IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 11, 2021, accepted May 3, 2021, date of publication May 6, 2021, date of current version May 14, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3077906

Deep Gated Recurrent Unit-Based 3D Localization for UWB Systems

DOAN TAN ANH NGUYEN[1], (Student Member, IEEE), JINGON JOUNG[1], (Senior Member, IEEE), AND XIN KANG[2], (Senior Member, IEEE)

[1]School of Electrical and Electronics Engineering, Chung-Ang University, Seoul 06974, South Korea
[2]Center for Intelligent Networking and Communications (CINC), University of Electronic Science and Technology of China (UESTC), Chengdu 611731, China

Corresponding author: Jingon Joung (jgjoung@cau.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) Grant through the Korean Government [Ministry of Science and ICT (MSIT)] under Grant NRF-2019R1A2C1084168 and Grant 2021R1A4A2001316.

ABSTRACT The localization system has been extensively studied because of its diverse applicability, for example, in the Internet of Things, automatic management, and unmanned aerial vehicle services. There have been numerous studies on localization in two-dimensional (2D) environments, but those in three-dimensional (3D) environments are scarce. In this paper, we propose a novel localization method that utilizes the gated recurrent unit (GRU) and ultra-wideband (UWB) signals. For the purpose of this study, we considered that the UWB transmitter (Tx) and many UWB receivers (Rx) were placed inside a confined space. The input of the proposed model was generated from the UWB signals that are sent from the Tx to the Rxs, and the output was the location of the Tx. The proposed GRU-based model converts the localization problem into a regression problem by combining the ranging and positioning phase. Thus, the proposed model can directly estimate the location of the Tx. Our proposed GRU-based method achieves 15 and four times shorter execution times for the training and testing, respectively, compared to the existing convolutional neural network (CNN)-based localization methods. The input data can also be easily generated with low complexity. The rows of the input matrix are the downsampled version of the UWB received signal. Throughout numerous simulation results, our novel localization method can achieve a lower root-mean-squared error up to 0.8 meters compared to the recently proposed existing CNN-based method. Furthermore, the proposed method operates well inside a confined space with fixed volume but varying width, height, and depth.

INDEX TERMS 3D localization, deep learning, gated recurrent unit (GRU), recurrent neural network (RNN), ultra-wideband (UWB) system.

I. INTRODUCTION

The rise of the Internet of Things (IoT) has led to an increase in the need for accurate indoor localization systems. A few applications of the location tracking system include robotics [1], location-based services [1], automatic warehouse management system [2], location-based authentication algorithms [3], and surveillance [4]. The information associated with the location of the device is important for indoor as well as outdoor navigation systems in both small-scale and large-scale areas. Indoor localization systems usually rely on wireless technologies, such as wireless-Fidelity (WiFi) [5], [6], Bluetooth, ultra-wideband (UWB), and radio frequency identification device (RFID) [7]. The WiFi technology is relatively easy to implement, possesses high accuracy, and can be implemented at low costs. However, WiFi systems can be remarkably affected by noise. To overcome this problem, it is necessary to implement complex algorithms in WiFi-based localization systems. Bluetooth technology is widely used to connect devices within a specified personal space. It provides large throughput with a wide signal reception range while incurring low energy consumption. However, real-time localization for Bluetooth is difficult owing to the significant delay and noise in the received signal. RFID systems are similar to Bluetooth devices in terms of wide signal reception range, low power consumption, and low localization accuracy. The UWB signal has possesses a few exceptional capabilities including its ability to penetrate various materials and robustness against multipath effect and interference from other devices. Thus, UWB technology

The associate editor coordinating the review of this manuscript and approving it for publication was Qilian Liang.

TABLE 1. Related work on localization systems.

Category	Method	Summary	References
Localization research for 3D environments	Positioning method	A weighted least square system was proposed for 3D positioning that uses information such as time of arrival (ToA), time difference of arrival (TDoA), and direction of arrival (DoA).	[8]
		An alternative positioning method for the trilateration method with lower computational complexity was proposed.	[9]
	RSS, Kalman filter, SVD,	A Kalman filter was used to estimate accurate received signal strength (RSS) to determine the distance between transmitters. A multilateration method that uses singular value decomposition (SVD) for 3D positioning was proposed.	[10]
	PoA, DBSCAN	An RFID-based 3D localization system, which utilizes phase of arrival (PoA) and density-based spatial clustering of applications with noise (DBSCAN) method to select the most likely position, was proposed.	[11]
	AoA, SVD	A 3D localization system, which uses commercial off-the-shelf WiFi infrastructures at low cost, was proposed. A cost-effective hardware scheme was designed to derive the azimuth and elevation angle of arrival (AoA) information for 3D localization. A fusion method of multiple channel state information measurements was adopted with SVD.	[12]
	AoA	A 3D anchor-free localization algorithm which uses both the distance and AoA information to achieve accurate physical node positions for large-scale wireless sensor network (WSN) in a 3D scenario, was proposed.	[13]
	AoA, RSS	A hybrid system that combines RSS and angle measurements in a WSN, respectively, was proposed. A novel nonconvex estimator was derived to find the location of a target node.	[14]
	RSS, autoencoder, KNN	A denoising autoencoder-based Bluetooth low energy (BLE) indoor localization method was proposed. A denoising autoencoder was adopted to extract robust fingerprint patterns from noisy RSS measurements. A 3D fingerprint database was also constructed offline for reference locations in 3D space. The target localtion is estimated using k-nearest-neighbor (KNN).	[15]
	RNN, neural network	A spectral clustering and weighted backpropagation neural networks RSS-based method was proposed for 3D indoor localization.	[16]
Localization research using UWB technology	ToA, AoA	A joint ToA and AoA estimator was proposed for UWB indoor localization systems. The estimator comprises an array of antennas, each feeding a demodulator consisting of a squarer, and a low-pass filter.	[17]
	TDoA	An impulse-based UWB localization system, which uses the time difference of arrival estimation and an error correction algorithm, was designed.	[18]
	PCA, SVM	A localization system, which was based on the principle component analysis (PCA), multiclass support vector machine (SVM), and UWB signals to detect the rooms where the UWB transmitter was located, was designed.	[19]
	RSS, autoencoder, ELM	A RSS-based fingerprinting method that combined an autoencoder and an extreme learning machine (ELM) scheme to predict the location of devices was designed.	[20]
	CNN	A convolutional neural network (CNN) model that estimated the distance between a UWB transmitter and a receiver was proposed. The input was a matrix generated from a received signal.	[21]
		A CNN-based method was designed to estimate the UWB transmitter location. The input was generated from three received signals.	[22]
	ToF, two-way ranging	A 3D localization system based on time of flight (ToF), symmetrical double-sided two-way ranging, and trilateration was proposed.	[23]
	ToF, round trip time	A time estimation method was designed based on the sliding correlation serial searching and polyphase filtering. The location of unknown terminal was acquired through round trip time estimation.	[24]
Recurrent neural network (RNN)-based localization research	RSS, RNN	A recurrent neural network model was designed for the WiFi fingerprinting indoor localization. This method focused on trajectory positioning and uses the correlation among the RSS measurements in a trajectory.	[32]
	LSTM	An LSTM model that estimated location by exploiting channel state information amplitude and phase data feature ratio was designed.	[33]
		A local feature-based deep LSTM approach was developed for WiFi fingerprinting indoor localization.	[34]
		An LSTM network-based localization method, which used distance between transmitter and receivers as input, was proposed.	[35]

is preferred in many localization systems. However, a few drawbacks are associated with UWB, such as high cost, short range, and complex hardware installation.

A. RELATED WORK

The related work for localization systems is summarized in Table 1. Most studies on localization are based on two-dimensional (2D) environments, and only a few are based on three-dimensional (3D) localization systems [8]–[16]. The UWB technology is widely used in indoor localization systems because of characteristics such as high definition, robustness against multipath effect and can penetrate various obstacles. Thus far, most UWB-based localization systems have been for 2D environments [17]–[22]. There is a paucity of studies concerning UWB-based localization systems for 3D environments [23], [24]. A few localization systems used convolutional neural networks (CNNs) to estimate the location of the device from the input image that is generated from the received signal. While CNNs are popular in computer vision applications, recurrent neural network (RNN) is a more reasonable model for processing sequential data [25] such as natural language processing [26], sentiment analysis [27], and speech recognition [28]. Because UWB signal is a type of sequential data, RNN is preferred for processing UWB signals. Long short-term memory (LSTM) [29] and gated recurrent unit (GRU) [30] are two widely used RNN architectures because they can efficiently capture long term dependencies, as well as mitigate vanishing or exploding gradients in training [31]. Several RNN-based localization methods have been developed [32]–[35]. Compared with GRU, LSTM structure operates well with complicated time series data where multiple time steps must be considered, because it contains three gates (input, output and forget) with a cell state [31], [36]. In contrast, the GRU structure only has two gates (update and reset) and a hidden state, which is simpler than that of the LSTM; therefore, the GRU training progress converges faster, especially with low-complexity time-series input data. Given that the UWB signal is a simple sequence data type, GRU is preferred over LSTM for processing UWB signals.

B. OUR CONTRIBUTION

In this paper, we propose a novel GRU-based localization method that utilizes the UWB technology.

- The proposed model uses UWB signal from different receivers (Rxs) as input. Because the environment is dense and large, the signals are prone to noise interferences. This can be addressed by implementing a specific procedure that extracts important information from the signals and uses it to mitigate the noise. To generate input, the signal received from each receiving antenna is downsampled. The number of input features is determined by the number of received signals. Furthermore, downsampling helps decrease the computational complexity of the proposed model.
- A deep GRU-based 3D indoor localization method, which utilizes the UWB signals, is proposed herein. After extracting the crucial information from the data to reduce the noise and obtaining a robust representation of the received signal, the deep GRU-based model is operated to encode temporal information and learn complex representation of the data, which improves the performance of the 3D localization.
- The proposed model estimates the location directly, as opposed to previous methods such as time of arrival (ToA)-based and CNN-based distance estimation [21], which require an additional positioning phase to find the location. Furthermore, the proposed method is robust in an asymmetrical space, which is traditionally problematic for the positioning phase.
- Several simulations were performed in many different 3D environments, across a wide range of signal-to-noise ratio (SNR) values and space volumes. The proposed method improves the localization accuracy as compared to previous methods.

The remainder of this paper is organized as follows. In Section II the system model, UWB model, channel models (CMs), operating space, and the antenna setup are demonstrated. In Section III, the novel GRU-based localization method is detailed. In Section IV, several conventional localization methods are described. In Section V, two positioning phase methods are explained. In Section VI, the localization performance of the conventional localization methods and the proposed localization method are analyzed with respect to various parameters. Finally, Section VII presents the conclusions as well as future works.

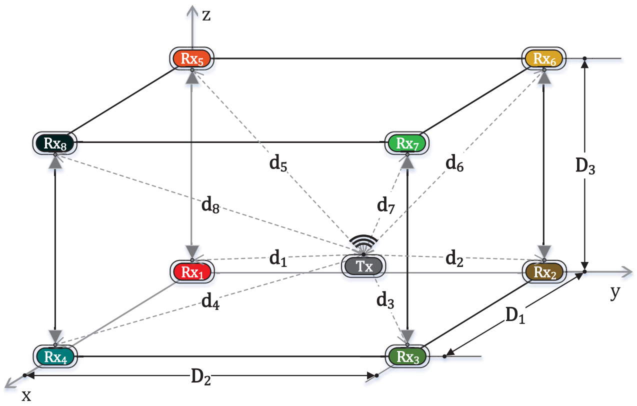


FIGURE 1. Model of 3D localization environment with one transmitter (Tx) and P receivers (Rxs), i.e., $P = 8$, where the position of Tx is estimated based on the signals received at the eight Rxs.

II. SYSTEM AND SIGNAL MODELS

In this section, we explain the system model, signal model, and the environments for 3D localization.

A. LOCALIZATION SYSTEM MODEL

In this study, the localization system was operated inside a cubic space as illustrated in Fig. 1. The space has a volume $D_1 \times D_2 \times D_3$, where D_1, D_2, and D_3 represent the width, depth, and height of the space, respectively. The origin point $(0, 0, 0)$ is positioned at the far left bottom in the confined space, as shown in Fig. 1. There are P number of receivers (Rxs) fixed around this space. The coordinates of the transmitter (Tx) and Rx$_p$ are (x, y, z) and (x_p, y_p, z_p) respectively, where $p \in \{1, 2, \ldots, P\}$. The location of Tx is a random point inside this space, that is, $x \sim \mathcal{U}[0, D_1]$, $y \sim \mathcal{U}[0, D_2]$, and $z \sim \mathcal{U}[0, D_3]$, where $\mathcal{U}[a, b]$ denotes the uniform distribution inside the interval $[a, b]$. The distance between Tx and Rx$_p$ is denoted as d_p, where $d_p = \sqrt{(x - x_p)^2 + (y - y_p)^2 + (z - z_p)^2}$.

The aim of the localization system is to estimate the unknown position of the Tx based on the measurements gathered from all Rxs. The localization process is performed in the following two phases: the ranging phase, where the distances between Tx and Rxs are estimated from the measurements; and the positioning phase, where the actual location of the Tx is calculated from the estimated distances.

In the ranging phase, each Rx estimates the distance from the Tx by using the measurements extracted from the received signals, such as angle of arrival (AoA), ToA, and time difference of arrival (TDoA). In a UWB-based localization system, time-based measurements, such as ToA or TDoA, are usually deployed for ranging because of the ability of UWB signals to resolve the multipath effects and penetrate various obstacles [37]. However, in practice, the time resolution of Rx is inefficient, and it cannot resolve all multipath components. Furthermore, the strongest signal is often not the first component that arrives in cluttered environments such as indoor locations. The effect of dense multipath components is more apparent in UWB systems owing to the strongly dispersive nature of UWB channels [38]. Besides the multipath effect, the complex statistical characteristics of UWB channels make the mathematical analysis for UWB localization systems significantly complicated. A statistical model of small-scale fading is constructed based on the environments where the UWB system operates. Therefore, alternative amplitude distributions can be considered [39], indicating that advanced signal processing techniques, according to the CMs, would be required.

For the positioning phase, the 3D trilateration is the preferred solution. However, the positioning phase may propagate the error that emerges from the estimated distances. In contrast, machine learning algorithms do not require a positioning phase, as they utilize complex statistical characteristics of UWB signals and CMs. A machine learning-based localization system was trained in this study to estimate the location of Tx directly, rather than estimating the distance between the Tx and Rx; therefore, the positioning phase becomes redundant.

B. SIGNAL AND CHANNEL MODELS

In this study, the IEEE 802.15.4a standard [40] was considered for the 3D localization system. Assuming that the Tx sends an UWB signal to Rx_p, the received signal $r_p(t)$ captured at Rx_p is modeled as follows [40]:

$$r_p(t) = h_p(t) * s(t) + n_p(t), \quad \forall p \in \{1, 2, \ldots, P\}, \tag{1}$$

where $h_p(t)$ represents the impulse response of the channel between Tx and Rx_p, $s(t)$ is the transmitted signal at time t, $n_p(t)$ represents an additive white Gaussian noise at Rx_p, and $*$ denotes the convolution operation. Here, the channels are modeled as follows [40]:

$$h_p(t) = \sum_{l=0}^{L_p-1} \sum_{k=0}^{K_l-1} a_{k,l} \exp(j\phi_{k,l})\delta(t - T_l - \tau_{k,l}), \tag{2}$$

where L_p is the total number of clusters that conforms to Poisson distribution; K_l is the total number of multipath components of the lth cluster that is determined by the power delay profile of the channels; $a_{k,l}$ is the tap weight of the kth multipath component of the lth cluster; $\phi_{k,l}$ is the uniformly distributed phase of the kth multipath component of the lth cluster; $\delta(\cdot)$ denotes the delta function; T_l denotes the delay for the lth cluster; and $\tau_{k,l}$ is the intra-cluster delay for the kth multipath component of the lth cluster. The channel impulse response $h_p(t)$ is modeled based on the environment of the localization. In [40], the description for each CM is specified, wherein the residential, office, suburban, industrial, and the open outdoor channels are considered with the case of line-of-sight (LoS) and non-line-of-sight (NLoS). The CMs used in this study are classified in Table 2. Here, a mean value of the total number of clusters (denoted by $\overline{L_p}$) and the average value of the total number of multipath components (denoted by $\overline{K_l}$), which are obtained from the considered channel models, are shown. The signal obtained after matched filtering at Rx_p, denoted by $y_p(t)$, is represented as [41]:

$$y_p(t) = s(T_d - t) * r_p(t), \tag{3}$$

where T_d is the delay for causality of the signals.

TABLE 2. Channel model (CM) notation and environments [40].

CM	Environment	$\overline{L_p}$	$\overline{K_l}$
CM1	Residential, LoS	3	21.6
CM2	Residential, NLoS	3.5	28.4
CM3	Office, LoS	5.4	153
CM4	Office, NLoS	1	208
CM5	Outdoor (sub-urban), LoS	13.6	81.4
CM6	Outdoor (sub-urban), NLoS	10.5	77
CM7	Industrial, LoS	4.75	352
CM8	Industrial, NLoS	1	1183
CM9	Open Outdoor (Snow-covered, farm)	3.31	1.2

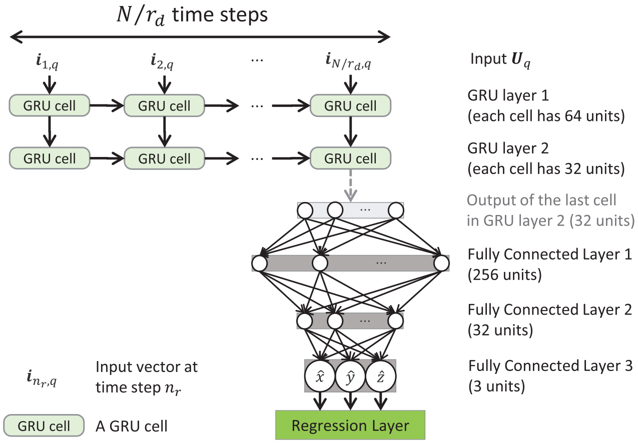


FIGURE 2. GRU-LE model architecture. The input is generated from *P* received UWB signals, and the output is the 3D location of Tx.

III. PROPOSED LOCALIZATION METHODS

In this section, a novel GRU-based method for 3D localization (referred to as the GRU-LE method hereinafter) is introduced.

The architecture of the GRU-LE model is illustrated in Fig. 2. The model consists of an input layer, two GRU layers, three fully connected layers, and a regression layer. The output of the model is the estimated 3D location of Tx. The input data, which is generated from the received UWB signals, is a matrix $U_q \in \mathbb{R}^{P\times(N/r_d)}$, where P represents the number of data features, which is the same number of

Rxs, and N/r_d represents the number of time steps for each training sample.

A. GRU-LE INPUT GENERATION

To generate Q samples for training the GRU-LE model, the received signal $y_{p,q}(t)$ is recorded at Rx_p, with $p \in \{1, \ldots, P\}$ and $q \in \{1, \ldots, Q\}$. The signal is then sampled with a frequency f_s into a complex valued sequence $y_{p,q}[n]$ with N elements, $n \in \{1, \ldots, N\}$. Here, N is set to be 3,600. Further, a standardized real-value vector $v_{p,q}$ is generated as follows:

$$v_{p,q}[n] = \frac{\left|y_{p,q}[n]\right| - \mu_{p,q}}{\sigma_{p,q}}, \tag{4}$$

$$\mu_{p,q} = \frac{1}{N}\sum_{n=1}^{N}\left|y_{p,q}[n]\right|, \tag{5}$$

$$\sigma_{p,q} = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N}\left(\left|y_{p,q}[n]\right| - \mu_{p,q}\right)^2}, \tag{6}$$

where $\mu_{p,q}$ and $\sigma_{p,q}$ are the mean and standard deviation of the real-valued sequence $\left|y_{p,q}[n]\right|$. The result of (4) is shown in Fig. 3(a). In the next step, the vector will be downsampled. This process not only helps eliminate the noise and capture the essence of the signal but also reduces the GRU-LE model complexity by decreasing the input size. A new vector $u_{p,q}$ is generated based on the downsampling rate r_d, as follows:

$$u_{p,q}[n_r] = \max_{n \in \{(n_r-1)r_d+1, \ldots, n_r r_d\}} v_{p,q}[n], \tag{7}$$

with $n_r \in \{1, \ldots, N/r_d\}$. The result of (7) with $r_d = 30$ is illustrated Fig. 3(b). Comparing $u_{p,q}$ to $v_{p,q}$, it is clear that the important values and shape of the signal are retained, while the noise and length of the vector are reduced significantly. Finally, the input matrix U_q is generated from all of $u_{p,q}$ recorded in P Rxs:

$$U_q = \begin{bmatrix} u_{1,q} \\ \vdots \\ u_{P,q} \end{bmatrix} \in \mathbb{R}^{P \times (N/r_d)}. \tag{8}$$

The result of (8) is used as the input for the GRU-LE model illustrated in Fig. 2. The visualization of U_q is shown in Fig. 3(c). Each row represents the signal received from the corresponding Rx. The bright spots indicate the high values while the dark areas indicate low values. Evidently, the pattern of the high values in every row is related to the distance between Tx and the corresponding Rx; therefore, the pattern of matrix U_q is dependent on the location of Tx. Thus, the GRU-LE model can learn from the input data and estimate the Tx location. The input matrix can be divided into many column vectors based on the time steps, that is, $U_q = \left[i_{1,q} \cdots i_{N/r_d,q}\right]$, where $i_{n_r,q} \in \mathbb{R}^{P\times 1}$ denotes the input vector at time step n_r, as shown in Fig. 2.

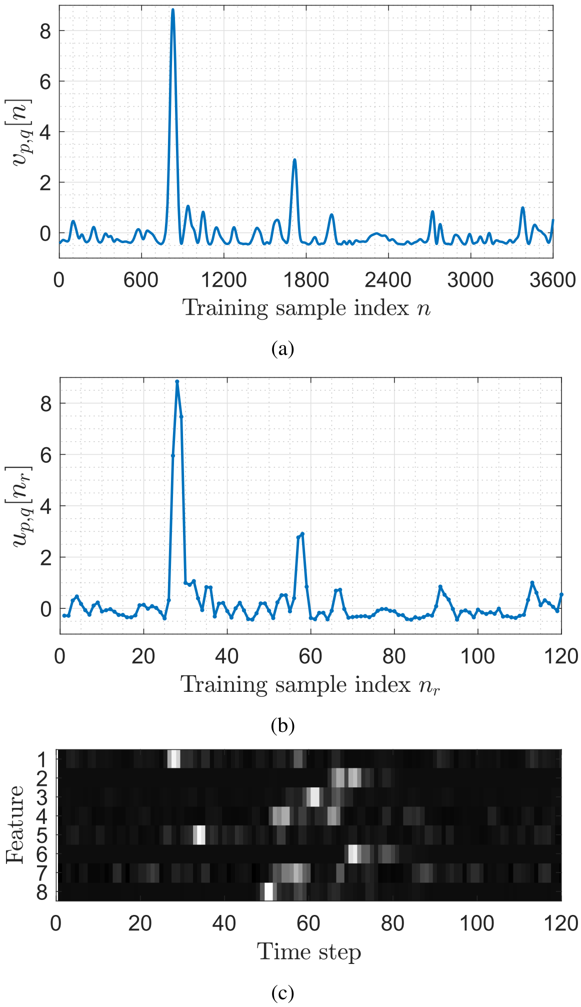


FIGURE 3. GRU-LE input generation. (a) The standardized vector $v_{p,q}$ generated from the received signal. (b) The vector $u_{p,q}$ downsampled from $v_{p,q}$. (c) The input matrix U_q for GRU-LE model.

B. GRU-LE ARCHITECTURE

We consider a GRU cell in lth GRU layer operating at a time step t whose structure is shown in Fig. 4. The vector $g_{l,t} \in \mathbb{R}^{I_g \times 1}$ is the input vector of the GRU cell at time step t. The vector $c_{l,t} \in \mathbb{R}^{C_g \times 1}$ is the output vector of this cell at time step t. The length of the output vector, C_g, is considered as the number of units for the GRU cell. The operation inside the cell is described as follows:

$$f_{l,t} = \sigma\left(W_f g_{l,t} + R_f c_{l,t-1} + b_f\right), \tag{9}$$

$$m_{l,t} = \sigma\left(W_m g_{l,t} + R_m c_{l,t-1} + b_m\right), \tag{10}$$

$$\tilde{c}_{l,t} = \tanh\left(W_{\tilde{c}} g_{l,t} + R_{\tilde{c}}\left(m_{l,t} \odot c_{l,t-1}\right) + b_{\tilde{c}}\right), \tag{11}$$

$$c_{l,t} = \left(1 - f_{l,t}\right) \odot c_{l,t-1} + f_{l,t} \odot \tilde{c}_{l,t}, \tag{12}$$

where $f_{l,t}, m_{l,t}, \tilde{c}_{l,t} \in \mathbb{R}^{C_g \times 1}$ indicate the update gate, reset gate and candidate activation of GRU cell in lth GRU layer at time step t; $W_f, W_m, W_{\tilde{c}} \in \mathbb{R}^{C_g \times I_g}$ denote the input weights for $f_{l,t}, m_{l,t}$, and $\tilde{c}_{l,t}$ respectively; $R_f, R_m, R_{\tilde{c}} \in \mathbb{R}^{C_g \times C_g}$ denote the recurrent weights for $f_{l,t}, m_{l,t}$, and $\tilde{c}_{l,t}$ respectively; $b_f, b_m, b_{\tilde{c}} \in \mathbb{R}^{C_g \times 1}$ indicate the biases for $f_{l,t}, m_{l,t}$,

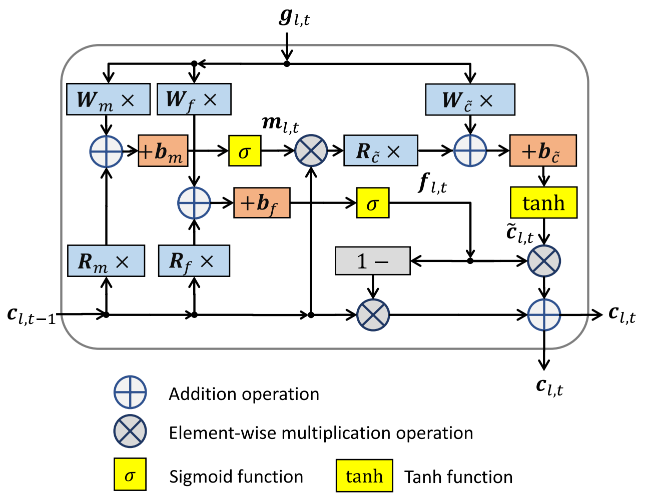


FIGURE 4. Structure of a GRU cell in the lth GRU layer at time step t.

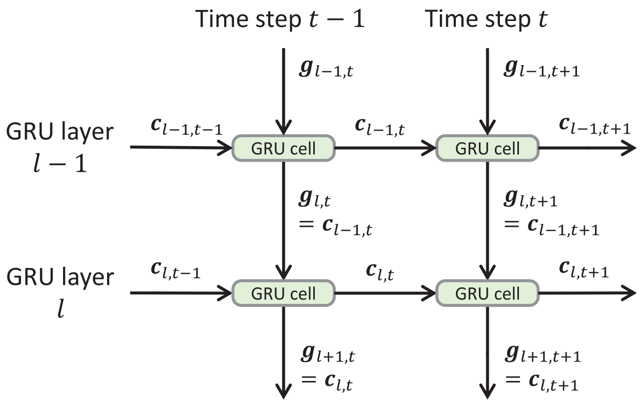


FIGURE 5. Deep GRU-based network structure. $g_{l,t}$ and $c_{l,t}$ denote the input and output for GRU cell in GRU layer l at time step t, respectively.

and $\tilde{c}_{l,t}$ respectively; $\sigma(\cdot)$ and $\tanh(\cdot)$ denote the sigma function and the tanh function, respectively; and $\odot$ denotes the element-wise multiplication operation.

A GRU cell can encode the non-temporal structure of the input in a single time step, and summarize the output from previous time steps. It is assumed that the GRU network is a "deep network" if it processes data with large number of time steps. This network can also extend the depth differently by stacking multiple GRU layers on top of each other. Fig. 5 illustrates the structure of a deep GRU network. The output of the previous GRU layer is the input of the next GRU layer at the same time step. Similar to RNN networks, stacking GRU layers helps the model learn data representation at different timescales [43]. For the GRU-LE model, the two GRU layers form the most crucial part because they are designed to learn the time series input. By stacking two GRU layers on top of each other, the levels of abstraction of input over time are added. As a result, the final cell of the last GRU layer learns all of the complex information not just from the previous cells but also from the previous layer. Thus, it is appropriate to use the output of this cell as the input for the next layer.

Algorithm 1 GRU-LE Localization Algorithm

```
Collect the received UWB signals $y_1(t)$, $y_2(t)$,..., and $y_P(t)$ in (3) from Rx$_1$, Rx$_2$,..., and Rx$_P$ respectively.
Generate the standardized real-valued sequences of the received signals, i.e., $v_1[n]$, $v_2[n]$,..., and $v_P[n]$ in (4).
Generate the downsampled sequences, i.e., $u_1[n_r]$, $u_2[n_r]$,..., and $u_P[n_r]$ in (7).
Generate the input matrix of GRU-LE, i.e., $U \in \mathbb{R}^{P\times(N/r_d)}$ in (8).
Provide $U$ to GRU-LE shown in Fig. 2.
Obtain the estimation $(\hat{x}, \hat{y}, \hat{z})$ of the location $(x, y, z)$ from the GRU-LE output.
```

As illustrated in Fig. 2, the GRU-LE model is designed with two GRU layers, and each layer contains the same number of GRU cells as the number of time steps in the input matrix U_q, that is, N/r_d. In time step n_r, the input vector of the cell in the first GRU layer is $i_{n_r,q}$. As regards the numerical results, the number of units for each cell in the first and second GRU layers is 64 and 32, respectively. Similarly, the number of units for the first two fully connected layers is selected to be 256 and 32, as they afford ideal results. Furthermore, stacking two fully connected layers on top of each other helps the model learn complex function more efficiently than having only one fully connected layer with the same number of units. The input for the first fully connected layer is the output of the GRU cell in the last time step in the second GRU layer. The final fully connected layer requires three units to represent the 3D coordinate of the Tx location. The output is the estimated location of the Tx. The regression layer is placed at the end of the model to calculate the loss between the output of the last fully connected layer and the ground truth of location, where the loss is the half mean squared error (MSE), as follows [42]:

$$L_{Tx} = \frac{1}{2Q}\sum_{q=1}^{Q}\left[\left(x_q-\hat{x}_q\right)^2+\left(y_q-\hat{y}_q\right)^2+\left(z_q-\hat{z}_q\right)^2\right], \qquad (13)$$

where (x_q, y_q, z_q) and $(\hat{x}_q, \hat{y}_q, \hat{z}_q)$ are the ground truth and the estimation of Tx position for the qth training sample, respectively. The model training process aims to minimize the loss L_{Tx} in (13) with Q training samples.

In summary, Algorithm 1 describes the localization procedure after the GRU-based model is trained, and the training index q is omitted.

IV. CONVENTIONAL LOCALIZATION METHODS

In this section, we briefly introduce three conventional localization methods:

i) ToA-based method [44].
ii) CNN-based distance estimation (CNN-DE) method [21].
iii) CNN-based location estimation (CNN-LE) method [22].

A. CONVENTIONAL ToA-BASED METHOD

The threshold-based ToA estimation method was proposed in [44]. From $y_p(t)$, which is calculated in (3), the Rx$_p$ estimates the ToA when the energy of the received signal is higher than the predetermined threshold. To increase the accuracy of ToA estimation method, a typical threshold value is decided such that it is inversely proportional to the average of the received SNR. As a result, in the ranging phase of this method, the Rx$_p$ can find the estimated distance $\hat{d}_p$ from the Tx by multiplying the estimated ToA with the speed of light, that is, $\hat{d}_p = \hat{\Delta}_p \times c$, where $\hat{\Delta}_p$ is the estimated ToA at Rx$_p$ and c denotes the speed of light. After the ranging phase, the estimated distances are used to calculate the actual Tx location in the positioning phase.

B. CONVENTIONAL CNN-DE METHOD

Similar to threshold-based ToA estimation method, the CNN-DE method based on [21] has a ranging and positioning phase. In the ranging phase, this method uses a CNN model to estimate the distance between the Tx and Rx$_p$ from a one-channel image that is generated from the received signal. As a result, P CNN models are required for P Rxs. This is a major drawback of CNN-DE method. Given that the GRU-LE method only needs one deep learning model to operate for all of Rxs, it is more beneficial. Because all of the Rxs perform the same function, we will demonstrate the CNN-DE method inside one Rx$_p$, without the loss of generality.

The CNN-DE model is illustrated in Fig. 6. The model consists of an input layer, four main groups of hidden layers, a fully connected layer with one unit, and a regression layer at the end. The input signal is a one-channel image that is generated from the received signal at Rx$_p$. Each hidden layer group contains a convolutional layer with filters size 3 × 3 and stride one, a batch normalization layer, and a rectified linear unit (ReLU) layer. The first three groups are connected to a max-pooling layer with filter size 2 × 2 and stride two. The output represents the estimated distance between the Tx and Rx$_p$.

The CNN model is trained on Q training samples that are generated from the received signal $y_{p,q}(t)$, $\forall q \in \{1, \ldots, Q\}$. At the Rx$_p$, the received signal is sampled with a frequency f_s into a complex valued sequence $y_{p,q}[n]$ with N elements ($n \in \{1, \ldots, N\}$). In this method, N is set to be 3,600. Further, a real-valued vector $v_{p,q}$ is generated as follows:

$$v_{p,q}[n] = \frac{\left|y_{p,q}[n]\right|}{\max\limits_{m\in\{1,\ldots,N\}} \left|y_{p,q}[m]\right|}. \tag{14}$$

The vector $v_{p,q}$ is shown is Fig. 7(a). This vector is reshaped into a $M \times (N/M)$ matrix $V_{p,q}$:

$$V_{p,q} = \mathrm{mat}[v_{p,q}]_M \in \mathbb{R}^{M\times(N/M)}, \tag{15}$$

where mat$[\cdot]_M$ denotes the matricization function of a column-major order and dimension M. In this method, M is set to 60 so that the CNN model input is a square image with the size of 60 × 60. The $V_{p,q}$ is a one-channel image, and is used as an input for the CNN model.

A monochrome image of the input is illustrated in Fig. 7(b). The bright and dark pixels indicate high and low values, respectively. Evidently, the bright stripe location in the image

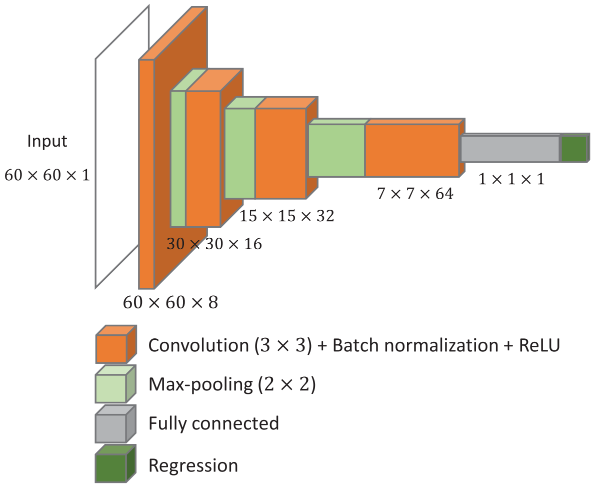
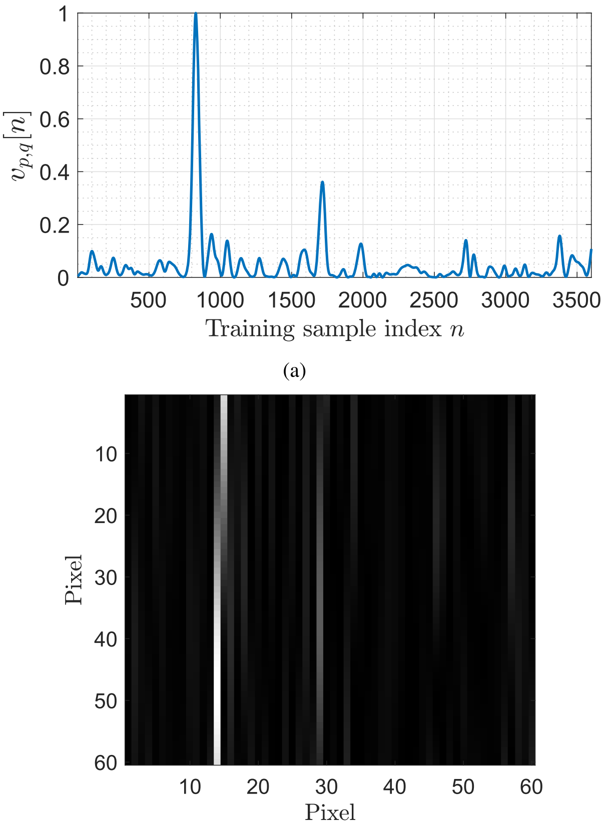


FIGURE 6. CNN-DE model architecture. Each Rx requires a model to estimate the distance, thus P CNN-DE models are necessary for the CNN-DE method.



(a)

(b)

FIGURE 7. CNN-DE input generation. (a) The normalized vector $v_{p,q}$ generated from the received signal. (b) The input image $V_{p,q}$ for the CNN-DE model.

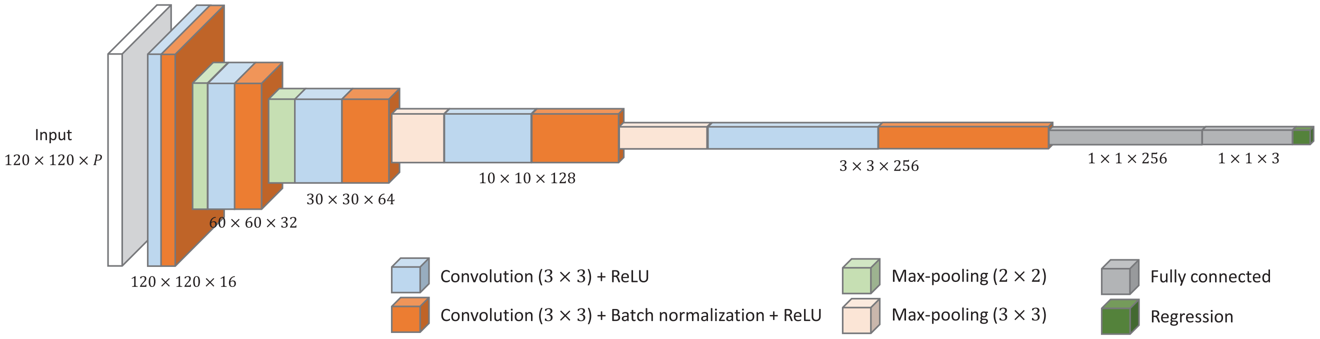


FIGURE 8. **CNN-LE model architecture. The input is generated from *P* received UWB signals, and the output is the Tx 3D location.**

is closely related to the location of the high values in $v_{p,q}$. Thus, the largest multipath component (MPC) and the number of local peaks before the global peak correspond to the bright and moderately light pixels in the image, and they are related to the distance between the Tx and Rx_p. As a result, the CNN model can provide accurate results through this pattern.

The output is the estimated distance $\hat{d}_{p,q}$ between the Tx and Rx_p. The regression layer is placed at the end of the model to calculate the loss between the output of the fully connected layer and the ground truth of distance, where the loss is the half MSE as follows [42]:

$$L_d = \frac{1}{2Q}\sum_{q=1}^{Q}\left[\left(d_{p,q} - \hat{d}_{p,q}\right)^2\right], \tag{16}$$

where $d_{p,q}$ is the ground truth of the distance between the Tx and Rx_p. The model training process is aimed at minimizing the loss L_d in (16) with Q training samples. The ranging phase of the CNN-DE method is completed once the estimated distances from all Rxs are gathered. Further, the positioning phase is implemented to calculate the location of Tx based on the estimated distances, similar to the threshold-based ToA method. The separation of the ranging and positioning phase is another disadvantage of the CNN-DE method, which is contrary to the GRU-LE method wherein the deep learning model includes both phases.

C. CONVENTIONAL CNN-LE METHOD

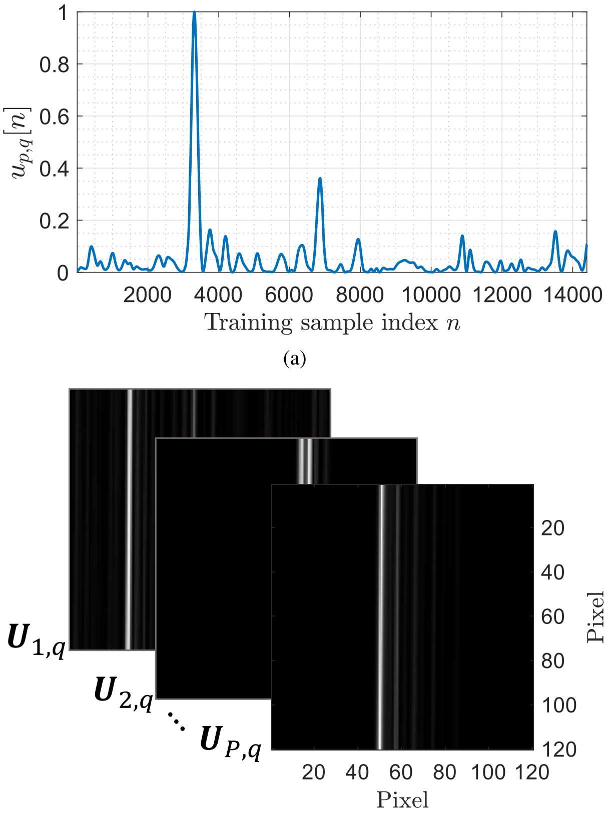
This method, which is mentioned in [22], uses a CNN model to estimate the location of the Tx from the *P*-channel input image that is generated from *P* UWB received signals. Thus, the CNN model combines both the ranging and positioning phase, and the operation needs to be performed only once for all Rxs. As a result, the CNN-LE method outperforms the CNN-DE method, and it is also robust against asymmetrical environment, which is contrary to the CNN-DE method.

The CNN-LE model is described in Fig. 8. This model contains an input layer, five main groups of hidden layers, two fully connected layers with 256 and three units respectively, and a regression layer, which is connected at the end. The input is a *P*-channel image that is generated from the signals received at all Rxs. Each hidden layer group contains two convolutional layers with filters size 3 × 3 and stride one, a batch normalization layer, and two ReLU layers, one between the two convolutional layers and one after the batch normalization layer. Each of the first two groups is connected to a max-pooling layer with filter size 2 × 2 and stride two, and each of the following two groups is connected to a max-pooling layer with filter size 3 × 3 and stride three. The output is the estimated location of Tx. Compared to the deep learning model of the proposed method, the deep learning model of the CNN-LE method is much more complicated, which causes the training time to increase immensely.

The input generation is described in Fig. 9. Based on the signal received at Rx_p, this method also extracts the same vector $v_{p,q}$ with the same elements as in (14). Further, a new vector $u_{p,q}$, shown in Fig. 9(a), is generated by over-sampling $v_{p,q}$ four times, and thus, $u_{p,q}$ has four times the number of elements compared to $v_{p,q}$ while both vectors have the same shape. The vector $u_{p,q}$ is reshaped into a matrix $U_{p,q}$ as follows:

$$U_{p,q} = \mathrm{mat}[u_{p,q}]_{2M} \in \mathbb{R}^{2M\times(2N/M)}. \tag{17}$$

Similar to the input image in CNN-DE, the positions of the bright values in the matrix $U_{p,q}$ are closely related to the distance between the Tx and Rx_p. The same procedure is carried out across all Rxs, thus resulting in *P* matrices from $U_{1,q}$ to $U_{P,q}$. Finally, all of these matrices are stacked on top of each other, as shown in Fig. 9(b), to create a *P*-channel image that is used as an input for the CNN model. As each channel represents the relation to the distance between Tx and the corresponding Rx, it is expected a whole *P*-channel image carries the information of the true location of Tx, which helps the CNN model estimate the Tx position accurately. Compared to the GRU-LE input, the CNN-LE input is 120 times larger, which is a major disadvantage for the CNN-LE method. Finally, the regression layer is aimed at calculating the same loss as the one in (13) between the output of the last fully connected layer and the ground truth of Tx location.



(a)

(b)

FIGURE 9. **CNN-LE input generation. (a) The over-sampled normalized vector $u_{p,q}$ generated from the received signal. (b) The input for the CNN-LE model by stacking all the matrices from $U_{1,q}$ to $U_{P,q}$.**

V. POSITIONING METHODS

The positioning phase is important for estimating the location of the Tx in the conventional threshold-based ToA and CNN-DE methods,. In this section we introduce two positioning methods:

i) 3D trilateration method.
ii) Complexity-reduced trilateration approach (COLA) method [9].

A. CONVENTIONAL TRILATERATION METHOD

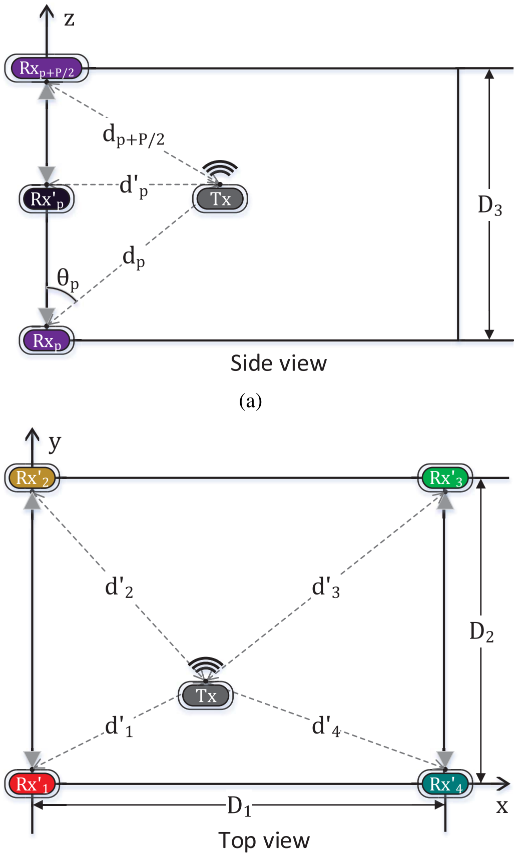
The trilateration is a simple positioning method that is applied after all distances from P Rxs are estimated. Here, we aim to estimate Tx location in 3D, and thus, this method will be referred to as 3D trilateration (3DT). We assume that the Tx is at the coordinate (x, y, z), the Rx$_p$ is located at (x_p, y_p, z_p), and the estimated distance between Tx and Rx$_p$ is $\hat{d}_p$. Thus, we have:

$$\hat{d}_p^2=(x_p-x)^2+(y_p-y)^2+(z_p-z)^2, \quad \forall p\in\{1,\ldots,P\}. \tag{18}$$

The system of quadratic equations in (18) can be converted into a system of linear equations as follows:

$$2x_{p,P}x+2y_{p,P}y+2z_{p,P}z=b_p, \quad \forall p\in\{1,\ldots,P-1\}, \tag{19}$$

where $x_{p,P}=(x_p-x_P)$, $y_{p,P}=(y_p-y_P)$, $z_{p,P}=(z_p-z_P)$, and $b_p=(x_p^2+y_p^2+z_p^2)-(x_P^2+y_P^2+z_P^2)-(\hat{d}_p^2-\hat{d}_P^2)$. The system of equations in (19) can be expressed in matrix form as follows:



(a)

(b)

FIGURE 10. **Model of 3D localization environment with one Tx and P Rx's following COLA method, i.e., $P = 8$. (a) The side view of the space. (b) The top view of the space.**

$$A\begin{bmatrix} x & y & z\end{bmatrix}^T = b, \tag{20}$$

$$A = 2\begin{bmatrix} x_{1,P} & y_{1,P} & z_{1,P} \\ \vdots & \vdots & \vdots \\ x_{P-1,P} & y_{P-1,P} & z_{P-1,P}\end{bmatrix} \in \mathbb{R}^{(P-1)\times 3}, \tag{21}$$

$$b = \begin{bmatrix} b_1 \\ \vdots \\ b_{P-1}\end{bmatrix} \in \mathbb{R}^{(P-1)\times 1}. \tag{22}$$

Thus, the estimated 3D coordinate of the Tx is calculated as follows:

$$\begin{bmatrix} \hat{x} & \hat{y} & \hat{z}\end{bmatrix}^T = (A^TA)^{-1}A^Tb. \tag{23}$$

Because there are three unknowns in the Tx coordinate, the minimum number $P-1$ should be three, i.e., $P \geq 4$, provided the Rxs are not coplanar.

B. CONVENTIONAL COLA METHOD

COLA is a positioning method proposed in [9]. The aim of this method is to reduce the complexity in the trilateration method mentioned above, and obtain higher accuracy. To successfully implement this method, the number P of Rxs must be even. The Rxs are setup according to Fig. 10. Without the loss of generality, we assume that Rx_p and $\mathrm{Rx}_{p+P/2}$ have the same horizontal coordinates but different vertical coordinates, that is, Rx_p is positioned at (x_p, y_p, z_p) and $\mathrm{Rx}_{p+P/2}$ is at $(x_p, y_p, z_{p+P/2})$, with $z_p \neq z_{p+P/2}$, $p \in \{1, \ldots, P/2\}$. After all distances $\hat{d}_p$ from Tx to all the Rxs are estimated, the vertical coordinate of Tx is calculated as follows:

$$\hat{z} = \frac{2}{P}\sum_{p=1}^{P/2}\left(z_p + \hat{d}_p \cos\hat{\theta}_p\right), \tag{24}$$

$$\cos\hat{\theta}_p = \frac{\left(z_{p+P/2} - z_p\right)^2 + \hat{d}_p^2 - \hat{d}_{p+P/2}^2}{2\left(z_{p+P/2} - z_p\right)\hat{d}_p}. \tag{25}$$

We consider that Rx'_p is a virtual Rx that has the same horizontal coordinates as Rx_p and the same vertical coordinates as Tx, as shown in Fig. 10(a). Thus, the Rx'_p is positioned at (x_p, y_p, z). From (25), the distance $\hat{d'}_p$ from Rx'_p to Tx is estimated as follows:

$$\hat{d'}_p = \hat{d}_p \sin\hat{\theta}_p = \hat{d}_p\sqrt{1 - \left(\cos\hat{\theta}_p\right)^2}. \tag{26}$$

The distance $\hat{d}'_p$ is shown in Fig. 10(b). Further, the horizontal coordinates of Tx will be estimated following the 2D trilateration. We have:

$$\hat{d'}_p^2 = (x_p - x)^2 + (y_p - y)^2, \forall p \in \{1, \ldots, P/2\}. \tag{27}$$

The system of quadratic equations in (27) can be transformed into a system of linear equations as follows:

$$2(x_p - x_{P/2})x + 2(y_p - y_{P/2})y = b'_p, \quad \forall p \in \{1, \ldots, P/2-1\}, \tag{28}$$

where $b'_p = (x_p^2 + y_p^2) - (x_{P/2}^2 + y_{P/2}^2) - (\hat{d'}_p^2 - \hat{d'}_{P/2}^2)$. The system of equations in (28) can be expressed in matrix form as follows:

$$H\begin{bmatrix} x & y \end{bmatrix}^T = b', \tag{29}$$

$$H = 2\begin{bmatrix} x_1 - x_{P/2} & y_1 - y_{P/2} \\ \vdots & \vdots \\ x_{P/2-1} - x_{P/2} & y_{P/2-1} - y_{P/2} \end{bmatrix} \in \mathbb{R}^{(P/2-1)\times 2}, \tag{30}$$

$$b' = \begin{bmatrix} b'_1 \\ \vdots \\ b'_{P/2-1} \end{bmatrix} \in \mathbb{R}^{(P/2-1)\times 1}. \tag{31}$$

Thus, the estimated horizontal coordinate of the Tx is calculated as follows:

$$\begin{bmatrix} \hat{x} & \hat{y} \end{bmatrix}^T = (H^T H)^{-1} H^T b'. \tag{32}$$

Because there are two unknowns in the horizontal coordinates of Tx, the minimum number $P/2 - 1$ should be two, that is, $P \geq 6$, provided the horizontal positions of the Rxs from Rx_1 to $\mathrm{Rx}_{P/2}$ are not co-linear.

TABLE 3. Locations of receiver antennas.

Receiver	Coordinate	Receiver	Coordinate
Rx_1	$(0, 0, 0)$	Rx_2	$(0, D_2, 0)$
Rx_3	$(D_1, D_2, 0)$	Rx_4	$(D_1, 0, 0)$
Rx_5	$(0, 0, D_3)$	Rx_6	$(0, D_2, D_3)$
Rx_7	(D_1, D_2, D_3)	Rx_8	$(D_1, 0, D_3)$

TABLE 4. Training options, training time, testing time and model size for each deep learning model.

Options	CNN-DE	CNN-LE	GRU-LE
Solver	SGDM	SGDM	ADAM
Mini batch size	200	200	200
Validation frequency (iterations)	50	100	175
Max epochs	30	30	80
Max epochs (retrain)	15	10	20
Initial learn rate	0.001	0.02	0.02
Initial learn rate (retrain)	0.0001	0.002	0.002
Learn rate drop period	1	2	2
Learn rate drop factor	0.9	0.9	0.9
Gradient threshold	infinite	1	1
# of epochs to converge	29	29	79
Training time (min)	2	105	7
Retraining time (min)	1	35	2
Testing time (sec)	3	12	3
Model size (MB)	0.1	6.3	0.14

VI. SIMULATION RESULTS

In this section, we present the simulated localization performances of the conventional methods and the proposed method. The simulated methods include the threshold-based ToA method with 3DT and COLA for positioning phase, CNN-DE method with 3DT and COLA for positioning phase, CNN-LE method, and the proposed GRU-LE method. Although the UWB localization system is more fitting for indoor environment, we simulate the localization system on all of the CMs in Table 2 to examine performance of the proposed model in different environments. For convenience, the GRU-LE model trained with the CM# data will be referred to as CM#-NET, with $\# \in \{1, \ldots, 9\}$. The environment setup is similar to Fig. 1, and the number of Rxs is set as $P = 8$. The location of each Rx is shown in Table 3. The simulation is performed in `MATLAB 2020b`. For each CM, 40,000 samples are generated for training, and an additional 4,000 samples are generated for testing. The sampling frequency is fixed at $f_s = 24$ GHz. The numbers N, M and r_d are set as 3,600, 60, and 30 respectively, indicating that the input size for CNN-DE, CNN-LE, and the GRU-LE model is 60×60, $120 \times 120 \times 8$, and 8×120, respectively.

The training options for each model are presented in Table 4. The solver is the algorithm that is used to train the deep learning model. Throughout the rigorous numerical verifications, it is observed the CNN-based models achieve better performance with a stochastic gradient descent with momentum (SGDM) optimizer. The GRU-based model, however, performs well with an adaptive moment

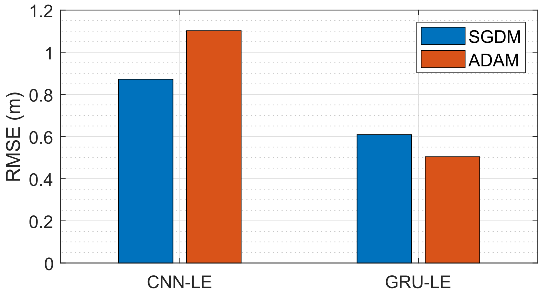


FIGURE 11. RMSEs of the CNN-LE and GRU-LE models with SGDM and ADAM optimizers.

estimation (ADAM) optimizer. For example, under the office environment with LoS, when the space size is 20 × 20 × 20 cubic meters and SNR = 30 dB, the simulation results in Fig. 11 verify that root MSE (RMSE) of the CNN-LE increases 0.2 m compared to that with SGDM. On the other hand, the RMSE of GRU-LE decreases 0.1 m when it is trained with ADAM. Each instance when the entire dataset passes through the network training is considered an epoch. To optimally train the network, the number of epochs must be much larger than one, which is set for the max epochs option. In each epoch, the data is divided into many mini batches, while the mini batch size is the number of samples used for training each iteration. The validation frequency determines the number of iterations for one network validation. The training data are shuffled after every epoch to avoid bias and ensure that the training converges quicker. The initial learn rate determines the size of the first training step. To avoid unstable training where the training loss jumps back and forth over a minimum, the learning rate drops to 90 percent of the prior rate after a certain number of epochs, as set for the learning rate drop period. To prevent the exploding of the gradient, a predefined threshold value is set so that the gradient is clipped if it exceeds the threshold. Throughout the simulations, the number of training samples and training options parameters are selected to ensure the optimal result. The localization RMSE is used as a performance metric, and it is calculated as follows:

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left[\left(x_t - \hat{x}_t\right)^2 + \left(y_t - \hat{y}_t\right)^2 + \left(z_t - \hat{z}_t\right)^2\right]}, \quad (33)$$

where T is the number of samples, $(\hat{x}_t, \hat{y}_t, \hat{z}_t)$ is the estimated Tx location, and (x_t, y_t, z_t) is the ground truth of Tx location of tth sample. The training time, retraining time, testing time, and model sizes in the bottom of Table 4 are recorded by monitoring the training and testing process. The number of epochs required to converge is determined when the validation RMSE does not drop by an amount larger than 0.01 m.

Two minor experiments are carried out for the proposed GRU-LE model to find the optimal number of Rxs and the downsampling rate r_d. However, the main experiments compare the performances of various localization methods in terms of SNR, space size, and asymmetry of the operated space. As a result, we examine the performances of the GRU-LE model that was trained on a different CM compared to the testing CM to find the possibility of transfer learning and subsequently decrease training time.

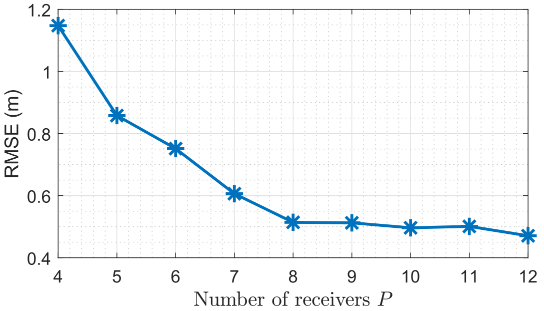


FIGURE 12. RMSE of the GRU-LE model across various number of Rxs.

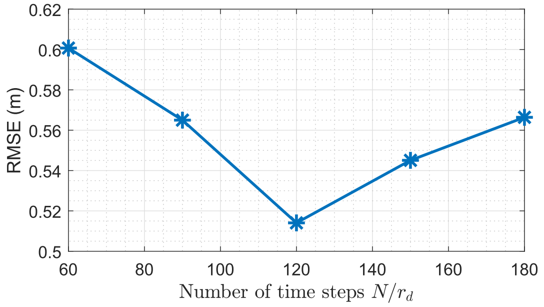


FIGURE 13. RMSE of the GRU-LE model across various number of time steps.

A. PERFORMANCE WITH RESPECT TO NUMBER OF RECEIVERS

This is a small experiment which examines the performance of the proposed GRU-LE model across different number of Rxs. The SNR is fixed at 30 dB, the space size is fixed at $D_1 = D_2 = D_3 = 20$ m, and the downsampling rates r_d is set as 30. As illustrated in Fig. 12, the RMSE decreases as the number of Rx's increases. However, the RMSE evens out after eight Rxs. Thus, the reasonable number of Rxs required for the operation of the GRU-LE model is eight.

B. PERFORMANCE WITH RESPECT TO DOWN-SAMPLING RATE

This experiment compares the RMSE of the proposed GRU-LE model across different downsampling rates r_d. The SNR is fixed at 30 dB, the space size is set at $D_1 = D_2 = D_3 = 20$ m, and the number of Rxs is set as $P = 8$. The simulation result is shown Fig. 13. Although the RMSE is within 0.1 m across different rates, the rate $r_d = 30$ results in the lowest RMSE, and thus $r_d = 30$ is chosen for this study.

C. PERFORMANCE WITH RESPECT TO SNR

In the first major experiment, the space size is fixed with $D_1 = D_2 = D_3 = 20$ m. Fig. 14 shows the localization accuracy of all methods across different SNRs. The proposed GRU-LE method outperforms all other methods across all

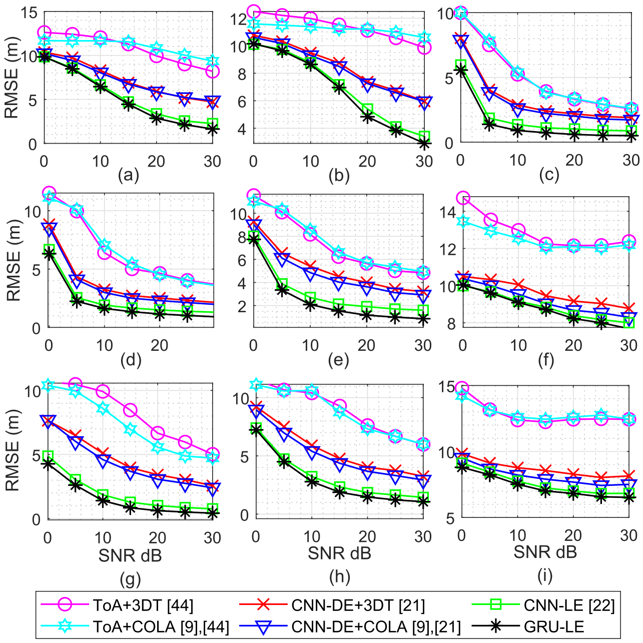


FIGURE 14. RMSE performance of the proposed GRU-LE (y-axis) of various schemes across SNRs (x-axis). Training and testing data are generated from the same CM. (a) CM1. (b) CM2. (c) CM3. (d) CM4. (e) CM5. (f) CM6. (g) CM7. (h) CM8. (i) CM9.

SNRs and CMs. Generally, the RMSE decreases with the increase in SNR for all channels and methods. This is because a high SNR signal is more robust against multipath effect and noise, which results in a clearer input pattern, thus resulting in a high accuracy. For indoor environments, the model works much better in the LoS CMs as compared to the NLoS CMs. The GRU-LE model operates the best in CM3 and CM7, and yields a RMSE below 1 m as the SNR increases over 10 dB. The performance of the proposed model in residential environments is the worst among all indoor CMs, which is caused by the number of local peaks in the UWB signal for the residential environment. It is worth being emphasized again that the proposed GRU-LE method outperforms other methods as shown in Table 5. Here, the average values of the RMSEs in Fig. 14 are summarized.

Fig. 15 shows the difference between the input generated in CM1 and CM3. The large number of bright spots per feature in CM1 input corresponding to the number of local peaks in the UWB signal results in a complex input pattern. In contrast, the pattern of the CM3 input is clearer which helps the deep learning model train efficiently. In outdoor cases, the performance of the model degrades significantly when operating in NLoS environment as compared to LoS one. By comparing the two positioning methods, it is observed the COLA operates slightly better than the 3DT when they are used in the CNN-DE method.

Fig. 16 illustrates the RMSE of the proposed GRU-LE with the CM#-NET when it is operating on various CMs test data across different SNRs. The RMSEs in Fig. 16 are averaged over SNR and shown in Table 6. As expected, for the test data CM#, the CM#-NET operates best (see the underlined RMSE value which is the smallest in each column.). On the other hand, a GRU-LE model trained for a specific indoor CM can perform well in other indoor CMs, and the same can be said for outdoor CMs. The CM1-NET and CM2-NET models show similar performances when they operate in each other's environment, contrary to the performances in respective environments. In other words, the CM1-NET and CM2-NET are a pair of interchangeable models. The same can be said for the pair of CM4-NET and CM8-NET. In summary, the GRU-LE performs the best when the training and testing data are from the same CM. From these results, we can surmise that transfer learning can be employed to reduce the learning time.

TABLE 5. RMSE (meters) of the proposed GRULE averaged over SNRs in Fig. 14.

Methods	CM1	CM2	CM3	CM4	CM5	CM6	CM7	CM8	CM9
ToA+3DT [44]	10.78	11.39	5.06	6.46	7.36	12.90	8.24	8.90	12.85
ToA+COLA [9], [44]	10.96	11.21	5.14	6.54	7.50	12.49	7.30	8.77	12.89
CNN-DE+3DT [21]	7.30	8.49	3.35	3.73	5.20	9.61	4.60	5.42	8.68
CNN-DE+COLA [9], [21]	7.19	8.33	3.16	3.50	4.82	9.21	4.32	5.08	8.13
CNN-LE [22]	5.42	6.95	1.87	2.42	3.13	8.89	1.96	3.25	7.60
Proposed GRU-LE	5.12	6.71	1.46	2.09	2.53	8.76	1.52	2.87	7.35

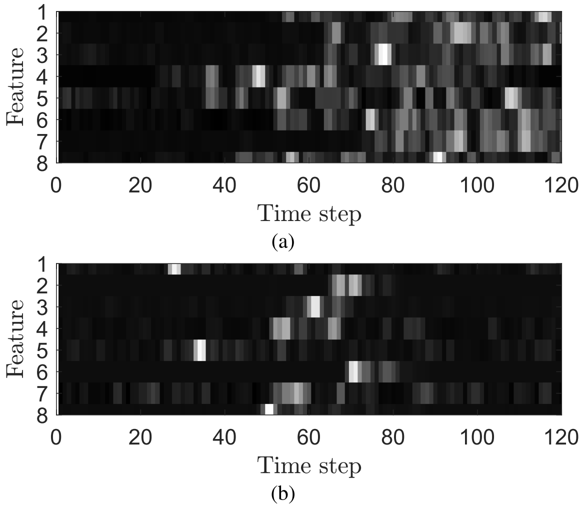


FIGURE 15. The difference between CM1 input and CM3 input for GRU-LE model. (a) CM1 input. (b) CM3 input.

D. PERFORMANCE WITH RESPECT TO VOLUME SIZE

Assuming that the localization methods for the UWB system mainly operate in indoor environments, SNR = 30 dB is ideal for the second major experiment. Fig. 17 demonstrates the RMSE of the conventional methods with the proposed GRU-LE method. Generally, the RMSE increases with the space size for all methods and CMs. The increase in the RMSE is expected because a longer distance between Tx and each Rx causes a larger path loss. Similar to the previous experiment, the COLA performs much better than 3DT when the distance estimation method is fairly accurate,

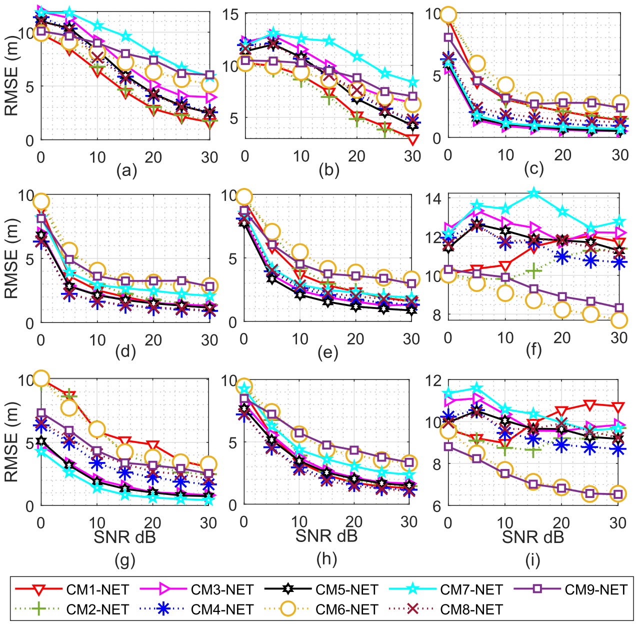


FIGURE 16. RMSE performances (y-axis) across SNRs (x-axis). The CM#-NET trained by CM# is tested with (a) CM1, (b) CM2, (c) CM3, (d) CM4, (e) CM5, (f) CM6, (g) CM7, (h) CM8, (i) CM9.

TABLE 6. RMSE (meters) of CM#-NET averaged over SNRs in Fig. 16 when they are tested with CM# test data.

GRU-LE	CM1	CM2	CM3	CM4	CM5	CM6	CM7	CM8	CM9
CM1-NET	5.12	6.99	3.52	3.01	3.96	11.14	5.88	3.55	9.98
CM2-NET	5.33	6.71	3.76	3.48	4.25	10.51	5.27	4.00	9.19
CM3-NET	7.50	9.70	1.46	2.58	2.86	12.44	2.06	3.70	10.16
CM4-NET	6.50	8.78	2.02	2.09	3.03	11.48	3.31	3.04	9.40
CM5-NET	6.51	8.50	1.58	2.51	2.53	11.86	2.00	3.48	9.75
CM6-NET	7.36	8.42	4.47	4.45	5.30	8.76	5.51	5.45	7.54
CM7-NET	9.21	11.20	1.73	3.50	3.46	13.13	1.52	4.55	10.44
CM8-NET	6.35	8.77	2.31	2.13	3.21	11.68	3.68	2.87	9.71
CM9-NET	8.05	9.13	3.76	4.18	4.71	9.36	4.24	5.41	7.35

TABLE 7. RMSE (meters) averaged over space sizes of every CM in Fig. 17.

Methods	CM1	CM2	CM3	CM4	CM5	CM6	CM7	CM8	CM9
ToA+3DT [44]	5.48	6.46	1.97	3.01	3.67	9.05	2.74	3.51	8.86
ToA+COLA [9], [44]	6.26	7.18	2.13	3.09	4.02	9.26	2.92	3.73	9.41
CNN-DE+3DT [21]	3.20	3.95	1.35	1.70	2.19	5.67	1.31	1.88	5.22
CNN-DE+COLA [9], [21]	3.24	3.94	1.23	1.55	1.99	5.43	1.23	1.74	4.93
CNN-LE [22]	1.74	2.49	0.80	1.18	1.28	5.21	0.62	1.13	4.60
Proposed GRU-LE	1.36	2.13	0.45	0.84	0.69	5.07	0.28	0.82	4.46

i.e., the CNN-DE method. Again, it is worth being emphasized that the proposed GRU-LE outperforms other existing methods irrespective of the space sizes. This is clearly shown in Table 7. Here, the RMSE values in Fig. 17 are averaged over space sizes. Indeed, the proposed GRU-LE outperforms other methods irrespective of the type of CMs as well.

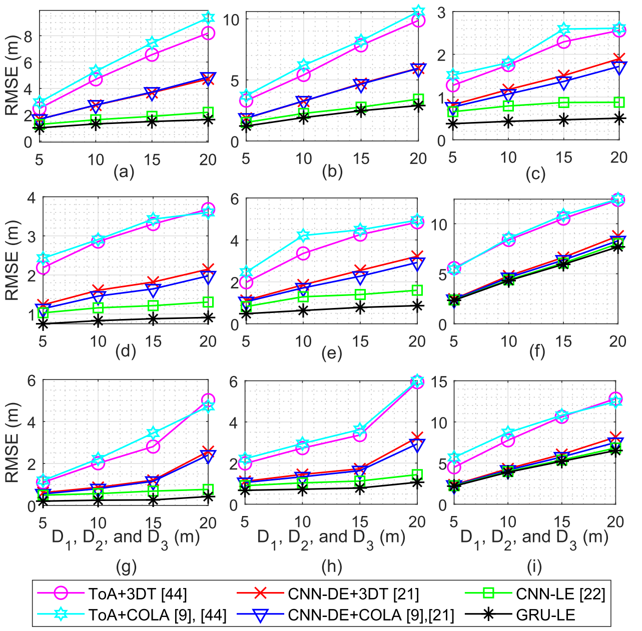


FIGURE 17. RMSE performances (y-axis) across different space sizes (x-axis: $D_1 = D_2 = D_3$). Training and testing data are generated from the same CM. (a) CM1. (b) CM2. (c) CM3. (d) CM4. (e) CM5. (f) CM6. (g) CM7. (h) CM8. (i) CM9.

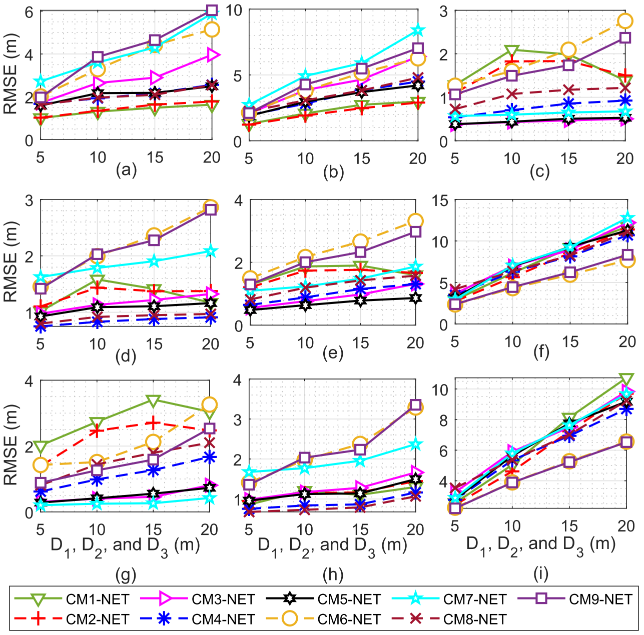


FIGURE 18. RMSE performances (y-axis) across different space sizes (x-axis: $D_1 = D_2 = D_3$). The CM#-NET trained by CM# is tested with (a) CM1, (b) CM2, (c) CM3, (d) CM4, (e) CM5, (f) CM6, (g) CM7, (h) CM8, (i) CM9.

Fig. 18 illustrates the RMSE of each GRU-LE CM#-NET when it operates each CM test data with respect to the area. Similar to the results above, the CM#-NETs trained in indoor and outdoor CMs work well in corresponding environments. Thus, these pairs of CM#-NETs can operate

TABLE 8. RMSE (meters) of CM#-NET averaged over space sizes in Fig. 18 when they are tested with CM# test data.

GRU-LE	CM1	CM2	CM3	CM4	CM5	CM6	CM7	CM8	CM9
CM1-NET	1.36	2.26	1.69	1.30	1.66	7.49	2.80	1.12	6.61
CM2-NET	1.44	2.13	1.57	1.32	1.59	7.08	2.26	1.18	5.99
CM3-NET	2.80	4.24	0.45	1.16	0.90	7.93	0.49	1.27	6.62
CM4-NET	2.08	3.28	0.76	0.84	1.00	7.15	1.14	0.90	5.93
CM5-NET	2.11	3.19	0.46	1.07	0.69	7.67	0.50	1.18	6.34
CM6-NET	3.72	4.34	1.93	2.17	2.41	5.07	2.08	2.26	4.49
CM7-NET	4.13	5.49	0.62	1.85	1.42	8.01	0.28	1.94	6.49
CM8-NET	2.06	3.48	1.05	0.91	1.25	7.48	1.54	0.82	6.29
CM9-NET	4.13	4.72	1.67	2.13	2.15	5.34	1.57	2.24	4.46

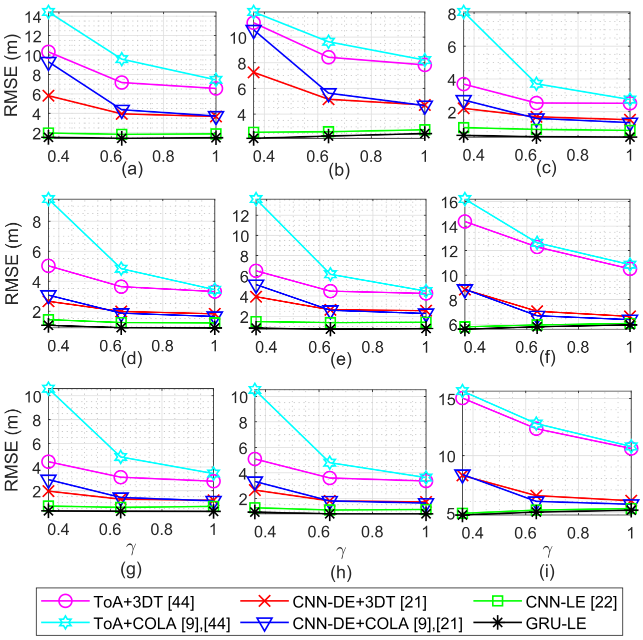


FIGURE 19. RMSE performances (y-axis) across different asymmetry factors (x-axis: γ) for space with the same volume. Training and testing data are generated from the same CM. (a) CM1. (b) CM2. (c) CM3. (d) CM4. (e) CM5. (f) CM6. (g) CM7. (h) CM8. (i) CM9.

in each other's environments. The pairs of interchangeable CM#-NETs include: CM1-NET and CM2-NET, CM4-NET and CM8-NET, and CM6-NET and CM9-NET. As also summarized in Table 8, for the test data CM#, CM#-NET operates best.

TABLE 9. RMSE (meters) averaged over asymmetry factor γ of every CM in Fig. 19.

Methods	CM1	CM2	CM3	CM4	CM5	CM6	CM7	CM8	CM9
ToA+3DT [44]	8.02	9.12	2.91	3.98	5.07	12.39	3.47	4.02	12.67
ToA+COLA [9], [44]	10.49	9.93	4.85	5.91	8.07	13.23	6.30	6.31	13.08
CNN-DE+3DT [21]	4.48	5.70	1.80	2.15	3.04	7.50	1.49	2.06	6.94
CNN-DE+COLA [9], [21]	5.80	6.95	1.88	2.19	3.33	7.30	1.84	2.25	6.70
CNN-LE [22]	1.90	2.66	0.94	1.29	1.41	5.93	0.67	1.16	5.18
GRU-LE	1.48	2.29	0.50	0.94	0.77	5.77	0.29	0.83	5.03

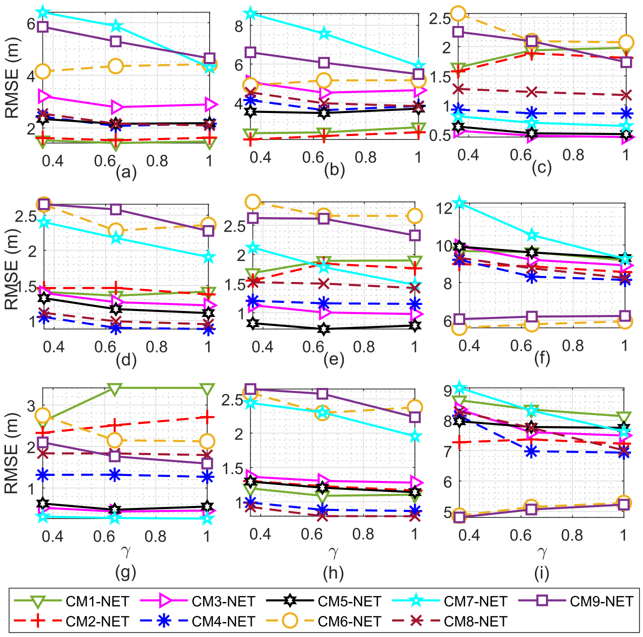


FIGURE 20. RMSE performances (y-axis) across different asymmetry factors (x-axis: γ) for space with the same volume. The CM#-NET trained by CM# is tested with (a) CM1, (b) CM2, (c) CM3, (d) CM4, (e) CM5, (f) CM6, (g) CM7, (h) CM8, (i) CM9.

TABLE 10. RMSE (meters) of CM#-NET averaged over asymmetry factor γ in Fig. 20 when they are tested with CM# test data.

GRU-LE	CM1	CM2	CM3	CM4	CM5	CM6	CM7	CM8	CM9
CM1-NET	1.48	2.54	1.85	1.39	1.82	9.49	3.14	1.13	8.38
CM2-NET	1.59	2.29	1.76	1.43	1.72	8.80	2.52	1.23	7.29
CM3-NET	2.96	4.74	0.50	1.29	1.04	9.35	0.47	1.32	7.82
CM4-NET	2.22	3.86	0.88	0.94	1.17	8.56	1.30	0.91	7.35
CM5-NET	2.23	3.57	0.55	1.20	0.77	9.61	0.55	1.21	7.83
CM6-NET	4.31	5.07	2.25	2.43	2.74	5.77	2.34	2.42	5.09
CM7-NET	5.53	7.36	0.72	2.16	1.79	10.67	0.29	2.23	8.34
CM8-NET	2.27	4.11	1.22	1.02	1.48	8.77	1.82	0.83	7.68
CM9-NET	5.26	6.05	2.03	2.50	2.52	6.16	1.82	2.48	5.03

E. PERFORMANCE WITH RESPECT TO ASYMMETRY OF THE SPACE

In Fig. 19, the performances of various localization methods are evaluated across different values of asymmetry factor $\gamma = D_3/D_1$. The SNR is fixed at 30 dB and the space size is unchanged at $15^3 = 3{,}375$ m^3, with $D_2 = 15$ m. The GRU-LE method outperforms all of the conventional methods, and it is unaffected by the asymmetry of the operating space. The ToA and CNN-DE methods are not robust against the space asymmetry. For positioning methods, the COLA suffers due to the asymmetry of the space, contrary to the 3DT method. Again, it is worth being emphasized that the proposed GRU-LE outperforms other existing methods irrespective of the asymmetry of the operating space. This is clearly shown in Table 9. Here, the RMSE values in Fig. 19

are averaged over different γ. Indeed, the proposed GRU-LE outperforms other methods irrespective of γ and the type of CMs as well.

Fig. 20 shows the RMSE of the CM#-NET when operating on the test data of various CMs. Similar to other experiments, the GRU-LE performs the best when the CMs for the training and testing data are the same. The CM6-NET and CM9-NET models perform better with CM6 and 9 test data than with other CMs. The CM1-NET and CM2-NET are interchangeable, as along with the pair of CM4-NET and CM8-NET. The RMSEs in Fig. 20 are averaged over γ and shown in Table 10. As expected, for the test data CM#, the CM#-NET operates best (see the underlined RMSE value which is the smallest in each column.).

VII. CONCLUSION

In this study, we developed a GRU-based model to estimate the Tx location in a 3D environment. This method integrates the ranging and positioning processes within a single model. The time series data generated from the received signals from all of the Rxs and the three coordinates of the Tx location are the input and output of this model, respectively. The localization performance was evaluated according to the RMSE values. As demonstrated by a numerical comparison, the GRU-LE model outperforms all other methods, and the proposed methods is also robust against asymmetrical 3D space, compared with the ToA and CNN-DE methods. Moreover, compared to the exiting CNN-LE method, our proposed GRU-LE method achieves 15 and four times shorter execution time in training and testing, respectively, and provides smaller RMSE up to 0.8 meters. In the future, we intend to apply the proposed localization method in real-time applications.

REFERENCES

[1] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, and E. Aboutanios, "Recent advances in indoor localization: A survey on theoretical approaches and applications," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1327–1346, 2nd Quart., 2017.

[2] X. Ma and T. Liu, "The application of Wi-Fi RTLS in automatic warehouse management system," in *Proc. IEEE Int. Conf. Autom. Logistics (ICAL)*, Chongqing, China, Aug. 2011, pp. 64–69.

[3] D. Liu and P. Ning, "Location-based pairwise key establishments for static sensor networks," in *Proc. 1st ACM Workshop Secur. Ad Hoc Sensor Netw. (SASN)*, New York, NY, USA, Oct. 2003, pp. 72–82.

[4] T. He, S. Krishnamurthy, J. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh, "Energy-efficient surveillance system using wireless sensor networks," in *Proc. ACM Int. Conf. Mobile Syst. Appl. Services*, Boston, MA, USA, Jun. 2004, pp. 270–283.

[5] N. Park, S. Jung, and D. Han, "An adaptive hybrid filter for practical WiFi-based positioning systems," *ICT Exp.*, vol. 1, no. 2, pp. 82–85, Sep. 2015.

[6] D. Han, S.-H. Jung, and S. Lee, "A sensor fusion method for Wi-Fi-based indoor positioning," *ICT Exp.*, vol. 2, no. 2, pp. 71–74, Jun. 2016.

[7] 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, 3rd Quart., 2019.

[8] N. Ben Halima and H. Boujemâa, "3D WLS hybrid and non hybrid localization using TOA, TDOA, azimuth and elevation," *Telecommun. Syst.*, vol. 70, no. 1, pp. 97–104, Jan. 2019.

[9] C. Y. Shih and P. J. Marrón, "COLA: Complexity-reduced trilateration approach for 3D localization in wireless sensor networks," in *Proc. 4th Int. Conf. Sensor Technol. Appl.*, Venice, Italy, Jul. 2010, pp. 24–32.

[10] J. Yang, H. Lee, and K. Moessner, "Multilateration localization based on singular value decomposition for 3D indoor positioning," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Alcala de Henares, Spain, Oct. 2016, pp. 1–8.

[11] L. Qiu, Z. Huang, N. Wirstrom, and T. Voigt, "3DinSAR: Object 3D localization for indoor RFID applications," in *Proc. IEEE Int. Conf. RFID (RFID)*, Orlando, FL, USA, May 2016, pp. 1–8.

[12] L. Zhang and H. Wang, "3D-WiFi: 3D localization with commodity WiFi," *IEEE Sensors J.*, vol. 19, no. 13, pp. 5141–5152, Jul. 2019.

[13] Y. Zhang, S. Liu, and Z. Jia, "Localization using joint distance and angle information for 3D wireless sensor networks," *IEEE Commun. Lett.*, vol. 16, no. 6, pp. 809–811, Jun. 2012.

[14] S. Tomic, M. Beko, and R. Dinis, "3-D target localization in wireless sensor networks using RSS and AoA measurements," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3197–3210, Apr. 2017.

[15] C. Xiao, D. Yang, Z. Chen, and G. Tan, "3-D BLE indoor localization based on denoising autoencoder," *IEEE Access*, vol. 5, pp. 12751–12760, 2017.

[16] L. Peng, J. Liu, M. Sheng, Y. Zhang, D. Hou, Y. Zheng, and J. Li, "3D indoor localization based on spectral clustering and weighted backpropagation neural networks," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Qingdao, China, Oct. 2017, pp. 1–6.

[17] L. Taponecco, A. A. D'Amico, and U. Mengali, "Joint TOA and AOA estimation for UWB localization applications," *IEEE Trans. Wireless Commun.*, vol. 10, no. 7, pp. 2207–2217, Jul. 2011.

[18] L. Zwirello, T. Schipper, M. Harter, and T. Zwick, "UWB localization system for indoor applications: Concept, realization and analysis," *J. Electr. Comput. Eng.*, vol. 2012, pp. 1–11, May 2012.

[19] S. P. Rana, M. Dey, H. U. Siddiqui, G. Tiberi, M. Ghavami, and S. Dudley, "UWB localization employing supervised learning method," in *Proc. IEEE 17th Int. Conf. Ubiquitous Wireless Broadband (ICUWB)*, Salamanca, Spain, Sep. 2017, pp. 1–5.

[20] Z. E. Khatab, A. Hajihoseini, and S. A. Ghorashi, "A fingerprint method for indoor localization using autoencoder based deep extreme learning machine," *IEEE Sensors Lett.*, vol. 2, no. 1, pp. 1–4, Mar. 2018.

[21] J. Joung, S. Jung, S. Chung, and E. R. Jeong, "CNN-based Tx–Rx distance estimation for UWB system localisation," *Electron. Lett.*, vol. 55, no. 17, pp. 938–940, Aug. 2019.

[22] D. T. A. Nguyen, H.-G. Lee, E.-R. Jeong, H. L. Lee, and J. Joung, "Deep learning-based localization for UWB systems," *Elect.*, vol. 9, no. 10, pp. 1712–1729, Oct. 2020.

[23] B. Silva, Z. Pang, J. Akerberg, J. Neander, and G. Hancke, "Experimental study of UWB-based high precision localization for industrial applications," in *Proc. IEEE Int. Conf. Ultra-WideBand (ICUWB)*, Paris, France, Sep. 2014, pp. 280–285.

[24] H. Xiong and J. Cheng, "Investigation of short-range high precision 3D localization via UWB radio," in *Proc. IEEE Global Commun. Conf.*, Austin, TX, USA, Dec. 2014, pp. 4090–4095.

[25] A. Karpathy, J. Johnson, and L. Fei-Fei, "Visualizing and understanding recurrent networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Juan, Puerto Rico, May 2016, pp. 1–11.

[26] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 604–624, Feb. 2021.

[27] L. Arras, G. Montavon, K.-R. Müller, and W. Samek, "Explaining recurrent neural network predictions in sentiment analysis," in *Proc. 8th Workshop Comput. Approaches Subjectivity, Sentiment Social Media Anal.*, Copenhagen, Denmark, Sep. 2017, pp. 159–168.

[28] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 6645–6649.

[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[30] K. Cho, B. Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Method Natural Lang. Process. (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1724–1734.

[31] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Proc. NIPS Deep Learn. Workshop*, Montréal, QC, Canada, Dec. 2014, pp. 1–9. [Online]. Available: https://arxiv.org/abs/1412.3555

[32] M. T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy, "Recurrent neural networks for accurate RSSI indoor localization," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10639–10651, Dec. 2019.

[33] Y. Zhang, C. Qu, and Y. Wang, "An indoor positioning method based on CSI by using features optimization mechanism with LSTM," *IEEE Sensors J.*, vol. 20, no. 9, pp. 4868–4878, May 2020.

[34] Z. Chen, H. Zou, J. Yang, H. Jiang, and L. Xie, "WiFi fingerprinting indoor localization using local feature-based deep LSTM," *IEEE Syst. J.*, vol. 14, no. 2, pp. 3001–3010, Jun. 2020.

[35] A. Poulose and D. S. Han, "UWB indoor localization using deep learning LSTM networks," *MDPI Appl. Sci.*, vol. 10, no. 18, pp. 6290–6312, 2020.

[36] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (GRU) neural networks," in *Proc. IEEE 60th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Boston, MA, USA, Aug. 2017, pp. 1597–1600.

[37] D. Dardari, A. Conti, U. Ferner, A. Giorgetti, and M. Z. Win, "Ranging with ultrawide bandwidth signals in multipath environments," *Proc. IEEE*, vol. 97, no. 2, pp. 404–426, Feb. 2009.

[38] Z. Sahinoglu and I. Guvenc, "Threshold-based TOA estimation for impulse radio UWB systems," in *Proc. IEEE Int. Conf. Ultra-Wideband*, Zurich, Switzerland, Sep. 2005, pp. 420–425.

[39] A. F. Molisch, "Ultrawideband propagation channels-theory, measurement, and modeling," *IEEE Trans. Veh. Technol.*, vol. 54, no. 5, pp. 1528–1545, Sep. 2005.

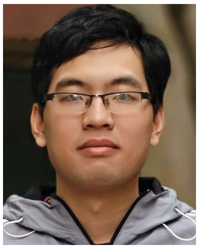
[40] A. F. Molisch, K. Balakrishnan, D. Cassioli, C. Chong, S. Emami, A. Fort, J. Karedal, J. Kunisch, H. Schantz, U. Schuster, and K. Siwiak, *IEEE 802.15.4a Channel Model—Final Report*, Standard IEEE 802.15 WPAN Low Rate Alternative PHY Task Group, IEEE 802.15-04-0662-02-004a, Nov. 2004. [Online]. Available: https://mentor.ieee.org/802.15/dcn/04/15-04-0662-04-004a-channel-model-final-report-r1.pdf

[41] G. Turin, "An introduction to matched filters," *IEEE Trans. Inf. Theory*, vol. 6, no. 3, pp. 311–329, Jun. 1960.

[42] *Create a Regression Output Layer—MATLAB Regression Layer*. Accessed: Dec. 3, 2020. [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/regressionlayer.html

[43] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," in *Proc. Int. Conf. Learn. Represent.*, Banff, AB, Canada, Apr. 2014, pp. 1–13. [Online]. Available: https://arxiv.org/abs/1312.6026

[44] W. Liu, H. Ding, X. Huang, and Z. Liu, "TOA estimation in IR UWB ranging with energy detection receiver using received signal characteristics," *IEEE Commun. Lett.*, vol. 16, no. 5, pp. 738–741, May 2012.



DOAN TAN ANH NGUYEN (Student Member, IEEE) was born in Hanoi, Vietnam, in 1996. He received the B.S. degree in electronics and telecommunications engineering from the Hanoi University of Science and Technology (HUST), Hanoi, in 2019. He is currently pursuing the master's degree with the School of Electrical and Electronics Engineering, Chung-Ang University, Seoul, South Korea. He is also a member of the Intelligent Wireless Systems Laboratory. His current research interests include localization systems and machine learning algorithms.

JINGON JOUNG (Senior Member, IEEE) received the B.S. degree in radio communication engineering from Yonsei University, Seoul, South Korea, in 2001, and the M.S. and Ph.D. degrees in electrical engineering and computer science from KAIST, Daejeon, South Korea, in 2003 and 2007, respectively.

He was a Postdoctoral Fellow with KAIST and UCLA, CA, USA, in 2007 and 2008, respectively. From 2009 to 2015, he was a Scientist with the Institute for Infocomm Research (I^{2}R), Agency for Science, Technology and Research (A*STAR), Singapore. In 2016, he joined Chung-Ang University (CAU), Seoul, as a Faculty Member. He is currently an Associate Professor with the School of Electrical and Electronics Engineering, CAU, where he is the Principal Investigator of the Intelligent Wireless Systems Laboratory. He is also an Inventor of a Space–Time Line Code (STLC) that is a fully symmetric scheme to a Space-Time Block Code, a.k.a. Alamouti code. His research interests include wireless communication signal processing, numerical analysis, algorithms, and machine learning.

Dr. Joung was a recipient of the First Prize of the Intel-ITRC Student Paper Contest, in 2006. He was recognized as an Exemplary Reviewer of the IEEE COMMUNICATIONS LETTERS, in 2012, and the IEEE WIRELESS COMMUNICATIONS LETTERS, in 2012, 2013, 2014, and 2019. From 2014 to 2019, he served as an Editorial Board Member for the *APSIPA Transactions on Signal and Information Processing*. He served as a Guest Editor for IEEE ACCESS, in 2016, and the *Electronics*, in 2019. He served as an Associate Editor for the *Sensors*, in 2020. Since 2018, he has been serving as an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY.

XIN KANG (Senior Member, IEEE) received the B.Eng. degree in electrical engineering from Xi'an Jiaotong University, China, in 2005, and the Ph.D. degree in electrical and computer engineering from the National University of Singapore, Singapore, in 2011. From 2011 to 2014, he was a Research Scientist with the Institute for Infocomm Research (I^{2}R), (A*STAR), Singapore. After that, he joined the Shield Laboratory, Huawei Singapore Research Center, as a Senior Researcher. He is currently a Full Professor working with the University of Electronic Science and Technology of China. He has filed more than 50 patents on security protocols, and contributed more than 30 technical proposals to 3GPP SA3, ITU-T SG17, and IETF. His research interests include optimization, game theory, energy harvesting, cognitive radio, AI, and network security protocol designs. He received the Best Paper Award from IEEE ICC 2017, and the best 50 papers of IEEE GlobeCom 2014.