Understanding Localization by a Tailored GPT

Xiaopeng Zhao†, Guosheng Wang†, Zhenlin An, Qingrui Pan, Lei Yang*
Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR
Shenzhen Research Institute of PolyU, Shenzhen, China
{zhao,an,pan,young}@tagsys.org,guoshwang@polyu.edu.hk

ABSTRACT

Conventional deep learning approaches for indoor localization often suffer from their reliance on high-quality training samples and display limited adaptability across varied scenarios. To address these challenges, we repurpose the Transformer model, celebrated for its profound contextual insights, to explore the underlying principles of indoor localization. Our microbenchmark results compellingly demonstrate the superiority of our approach, showing improvements of 30% to 70% across a diverse set of 50 scenarios compared to other state-of-the-art methods. In conclusion, we propose a specialized Generative Pre-training Transformer (GPT) variant, termed LocGPT, configured with 36 million parameters that are tailored to facilitate transfer learning. By fine-tuning this pre-trained model, we achieve near-par accuracy using merely half the conventional dataset, thereby heralding a pioneering stride in transfer learning within the indoor localization domain.

CCS CONCEPTS

• **Networks** → **Location based services**; **Mobile networks**.

KEYWORDS

Internet-of-Things, Wireless Localization, Deep Learning

ACM Reference Format:
Xiaopeng Zhao, Guosheng Wang, Zhenlin An, Qingrui Pan, Lei Yang. 2024. Understanding Localization by a Tailored GPT. In *The 22nd Annual International Conference on Mobile Systems, Applications and Services (MOBISYS '24), June 3–7, 2024, Minato-ku, Tokyo, Japan.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3643832.3661869

1 INTRODUCTION

In recent years, there has been an emerging interest in the construction of precise RF-based indoor localization systems that supplement the final hundred meters where GPS lacks reach. Numerous strides [1–19] have been conducted for this purpose in the past two decades, culminating in a rather promising level of accuracy. High-precision indoor localization has the potential to unlock a plethora of key applications encompassing indoor navigation, augmented reality, location-conscious pervasive computing, advertising, and social networking, among others. As a result, the practice of tracking IoT devices within structural enclosures has emerged as an expanding commercial interest.

However, only a small fraction of the proposed solutions have successfully transitioned to real-world implementations due to unforeseen challenges encountered across different deployment scenarios, some of which are discussed here. (1) *Hardware diversity*: the heterogeneity in hardware arising from circuitry specifications can lead to undesirable errors in RF signal measurements, thereby skewing the input to a localization algorithm with several unknown offsets. (2) *Spatial diversity*: the RF signal experiences spatial fluctuations due to uneven electromagnetic field distribution. (3) *Multipath effect*: amidst an unpredictable, intricate, and dynamic wireless landscape, RF signals suffer reflection from a multitude of static or mobile obstacles. This multipath effect has become a formidable barrier to indoor localization, especially when the localization algorithm is highly reliant on line-of-sight (LoS) propagation.

Indoor localization essentially boils down to solving a non-linear optimization problem, that is, finding the optimal position at which the RF device can generate signals that closely match those received by the base stations (see §2). This challenge squarely falls within the realm of machine learning. In the wake of the deep learning (DL) surge, some studies [20–24] have begun to exploit DL advancements to confront these challenges. These works have successfully showcased that DL surpasses traditional localization algorithms in terms of accuracy and stability. The success of previous attempts has inspired our investigation into the potential of DNNs, while addressing the following two main limitations:

- Current methods are largely reliant on supervised learning, in which the efficacy is intimately tied to the volume and quality of training samples (i.e., accurate location labels). To this end, existing methods require another set of high-precision optical localization systems, such as OptiTrack [21] or Lidar [25], to amass the training datasets. This necessity adds significant complexity to deployment. Additionally, the lack of high-quality, large-scale datasets specifically tailored for this purpose further exacerbates the challenges in the community.
- Current methods face the challenge of non-transferability. In particular, the training dataset is intimately tied to the specific layout of a scene, which means that the value of datasets collected in other environments or at earlier times drastically diminishes. This limitation hampers the universal deployment of a trained model across diverse environments, thus impeding scalability. This constrained adaptability represents a significant drawback in existing approaches, highlighting the demand for techniques with enhanced versatility across multiple scenarios.

To address the above limitations, we propose the Transformer-based localization (TBL) model, distinguished by its proficiency in contextual understanding and ability to discern relationships

*Corresponding author. †Co-primary student authors.

MOBISYS '24, June 3–7, 2024, Minato-ku, Tokyo, Japan

ACM ISBN 979-8-4007-0581-6/24/06…$15.00
https://doi.org/10.1145/3643832.3661869

between sequential elements. Our model is distinct from previous works [26, 27], which utilized individual components of the Transformer model (e.g., encoders or decoders). Instead, our study exploits the full potential of the Transformer architecture, aiming to understand both the contextual scenes and the intricate relationships between phase measurements and positions. To unleash the generalizability of the TBL model across scenes, we introduce LocGPT, a tailored variant of the Generative Pre-training Transformer (GPT) model for localization. LocGPT, which serves as the pre-training version for the TBL model, is pre-trained on 1.4 million localization data samples. Like linguistic models such as ChatGPT that generate text from provided contexts, LocGPT "generates" locations by processing contextual RF signals and previously determined positions. Specifically, we make the following efforts:

• Focusing on antenna array-based triangulation, we present an inventive hierarchical neural network architecture adapted from the Transformer model. The architecture comprises multiple A-Subnetworks and a singular T-Subnetwork. Each A-Subnetwork ingests phase measurements acquired from a specific antenna array as tokens, generating Angle-of-Arrival (AoA) results and directional features. The T-Subnetwork integrates the combined feature vectors along with the historical positions of the target device to generate its current position. The A-Subnetworks and T-Subnetwork are built using Transformer encoders and decoders, respectively. This arrangement harmonizes ideally with the core architecture of the original Transformer.

• Second, we propose a semi-supervised training methodology. Past DL-based localization systems [20–24] require the use of absolute location labels to guide the neural network's training in a supervised manner. As previously mentioned, this strict requirement for label acquisition necessitates the use of a secondary high-precision localization system, thereby increasing deployment costs and operational complexity. In this work, we introduce two novel loss functions that are designed to minimize the discrepancy between the actual and predicted distances of two sampled locations. Without the need for absolute position labels, the network model can predict two locations whose distances closely align with the ground truth. This subtle modification allows the collection of RF signals using a pair of fixed target devices with a known separation.

• Third, the current datasets for indoor localization face several shortcomings in terms of data scale, diversity of scenes, label accuracy, and comprehensive coverage. These shortcomings prevent them from serving as a universal resource for benchmarking, training, and transfer learning, irrespective of the datasets' quality or quantity. Thus, to address this pressing need, we introduce Ray, a first-ever 3D indoor localization dataset on the scale of millions. We have established an open benchmark database for localization, comprising approximately 1.6 million samples across 50 scenarios, spanning RFID, Wi-Fi, and BLE modalities. Each entry in the database encompasses comprehensive data, such as IQ signals, alongside the mm-level location labels. With the aim of facilitating the development of robust and versatile indoor localization solutions, we intend to make this significant resource freely accessible.

• Finally, the micro-benchmark results compellingly demonstrate that our proposed Transformer model consistently outperforms other state-of-the-art solutions, with improvements ranging between 30% and 70%. In light of these promising outcomes, we are excited to release a pre-training version of the Transformer-based localization (TBL) model, LocGPT, which is equipped with 36 million parameters. As a robust initial model for localization, LocGPT can be effectively fine-tuned for a specific scene using less than 50% of the sample data ordinarily required. This design has the potential to significantly lower the barriers to adoption, making high-accuracy indoor localization more accessible across a wide range of applications and environments.

Summary. Our main contributions are fourfold. First, we repurpose the Transformer model to achieve the antenna array-based triangulation. Second, we propose a semi-supervised training methodology to reduce deployment costs and operational complexity. Third, we present Ray, a large-scale dataset. Finally, we release the first pre-training model LocGPT to achieve widespread transfer learning in the indoor localization domain.

2 PRELIMINARY

This section introduces the background knowledge about the spatial spectrum and the triangulation-based localization.

2.1 Spatial Spectrum

Let us consider an antenna array equipped with $\sqrt{K}\times\sqrt{K}$ uniformly spaced elements, with a spacing of half a wavelength between each pair of adjacent elements. We can establish a local Cartesian coordinate system for the array by designating the element E_0 at a corner as the origin. With E_0 chosen as a reference point, the relative power of projecting the received signal into the direction of (α, β) can be computed as follows:

$$\mathcal{P}(\alpha,\beta) = \frac{1}{K}\left|\sum_{k=1}^{K} w_k(\alpha,\beta)\cdot \tilde{s}_k^*\right| \tag{1}$$

where $w_k(\alpha,\beta) = e^{\mathrm{J}\Delta\theta_k}$ denotes the complex weight assigned for steering a beam towards a specific angle of (α,β), which represent the azimuthal and elevation angles, respectively. The term $\tilde{s}_k = \tilde{s}_k/\tilde{s}_0 = e^{\mathrm{J}\Delta\tilde{\theta}_k}$ is the ratio of signal $\tilde{s}_k$ and $\tilde{s}_0$, which are received by E_k and E_0. The $\Delta\theta_k$ and $\Delta\tilde{\theta}_k$ refer to their theoretical and actual phase difference, respectively. The sum aggregates the relative power across all K pairs of elements. When $\Delta\tilde{\theta}_k$ aligns with $\Delta\theta_k$ (i.e., the signal does originate from the direction of (α,β)), the normalized relative power $\mathcal{P}(\alpha,\beta)$ should reach its maximum. The relative power at the direction ω can then be represented as a vector in the following manner:

$$\mathcal{P}(\omega) = \left[w_1(\omega), w_2(\omega), \cdots, w_K(\omega)\right]\left[\tilde{s}_1, \tilde{s}_2, \cdots, \tilde{s}_K\right]^T \tag{2}$$

where T stands for the transpose operation. Next, a heatmap can be produced to illustrate the relative power at N potential directions from which the received RF signal may have originated. We term this heatmap a *spatial spectrum*, which is formalized as:

$$\begin{aligned}\Omega &= \begin{bmatrix}\mathcal{P}(\omega_1) & \mathcal{P}(\omega_2) & \cdots & \mathcal{P}(\omega_N)\end{bmatrix}^T + \begin{bmatrix}Z_1 & Z_2 & \cdots & Z_N\end{bmatrix}^T \\ &= \mathbf{W}\mathbf{S} + \mathbf{Z}\end{aligned} \tag{3}$$

where $\mathbf{\Omega}$, $\mathbf{W}$, $\mathbf{S}$, and $\mathbf{Z}$ signify the spatial spectrum, the weight matrix (aka steering matrix), the received signals, and the noise, respectively. The value of N can be customized based on the required angle resolution. Hence, $N = 360\times 90$ is appropriate if a one-degree resolution is acceptable. The spatial spectrum should reach a peak in the direction from which the RF signal originates. Unfortunately,

the multipath effect can cause the RF signal to arrive from multiple directions, which may result in multiple peaks in the spectrum, thus leading to ambiguity. In such a case, it becomes challenging to determine the true source of the signal. Formally, the direction is calculated by solving the following optimization problem:

$$\omega^* = \underset{\omega}{\text{argmax}}\ \Omega = \underset{\omega}{\text{argmax}}\ (\mathbf{WS} + \mathbf{Z}) \tag{4}$$

The above equation is to find a direction that can produce the real measurements.

2.2 Triangulation

Each array can determine a distinct direction. The device's position is discerned either at the intersection of two such directions or the centroid of the area created by overlapping multiple directions. This technique for identifying location is referred to as *triangulation*. This procedure can be formalized to solve the following optimization problem:

$$p^* = \underset{p}{\text{argmin}} \sum_{i=1}^{G} \mathbf{d}(p, l(\mathbf{R}_i \cdot \vec{\omega}_i^* + \mathbf{O_i})) \tag{5}$$

where $\mathbf{d}(\cdot)$ is the Euclidean distance between a point p and a line l, and G is the number of antenna arrays. In addition, the $l(\mathbf{R}_i \cdot \vec{\omega}_i^* + \mathbf{O}_i)$ represents the direction (i.e., a ray) estimated by the i^{th} antenna array, and $\mathbf{R}_i$ is the transform matrix and $\mathbf{O}_i$ is the coordinate of the array.

2.3 Why Deep Learning Helps?

In essence, the ultimate goal of indoor localization is to resolve the two intricate non-linear optimization problems expressed in Eqns. 4 and 5. Theoretically, the DNN can fit any function similar to ours through the use of a large number of training samples. DNNs can autonomously extract and leverage hidden variables from input data, enabling them to capture complex signal features and environmental factors. This extraction process is robust against ambient interferences, which can be advantageous when dealing with the diverse and dynamic conditions present in indoor scenarios. Therefore, given the inherent capabilities of DNNs and the great success of DL techniques in related tasks demonstrated in previous research [20–24], we posit that DNN holds significant potential for advancing the field of indoor localization.

3 OVERVIEW

Antenna arrays, commonly used in high-capacity communications, such as MIMO and beamforming, are now being adapted for indoor localization by leading standard organizations, as seen in Wi-Fi 802.11 az and Bluetooth 5.1+. Their strength lies in enhancing signal strength and localization accuracy through spatial diversity and directional signal propagation. In line with this, our system uses antenna arrays to pinpoint various wireless terminals, offering a technology-agnostic solution.

We also harness the power of the Transformer to address the problem of indoor localization. The Transformer, composed of encoders and decoders, is originally designed for natural language processing (NLP) tasks, such as language translation. It has now become the de facto standard model for NLP that is widely adopted for many applications, such as ChatGPT [28] and LLaMA [29]. On a broader scale, our entire system operates analogously to the task of sequence-to-sequence language translation. The Transformer encoders ingest a sequence of phase measurements (recorded by antennas), whereas the Transformer decoders convert this sequence to a sequence of positions. Toward this objective, we first introduce how the Transformer has been repurposed to serve our localization task in §4. Then, we demonstrate the feasibility and effectiveness of the proposed model via a comprehensive micro-benchmark in §6 across the high-quality dataset (see §5). Finally, we introduce the first pre-training Transformer model, LocGPT, in §7 and evaluate it in §8.

4 TRANSFORMER-BASED LOCALIZATION

This section provides a detailed account of the repurposed Transformer for the localization task. The approach is termed "Transformer-based Localization" (TBL).

4.1 Network Architecture

As illustrated in Fig. 1, the architecture of our model is underpinned by the Transformer neural network, which mainly consists of two types of subnetworks: the AoA subnetwork (A-Subnetwork) and the triangulation subnetwork (T-Subnetwork). **(1) A-Subnetwork**: Each A-Subnetwork is dedicated to fit Eqn. 4 by accepting the phase values collected by an antenna array. An A-Subnetwork is realized by exploiting the powerful capabilities of the Transformer encoders, which are renowned for their proficiency in discerning intricate patterns and hierarchically structured representations within the input data. The model architecture accommodates an arbitrary number of A-Subnetworks, each corresponding to a distinct base station. These A-Subnetworks operate independently of one another, transforming their respective inputs into an AoA and corresponding directional features. These outputs then collectively serve as the input to the T-Subnetwork. **(2) T-Subnetwork**: Equipped with Transformer decoders, the T-Subnetwork collates the directional and feature vectors provided by the A-Subnetworks to yield a precise estimate of the target's location. Namely, it is designed to fit Eqn. 5 mathematically. Owing to the sequence generation capabilities of the Transformer decoders, the T-Subnetwork also effectively incorporates historical location data, thereby refining the accuracy of location predictions. In the following, we delve into a detailed examination of each subnetwork.

4.2 A-Subnetwork

Inspired by the Transformer encoder stack, the A-Subnetwork is designed to estimate the AoA of the RF signals that are transmitted from the target device. More formally, the i^{th} A-Subnetwork denoted by $\mathcal{A}_i$ is defined as follows:

$$\mathcal{A}_i : \left(A_i, \{ \underbrace{(t_j, \Phi_j)}_{\text{Current}}, \underbrace{(t_{j-1}, \Phi_{j-1}), \ldots, (t_{j-M}, \Phi_{j-M})}_{\leq M \text{ recent measurements}} \} \right) \rightarrow (\vec{\omega}_i, \mathcal{F}_i) \tag{6}$$

where A_i denotes the location of the i^{th} station, and (t_j, Φ_j) represents the phase values $\Phi_j = \{\phi_j^1, \phi_j^2, \cdots, \phi_j^K\}$, which are measured by the station equipped with K antennas at the time t_j. The subnetwork takes the required current phase measurement Φ_j and optional recent historical measurements, namely $\Phi_{j-1}, \Phi_{j-2}, \ldots, \Phi_{j-M}$. The total number of measurements does not exceed $M + 1$. The direction vector $\vec{\omega}_i$ indicates the estimated AoA, while $\mathcal{F}_i$ is the

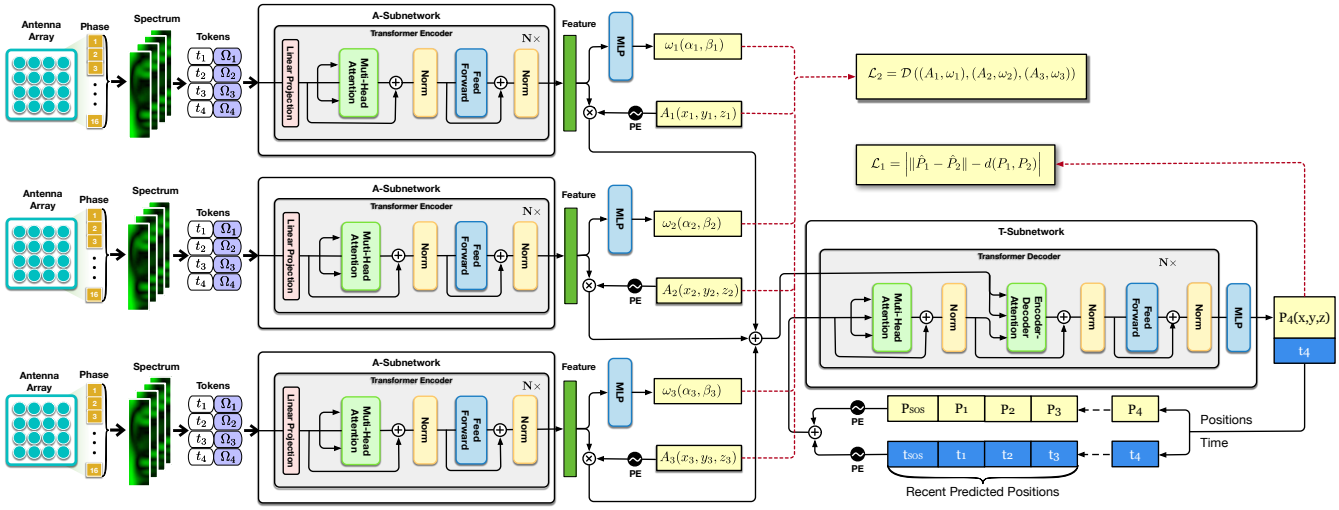

Fig. 1: TBL Network Architecture. It consists of two layers of neural networks. The first layer is the A-Subnetwork assigned to each base station with the purpose of estimating the AoA relative to the current antenna array. The second layer is the T-Subnetwork aiming to pinpoint the final position by feeding the AoA feature from three A-Subnetworks.

feature vector that is utilized by the subsequent subnetwork. Next, we elaborate on the subnetwork.

4.2.1 Input Representation

Each A-Subnetwork processes a spatial spectrum. We favor the use of spatial spectrum over raw phase values as input due to several key considerations. First, fluctuations in the antenna array size integrated into the base station (e.g., 3×3 or 4×4) can lead to inconsistent input dimensions. Second, the device's operational frequencies may vary (e.g., 800MHz or 2.4GHz), such that even the sample phase value reflects the different distances traversed by the RF signals at varying frequencies. Finally, using raw phase values as inputs fails to consider the structure of the antenna array. For instance, 16 phase values could be measured by arrays of 1×16, 4×4, and 2×8, but this structural information would be lost in the phase sequence. These variations in hardware configurations could undermine the model's universality.

(1) Spectra as Tokens: In contrast, spatial spectra are closely linked with the hardware configurations. Regardless of the hardware setup, the spatial spectra of a consistent size can be generated using Eqn. 3. The size of the spatial spectrum depends solely on the granularity with which we traverse the spatial domain. Thus, we partition the space into 36×10 directions with a granularity of 10° to strike a balance between resolution and computational load. Directions with elevation angles less than 10° should be disregarded due to the limited directionality of directional antennas, which is typically less than 80°. Hence, the spatial spectrum size is standardized to 36×9, which we subsequently reshape into a 324-dimension vector that serves as a token. This can be formally expressed as:

$$\Omega = \left[\mathcal{P}(0^\circ, 0^\circ), \mathcal{P}(0^\circ, 10^\circ), \cdots, \mathcal{P}(350^\circ, 70^\circ), \mathcal{P}(350^\circ, 80^\circ)\right] \quad (7)$$

where $\mathcal{P}(\alpha, \beta)$ denotes the normalized relative power at the direction $\omega(\alpha, \beta)$ (refer to Eqn. 2).

Contrary to previous works [21, 30, 31] that transform spectra into images and employ CNN or AutoEncoder for feature extraction, we opted to preserve a whole spectrum as a single token form for three primary reasons. First, image recognition is a highly time-intensive process. Second, our experience suggests that the partitioning of images (for instance, ViT patches) causes disorganization within the spectrum, leading to training misconvergence. Finally, an image-centric approach tends to fall into the approaches of signature-based localization, in which an image feature corresponds to a specific location, making it extremely susceptible to environmental factors.

(2) Learning from History: It is important to note that phase measurements from the same object are intricately interconnected over time. Given that historical measurements may help eliminate transient interference, we thus assemble the current measurement and M preceding ones into a token sequence that are fed into the subnetwork. Prior measurements are not mandatory. In cases where past measurements are lacking, their corresponding tokens can simply be set to zero. Nevertheless, the maximum number of tokens fed into the subnetwork is set to M. The measurement time could be seen as the "position" of a measurement within the sequence. To incorporate this aspect, we encode the time using the positional encoding scheme as follows.

$$\mathrm{PE}(x) = \left[\sin\left(2^{0/L}\pi x\right), \cos\left(2^{0/L}\pi x\right), \ldots, \sin\left(2^{(L-1)/L}\pi x\right), \cos\left(2^{(L-1)/L}\pi x\right)\right] \quad (8)$$

Originally devised to encode positions within the Transformer model, $\mathrm{PE}(x)$ has evolved to augment the dimension of any number x from one to $2L$. Today, it has been widely used across various DL contexts. In our scenario, we first encode the measurement time using $\mathrm{PE}(t)$ into a 324-dimensional vector, which is then added to the token. Hence, the input for an A-Subnetwork at time t_j can be formally represented as:

$$\mathbf{I}(t_j) = \left[\Omega_j + \mathrm{PE}(t_j - t_j), \Omega_{j-1} + \mathrm{PE}(t_j - t_{j-1}), \ldots, \Omega_{j-M} + \mathrm{PE}(t_j - t_{j-M})\right]$$

where the time is expressed relative to the current time t_j. In addition, $\Omega_j, \ldots, \Omega_{j-M}$ are the spectra obtained at time $t_j, \ldots, t_{j-M}$, i.e., $t_j > t_{j-1} > \cdots > t_{j-M}$. The input consistently comprises $M+1$ tokens. Missing tokens are filled with zeros. Our evaluation advises $M = 5$ in practice.

Discussion: In our approach, positional encoding is applied exclusively to the synchronized timestamps of data from each base station. The spatial spectra derived from each base station naturally serve as a form of "spatial encoding" for phase, as depicted in Section 2.1. Therefore, we omit additional positional encoding for spatial spectra.

4.2.2 Network Body

Each A-subnetwork consists of multiple Transformer encoders. The encoder is a key building block in the Transformer architecture, primarily designed for processing sequential data in NLP tasks. This encoder operates through multiple layers, each consisting of two main components: a self-attention mechanism and a position-wise feed-forward neural network. The self-attention mechanism allows the model to weigh the relevance of each element in a sequence relative to all other elements, thus capturing dependencies regardless of their distance in the sequence. The position-wise feed-forward networks, which are applied independently to each position, involve two linear transformations with a GELU activation in between. Layer normalization and residual connections are employed around both the self-attention and feed-forward sub-layers to stabilize the learning process. Then, these layers are stacked to construct the complete Transformer encoder, which transforms input data into a higher-level feature representation that captures complex patterns within the data.

4.2.3 Network Output

Unlike CNNs, the Transformer retains the dimensionality of the input to ensure consistency and compatibility between layers. This means each layer's input and output dimensions remain the same. As we feed in $M+1$ tokens, the outputs are $M+1$ feature vectors with 324 dimensions, denoted by $\mathcal{F}_i$. An additional Multilayer Perceptron (MLP, a two-layer fully connected network) is appended to decode the features into a directional vector $\vec{\omega}_i(\alpha, \beta)$, which represents the estimated AoA. Considering the location of the station A_i, the direction can be formulated as a ray:

$$\vec{l}_i = A_i + \vec{\omega}_i \cdot u = A_i + \text{MLP}(\mathcal{F}_i) \times u \tag{9}$$

where u is the distance between the A_i and a point on the ray. Given three A-Subnetworks, we finally obtain three rays, namely, $\vec{l}_1$, $\vec{l}_2$, and $\vec{l}_3$.

4.3 T-Subnetwork

Our model incorporates a single T-Subnetwork to compute the final position using the resulting AoAs. A T-Subnetwork denoted by $\mathcal{T}$ can be formally expressed as follows:

$$\mathcal{T} : \Big((A_1, \mathcal{F}_1(t_j)), (A_2, \mathcal{F}_2(t_j)), (A_3, \mathcal{F}_3(t_j)) \Big) \rightarrow P_j(x, y, z) \tag{10}$$

where $P_j(x, y, z)$ is the 3D location of the wireless terminal at time t_j. One might question the necessity for the T-Subnetwork, given that the previous three A-Subnetworks have already provided AoA directions. Theoretically, we could compute the intersection of these three rays using a geometric approach, if they intersect at all. However, our observations reveal that these three rays rarely intersect at a single point in 3D space, as shown in Fig. 2(a). In such scenarios, the goal is to identify a point closest to the three rays, thereby transforming the problem into another optimization task (see Eqn. 5). Notably, recent location estimations can greatly aid in current predictions, especially when tracking a moving target. Therefore, we must devise a strategy to incorporate this motion context into our model. In response to this, employing another neural network, referred to as the T-Subnetwork, has demonstrated its effectiveness in tackling the issues.

4.3.1 Input Representations

As aforementioned, the network should consider two factors: the directional rays and the historically estimated locations. Thus, we have two types of inputs for the T-Subnetwork, which we discuss below.

(1) Input I: Triangulation determines the intersection of the three rays, each originating from a known location and representing a direction. This necessitates the inclusion of both the base station's location $A_i(x_i, y_i, z_i)$ and the directional features $\mathcal{F}_i(t_j)$ into the T-Subnetwork. Specifically, the station's location A_i is first encoded using $\text{PE}(\cdot)$ and then multiplied with the feature $\mathcal{F}_i(t_j)$. Finally, the updated feature vectors from the three A-Subnetworks are combined into a single vector, which is then inputted into the T-Subnetwork. This can be formally represented as:

$$\mathbf{I}_1(t_j) = \text{PE}(A_1) \otimes \mathcal{F}_1(t_j) + \text{PE}(A_2) \otimes \mathcal{F}_2(t_j) + \text{PE}(A_3) \otimes \mathcal{F}_3(t_j) \tag{11}$$

where $\text{PE}(A_i) = [\text{PE}(x_i), \text{PE}(y_i), \text{PE}(z_i)]$. Here, x_i, y_i, and z_i are enhanced to a 108D vector using the $\text{PE}(\cdot)$ and subsequently merged into a singular 324D vector. The symbol $\otimes$ stands for the Hadamard product. This approach ensures the input is standardized, regardless of the number of deployed base stations.

(2) Input II: Given that the positions of a target are intrinsically interrelated, the T-Subnetwork input additionally comprises M historical position tokens along with a "start-of-sequence" (SOS) token, specifically employed to initiate the decoding procedure. Similarly, we treat the time as the "position" within the sequence context when the target is found at P_j. Both are encoded using $\text{PE}(x)$ and then combined to form an embedding token. Thus, the second input can be represented as

$$\mathbf{I}_2(t_j) = \left[\text{PE}(t_j - t_{j-1}) + \text{PE}(P_{j-1}), \cdots, \text{PE}(t_j - t_{\text{SOS}}) + \text{PE}(P_{\text{SOS}}) \right] \tag{12}$$

with the phase values measured at time t_j, wherein we are predicting the position of the device at time t_j. Both the time and location are expanded to a 324D vector. Hence, every element in $\mathbf{I}_2$ is a 324D token, which matches the dimensionality in the A-Subnetwork. The consistent dimensionality is particularly relevant during the cross-attention phase where the decoder attends to the encoder's outputs. The second input contains a maximum of $M + 1$ tokens. Similarly, the history is not mandatory. In cases where we are first localizing the object, the initial context P_{SOS} and t_{SOS} can be set to zero. Even for the stationary object, the historical context could help reduce potential errors.

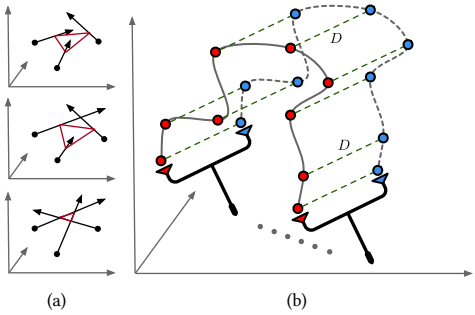

Fig. 2: Intersection and Dataset Collection. (a) shows the different intersection cases by three rays. (b) shows our proposed dataset collection approach where two wireless terminals are fixed at the two heads of the Y-shaped handle with a known constant distance.

4.3.2 Network Body

The T-Subnetwork is composed of multiple Transformer decoders, which are instrumental in sequence generation tasks. Each decoder layer comprises three principal components: self-attention, encoder-decoder attention, and a position-wise feed-forward network. (1) The self-attention mechanism allows each token in the input sequence to focus on other tokens within the same sequence, which helps in understanding the sequence's context. As depicted in Fig. 1, the input $\mathbf{I}_2(t_j)$ is fed into the self-attention block, which attends to the historical trajectory. Once a position is predicted by the T-Subnetwork, the output position is added to $\mathbf{I}_2(t_{j+1})$, contributing to the tracking context. This mode is often referred to as "autoregressive". (2) The encoder-decoder attention allows every position in the decoder to attend to all features in the input sequence captured by the encoder, helping the decoder concentrate on the input's relevant parts. In the figure, input $\mathbf{I}_1(t_j)$ is integrated into the encoder-decoder attention block. (3) The output of features from the decoder is directed through an MLP for the final positions prediction.

4.3.3 Output

The final output of the T-Subnetwork represents the ultimate spatial location of the wireless terminal at time t_j. This is achieved through the synergistic operations of both the subnetworks, which analyze spatial spectrum data and execute a sequence of historical location estimations to determine the terminal's position in the space. To enable autoregressive learning, the current predicted position P_j will be appended to $\mathbf{I}_2(t_{j+1})$ for the next prediction. If the total number of tokens is greater than $M+1$, the oldest predictions are removed.

4.4 Semi-Supervised Training

Next, we introduce a novel semi-supervised training approach along with the loss functions.

(1) Position Loss: All past DL-based indoor solutions, such as DLoc [20] and iArk [21], adopt the supervised training approach, i.e., providing the absolute location labels and using the distance between the predictions and true location labels for the loss function. In particular, their loss functions are defined as follows:

$$\mathcal{L}_0 = \|\widehat{P}_j - P_j\| \tag{13}$$

where $\widehat{P}_j$ and P_j are the predicated location and the ground truth, respectively. This approach requires us to acquire the absolute locations using a second high-precision positioning system, such as the OptiTrack or Lidar. However, such a cumbersome deployment mode prevents the spread of DL-driven localization. This limitation motivates us to find new loss functions that do not require absolute position labels.

(2) Distance Loss: Our first loss involves the distance between two sampled locations. We design a Y-shaped handle with two terminals fixed at the two heads at a known distance D, as shown in Fig. 2(b). As this handle is maneuvered in space, base stations collect two RF signals, each from a terminal. The data are then converted into two spatial spectra and processed by the model. Consequently, the model can predict two results, $\widehat{P}_j^1$ and $\widehat{P}_j^2$, for the two devices at time t_j, respectively. Then, we can define the distance loss as follows:

$$\mathcal{L}_1 = \left| \|\widehat{P}_j^1 - \widehat{P}_j^2\| - D \right| \tag{14}$$

This subtle shift obviates the necessity for recording the absolute locations of the two sampled points. This approach greatly simplifies the process of data collection by eliminating the need for additional positioning systems.

(3) Variance Loss: Prior solutions tend to emphasize minimizing $\mathcal{L}_0$, often overlooking the variability in the resultant errors. When it comes to triangulation, this variability manifests as the area of intersection of the three rays. Ideally, these three rays would converge at a single point. Yet, they often form a triangle, within which the wireless device might be located. Therefore, it is ideal to minimize the area of this error triangle as much as possible. With this perspective in mind, we propose another loss function that is intended to reduce the area of the intersection region. This issue might be amplified in three dimensions as the three rays might not intersect at all, as shown in Fig. 2(a). In such instances, computing the area is unfeasible. Thus, we use the perimeter instead for the loss:

$$\mathcal{L}_2 = \mathcal{D}(\vec{l}_1, \vec{l}_2) + \mathcal{D}(\vec{l}_2, \vec{l}_3) + \mathcal{D}(\vec{l}_1, \vec{l}_3) \tag{15}$$

where $\mathcal{D}(\cdot)$ signifies the distance of two rays. The rays of $\vec{l}_1$, $\vec{l}_2$, and $\vec{l}_3$ are the outputs of the A-Subnetwork (see Eqn. 9). The minimization of distances can bring these rays closer together in a manner that they are near-intersecting or intersect within a small region, which can be considered an approximation of an intersection point under practical circumstances. It is often sufficient for indoor localization applications.

The $\mathcal{L}_2$ is solely back-propagated into the A-Subnetworks for parameter adjustment without affecting the T-Subnetwork. This characteristic can further instruct the A-subnetworks in the acquisition of AoA knowledge. Previous studies [21, 30, 31] have reported the use of triangulation-based deep learning with the inclusion of CSI images or spatial spectra as inputs. Regrettably, these studies lack a mechanism to verify the model's triangulation fitting. Most of the time, the model functions similarly to a fingerprint-based localization, outputting a location given an image resembling a previously learned one. Our novel loss function, solely back-propagated

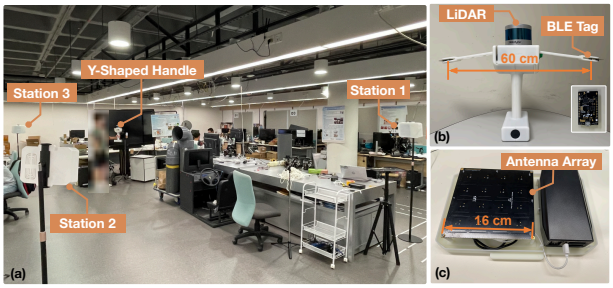

Fig. 3: The deployment of BLE localization platform.

to the A-subnetwork, guarantees that the produced features are truly relevant to the AoA.

(4) Joint Loss: A joint loss function can be formulated to combine the two loss components, i.e., the deviation between predicted and true distances and the area of intersection region. The overall loss function can be written as follows:

$$\mathcal{L}_3 = \lambda \mathcal{L}_1 + (1-\lambda)\left(\mathcal{L}_2^1 + \mathcal{L}_2^2\right) \tag{16}$$

where $\mathcal{L}_1$ is the distance loss, and $\mathcal{L}_2^{1(2)}$ are the variance loss for the two devices' predicted positions $P^{1(2)}$, respectively. λ is a hyperparameter that determines the relative weighting of the two loss terms and can be tuned for optimal performance.

Discussion: Compared to methods that employed Lidar or OptiTrack, this semi-supervised approach substantially reduces both the cost and complexity of training dataset collection. Since the model never receives absolute position labels as input, it develops its own global coordinate system. The coordinates generated by the model need to be aligned with the real-world coordinate system through a coordinate transformation, which can be achieved by positioning several anchors at known locations.

5 RAY DATASET

The performance of neural networks highly relies on the quantity and quantity of training datasets. Current datasets have been hindered by various limitations in terms of data scale, scene diversity, label precision, and coverage. To address these limitations, we build a multi-technology, cross-scene, million-scale 3D localization database, named *Ray*, after a three-year effort. A summary of the gathered data is displayed in Table 1. We collect data from 17 scenes with 50 different scenarios (i.e., settings). The database contains a total of 1,617,142 records from RFID tags (80.6%), Wi-Fi devices (7.2%), and BLE beacons (12.2%).

(1) RFID: Our design incorporates a dual-channel station with an array composed of 4×4 antennas. We use a USRP X310 SDR from NI [32] to handle the baseband signal processing of a station. Each X310 is equipped with two TwinRX daughterboards. One RX channel is linked to the antenna array, while the other is connected to a circularly polarized patch antenna, serving as the side channel. The RF source is fixed approximately 5 m from the tag; it is used to power the tag and prompt it to continuously send `RN16` packets. The advantage of such a bistatic design is that three antenna arrays can be placed at a considerable distance from the tag (e.g., 50 m). Notably, the focus here is not on ultra-long-range communication of RFID systems. We recommend readers refer to a previous study [33] to further understand how the tag-to-receiver range can exceed 130 m at 1 kbps and 30 dBm transmitting power. Using the experimental platform, we collect data from 7 different types of scenes with 28 distinct settings.

Table 1: Summary of Ray Dataset. RSS represents the average signal strength; Total. indicates the total number of samples; Den. provides the sample count per cubic meter; Sta. denotes the count of base stations; Dist. reflects the average distance between sampled points and the base stations; Temp. is the ambient temperature.

Type. (#)	Sc. (#)	St. (#)	RSS (dBm)	Total (#)	Den. (p/m^3)	Sta. (#)	Dist. (m)	Temp. (°C)
RFID	A	S1	−62.5	84,392	3,843	3	5	31.2
		S2	−66.4	57,311	4,689	3	10	30.3
		S3	−66.7	55,527	5,274	3	15	29.9
		S4	−69.4	54,518	3,787	3	20	29.4
		S5	−71.0	50,302	4,336	3	25	27.2
		S6	−75.0	51,241	5,866	3	30	27.4
		S7	−77.4	51,289	5,871	3	35	27.7
		S8	−78.8	74,521	7,834	3	40	28.1
		S9	−79.3	61,909	4,236	3	45	28.3
		S10	−79.1	76,475	5,224	3	50	29
		S11	−88.6	50,186	10,490	3	55	28.7
	B	S12	−71.8	23,028	539	3	25	30.1
		S13	−76.9	21,357	702	3	35	30.4
		S14	−78.1	38,303	1,382	3	40	30.9
	C	S15	−68.3	18,726	6,080	3	20	33.1
	D	S16	−68.9	77,538	4,345	3	13	29.2
		S17	−67.0	40,571	2,546	3	13	28.8
	E	S18	−66.2	160,494	38,213	3	10	18.4
		S19	−65.3	78,635	27,924	2	10	24.9
		S20	−63.7	30,103	10,907	2	10	25.1
		S21	−64.9	26,916	8,901	2	10	27.6
		S22	−65.4	32,042	5,057	2	10	24.8
	F	S23	−65.1	48,467	22,627	3	7	25.8
	G	S24	−61.4	10,521	4,912	3	5	27.5
		S25	−60.2	5,291	938	3	5	30.1
		S26	−61.9	6,723	2,394	3	5	27.3
		S27	−61.9	8,413	1,829	3	5	28.2
		S28	−63.7	8,911	1,600	3	5	27.9
WiFi	H	S29	−58.6	11,288	123	4	7	N/A
		S30	−58.6	14,543	164	4	7	N/A
		S31	−60.0	10,579	106	4	7	N/A
		S32	−60.1	8,287	76	4	7	N/A
		S33	−60.1	5,233	58	4	7	N/A
	I	S34	−48.2	25,976	701	3	4	N/A
		S35	−48.2	23,677	656	3	4	N/A
		S36	−48.3	16,286	497	3	4	N/A
BLE	J	S37	−72.5	8,030	138	4	5	26.4
		S38	−84.5	11,123	199	4	5	26.4
		S39	−89.2	8,967	309	4	5	27.1
	K	S40	−62.4	8,419	1,011	4	5	29.3
		S41	−61.1	8,959	1,134	4	5	28.8
	L	S42	−60.4	6,386	823	4	5	29.9
		S43	−61.7	8,375	607	4	5	28
	M	S44	−60.9	11,235	1,489	4	5	18.5
		S45	−64.9	8,647	1,259	4	5	18.3
	N	S46	−71.0	15,412	1,381	4	20	17.8
		S47	−75.1	19,048	1,311	4	20	17.8
	O	S48	−78.3	32,039	1,059	4	25	17.6
	P	S49	−78.7	31,766	1,251	4	25	17.3
	Q	S50	−69.2	18,975	1073	4	10	16.8

(2) Wi-Fi: We integrate the dataset released by DLoc [20] into the dataset, which aims to position a Wi-Fi receiver using CSI. It covers two scenes and eight settings. Each station is equipped with a four-antenna linear array. For more hardware configuration details, readers are directed to [20]. Unlike narrowband-enabled RFID or BLE, Wi-Fi adopts the wide band, so the CSI contains 64 subcarrier information (i.e., phase and RSSI). To utilize this extra information,

Table 2: Accuracy of RFID Localization (mean errors in cm)

#	TBL (w/ history)	TBL (w/o history)	iArk	DLoc
S01	8.2±6.4	10.9 (24.8% ↑)	13.2 (37.9% ↑)	23.4 (65.0% ↑)
S02	9.7±7.1	13.5 (28.1% ↑)	31.7 (69.4% ↑)	32.6 (70.2% ↑)
S03	18.3±10.8	25.1 (27.1% ↑)	46.4 (60.6% ↑)	37.7 (51.5% ↑)
S04	27.4±16.4	33.7 (18.7% ↑)	70.9 (61.4% ↑)	46.1 (40.6% ↑)
S05	33.7±19.8	47.9 (29.6% ↑)	91.2 (63.0% ↑)	56.6 (40.5% ↑)
S06	37.5±34.9	62.5 (40.0% ↑)	99.1 (62.2% ↑)	63.5 (40.9% ↑)
S07	38.1±24.6	51.2 (25.6% ↑)	110.7 (65.6% ↑)	63.2 (39.7% ↑)
S08	55.2±31.3	79.0 (30.1% ↑)	118.2 (53.3% ↑)	85.0 (35.1% ↑)
S09	60.8±36.0	88.2 (31.1% ↑)	133.1 (54.3% ↑)	87.3 (30.4% ↑)
S10	35.8±27.3	57.4 (37.6% ↑)	167.1 (78.6% ↑)	78.1 (54.2% ↑)
S11	41.9±28.4	52.4 (20.0% ↑)	179.0 (76.6% ↑)	72.6 (42.3% ↑)
S12	43.5±29.1	63.7 (31.7% ↑)	91.2 (52.3% ↑)	88.2 (50.7% ↑)
S13	52.9±34.1	65.3 (19.0% ↑)	108.5 (51.2% ↑)	101.2 (47.7% ↑)
S14	39.3±26.6	60.5 (35.0% ↑)	114.0 (65.5% ↑)	76.3 (48.5% ↑)
S15	10.2±6.8	16.4 (37.8% ↑)	60.5 (83.1% ↑)	13.2 (22.7% ↑)
S16	14.2±9.7	23.8 (40.3% ↑)	38.8 (63.4% ↑)	29.6 (52.0% ↑)
S17	14.5±10.1	20.2 (28.2% ↑)	36.0 (59.7% ↑)	41.4 (65.0% ↑)
S18	7.0±3.7	9.9 (29.3% ↑)	29.1 (75.9% ↑)	13.2 (47.0% ↑)
S19	5.1±2.4	9.1 (44.0% ↑)	31.2 (83.7% ↑)	9.7 (47.4% ↑)
S20	5.9±2.7	8.9 (33.7% ↑)	25.3 (76.7% ↑)	11.3 (47.8% ↑)
S21	9.7±5.6	8.9 (9.0% ↓)	19.8 (51.0% ↑)	9.4 (3.2% ↓)
S22	8.1±6.9	13.7 (40.9% ↑)	33.2 (75.6% ↑)	15.3 (47.1% ↑)
S23	4.4±3.1	8.2 (46.3% ↑)	18.6 (76.3% ↑)	9.4 (53.2% ↑)
S24	6.6±4.3	9.9 (33.3% ↑)	16.4 (59.8% ↑)	13.6 (51.5% ↑)
S25	10.8±7.6	13.0 (16.9% ↑)	16.7 (35.3% ↑)	23.7 (54.4% ↑)
S26	8.6±5.0	10.7 (19.6% ↑)	14.2 (39.4% ↑)	13.2 (34.8% ↑)
S27	7.8±5.4	13.1 (40.5% ↑)	20.5 (62.0% ↑)	22.2 (64.9% ↑)
S28	14.7±9.5	16.3 (9.8% ↑)	15.7 (6.4% ↑)	21.5 (31.6% ↑)
Mean	**22.5±14.8**	**31.9 (28.9% ↑)**	**62.5 (60.7% ↑)**	**41.4 (45.5% ↑)**

Table 3: Accuracy of Wi-Fi Localization (mean errors in cm).

#	TBL (w/o history)	TBL (w/ history)	iArk	DLoc
S29	23.8±14.2	29.1 (18.2% ↑)	74.8 (68.2% ↑)	63.2 (62.3% ↑)
S30	21.2±12.5	28.1 (24.6% ↑)	73.2 (71.0% ↑)	61.8 (65.7% ↑)
S31	21.8±13.1	30.4 (28.3% ↑)	68.2 (68.0% ↑)	59.5 (63.4% ↑)
S32	20.3±12.9	28.2 (28.0% ↑)	71.5 (71.6% ↑)	58.8 (65.5% ↑)
S33	22.4±14.0	30.5 (26.6% ↑)	73.9 (69.7% ↑)	60.4 (62.9% ↑)
S34	16.4±8.5	19.4 (15.5% ↑)	54.2 (69.7% ↑)	37.2 (55.9% ↑)
S35	11.2±6.3	14.7 (23.8% ↑)	48.7 (77.0% ↑)	32.1 (65.1% ↑)
S36	14.1±7.7	18.0 (21.7% ↑)	45.2 (68.8% ↑)	36.6 (61.5% ↑)
Mean	**18.9±11.1**	**24.8 (23.3% ↑)**	**63.7 (70.5% ↑)**	**51.2 (62.8% ↑)**

Table 4: Accuracy of BLE Localization (mean errors in cm)

#	TBL (w/o history)	TBL (w/ history)	iArk	DLoc
S37	19.8±10.4	24.2 (18.2% ↑)	35.9 (44.8% ↑)	30.5 (35.1% ↑)
S38	37.5±25.2	43.2 (13.2% ↑)	58.2 (35.6% ↑)	54.9 (31.7% ↑)
S39	39.2±27.9	47.7 (17.8% ↑)	63.5 (38.3% ↑)	58.0 (32.4% ↑)
S40	7.0±4.8	11.5 (39.1% ↑)	10.3 (32.0% ↑)	27.0 (74.1% ↑)
S41	6.6±2.7	10.3 (35.9% ↑)	9.9 (33.3% ↑)	24.4 (73.0% ↑)
S42	6.6±5.0	8.1 (18.5% ↑)	27.8 (76.3% ↑)	19.9 (66.8% ↑)
S43	5.1±4.7	7.8 (34.6% ↑)	26.2 (80.5% ↑)	21.7 (76.5% ↑)
S44	32.5±22.2	36.5 (11.0% ↑)	65.2 (50.2% ↑)	48.3 (32.7% ↑)
S45	36.2±24.8	40.1 (9.7% ↑)	68.7 (47.3% ↑)	51.6 (29.8% ↑)
S46	50.2±45.8	65.8 (23.7% ↑)	80.0 (37.2% ↑)	103.8 (51.6% ↑)
S47	52.9±47.7	67.3 (21.4% ↑)	84.5 (37.4% ↑)	105.2 (49.7% ↑)
S48	101.1±51.9	107.0 (5.5% ↑)	122.6 (17.5% ↑)	137.8 (26.6% ↑)
S49	72.3±40.5	90.5 (20.1% ↑)	114.6 (36.9% ↑)	97.5 (25.8% ↑)
S50	45.9±34.7	56.2 (18.3% ↑)	82.1 (44.1% ↑)	75.5 (39.2% ↑)
Mean	**36.6±24.9**	**44.0 (20.5% ↑)**	**60.7 (43.7% ↑)**	**61.2 (46.1% ↑)**

we compute a spatial spectrum using Eqn. 3 for each subcarrier and simply add the 64 spatial spectra together as the final one.

(3) Bluetooth: The deployment of the BLE localization platform is shown in Fig. 3. The platform, developed by Silicon Labs, operates on 2.4 GHz in accordance with BLE V4.2. The platform utilizes a 4×4 patch antenna array (model: BRD4191A) [34] and has a size of 16×16 cm^2, with each individual patch antenna being 2.42×2.42 cm^2. The average errors in azimuth and elevation are $\pm 3.2^\circ$ and $\pm 4^\circ$, respectively. Each antenna comes with dual-input ports for receiving both horizontally and vertically polarized signals. Built with a JL-2800 laminate type, the antenna array board underwent extensive optimization and testing using IT-180A and IS400 laminates. We made phase measurements using Gecko SDK 4.1 and the RTL library. The Bluetooth tags (model: RD4184A) are also from Silicon Labs.

6 MICRO-BENCHMARK

In this section, we evaluate the TBL solution using the collected datasets, focusing on the errors in individual scenes.

Experimental Setup. We construct individual models for each distinct setting without addressing the transferability of these models. Thus, a total of 50 TBL models are trained for the corresponding 50 scenarios. In Section 6.4, we evaluate various model configurations and ultimately adopt a structure comprising two standard encoders for the A-Subnetwork and two decoders for the T-Subnetwork in micro-benchmark. Within this setup, each attention layer features eight heads, each with 64 dimensions. This architecture results in a total of approximately 12 million parameters for the entire model. In the training process, we split the collected dataset into segments in the time domain, each containing 10 samples to facilitate learning from historical data. The samples in each segment are fed into the model sequentially for training or testing. We allocate 80% of segments for training purposes and reserve the remaining 20% for testing. During the training process, the joint loss with $\lambda = 0.05$ is employed, in which the $\mathcal{L}_2$ is sole-propagated to the encoder. During model training, we maintain a batch size of 4096 and subject each dataset to 5,000–7,000 iterations of training. Our model is trained on a server equipped with an AMD 3955WX processor, 64 GB of RAM, and two NVIDIA 4090 GPUs. The training process in each setting takes approximately 4–6 hours each. We accomplished the complete training tasks in 11 days.

6.1 Accuracy

The accuracy of our model is gauged by the position error, which is conceptualized as the Euclidean distance between the predicted and actual positions. For a comparative analysis, we adopt the SOTA solutions, namely, iArk [21] and DLoc [20] – as benchmark references. Using spatial spectra in image format, both approaches leverage ResNet coupled with MLP to determine the positions. Particularly, DLoc integrates an additional ResNet-based decoder for consistency verification. In contrast, the Transformer ingests spatial spectra as tokens, which may include or exclude historical data. To ensure unbiased comparisons, input spatial spectra dimensions are consistently set at 36×9. Both iArk and DLoc were trained using the same dataset and under the same train/test sets settings as TBL. The outcomes for the three technologies are tabulated in Table 2, Table 3, and Table 4, respectively. In the table, the red arrow (↑) indicates an increase in percentage error, reflecting a decrease in

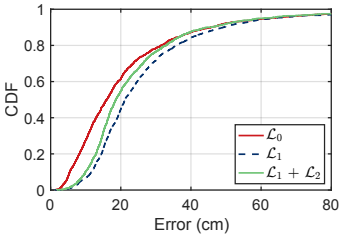

Fig. 4: Ablation Study

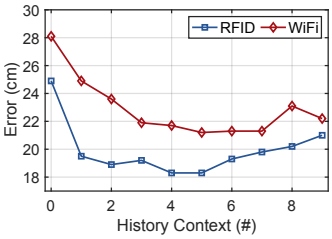

Fig. 5: Historical Context

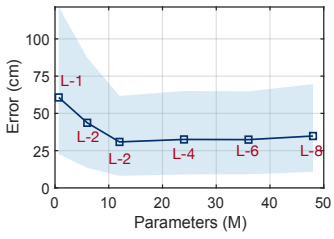

Fig. 6: Model Variant

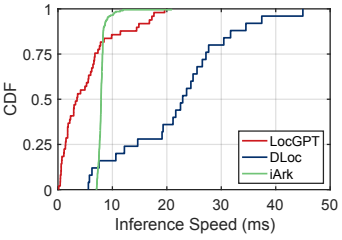

Fig. 7: Inference Speed.

performance compared to the TBL with historical context. Conversely, the green arrow (↓) represents a reduction in percentage error, indicating improved performance relative to the TBL with historical context. Overall, the TBL model consistently outperforms the other solutions across all three datasets, regardless of the inclusion of historical data. Given the space constraints, we focus solely on the RFID results for analysis. The findings from the other two datasets indicate similar results. Specifically, we observe the following:

• First, the TBL reduces errors by 60.7% and 45.5% on average when compared with iArk and DLoc, respectively. An outlier (9.4 cm vs. 8.9 cm) is evident in S21. This marginal 3.2% deviation lies within the standard deviation. • Second, the errors of the TBL without historical context are derived from the localization errors of the first samples in the test segments, where no historical context is available. The TBL results with historical context cover the cases with the context from 1 to 10 samples. The results demonstrate that the inclusion of past data enhances accuracy by an estimated 28.9%. • Third, a pronounced correlation exists between accuracy and the RSS. This correlation is evident in the results from the S01 to S11 scenarios, all gathered within the same environment but at varying distances. As the range increases (specifically, between 5 and 55 m), the RSS weakens, spanning from -62.5 dBm to -88.6 dBm, which in turn elevates the mean error values, ranging from 8.2 cm to 41.9 cm. • Fourth, the best outcomes are discerned in S23, S19, and S20. These results are attributable to their augmented densities (e.g., 22,627 samples per m^3) and reduced distances (e.g., 7 m). Conversely, the least accurate outcomes are recorded in segments S8–S11, stemming from notably feeble signals (e.g., -88.6 dBm) and extended distances (e.g., 40 m). • Finally, comparable configurations within identical scenes (e.g., S18–S22 within Scene E and S24–S28 within Scene G) display closely aligned errors (e.g., deviations of 22.6% and 29.4%, respectively). This finding suggests that localization precision is predominantly influenced by scene layouts rather than specific settings, such as station placements.

Discussion. The superior performance of TBL when compared with SOTA methods can be attributed to several key factors. Primarily, the self-attention mechanism allows the model to intelligently weigh signals across various time intervals. This is further enhanced by the contextual processing capability of Transformers, which processes data in parallel, fostering a deeper understanding of inter-measurement relationships. Moreover, the model's hierarchical feature abstraction strategy, where encoders in the A-Subnetwork are tasked with AoA estimation and decoders in the T-Subnetwork focus on triangulation, facilitates a refined analysis of environmental dynamics. The integration of historical context aids in mitigating instantaneous interference and noise, ensuring consistent outcomes. Lastly, the introduction of a joint loss function integrates the distinct roles of encoders and decoders, culminating in enhanced localization precision.

6.2 Ablation Study

Next, we investigate the influence of various loss functions on localization accuracy. We measure errors using three distinct loss functions: $\mathcal{L}_0$ (Eqn. 13), $\mathcal{L}_1$ (Eqn. 14), and the joint loss $\mathcal{L}_3 = \mathcal{L}_1 + \mathcal{L}_2$ (Eqn. 16). In our subsequent experiments, we primarily utilize the S03 dataset, chosen for its characteristic base station deployment distances typical of indoor localization scenarios, unless specified otherwise. For the ablation study, to ensure fairness, we maintain consistent hyper-parameters across various loss functions. The respective CDFs of these errors are presented in Fig. 4. Leveraging absolute location labels results in a reduced mean error of 16 cm (the 90th percentile: 45.7 cm) relative to other setups. The accuracy of joint loss is about 24% lower than $\mathcal{L}_0$. This is anticipated, given that supervised learning typically exhibits superior performance over semi-supervised techniques. Nevertheless, this comes at the cost of deploying an additional high-precision localization system for acquiring labels. On the other hand, the median errors for $\mathcal{L}_1$ and $\mathcal{L}_1 + \mathcal{L}_2$ stand at 21.2 cm (90th percentile: 49.2 cm) and 18.6 cm (90th percentile: 44.6 cm), respectively, suggesting that incorporating variance loss can reduce errors by approximately 13%.

6.3 Impact of the Historical Context

We further analyze the impact of the historical context by focusing on two chosen scenes: S03 (RFID) and S30 (Wi-Fi). During the experiments, we compute the average errors incorporating M historical contexts, i.e., encompassing M past spatial spectra for the A-Subnetwork and M location outcomes for the T-Subnetwork, where M ranges from 0 to 9. As shown in Fig. 5, the findings reveal a compelling trend: errors decrease notably when incorporating $5 - 6$ historical context steps. This finding underscores the value of leveraging historical data, which can help in counteracting the sporadic signal fluctuations instigated by transient interferences, such as moving entities. Interestingly, there is an uptick in errors for $M > 6$, possibly due to the error accumulation. Thus, a setting of $M = 5$ is recommended.

6.4 Model Variant

In our default configuration, we employ two encoder layers for the A-Subnetwork and two decoder layers for the T-Subnetwork.

This experiment aims to assess the performance of various architectural variants. Specifically, we explore configurations with one layer (with two 8D heads each), two layers (with two 8D heads each), two layers (with eight 64D heads each), four layers (with eight 64D heads each), six layers (with eight 64D heads each), and eight layers (with eight 64D heads each). The respective parameter counts for these configurations are 0.8, 6, 12, 24, 36, and 48 M. The results are shown in Fig. 6. The six configurations achieve mean errors of 60.7, 43.6, 30.9, 32.5, 32.4, and 34.8 cm, respectively. We also observe a linear reduction in error corresponding to parameter counts less than 10 M. Yet, the error plateaus when the parameter count ranges from 10 to 50 M. This observation is in accordance with the emergence phenomenon or phase transition in large AI models—namely, a pronounced behavioral shift not anticipated from training the system at smaller scales. Without supplementing with additional training data, increasing model size may not necessarily yield enhanced accuracy. Thus, 10 M parameters are advised for the TBL for an individual scenario.

6.5 Inference Speed

Finally, we evaluate the inference speeds of TBL, iArk, and DLoc. For a fair comparison, all models are implemented within the PyTorch framework and evaluated using the S03 dataset to measure inference times. These evaluations are conducted on the same computational setup as detailed in the experimental setup section, ensuring consistency across all tests. The CDFs of the inference time for each approach are shown in Fig. 7. As can be seen, their median inference time stands at 3.5 ms, 7.9 ms, and 22.9 ms, respectively. Notably, TBL demonstrates superior speed compared with the others. This efficiency can be attributed to TBL's treatment of spatial spectra as tokens, rather than the more resource-intensive image inputs used by iArk and DLoc. Furthermore, the latter two rely on computationally demanding CNNs for processing. Additionally, the inherent parallel processing capabilities of the Transformers contribute to TBL's faster response time.

7 LocGPT: PRE-TRAINING MODEL

Prior micro-benchmark has demonstrated both the feasibility and efficacy of implementing the Transformer model for localization purposes. In this section, we address the challenge of non-transferability.

7.1 Pre-Training

RF signal propagation is highly dependent on environmental specifics, meaning models trained in one setting may not perform well in others, thus limiting their generalizability. This challenge requires the recollection of large volumes of data whenever a new scene is encountered. Inspired by the remarkable success of GPT in the field of NLP, we propose LocGPT to tackle the generalizability issue. LocGPT is a pre-trained TBL model with millions of high-quality data. When encountering a distinct scene, LocGPT is capable of being fine-tuned with a limited amount of data, thereby fitting effectively into the new dynamics.

The network architecture of LocGPT, shown in Fig. 8, is almost the same as that of the previously proposed TBL model, except for the number of A-Subnetworks. In the previous architecture, each base station is designated an A-Subnetwork. However, the exact number of base stations deployed can vary depending on

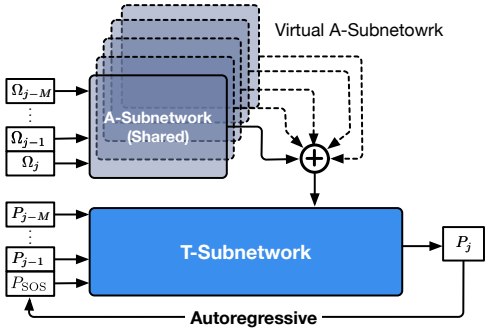

Fig. 8: Architecture of LocGPT

the specific setting. To deal with the scalability, we train only a single A-Subnetwork, term as the "parent A-Subnetwork", during the pre-training phase. Doing this allows all trained base stations to share parameters from this singular A-Subnetwork. In contrast to our micro-benchmark setup, for pre-training LocGPT, we employ 1.4 million localization data points. The initial configuration of a two-encoder/decoder model is inadequate for addressing generalization challenges. Through empirical analysis, we determined that a configuration comprising six layers in both the encoder and decoder subnetworks strikes an optimal balance between convergence speed and generalization capabilities, with each layer featuring eight heads of 64 dimensions. Consequently, the total amount of parameters of LocGPT is about 20M, effectively reduced from 36M through parameter sharing of A-Subnetwork. While models with an increased number of encoder/decoder layers exhibit marginally enhanced transferability, they also require extended training durations. LocGPT is pre-trained using all datasets, excluding S02, S13, S20, S27, S31, S32, S33, S35, S36, S37, S40, S46, and S50. These datasets are reserved for assessing the model's transfer learning performance. This selected dataset enables us to evaluate LocGPT's adaptability across various technologies, including RFID, Wi-Fi, and BLE, and to showcase its transfer learning potential both within identical scenes under varying scenarios and across completely distinct scenes. LocGPT is trained with $\mathcal{L}_0$ loss. The hardware setup for pre-training is the same as that used for the micro-benchmark. The Adam optimizer with a cosine learning rate schedule is employed, ranging from $5e^{-4}$ to e^{-6}. We conducted the pre-training within 5 days.

7.2 Fine-Tuning

During the fine-tuning phase, the shared A-Subnetwork is further specialized into distinct A-Subnetworks tailored for individual base stations. The rationale behind this design is to harness the generalized capabilities of the primary A-Subnetwork, which captures universal AoA estimation features and then adapts these features to the specificities of each base station. This model can be fine-tuned through three predominant techniques: Full fine-tuning, LoRA, and Adapter.

(1) Full Fine-tuning: This approach updates all 36M parameters of the pre-trained model based on the target task's dataset. It is a comprehensive method that adjusts the entire model to the new data.

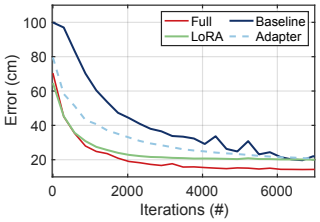

Fig. 9: Convergence Efficiency

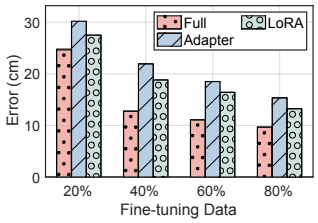

Fig. 10: Accuracy

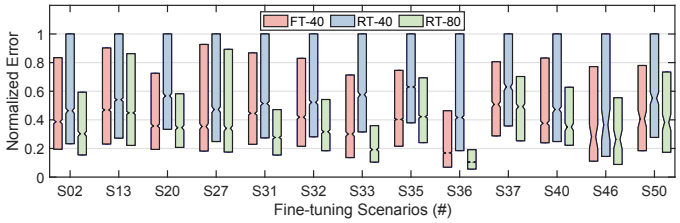

Fig. 11: Transfer Learning from LocGPT v1.0

(2) LoRA: Leveraging a recalibration strategy, LoRA modifies the activations of the pre-trained model at each layer. Instead of revising all parameters, LoRA recalibrates existing model knowledge to better fit the new tasks. In this method, about 0.9M new parameters are updated [35].

(3) Adapter: This method embeds small, specialized adapter modules between the model's layers. Rather than adjusting all the parameters, only these adapters, consisting of 0.8M parameters, undergo training, while the primary model parameters stay frozen.

8 EVALUATION

This section evaluates the advancement of LocGPT and the performance of various fine-tuning techniques.

8.1 Convergence Efficiency

We selected the S02 dataset to fine-tune LocGPT, allocating 40% for training and 60% for testing. We measured fine-tuning efficiency by counting iterations (each processing 4096 samples) and tracking mean error. After each iteration, we assess the mean error using the test set. For comparison, we also trained a TBL model from scratch as the baseline, bypassing the benefits of the pre-trained model. The results are depicted in Fig. 9. The ideal trajectory for these curves is towards the bottom-left corner, which would indicate achieving lower errors in fewer iterations. The graph clearly shows that leveraging a pre-trained model accelerates this convergence, regardless of the specific fine-tuning approach used. Specifically, to reach a mean error of 25cm, full fine-tuning, LoRA, adapter, and baseline methods took 1450, 2030, 4350, and 5220 iterations, respectively. Thus, in terms of convergence speed, the ranking from slowest to fastest is the baseline, adapter, LoRA, and full fine-tuning. Furthermore, this experiment demonstrates that training deep models from scratch can be computationally intensive and time-consuming. By starting with a pretrained model, one can achieve faster convergence, thus saving computational resources and time.

8.2 Accuracy

Next, we employ varying portions of the S02 dataset for fine-tuning, specifically 20%, 40%, 60%, and 80%, while maintaining a consistent 20% for testing. The comparative accuracies of the three fine-tuning techniques are depicted in Fig. 10. As can be seen, the trends remain consistent across all four cases, with full fine-tuning consistently achieving the lowest mean errors, followed by the adapter, and then LoRA. For instance, when utilizing 60% of the dataset, the resultant mean errors for the three techniques are 11.1, 18.5, and 16.4 cm, respectively. Evidently, in terms of precision, full fine-tuning stands out as the superior method compared with the others. The advantage of full fine-tuning can be attributed to two main reasons. First, full fine-tuning can provide a good balance between leveraging the pre-trained knowledge and fitting the new data. This can lead to a model that generalizes well to new, unseen data. Second, unlike other methods that add additional components (like adapters) or adjust activations (e.g., LoRA), full fine-tuning does not introduce new architectural components. Instead, it utilizes the existing architecture, which often has been optimized for performance during pre-training.

8.3 Transfer Learning

Finally, we assess the transfer learning capabilities of LocGPT in new environments using a full fine-tuning method, termed "FT-40," where it was fine-tuned with 40% of data from different scenarios. For comparison, we trained separate models from scratch with either 80% or 40% of data, named "RT-80" and "RT-40," respectively, using the same 20% test dataset. To assess transfer learning effectiveness beyond just error rates, we used robust scaling normalization for localization errors, defined as $X_{\text{norm}} = X/\text{IQR}$, with IQR being the interquartile range. The resulting error distributions are shown in Figure 11. Our observations highlight two key insights. On one hand, the fine-tuned model consistently surpassed the performance of RT-40, even though both are trained on a similar 40% data subset. For instance, the median errors in the 13 scenarios for RT-40 are reduced by 16.7%, 16.1%, 37.1%, 24.8%, 13.3%, 20.5%, 47.7%, 28.5%, 59.5%, 19.4%, 20.2%, 22.8%, and 23.6% respectively, when compared with FT-40. On the other hand, the outcomes for FT-40 closely mirror the results of RT-80, with median error disparities being 13.7%, 4.5%, 3.2%, 2.8%, 46.0%, 23.8%, 36.1%, 6.5%, 38.2%, 3.1%, 7.0%, 6.2%, and 4.9%, respectively. In short, the results of FT-40 closely align with that of RT-80 (diff: 15%), but are far higher than that of RT-40 (diff: 27%). These comparisons fully demonstrate the efficacy of transferring knowledge from pre-training for specific scenarios, even when faced with constrained data availability.

This efficacy of LocGPT is due to several key factors. First, the pre-trained model's exposure to diverse data enables it to discern various patterns, structures, and data relationships. Second, the pre-trained model's weights offer a robust starting point, encapsulating beneficial features that are typically transferable across tasks, leading to enhanced performance against models that are arbitrarily initialized. Third, by commencing with a pre-trained model and subsequently fine-tuning with limited data, the risks associated with overfitting are reduced, as the model has already generalized over comprehensive data during its pre-training phase. Finally, the use of spatial spectra-based tokens ensures effective

abstraction of hardware diversity, ensuring that LocGPT can be adapted to various localization technologies.

9 LIMITATIONS AND FUTURE WORKS

We discuss the limitations and future improvement of LocGPT as follows:

Zero/Few-shot generalizability: The transfer learning experiments suggest that 40% of new environmental data is essential for effective fine-tuning. This need arises from the intricate and unpredictable nature of indoor environments, amplified by the multipath effect, where signals bounce and create diverse and unpredictable spatial contexts. This complexity demands a deeper learning and adaptation strategy. Integrating additional modalities, like environmental visual data, could offer a viable strategy, enhancing the model's understanding of the context for RF signal propagation. This enhancement has the potential to boost the model's performance in zero/few-shot learning scenarios.

Dynamic adaptation: The performance of the trained model is susceptible to environmental configurations, such as the placement of furniture items like tables and sofas. When primary obstacles or reflectors change their positions, it becomes necessary to fine-tune models using a small number of freshly gathered samples. Regrettably, during the operational phase, we do not have a clear indicator of a drop in prediction accuracy due to the absence of ground truth or a benchmark. Currently, with our model, we can employ the $\mathcal{L}_2$ loss as an indirect measure to monitor performance degradation. A persistently high $\mathcal{L}_2$ loss indicates that the rays estimated by the A-Subnetworks fail to intersect. It suggests that the model might be losing its effectiveness and needs fine-tuning. Future work will explore additional strategies to mitigate the impact of dynamic environmental changes.

Multi-modality scalability: Leveraging the inherent flexibility of the Transformer architecture, LocGPT is well-suited for integrating multi-modal inputs, thereby enhancing its applicability to a broader range of downstream tasks. It can easily accommodate additional sensor data streams, such as those from Inertial Measurement Units (IMU) or visual sources, through the integration of supplementary encoders. These additional encoders, tailored for IMU or visual data, would enable unified feature extraction alongside the existing framework. Given that the A-subnetwork in LocGPT is tailored for phase information extraction, the model could potentially be extended to support CSI-based sensing applications by replacing the MLP head in the A-subnetwork, and modifying the T-subnetwork accordingly. This extension represents a promising direction for future work.

10 RELATED WORK

Our work falls under several broad categories of indoor localization studies.

(1) Indoor RF Localization. RF localization is a long-studied topic with extensive works [1, 36]. Various metrics of RF signals are widely used for localization, including RSSI [36], carrier phase [1, 37], CSI [3, 38], Time-of-Flight (ToF) [39], and AoA [4, 20, 21, 40–43]. In this study, we focus on AoA-based methods, which suffer from problems caused by the environment (e.g., complex multipath effect, and non-line-of-sight.) Past works have solved these problems by identifying the direct path and trying to eliminate the effects caused by the environment [3, 40]. For instance, mD-Track [3] improves the AoA accuracy under multipath scenarios by using ToF measurement for direct path identification. However, these approaches tend to fail when the multipath signals are extremely close or when the direct path is completely blocked.

(2) Deep learning for localization. The pursuit of enhancing accuracy in intricate indoor environments has guided recent SOTA efforts towards deploying deep learning (DL) models[20, 21, 44, 45]. This trend is paralleled by the introduction of numerous localization datasets aimed at supporting the training of such models [20, 21, 44, 46]. In response to these developments, our work takes a significant step forward by not only compiling the most extensive dataset characterized by high-accuracy labels and a wide array of scenarios but also by introducing a Transformer-based solution for localization. Our approach capitalizes on the strengths of Transformer architectures, renowned for their success in various domains, to forge a novel path in localization tasks. Furthermore, by presenting a pre-trained model, we directly tackle the challenges associated with transfer learning. This initiative aims to reduce the effort required to adapt the model to new environments or conditions, thus broadening the practical applicability of DL in localization and setting a new benchmark for future research in this field.

(3) Transfer learning for generalization. Many efforts have been made to solve the environment-dependent problems. [47, 48] significantly reduces the data collection and labeling cost for localization in each new environment. However, [47] is based on the fingerprint method, which is different from ours, and [48] is implemented by RF-propagation simulations, lacks practicality and is not as convincing as our experiments, which are obtained from extensive real-world data. [49] achieves great performance in localization in new target environments, but these environments are few and similar to source environments, which differs from our aim for a generalized solution without such constraints.

11 CONCLUSION

This paper presents a Transformer-based Localization (TBL) model designed for wireless indoor localization encompassing RFID, Wi-Fi, and BLE technologies. The hierarchical structure of TBL consists of multiple A-Subnetworks, each tasked with determining the AoA from different base stations. These are then collectively processed by the T-Subnetwork to predict the localization results along with historical positions. TBL outperforms existing methods in 50 different scenarios. To enhance the generalizability of the TBL model across scenarios, we introduce LocGPT, which is pre-trained on 1.4 million data samples. It demonstrates the capability to maintain near-optimal accuracy even with considerably reduced datasets.

ACKNOWLEDGMENTS

This work is supported by the NSFC Key Program (No. 61932017), UGC/GRF (No. 15204820, 15215421), Innovation and Technology Fund (ITS/099/21), and Shenzhen Fundamental Research Program (No. JCYJ20230807140410022). We thank all the anonymous reviewers and our shepherd, Dr. Inseok Hwang, for their valuable comments and helpful suggestions.

REFERENCES

[1] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, and Y. Liu, "Tagoram: Real-time tracking of mobile rfid tags to high precision using cots devices," in *Proc. of ACM MobiCom*, 2014, pp. 237–248.

[2] Y. Ma, N. Selby, and F. Adib, "Minding the billions: Ultra-wideband localization for deployed rfid tags," in *Proc. of ACM MobiCom*, 2017, pp. 248–260.

[3] Y. Xie, J. Xiong, M. Li, and K. Jamieson, "md-track: Leveraging multi-dimensionality for passive indoor wi-fi tracking," in *Proc. of ACM MobiCom*, 2019, pp. 1–16.

[4] Y. Xie, Y. Zhang, J. C. Liando, and M. Li, "Swan: Stitched wi-fi antennas," in *Proc. of ACM MobiCom*, 2018.

[5] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller, "3d tracking via body radio reflections," in *Proc. of USENIX NSDI*, vol. 14, 2013.

[6] F. Adib, Z. Kabelac, and D. Katabi, "Multi-person motion tracking via rf body reflections," in *Proc. of USENIX NSDI*, 2015.

[7] M. Zhao, Y. Tian, H. Zhao, M. A. Alsheikh, T. Li, R. Hristov, Z. Kabelac, D. Katabi, and A. Torralba, "Rf-based 3d skeletons," in *Proc. of ACM SIGCOMM*, 2018, pp. 267–281.

[8] M. Zhao, T. Li, M. Abu Alsheikh, Y. Tian, H. Zhao, A. Torralba, and D. Katabi, "Through-wall human pose estimation using radio signals," in *Proc. of IEEE/CVF CVPR*, 2018, pp. 7356–7365.

[9] Y. Ma and E. C. Kan, "Accurate indoor ranging by broadband harmonic generation in passive nltl backscatter tags," *IEEE Transactions on Microwave Theory and Techniques*, vol. 62, no. 5, pp. 1249–1261, 2014.

[10] X. Hui and E. C. Kan, "Radio ranging with ultrahigh resolution using a harmonic radio-frequency identification system," *Nature Electronics*, vol. 2, no. 3, p. 125, 2019.

[11] A. Haniz, G. K. Tran, K. Saito, K. Sakaguchi, J.-i. Takada, D. Hayashi, T. Yamaguchi, and S. Arata, "A novel phase-difference fingerprinting technique for localization of unknown emitters," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 9, pp. 8445–8457, 2017.

[12] M. Youssef and A. Agrawala, "The horus wlan location determination system," in *Proc. of ACM MobiSys*, 2005, pp. 205–218.

[13] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka, "You are facing the mona lisa: Spot localization using phy layer information," in *Proc. of ACM MobiSys*, 2012, pp. 183–196.

[14] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: wireless indoor localization with little human intervention," in *Proc. of ACM MobiCom*, 2012, pp. 269–280.

[15] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, "Push the limit of wifi based localization for smartphones," in *Proc. of ACM MobiCom*, 2012, pp. 305–316.

[16] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: Unsupervised indoor localization," in *Proc. of ACM MobiSys*, 2012, pp. 197–210.

[17] L. Ni, Y. Liu, Y. Lau, and A. Patil, "Landmarc: Indoor location sensing using active rfid," *Wireless networks*, 2004.

[18] J. Wang and D. Katabi, "Dude, where's my card? rfid positioning that works with multipath and non-line of sight," in *Proc. of ACM SIGCOMM*, 2013, pp. 51–62.

[19] S. J. Pan, V. W. Zheng, Q. Yang, and D. H. Hu, "Transfer learning for wifi-based indoor localization," in *Proc. of ACM AAAI workshop*, vol. 6, 2008.

[20] R. Ayyalasomayajula, A. Arun, C. Wu, S. Sharma, A. R. Sethi, D. Vasisht, and D. Bharadia, "Deep learning based wireless localization for indoor navigation," in *Proc. of ACM MobiCom*, 2020, pp. 1–14.

[21] Z. An, Q. Lin, P. Li, and L. Yang, "General-purpose deep tracking platform across protocols for the internet of things," in *Proc. of ACM MobiSys*, 2020, pp. 94–106.

[22] C. Li, Z. Cao, and Y. Liu, "Deep ai enabled ubiquitous wireless sensing: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–35, 2021.

[23] W. Qian, F. Lauri, and F. Gechter, "Supervised and semi-supervised deep probabilistic models for indoor positioning problems," *Neurocomputing*, vol. 435, pp. 228–238, 2021.

[24] C. Zhan, M. Ghaderibaneh, P. Sahu, and H. Gupta, "Deepmtl: Deep learning based multiple transmitter localization," in *Proc. of IEEE WoWMoM*, 2021, pp. 41–50.

[25] Y. Ma, Z. Luo, C. Steiger, G. Traverso, and F. Adib, "Enabling deep-tissue networking for miniature medical devices," in *Proc. of ACM SIGCOMM*, 2018, pp. 417–431.

[26] S. M. Nguyen, D. V. Le, and P. J. Havinga, "Learning the world from its words: Anchor-agnostic transformers for fingerprint-based indoor localization," in *Proc. of IEEE PerCom*, 2023, pp. 150–159.

[27] X. Wang, J. Zhang, S. Mao, S. C. Periaswamy, and J. Patton, "Locating multiple rfid tags with swin transformer-based rf hologram tensor filtering," in *Proc. of IEEE VTC*, 2022, pp. 1–2.

[28] OpenAI, "Chatgpt," https://openai.com/chatgpt, 2023.

[29] Meta, "Large language model (llama) at meta ai," https://ai.meta.com/blog/large-language-model-llama-meta-ai/, 2023.

[30] M. Comiter and H. Kung, "Localization convolutional neural networks using angle of arrival images," in *Proc. of IEEE GLOBECOM*. IEEE, 2018, pp. 1–7.

[31] X. Wang, X. Wang, and S. Mao, "Deep convolutional neural networks for indoor localization with csi images," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 316–327, 2018.

[32] "USRP X310," https://www.ettus.com/all-products/x310-kit/, 2020.

[33] J. Kimionis, A. Bletsas, and J. N. Sahalos, "Increased range bistatic scatter radio," *IEEE Transactions on Communications*, vol. 62, no. 3, pp. 1091–1104, 2014.

[34] S. Labs, "Direction finding using bluetooth low energy," Application Note, 2021. [Online]. Available: https://www.silabs.com/documents/public/application-notes/an1298-direction-finding-using-bluetooth-low-energy.pdf

[35] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[36] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "Landmarc: Indoor location sensing using active rfid," in *Proc. of IEEE PerCom*. IEEE, 2003, pp. 407–415.

[37] Y. Ma, N. Selby, and F. Adib, "Drone relays for battery-free networks," in *Proc. of ACM SIGCOMM*, 2017, pp. 335–347.

[38] Z. Yang, Z. Zhou, and Y. Liu, "From rssi to csi: Indoor localization via channel response," *ACM Computing Surveys (CSUR)*, vol. 46, no. 2, pp. 1–32, 2013.

[39] A. T. Mariakakis, S. Sen, J. Lee, and K.-H. Kim, "Sail: Single access point-based indoor localization," in *Proc. of ACM MobiSys*, 2014, pp. 315–328.

[40] R. Ayyalasomayajula, D. Vasisht, and D. Bharadia, "Bloc: Csi-based accurate localization for ble tags," in *Proc. of ACM CoNEXT*, 2018, pp. 126–138.

[41] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "Spotfi: Decimeter level localization using wifi," in *Proc. of ACM SIGCOMM*, 2015, pp. 269–282.

[42] J. Xiong and K. Jamieson, "Arraytrack: A fine-grained indoor location system," in *Proc. of USENIX NSDI*, 2013, pp. 71–84.

[43] J. Gjengset, J. Xiong, G. McPhillips, and K. Jamieson, "Phaser: Enabling phased array signal processing on commodity wifi access points," in *Proc. of ACM Mobicom*, 2014, pp. 153–164.

[44] Y. Zheng, Y. Zhang, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, "Zero-effort cross-domain gesture recognition with wi-fi," in *Proc. of ACM MobiSys*, 2019, pp. 313–325.

[45] D. Li, J. Xu, Z. Yang, Y. Lu, Q. Zhang, and X. Zhang, "Train once, locate anytime for anyone: Adversarial learning based wireless localization," in *Proc. of IEEE INFOCOM*. IEEE, 2021, pp. 1–10.

[46] U. Raza, A. Khan, R. Kou, T. Farnham, T. Premalal, A. Stanoev, and W. Thompson, "Dataset: Indoor localization with narrow-band, ultra-wideband, and motion capture systems," in *Proceedings of the 2nd Workshop on Data Acquisition to Analysis*, 2019, pp. 34–36.

[47] B.-J. Chen and R. Y. Chang, "Few-shot transfer learning for device-free fingerprinting indoor localization," in *Proc. of IEEE ICC*, 2022, pp. 4631–4636.

[48] I. O. Korkmaz, T. Özateş, E. Koç, E. Aydın, E. Kor, D. Dilek, M. A. Güngen, I. G. Köse, and Ç. Akman, "Indoor localization with transfer learning," in *Proc. of IEEE SIU*, 2022, pp. 1–4.

[49] M. I. AlHajri, R. M. Shubair, and M. Chafii, "Indoor localization under limited measurements: A cross-environment joint semi-supervised and transfer learning approach," in *Proc. of IEEE SPAWC Workshop*. IEEE, 2021, pp. 266–270.