area typically has more APs; but since the complexity of the proposed algorithm is $\mathcal{O}(DN^2)$, i.e., linearly in the number of APs D, the proposed method can be easily scaled to a large indoor area. As shown in Table IV, our method in the extended scenario acquires a 0.34-meter region localization error, and outperforms all baseline methods.

In Table IV, the localization errors of our algorithm in corridor regions (including 8 out of 24 regions) were 0.36 meters, while the localization error in room regions was measured at 0.32 meters. The localization error in corridor regions is slightly higher than in room regions. The primary reason is the absence of clear obstacles between corridors, such as doors or walls, making it challenging for our signal subspace model to classify the RSS data measured between two corridors.

VI. CONCLUSION

In this paper, a subspace clustering method with a sequential prior is proposed to construct a region-based radio map from sequentially collected RSS measurements. A maximum-likelihood estimation problem with a sequential prior is formulated, and solved by a proposed merge-and-split algorithm that is proven to converge to a globally optimal solution for a special case. Furthermore, a graph model for a set of possible routes is constructed which leads to a Viterbi algorithm for the region matching and achieves less than 1% matching error. The numerical results demonstrated that the proposed unsupervised scheme achieves an even better localization performance than several supervised learning schemes, including KNN, SVM, and DNN which use location labels during the training.

APPENDIX A
PROOF OF PROPOSITION 1

We have

$$\tilde{f}(\boldsymbol{\tau}) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1,j\neq k,k+1}^{K} z_\beta(i,\tau_{j-1},\tau_j)\big\|\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k(\tau_{k-1},\tau_k)\big\|_2^2 + \tilde{f}_k(\tau_k)$$

where

$$\tilde{f}_k(\tau_k) = \frac{1}{N}\sum_{i=1}^{N}\Big[z_\beta(i,\tau_{k-1},\tau_k)\big\|\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k(\tau_{k-1},\tau_k)\big\|_2^2 + z_\beta(i,\tau_k,\tau_{k+1})\big\|\mathbf{x}_i - \hat{\boldsymbol{\mu}}_{k+1}(\tau_k,\tau_{k+1})\big\|_2^2\Big].$$

Recall the uniform convergence for $\hat{\boldsymbol{\mu}}_k(\tau_{k-1},\tau_k)$ as $\beta\to 0$ in (21), i.e., $\hat{\boldsymbol{\mu}}_k(\tau_{k-1},\tau_k)\to\tilde{\boldsymbol{\mu}}(\tau_{k-1},\tau_k)$. We have

$$\tilde{f}_k(\tau_k)\to\bar{f}_k(\tau_k) \triangleq \frac{1}{N}\sum_{i=1}^{N}\Big(z_\beta(i,\tau_{k-1},\tau_k)\big\|\mathbf{x}_i - \tilde{\boldsymbol{\mu}}(\tau_{k-1},\tau_k)\big\|_2^2 + z_\beta(i,\tau_k,\tau_{k+1})\big\|\mathbf{x}_i - \tilde{\boldsymbol{\mu}}(\tau_k,\tau_{k+1})\big\|_2^2\Big)$$

where $\bar{f}_k(\tau_k)$ is the asymptotic form of $\tilde{f}_k(\tau_k)$ as $\beta\to 0$.

Since $z_\beta(i,t_{k-1},t_k)\to\mathbb{I}\{t_{k-1}<i\le t_k\}$ as $\beta\to 0$, where the indicator function $\mathbb{I}\{t_{k-1}<i\le t_k\}=1$ if $t_{k-1}<i\le t_k$, and 0 otherwise, we have

$$\sum_{i=1}^{N} z_\beta(i,\tau_{k-1},\tau_k)\big\|\mathbf{x}_i - \tilde{\boldsymbol{\mu}}(\tau_{k-1},\tau_k)\big\|_2^2 \to \sum_{i=\tau_{k-1}+1}^{\tau_{k+1}}(1-\sigma_\beta(i-\tau_k))\big\|\mathbf{x}_i - \tilde{\boldsymbol{\mu}}(\tau_{k-1},\tau_k)\big\|_2^2$$

and

$$\sum_{i=1}^{N} z_\beta(i,\tau_k,\tau_{k+1})\big\|\mathbf{x}_i - \tilde{\boldsymbol{\mu}}(\tau_k,\tau_{k+1})\big\|_2^2 \to \sum_{i=\tau_{k-1}+1}^{\tau_{k+1}}\sigma_\beta(i-\tau_k)\big\|\mathbf{x}_i - \tilde{\boldsymbol{\mu}}(\tau_k,\tau_{k+1})\big\|_2^2$$

as $\beta\to 0$.

Thus, as $\beta\to 0$, we have $\tilde{f}_k(\tau_k)\to f_k(\tau_k)$. So, we have $\tilde{f}(\boldsymbol{\tau})\to\frac{1}{2}\sum_{k=0}^{K} f_k(\tau_k)$ as $\beta\to 0$.

APPENDIX B
PROOF OF PROPOSITION 2

Without loss of generality, we study the case $K=2$ and $k=1$. From (22) and (25), $\bar{F}_k(\gamma_k)$ and $\bar{f}_k(\gamma_k)$ can be written as $\bar{F}_1(\gamma_1)=\mathbb{E}\{\bar{f}_1(\gamma_1)\}$ and

$$\bar{f}_1(\gamma_1) = \lim_{\beta\to 0}\frac{1}{N}\sum_{i=1}^{N}\Big[\Big(1-\sigma_\beta(i-\gamma_1 N)\Big)\big\|\mathbf{x}_i - \tilde{\boldsymbol{\mu}}(0,\gamma_1 N)\big\|_2^2 + \sigma_\beta(i-\gamma_1 N)\big\|\mathbf{x}_i - \tilde{\boldsymbol{\mu}}(\gamma_1 N, N)\big\|_2^2\Big]. \quad (30)$$

Firstly, we prove that $\bar{f}_1(\gamma_1)\xrightarrow{\mathrm{P}}\bar{F}_1(\gamma_1)$ uniformly for all γ_1, i.e., $\sup_{\gamma_1\in\Gamma}|\bar{f}_1(\gamma_1)-\bar{F}_1(\gamma_1)|\xrightarrow{\mathrm{P}}0$ as $N\to\infty$, where $\xrightarrow{\mathrm{P}}$ denotes convergence in probability.

For any $\gamma_1 \le \bar{\gamma}_1 = t_1/N$, one can easily derive from (21) and (30) that, with $s_0^2 \triangleq Ds^2$,

$$\begin{aligned}\bar{f}_1(\gamma_1) - \bar{F}_1(\gamma_1) &= \lim_{\beta\to 0} f_1(\gamma_1 N) - \lim_{\beta\to 0} F_1(\gamma_1 N) \\ &= 2(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^{\mathrm{T}} \frac{1}{N}\left[\frac{\gamma_1 - \bar{\gamma}_1}{1-\gamma_1}\sum_{i=\bar{\gamma}_1 N+1}^{N} \boldsymbol{\epsilon}_i \right. \\ &\quad \left. + \frac{1-\bar{\gamma}_1}{1-\gamma_1}\sum_{i=\gamma_1 N+1}^{\bar{\gamma}_1 N} \boldsymbol{\epsilon}_i\right] + \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\epsilon}_i^{\mathrm{T}}\boldsymbol{\epsilon}_i - s_0^2. \end{aligned} \tag{31}$$

Lemma 2: It holds that $\sup_{\gamma_1\in\bar{\Gamma}_1} |\bar{f}_1(\gamma_1) - \bar{F}_1(\gamma_1)| \xrightarrow{\mathrm{P}} 0$ with $N\to\infty$, where $\bar{\Gamma}_1 \triangleq \{1/N, 2/N, ..., \bar{\gamma}_1\}$.

Proof: The absolute value of $\bar{f}_1(\gamma_1) - \bar{F}_1(\gamma_1)$ in (31) can be upper bounded by

$$\begin{aligned}&|\bar{f}_1(\gamma_1) - \bar{F}_1(\gamma_1)| \\ &\le \left|\frac{2(\gamma_1 - \bar{\gamma}_1)}{1-\gamma_1}\right| \cdot |\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2|^{\mathrm{T}} \left|\frac{1}{N}\sum_{i=\bar{\gamma}_1 N+1}^{N} \boldsymbol{\epsilon}_i\right| + \left|\frac{2(1-\bar{\gamma}_1)}{1-\gamma_1}\right| \\ &\quad \times |\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2|^{\mathrm{T}} \left|\frac{1}{N}\sum_{i=\gamma_1 N+1}^{\bar{\gamma}_1 N} \boldsymbol{\epsilon}_i\right| + \left|\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\epsilon}_i^{\mathrm{T}}\boldsymbol{\epsilon}_i - s_0^2\right|. \end{aligned} \tag{32}$$

where $|\mathbf{x}|$ means $(|x_1|, |x_2|, ..., |x_n|)^{\mathrm{T}}$, i.e., taking absolute values for each element of the vector $\mathbf{x}$.

For the first term on the right hand side (R.H.S.) of (32), we have

$$\begin{aligned}&\sup_{\gamma_1\in\bar{\Gamma}_1}\left|\frac{2(\bar{\gamma}_1 - \gamma_1)}{1-\gamma_1}\right| \cdot |\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2|^{\mathrm{T}} \left|\frac{1}{N}\sum_{i=\bar{\gamma}_1 N+1}^{N} \boldsymbol{\epsilon}_i\right| \\ &\le \frac{2(\bar{\gamma}_1 N - 1)}{N-1}\Big(\max_j |\mu_{1,j} - \mu_{2,j}|\Big)\left\|\frac{1}{N}\sum_{i=\bar{\gamma}_1 N+1}^{N} \boldsymbol{\epsilon}_i\right\|_1 \to 0 \end{aligned}$$

as $N\to\infty$. This is because $\frac{2(\bar{\gamma}_1 N-1)}{N-1}(\max_j |\mu_{1,j} - \mu_{2,j}|)$ is bounded and

$$\frac{1}{N}\sum_{i=\bar{\gamma}_1 N+1}^{N} \boldsymbol{\epsilon}_i = \frac{(1-\bar{\gamma}_1)N}{N}\frac{1}{(1-\bar{\gamma}_1)N}\sum_{i=\bar{\gamma}_1 N+1}^{N} \boldsymbol{\epsilon}_i \to \mathbf{0}$$

due to the strong law of large numbers, where we recall that $\boldsymbol{\epsilon}_i$ are i.i.d. with zero mean and bounded variance.

For the second term on the R.H.S. of (32), we have

$$\begin{aligned}&\sup_{\gamma_1\in\bar{\Gamma}_1}\left|\frac{2(1-\bar{\gamma}_1)}{1-\gamma_1}\right| \cdot |\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2|^{\mathrm{T}} \left|\frac{1}{N}\sum_{i=\gamma_1 N+1}^{\bar{\gamma}_1 N} \boldsymbol{\epsilon}_i\right| \\ &\le 2\Big(\max_j |\mu_{1,j} - \mu_{2,j}|\Big)\sup_{\gamma_1\in\bar{\Gamma}_1}\left\|\frac{1}{N}\sum_{i=\gamma_1 N+1}^{\bar{\gamma}_1 N} \boldsymbol{\epsilon}_i\right\|_1 \to 0. \end{aligned}$$

To see this, we compute

$$\begin{aligned}\sup_{\gamma_1\in\bar{\Gamma}_1}\left\|\frac{1}{N}\sum_{i=\gamma_1 N+1}^{\bar{\gamma}_1 N} \boldsymbol{\epsilon}_i\right\|_1 &= \sup_{\gamma_1\in\bar{\Gamma}_1}\sum_{j=1}^{D}\left|\frac{1}{N}\sum_{i=\gamma_1 N+1}^{\bar{\gamma}_1 N} \epsilon_{i,j}\right| \\ &\le \sum_{j=1}^{D}\sup_{\gamma_1\in\bar{\Gamma}_1}\left|\frac{1}{N}\sum_{i=\gamma_1 N+1}^{\bar{\gamma}_1 N} \epsilon_{i,j}\right|. \end{aligned}$$

Note that, $\forall\ \ > \lambda > 0$,

$$\begin{aligned}&\mathbb{P}\left\{\sup_{\gamma_1\in\bar{\Gamma}_1}\left|\frac{1}{N}\sum_{i=\gamma_1 N+1}^{\bar{\gamma}_1 N} \epsilon_{i,j}\right| > \lambda\right\} \\ &= \mathbb{P}\left\{\max_{2\le k\le \bar{\gamma}_1 N}\left|\sum_{i=k}^{\bar{\gamma}_1 N}\frac{\epsilon_{i,j}}{N}\right| > \lambda\right\} \\ &\le \frac{1}{\lambda^2}\sum_{i=2}^{\bar{\gamma}_1 N}\mathbb{V}\left\{\frac{\epsilon_{i,j}}{N}\right\} = \frac{1}{\lambda^2}\frac{1}{N^2}(\bar{\gamma}_1 N - 1)s^2 \to 0 \end{aligned}$$

as $N\to\infty$, where the inequality above is due to the Kolmogorov's inequality. This confirms that the second term of the R.H.S. of (32) indeed converges to 0 in probability as $N\to\infty$.

For the last term on the R.H.S. of (32) that is independent of γ_1, it is easy to verify $\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\epsilon}_i^{\mathrm{T}}\boldsymbol{\epsilon}_i - s_0^2 \to 0$ as $N\to\infty$.

Since $\sup_{\gamma_1\in\bar{\Gamma}_1}|\bar{f}_1(\gamma_1) - \bar{F}_1(\gamma_1)|$ is less than or equal to the supreme of the R.H.S. of (32), and the R.H.S. of (32) converges to 0 in probability as $N\to\infty$, we have the conclusion that $\sup_{\gamma_1\in\bar{\Gamma}_1}|\bar{f}_1(\gamma_1) - \bar{F}_1(\gamma_1)| \xrightarrow{\mathrm{P}} 0$ as $N\to\infty$. □

For any $\bar{\gamma}_1 \le \gamma_1 \le 1$, we can derive a formula similar to (31) and a result similar to Lemma 2, based on which, the convergence of $\sup_{\gamma_1\in\{\bar{\gamma}_1, \bar{\gamma}_1+1/N, ..., 1\}}|\bar{f}_1(\gamma_1) - \bar{F}_1(\gamma_1)| \xrightarrow{\mathrm{P}} 0$ as $N\to\infty$ can be established in a similar way.

Recall that γ_1^* is the minimizer of $\bar{F}_1(\gamma_1)$, and $\hat{\gamma}_1$ is the minimizer of $\bar{f}_1(\gamma_1)$. We have $\bar{f}_1(\hat{\gamma}_1) \le \bar{f}_1(\gamma_1^*)$ and $\bar{F}_1(\gamma_1^*) \le \bar{F}_1(\hat{\gamma}_1)$. Recall that $\bar{f}_1(\gamma_1) \xrightarrow{\mathrm{P}} \bar{F}_1(\gamma_1)$ uniformly for all γ_1. We have

$$\begin{aligned}\bar{f}_1(\hat{\gamma}_1) &= \bar{F}_1(\hat{\gamma}_1) + o_p(1) \\ &\ge \bar{F}_1(\gamma_1^*) + o_p(1) = \bar{f}_1(\gamma_1^*) + o_p(1)\end{aligned}$$

where $o_p(1)$ is a term that converges to 0 in probability as $N\to\infty$. Therefore, we must have $\bar{f}_1(\hat{\gamma}_1) = \bar{f}_1(\gamma_1^*) + o_p(1)$.

In addition, as is studied in Proposition 3, $F_1(\tau)$ has the property that it has a unique local minimizer with a small enough β, and thus, so is $\bar{F}_1(\gamma) = F_1(\gamma N)$ for all finite N. Finally, we invoke the following result from [38].

Lemma 3 (Theorem 5.7, [38]): With $\beta\to 0$, if $\sup_{\gamma_1\in\Gamma}|\bar{f}_1(\gamma_1) - \bar{F}_1(\gamma_1)| \xrightarrow{\mathrm{P}} 0$, and $\inf_{\gamma_1: d(\gamma_1,\gamma_1^*)\ge\varepsilon_1}\bar{F}_1(\gamma_1) > \bar{F}_1(\gamma_1^*)$ for every $\varepsilon_1 > 0$, where $d(\gamma_1, \gamma_1^*) = |\gamma_1 - \gamma_1^*|$. Then any sequence of estimators $\hat{\gamma}_1$ with $\bar{f}_1(\hat{\gamma}_1) \le \bar{f}_1(\gamma_1^*) + o_p(1)$ converges in probability to γ_1^*.

Therefore, we have $\hat{\gamma}_1 \to \gamma_1^*$ in probability as $N\to\infty$.

APPENDIX C
PROOF OF PROPOSITION 3, 4, AND 5

1. Proof of Proposition 3

Following the signal model (1), under $d_j = d_{j+1} = 0$, $s_j = s_{j+1} = s$, we have

$$\mathbf{x}_i = \begin{cases}\boldsymbol{\mu}_j + \boldsymbol{\epsilon}_i, & \tau_{k-1} < i \le t_j \\ \boldsymbol{\mu}_{j+1} + \boldsymbol{\epsilon}_i, & t_j < i \le \tau_{k+1}.\end{cases} \tag{33}$$

Case 1: Consider $\tau_{k-1} < i \leq t_j$. Using (33), one can compute the expectation of $f_k(\tau)$ and then take the difference, i.e., $\triangle F_k(\tau) = F_k(\tau) - F_k(\tau-1)$ can be computed as

$$\triangle F_k(\tau) = \frac{1}{N}\|\boldsymbol{\mu}_j - \boldsymbol{\mu}_{j+1}\|_2^2 u(\tau, t_j, \beta) + \frac{1}{N}s_0^2\gamma(\tau,\beta) \quad (34)$$

where $u(\tau, t_j, \beta)$ and $\gamma(\tau, \beta)$ are functions w.r.t. the segmentation variable τ, the segment boundary t_j, and the parameter β.

To simplify the expression (34), consider the definition of $\sigma_\beta(x) = \sigma((x-1/2)/\beta)$ based on the sigmoid function $\sigma(x) = (1+\exp(-x))^{-1}$. It follows that

$$\begin{cases} 1-\varepsilon < \sigma_\beta(i-\tau) < 1, & i \geq \tau+1 \\ 0 < \sigma_\beta(i-\tau) < \varepsilon, & i \leq \tau \end{cases} \quad (35)$$

for $\beta < (2\ln(1/\varepsilon - 1))^{-1}$. Using the bounds in (35), there exist positive constants $B_1, B_2 < \infty$, such that $u(\tau, t_j, \beta) \leq \varepsilon B_1 - B_2$.

Likewise, using (35), the term $\gamma(\tau,\beta)$ can be upper bounded as $\gamma(\tau,\beta) \leq \varepsilon(4 + \frac{(\tau_{k+1}-\tau_{k-1})^2}{\tau_{k+1}-\tau_{k-1}-1})$. As a result, there exist constants $C_1, C_2 < \infty$, such that the difference in (34) $\triangle F_k(\tau) < \varepsilon C_1 - C_2 < 0$ for a small enough β.

Case 2: Now, consider $t_j < i \leq \tau_{k+1}$. The derivation for a lower bound of $\triangle F_k(\tau)$ in (34) is similar to the derivation of the upper bound of $\triangle F_k(\tau)$ in Case 1 for $\tau_{k-1} < i \leq t_j$. The lower bound of $u(\tau, t_j, \beta)$ is given by $B_4 - \varepsilon B_3$ for some positive and finite B_3, B_4. In addition, the lower bound of $\gamma(\tau,\beta)$ is $\gamma(\tau,\beta) \geq -\frac{1}{2}(\tau_{k+1}-\tau_{k-1})\varepsilon$. As a result, there exist constants $C_1', C_2' < \infty$, such that $\triangle F_k(\tau) > C_2' - \varepsilon C_1' > 0$ for a small enough β.

Combining the above two cases yields the results of Proposition 3.

2. Proof of Proposition 4

Under $d_k = 0$, $s_k = s$, and no partition index in $(\tau_{k-1}, \tau_{k+1}]$, it corresponds to the case where there exists a t_j in $\{t_1, t_2, \ldots, t_{K-1}\}$ such that $t_j \geq t_{k+1}$. Thus, the derivation of $\triangle F_k(\tau)$ follows (31) by replacing t_j with τ_{k+1}. Then, following the same derivation as in Appendix C-1, one can easily arrive at $\triangle F_k(\tau) < \varepsilon s^2 \frac{D(\tau_{k+1}-\tau_{k-1})^2}{(\tau-\tau_{k-1})(\tau_{k+1}-\tau)}$, and $\triangle F_k(\tau) > -\varepsilon s^2 \frac{D(\tau_{k+1}-\tau_{k-1})^2}{(\tau_{k+1}-\tau+1)(\tau-\tau_{k-1}-1)}$, which are bounded since $\tau \in (\tau_{k-1}, \tau_{k+1}]$. As a result, $|\triangle F_k(\tau)| < \varepsilon s^2 C_0$, for some finite constant $C_0 > 0$.

3. Proof of Proposition 5

Following the signal model (1) under $d_k = 0$, $s_k = s$ for any $k \in \{1, 2, ..., K\}$, where $K \geq 2$, we have

$$\mathbf{x}_i = \boldsymbol{\mu}_k + \boldsymbol{\epsilon}_i, t_{k-1} < i \leq t_k \quad (36)$$

Case 1: Consider $t_{k-1} < \tau \leq t_j$. Define $\boldsymbol{\eta}(o,l) = \frac{1}{t_l - t_{o-1}}\sum_{a=o}^{l}(t_a - t_{a-1})\boldsymbol{\mu}_a$ as the mean of the sample located in the kth subspace with $s_k = s$, and $d_k = 0$, $k \in [o, l]$. Using (36), one can compute the expectation of $f_k(\tau)$ and then take the difference, i.e., $\triangle F_k(\tau) = F_k(\tau) - F_k(\tau-1)$. Using the bounds in (35), there exist constants C_3, C_4, such that $\triangle F_k(\tau)$ is upper bounded by $\triangle F(\tau) < \varepsilon C_3 - C_4$, which is smaller than zero if β is small enough, and $\|\boldsymbol{\mu}_j - \boldsymbol{\eta}(j+1, j+J+1)\|_2^2 \neq 0$, i.e., $\boldsymbol{\mu}_j, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_{j+J+1}$ are linearly independent.

Case 2: Now, consider $t_{j+J} < \tau \leq \tau_{k+1}$. The derivation for a lower bound of $\triangle F_k(\tau)$ is similar to the derivation of the upper bound of $\triangle F_k(\tau)$ in Case 1 for $t_{j+J} < \tau \leq \tau_{k+1}$. Using the bounds in (35), there exist constants C_3', C_4', such that $\triangle F_k(\tau)$ is lower bounded by $\triangle F(\tau) > -\varepsilon C_3' + C_4'$, which is larger than zero if β is small enough and $\|\boldsymbol{\eta}(j, j+J) - \boldsymbol{\mu}_{j+J+1}\|_2^2 \neq 0$.

Combining the above two cases, if $\boldsymbol{\mu}_j, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_{j+J+1}$ are linear independent, we have $\triangle F_k(\tau) > 0$ for $\tau_{k-1} + 1 < \tau \leq t_j$ and $\triangle F_k(\tau) < 0$ for $t_{j+J} < \tau \leq \tau_{k+1}$ with a small enough.

Appendix D
Proof of Lemma 1

Since there exists at least one index $t_{k'} \in \{t_1, t_2, ..., t_{K-1}\}$ in the cluster $\tilde{\mathcal{C}}_j^{(m,k)}$, it follows that there are at least one such index $t_{k'}$ within the interval $(\tau_{j-1}^{(m,k,j)}, \tau_{j+1}^{(m,k,j)}]$. It thus follows from Proposition 2 and 5 that for any $\varepsilon > 0$, there exists a small enough β and some finite constants C and $C' > 0$, such that

$$\begin{aligned} \triangle F_*^{(m,k,j)} &= F_j\left(\tau_{j-1}; \boldsymbol{\tau}_{-j}^{(m,k,j)}\right) - F_j\left(\tau_j^*; \boldsymbol{\tau}_{-j}^{(m,k,j)}\right) \\ &> -\varepsilon C + C' \end{aligned}$$

which is positive from a small enough ε (hence, small enough β). For the cluster $\tilde{\mathcal{C}}_{j'}^{(m,k)}$, since there is no partition such index $t_{k'}$ in the interval $(\tau_{j'-1}^{(m,k,j')}, \tau_{j'+1}^{(m,k,j')}]$, Proposition 4 suggests that $|F_{j'}(\tau; \boldsymbol{\tau}_{-j'}^{(m,k,j')}) - F_{j'}(\tau_j^*; \boldsymbol{\tau}_{-j'}^{(m,k,j')})| < \varepsilon B$ which leads to $|\triangle F_*^{(m,k,j')}| < \varepsilon B$.

As a result, for a small enough β (hence, small enough ε), we must have $\triangle F_*^{(m,k,j)} > \triangle F_*^{(m,k,j')}$.

Appendix E
Proof of Theorem 1

If $\exists k \in \{1, 2, ...K-1\}$ satisfying $\tau_k^{(m)} \neq t_k$, then for any partition assignment $\{\tau_1, \tau_2, ..., \tau_{k-1}, \tau_{k+1}, ..., \tau_{K-1}\}$, there always exists $l \in \{1, 2, ..., K-1\}$ such that the interval (τ_{l-1}, τ_l) contains at least one of $\{t_k\}_{k=1}^{K-1}$. We first prove that the following two cases will not occur if $\tilde{F}(\boldsymbol{\tau}^{(m+1)}) = \tilde{F}(\boldsymbol{\tau}^{(m)})$.

1) If there exists $l \in \{1, 2, ..., K-1\}$ such that the interval (τ_{l-1}, τ_{l+1}) contains none of $\{t_k\}_{k=1}^{K-1}$, then, there must exist a interval (τ_{a-1}, τ_a), $a \neq l, l+1$ containing at least one of $\{t_k\}_{k=1}^{K-1}$. According to Lemma 1, we will obtain *a lower total cost* $\tilde{F}(\cdot)$ if we replace the partition τ_l with one of the partition in (τ_{a-1}, τ_a). Thus, the algorithm will not stop if there exists $l \in \{1, 2, ..., K-1\}$ such that the interval (τ_{l-1}, τ_{l+1}) contains none of $\{t_k\}_{k=1}^{K-1}$.

2) If there exists $l \in \{1, 2, ..., K-1\}$ such that the interval (τ_{l-1}, τ_{l+1}) contains at least two of $\{t_k\}_{k=1}^{K-1}$. Then, there must exist an interval among $(\tau_{i-1}, \tau_i]$, $i \in \{1, 2, ...l-1, l+2, ..., K\}$ containing none of $\{t_k\}_{k=1}^{K-1}$. This is conflicted with the conclusion of the above case 1), i.e., the interval $(\tau_{l-1}, \tau_l]$

and $[\tau_l, \tau_{l+1})$, for any $l \in \{1, 2, ...K-1\}$, aways exist at least one of $\{t_k\}_{k=1}^{K-1}$ if the algorithm is convergent. So, there exists none interval containing at least two of $\{t_k\}_{k=1}^{K-1}$ if $\tilde{F}(\boldsymbol{\tau}^{(m+1)}) = \tilde{F}(\boldsymbol{\tau}^{(m)})$.

Based on the above analysis, we have the conclusion that the interval (τ_{l-1}, τ_{l+1}) for any $l \in \{1, 2, ...K-1\}$ contains exactly one of $\{t_k\}_{k=1}^{K-1}$ if $\tilde{F}(\boldsymbol{\tau}^{(m+1)}) = \tilde{F}(\boldsymbol{\tau}^{(m)})$.

However, when Algorithm 1 converges, Proposition 3 implies that $\tau_l = t_l$ if t_l is the only $\{t_k\}_{k=1}^{K-1}$ in (τ_{l-1}, τ_{l+1}). With this, we conclude that if $\tau_k^{(m+1)} = \tau_k^{(m)} = t_k$ for all k when Algorithm 1 has converged.

REFERENCES

[1] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2568–2599, Thirdquarter 2019.

[2] X. Chen, H. Li, C. Zhou, X. Liu, D. Wu, and G. Dudek, "Fidora: Robust WiFi-based indoor localization via unsupervised domain adaptation," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9872–9888, Jun. 2022.

[3] N. Guo, S. Zhao, X.-P. Zhang, Z. Yao, X. Cui, and M. Lu, "New closed-form joint localization and synchronization using sequential one-way TOAs," *IEEE Trans. Signal Process.*, vol. 70, pp. 2078–2092, 2022.

[4] Z. Hajiakhondi-Meybodi, A. Mohammadi, M. Hou, and K. N. Plataniotis, "DQLEL: Deep Q-learning for energy-optimized LoS/NLoS UWB node selection," *IEEE Trans. Signal Process.*, vol. 70, no. 4, pp. 2532–2547, 2022.

[5] Y. Sun, K. Ho, L. Gao, J. Zou, Y. Yang, and L. Chen, "Three dimensional source localization using arrival angles from linear arrays: Analytical investigation and optimal solution," *IEEE Trans. Signal Process.*, vol. 70, pp. 1864–1879, 2022.

[6] A. Pandey, P. Tiwary, S. Kumar, and S. K. Das, "FadeLoc: Smart device localization for generalized $\kappa - \mu$ faded IoT environment," *IEEE Trans. Signal Process.*, vol. 70, no. 4, pp. 3206–3220, 2022.

[7] K. S. V. Prasad and V. K. Bhargava, "RSS localization under Gaussian distributed path loss exponent model," *IEEE Wireless Commun. Lett.*, vol. 10, no. 1, pp. 111–115, 2020.

[8] S. Phoemphon, C. So-In, and N. Leelathakul, "Fuzzy weighted centroid localization with virtual node approximation in wireless sensor networks," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4728–4752, Dec. 2018.

[9] S. Shrestha, X. Fu, and M. Hong, "Deep spectrum cartography: Completing radio map tensors using learned neural models," *IEEE Trans. Signal Process.*, vol. 70, pp. 1170–1184, 2022.

[10] X. Du, X. Liao, M. Liu, and Z. Gao, "CRCLoc: A crowdsourcing-based radio map construction method for WiFi fingerprinting localization," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 12 364–12 377, Jul. 2022.

[11] X. Wang, X. Wang, S. Mao, J. Zhang, S. C. Periaswamy, and J. Patton, "Indoor radio map construction and localization with deep Gaussian processes," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 11238–11249, Nov. 2020.

[12] K. M. Chen and R. Y. Chang, "A comparative study of deep-learning-based semi-supervised device-free indoor localization," in *Proc. IEEE Global Commun. Conf.*, Dec. 2021, pp. 1–6.

[13] J. Gao, D. Wu, F. Yin, Q. Kong, L. Xu, and S. Cui, "MetaLoc: Learning to learn wireless localization," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 12, pp. 3831–3847, Dec. 2023.

[14] Y. Li, Y. Nishikawa, and T. Nobukiyo, "Dividing-and-kriging method for wireless RSS fingerprint based indoor localization," in *Proc. IEEE Global Commun. Conf.*, Dec. 2019, pp. 1–6.

[15] P. Ferrand, A. Decurninge, L. G. Ordonez, and M. Guillaud, "Triplet-based wireless channel charting: Architecture and experiments," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2361–2373, Aug. 2021.

[16] S.-h. Jung, B.-c. Moon, and D. Han, "Unsupervised learning for crowd-sourced indoor localization in wireless networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 11, pp. 2892–2906, Nov. 2016.

[17] C. Wu, Z. Yang, Y. Liu, and W. Xi, "WILL: Wireless indoor localization without site survey," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 4, pp. 839–848, Apr. 2013.

[18] C. Gao and R. Harle, "Semi-automated signal surveying using smartphones and floorplans," *IEEE Trans. Mobile Comput.*, vol. 17, no. 8, pp. 1952–1965, Aug. 2018.

[19] J. Choi, "Sensor-aided learning for Wi-Fi positioning with beacon channel state information," *IEEE Trans. Wireless Commun.*, vol. 21, no. 7, pp. 5251–5264, Jul. 2022.

[20] R. Vidal, "Subspace clustering," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 52–68, Mar. 2011.

[21] A. Collas, F. Bouchard, A. Breloy, G. Ginolhac, C. Ren, and J.-P. Ovarlez, "Probabilistic PCA from heteroscedastic signals: Geometric framework and application to clustering," *IEEE Trans. Signal Process.*, vol. 69, pp. 6546–6560, 2021.

[22] J. Fan, C. Yang, and M. Udell, "Robust non-linear matrix factorization for dictionary learning, denoising, and clustering," *IEEE Trans. Signal Process.*, vol. 69, pp. 1755–1770, 2021.

[23] Z. Du, X. Wang, G. Zhou, and Q. Wang, "Fast and unsupervised action boundary detection for action segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 3323–3332.

[24] S. Tierney, J. Gao, and Y. Guo, "Subspace clustering for sequential data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1019–1026.

[25] G. Xia, P. Xue, H. Sun, Y. Sun, D. Zhang, and Q. Liu, "Local self-expression subspace learning network for motion capture data," *IEEE Trans. Image Process.*, vol. 31, no. 4, pp. 4869–4883, 2022.

[26] Z. Zhou et al., "Sequential order-aware coding-based robust subspace clustering for human action recognition in untrimmed videos," *IEEE Trans. Image Process.*, vol. 32, no. 4, pp. 13–28, 2023.

[27] T. Zhou, H. Fu, C. Gong, J. Shen, L. Shao, and F. Porikli, "Multi-mutual consistency induced transfer subspace learning for human motion segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10277–10286.

[28] Z. Xing, J. Chen, and Y. Tang, "Integrated segmentation and subspace clustering for RSS-based localization under blind calibration," in *Proc. IEEE Global Commun*. Conf., 2022, pp. 5360–5365.

[29] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Comput.*, vol. 11, no. 2, pp. 443–482, 1999.

[30] E. Tohidi, J. Chen, and D. Gesbert, "Sensor selection for model-free source localization: Where less is more," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Barcelona, Spain, May 2020, pp. 4831–4835.

[31] J. Wang, P. Urriza, Y. Han, and D. Cabric, "Weighted centroid localization algorithm: Theoretical analysis and distributed implementation," *IEEE Trans. Wireless Commun.*, vol. 10, no. 10, pp. 3403–3413, Oct. 2011.

[32] L. Bai, J. Liang, and Y. Zhao, "Self-constrained spectral clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 5126–5138, Apr. 2023.

[33] S. Sadowski, P. Spachos, and K. N. Plataniotis, "Memoryless techniques and wireless technologies for indoor localization with the Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 10996–11005, Nov. 2020.

[34] J. Hu, D. Liu, Z. Yan, and H. Liu, "Experimental analysis on weight K-nearest neighbor indoor fingerprint positioning," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 891–897, Feb. 2019.

[35] Z.-l. Wu, C.-h. Li, J. K.-Y. Ng, and K. R. Leung, "Location estimation via support vector regression," *IEEE Trans. Mobile Comput.*, vol. 6, no. 3, pp. 311–321, Mar. 2007.

[36] L. Hao, B. Huang, B. Jia, and G. Mao, "DHCLoc: A device-heterogeneity-tolerant and channel-adaptive passive WiFi localization method based on DNN," *IEEE Internet Things J.*, vol. 9, no. 7, pp. 4863–4874, Apr. 2022.

[37] C.-Y. Chen, I. Alexander, C. Lai, P.-Y. Wu, and R.-B. Wu, "Optimization and evaluation of multidetector deep neural network for high-accuracy Wi-Fi fingerprint positioning," *IEEE Internet Things J.*, vol. 9, no. 16, pp. 15204–15214, Aug. 2022.

[38] A. W. Van der Vaart, *Asymptotic Statistics*, vol. 3. Cambridge, U.K.: Cambridge Univ. Press, 2000.