Transfer Learning for UWB error correction and (N)LOS classification in multiple environments

Jaron Fontaine*, Fuhu Che†, Adnan Shahid*, Ben Van Herbruggen*, Qasim Zeeshan Ahmed†, Waqas Bin Abbas†, Eli De Poorter*.
* IDLab, Department of Information Technology at Ghent University - imec, Belgium
† School of Computing and Engineering, University of Huddersfield, Huddersfield, UK
Email: jaron.fontaine@ugent.be

***Abstract*—Ultra Wideband (UWB) is a popular technology to address the need for high precision indoor positioning systems in challenging industry 4.0 use cases. In line-of-sight (LOS) environments, UWB positioning errors in the order of $1-10$ cm can be achieved. However, in non-line-of-sight (NLOS) conditions, this precision drops significantly, with errors typically >30 cm. Machine learning has been proposed to improve the precision in such NLOS conditions, but is typically environment-specific and lacks generalization to new environments and UWB configurations. As such, it is necessary to collect large datasets to train a neural network for each new environment or UWB configuration. To remedy this, this paper proposes automatic optimizations for transfer learning (TL) deep neural networks towards new environments and UWB configurations. We analyze error correction and (non)-line-of-sight ((N)LOS) classification models, using either feature- or channel impulse response-based (CIR) input data. Our TL solutions show a 50% error improvement and 15% (N)LOS classification accuracy improvement (for both feature- and CIR-based approaches) compared to a model trained in a different environment. We also analyze the impact on TL using a limited number of samples (25 to 400 samples). The highest accuracy is typically achieved by the CIR-based approach, where with only 50 samples from the new mixed (N)LOS environment, we show ± 10 cm precision after error correction with 93% (N)LOS detection. The presented results demonstrate high precision UWB localization (from 643 mm to 245 mm) through ML with minimal data collection effort in challenging NLOS environments.**

***Index Terms*—Transfer learning, UWB, localization systems, error correction, (N)LOS classification**

I. Introduction

High precision positioning has been extensively researched in recent years as it is a core technology to support multiple Internet of things (IoT) use cases such as forklifts control [1], [2], assistive healthcare systems [3], automated guided vehicles [4], positioning systems in industrial environments [5]–[8] and to maintain the social distancing as observed in the COVID-19 scenario [9]–[11]. According to this trend, the Ultra Wideband (UWB)-based indoor positioning system (IPS) is one of the most promising techniques in comparison with several other technologies (e.g. RFID, Wi-Fi- and Bluetooth-based positioning using received signal strength, etc.). UWB IPS can achieve centimeter-level positioning accuracy due to its capability of transmitting an extremely narrow pulse with extremely wide bandwidth (≥ 500 MHz) and very fine time resolution (around 1 ns) [12], [13]. In addition, due to the large bandwidth of the transmitted signal, very high frequency diversity is achieved which makes the UWB signal resistant to phenomena such as severe multipath fading, propagation penetrating through different types of materials and interference [14], [15]. However, in the indoor environments, the high accuracy performance of UWB is still not always guaranteed due to the presence of a) non-line-of-sight (NLOS) and b) multipath conditions, which are usually considered as important factors affecting the positioning accuracy of any IPS [16]–[18]. UWB suffers in NLOS conditions as its direct path is blocked which introduces a positive bias in the range estimates between the transceivers [19]. In indoor environments, NLOS is very common with propagating signals easily blocked by walls (drywalls or concrete), pedestrians, industrial equipment, robots, etc [20]. In addition, multipath conditions in UWB also play a key role in deteriorating the positioning performance. In such conditions, signals are reflected by the wall and might be stronger than the (attenuated) direct path or arrive shortly after each other. Therefore, it is important to minimize the impact of NLOS and multipath conditions for improving the accuracy of UWB IPS.

There has been an ongoing research addressing error correction and NLOS detection to improve the performance of UWB IPS [17], [18], [21], [22]. In brief, most works focus on machine learning (ML) techniques, which can use either raw physical data or use a feature-based method [15], [23]. For raw channel impulse response (CIR)-based ML, typical techniques are based on using the CIR signal, logged by the UWB receiver, and train a ML model to identify NLOS and ranging errors. These raw-CIR-based techniques can become computationally complex. Instead, less complex feature-based methods are proposed and rely on calculating the relevant features (e.g., the amplitude of the signal, power difference, power ratio, etc.) which can be extracted from the CIR or directly registered by the UWB hardware. NLOS can then be classified by ML approaches such as support vector machine (SVM), decision tree, neural network (NN), convolutional neural network (CNN) and so on [24]–[27].

It is worth noticing that both raw CIR- and feature-based approaches show high error correction performance and (N)LOS classification accuracy in known environments. However, the performance of the proposed approaches is often not evaluated in an unseen environment that the NN is not trained for [24]–[27]. Some works have addressed the performance from anchors not present in the training dataset

and NLOS generalization towards one new environment [7], where good generalization is achieved. Very recently, in [28] the authors have noted generalization issues with CNNs in unseen NLOS detection datasets, while using transfer learning with 20%-70% of this dataset improves the performance. Such unseen environments correspond to different UWB device topologies, size of the room, and presence of objects, etc. as compared to seen environments of the trained models. These differences deteriorate the localization performance of the UWB IPS because of two main reasons. Firstly, the UWB collected signals in various environments could be different due to diverse obstacles and differently sized rooms. Secondly, the UWB devices used, configurations and deployment of the UWB devices, together with the data collection process will depend upon the trained UWB setup. This will result in unsatisfactory performance of the previously-trained ML models in unseen environments. However, collecting data and training different models for each unique environment and UWB configuration requires considerable work and time (i.e. setting up the devices, perform large data collections and labeling, and execute the training model process). Even then, the environment could already have changed as time progresses, requiring to update the models frequently. Therefore, conventional techniques requiring new big datasets and completely new models are limited in their versatility to unseen and changing environments with or without different UWB configurations.

To address this shortcoming, we propose a transfer learning (TL) framework for UWB error correction and (non)-line-of-sight ((N)LOS) detection using feature- and CIR-based NNs. In the context of deep learning, TL typically requires careful consideration of updating specific NN layers and finetuning the model with data from a new environment. To reduce the effort needed to perform TL, in the proposed TL framework we consider these variables as hyperparameters and use Bayesian optimization for automatic tuning of these hyperparameters.

As such, the main contributions of this paper are as follows.

- We propose a TL framework for UWB error correction and (N)LOS detection using feature- and raw CIR-based approaches.
- We construct various TL-based hyperparameters including frozen layers, unfrozen finetuning and training with small datasets in different environments and UWB configurations.
- We propose an automatic hyperparameter tuning strategy for TL using Bayesian optimization.
- We analyze the performance of both feature- and raw CIR-based approaches towards various unseen environments and configurations for UWB error correction and (N)LOS detection. In addition, we investigate different number of training samples required from new environments. The complexity of the proposed TL raw CIR- and feature-based approaches is also presented.

The remainder of this paper is organized as follows. The related work is described in Section II. Next, we present the system description and problem statement of error correction and (N)LOS detection in Section III, followed by a detailed discussion on the dataset description and collection in Section IV. In Section V, we present the proposed TL feature- and raw CIR-based approaches for error correction and (N)LOS detection. Section VI presents the results. Finally, the conclusions and future work are presented in Section VII.

II. Related work

In this section, we provide an overview of several related papers in the literature as shown in Table I. Firstly, we categorize existing UWB research into (i) error correction and (ii) (N)LOS detection approaches. For each of the related papers, we discuss if their approaches are feature- or raw CIR-based and discuss TL in terms of automatic hyperparameter tuning, number of samples analysis, new environments, and UWB configurations in different domains. (iii) Next, we discuss other UWB related domains, i.e. fingerprinting and elaborate that TL is currently underexplored even in these domains.

A. UWB Error Correction

UWB error correction based on raw CIR-based are mentioned in [29]–[31]. Similar to the (N)LOS approaches, some research papers have focused on extracting features from the CIR data. The authors of [29] extracted the features based on distance measurement and received signal strength. Then the authors proposed local spatial feature extraction, temporal feature extraction and position prediction to improve the positioning accuracy. [30] main focus is on UWB measured ranging associated with NLOS and multipath. A large dataset comprising of the measured distance and 7 different signal features was trained by a NN to perform error prediction. The focus of [14] is on UWB feature-based error correction. Two classes of non-parametric regression techniques include a SVM and the Gaussian process and were applied by the authors to directly mitigate the ranging error in the physical layer, based on 6 signal features from the received waveform and the estimated distance. The fraction of residual errors less than $1m$ have increased from 63% to around 90% by using SVM- and Gaussian process-based mitigation. Finally, in paper [31], a semi-supervised autoencoder-based ML approach is proposed by the authors, based on raw CIR data, to achieve high IPS accuracy for low-cost edge devices. The results achieved 29% higher localization accuracy than state-of-the-art deep neural networks (DNNs) in complex environments.

Several papers have also explored both error correction and NLOS detection [7], [32]–[35]. The authors in [32] put forward a bagging-based ensembled tree classifier for NLOS classification and treebagger regression for NLOS error mitigation. The result showed an accuracy of 95.65% for NLOS classification and a root mean square error of $0.4790\ m$ for error correction. In [33], the authors presented a fuzzy logic control decision method to identify the NLOS in multiple classes (LOS, soft NLOS and hard NLOS). Similarly, in [34], the authors collected 9 signal features of the data in terms of LOS, weak LOS and NLOS. A SVM classifier is used for detecting NLOS conditions as well as error correction in different combination of features. In paper [35], the authors

TABLE I: Comparison of the proposed TL approach and analysis performed in this paper with related work. If TL was used, the table mentions: (i) if automatic hyperparameter tuning was applied (auto.), (ii) the number of considered TL samples that are used as input for TL, (iii) if multiple configurations (conf.) and (iv) environments (env.) were used.

				Transfer learning			
Paper	**Domain and problem**	**Feature-based**	**Raw-based**	**Auto.**	**# Samples**	**>1 Conf.**	**>1 Env.**
[14], [29], [30]	Error Correction	✓					
[31]	Error Correction		✓				
[32]–[34]	Error correction and (N)LOS detection	✓					
[7], [35]	Error correction and (N)LOS detection		✓				
[36]–[38]	(N)LOS detection	✓					
[39]–[42]	(N)LOS detection		✓				
[43]	(N)LOS detection		✓		100		✓
[28]	(N)LOS detection		✓		10000		✓
[44]	Fingerprinting (UWB)		✓		4-225		✓
[45]	Fingerprinting (WiFi)		✓		28-250		
[46]	Fingerprinting (WiFi)		✓		200-600		
Our work	**Error correction and (N)LOS detection**	✓	✓	✓	**25-100**	✓	✓

investigated subsets of anchor pairs selection with 1 tag and 8 anchors for a time difference of arrival (TDOA) position algorithm. Using ML-based error correction and detection, the positioning error can be reduced by 75% on top of selecting the best combination of the anchors. In [7], the authors trained three CNN based architectures (Residual network, Encoder and Fully convolutional network) to detect NLOS conditions directly from the raw CIR data. However, all the aforementioned papers also did not evaluate their approaches in unseen environments. Therefore, the applicability of the approaches in unseen environments cannot be justified.

B. UWB NLOS Detection

Finally, the following papers did only consider NLOS classification. Although these papers retain a high classification accuracy, they do not investigate the improvement of UWB error correction.

UWB feature based methods are mentioned in [36]–[38]. In [36], the authors address three ML approaches to classify NLOS in multiple classes (LOS, NLOS and multipath). 12 features were extracted and the overall accuracy reaches up to 91.9% in the best case. Similarly, the authors in [37] proposed Gaussian distribution and generalized Gaussian distribution algorithms to identify NLOS components in an imbalanced dataset with 10 features. The NLOS classification accuracy reached 96% and 98% respectively. Furthermore, [38] applied a genetic algorithm to find the best combination of 18 features in an office environment.

In contrast, the authors in [39]–[42] focused on UWB raw CIR-based NLOS detection. Specifically, in [39], the authors apply a CNN to identify the NLOS signal after denoising the raw CIR data using a reversible transformation method. The evaluation results show an averaged accuracy increase of 27.9% for NLOS classification accuracy. Similarly, the authors of [40] also apply a CNN to classify the NLOS after using a Gate Recurrent Unit to extract spatial features of raw CIR data. Moreover, an unsupervised ML approach based on Gaussian mixture models to identify the NLOS links from the unlabelled data is proposed in [41]. The authors of [42] proposed a CNN to extract non-temporal features from UWB raw CIR data, and then, the features were fed into the Long-Short Term Memory network for NLOS classification which showed an accuracy of 82.14%. Compared to the feature based methods, the above mentioned papers based on raw CIR measurements provide a superior performance for NLOS detection.

However, the aforementioned papers did not evaluate the performance of the approaches in an unseen environment which limits their suitability in practical settings. To the best of our knowledge, until now only two research papers address paper TL for UWB (N)LOS detection. In [43], the authors proposed TL based on NN and CNNs to identify UWB NLOS signals in unseen environments. 100 training samples were selected for TL. The accuracy was improved approximately 10% higher accuracy and achieved approximately five times faster training time by using the TL approach. However, the paper does not consider a) feature based NN, b) the impact of different UWB configurations and c) evaluating the impact of different training samples. Finally, in [28] the authors propose a Stockwell transform-based CNN (ST-CNN) method for predicted (N)LOS conditions. This method is different from most CIR-based models and uses both frequency and time domain information as an input to the ST-CNN. The authors note a 7-10% performance increase over more traditional CNN-based approaches. However, when presenting the ST-CNN data captured in a different environment, the accuracy drops severely, to below 50% accuracy. To overcome this generalization problem, the authors use a transfer learning method with 20-40% of the dataset captured in the new environment. As a result, accuracies above 97% are obtained with a smaller training time. However, the paper does not consider a) feature based NN, b) the impact of different UWB configurations and c) evaluating the impact of a very small TL dataset, as the results are still using up to 10000 training samples.

C. TL based on Fingerprinting

Although, TL for UWB error correction and (N)LOS detection has rarely been researched, multiple papers have adopted TL to increase the accuracy of IPS using fingerprinting approaches.

In [44]–[46], the authors proposed TL based on the fingerprinting samples. The authors in [44] proposed a NN

to learn precise fingerprinting of UWB access points in an environment, then apply TL by updating the NN in a new environment. Only 7% of fingerprinting samples is needed for a new environment. Similar work was conducted in [45] where the authors also used a NN based TL approach. But they combine 15 different WiFi datasets to train the model. The positioning error was reduced to 25% in most cases. Finally, the authors in [46] proposed TL to efficiently transfer the knowledge between source domains and the single target domain for WiFi fingerprinting-based target positioning. They applied the proposed approach with multiple sources jointly when the datasets lacks of partial access point data or geomagnetic data. In conclusion, throughout the literature, none of the aforementioned work on UWB discusses the stability of TL models to an unseen environment when the devices and configurations of devices are different. To the best of our knowledge no previous work has been done on the comparison of using both feature- and raw-based using TL. Moreover, hyperparameter tuning is often manually executed which is another drawback in the existing approaches.

To overcome these shortcoming, we fill these gaps and propose a TL framework for UWB error correction and (N)LOS detection using both raw CIR-based and feature-based data. Moreover, we identify TL variables as hyperparameters and propose an automatic hyperparameter tuning strategy for TL using Bayesian optimization. We also compare the evaluation results in various environments and UWB configurations for both feature- and raw CIR-based approaches. Furthermore, we investigate different number of training samples required from new environments and the complexity of the proposed raw CIR-based and feature-based approaches.

III. System description and problem statement

To give a better understanding of the problem we try to solve in this paper, we first present the UWB localization system and its challenges in a detailed manner. As such, in this section we mathematically formulate (i) UWB localization systems (as shown in Figure 1 (a)) and (ii) different environments and UWB configurations and their impact on the UWB localization performance (as shown in Figure 1 (b)). For improved readability, Table II gives an overview of all mathematical symbols used throughout this paper.

A. UWB Localization Systems

UWB IPS need a coordinate system with a reference point $ref = (0, 0, 0)$ to offer relative 3-dimensional $(3D)$ positions for both anchors a_q and tags t_r,

$$a_q \in \{a_1, a_2, \cdots, a_Q\}, \tag{1}$$

$$t_r \in \{t_1, t_2, \cdots, t_R\}, \tag{2}$$

where Q is the total number of anchors (fixed node with known coordinates) and R is the total number of tags (mobile node with unknown coordinates) in the UWB IPS.

The position of an anchor and a tag within the UWB localization system can be represented in (3) and (4).

TABLE II: Mathematical symbols used throughout this paper

Symbols	Description
ref	Localization system reference point
a_q	UWB anchor
Q	Number of UWB anchors
t_r	UWB tag
R	Number of UWB tags
$\Delta a_q t_r$	Euclidean distance between a_q and t_r
$ToF_{a_q t_r}$	Time of flight between a_q and t_r
$CIR_{a_q t_r}(t)$	Channel impulse response provided by the UWB chip
$\widehat{ToF}_{a_q t_r}$	The estimated ToF by the UWB chip
$e_{a_q t_r}$	The distance error between a_q and t_r
$\widehat{\Delta} a_q t_r$	The estimated range between a_q and t_r
X_{base}	The base dataset, with samples from existing samples and environments (in this paper from the IIoT testbed)
P	Number of samples in X_{base}
N	Number of different environments
L	Number of different configurations
ML_{base}	The ML model trained on X_{base}
M	Number of samples in a new environment and UWB configuration
K	Number of TL samples from the new environment and UWB configuration
$\acute{X}_{new,k}$	New small dataset intended for TL with K samples
$\acute{M}L_{new}$	New model trained from $\acute{X}_{new,k}$
X_O	OfficeLab dataset, collected in a real office testbed
X_U	University dataset, collected in a university setting [34]
lr	Learning rate of the ML model optimizer
H	The collection of chosen trainable layers during TL
H'	Represents the H hyperparameter chosen by the Bayesian optimizer
$\delta_{finetuning}$	This Boolean parameter decides to perform additional finetuning epochs including the previously untrainable layers during the TL process
$\delta'_{finetuning}$	Represents the $\delta_{finetuning}$ hyperparameter chosen by the Bayesian optimizer

$$a_{q_{pos}} = (a_{q_x}, a_{q_y}, a_{q_z}), \tag{3}$$

$$t_{r_{pos}} = (t_{r_x}, t_{r_y}, t_{r_z}), \tag{4}$$

where x, y, and z are the Cartesian coordinates relative to the reference point ref, respectively.

To calculate $t_{r_{pos}}$, the r-th tag t_r will measure it's range with each anchor a_q in the localization system. The ground truth range $\Delta a_q t_r$ between a_q and t_r can be represented by the Euclidean distance formula as follows.

$$\Delta a_q t_r = \sqrt{(a_{q_x} - t_{r_x})^2 + (a_{q_y} - t_{r_y})^2 + (a_{q_z} - t_{r_z})^2}. \tag{5}$$

As the goal is to find the position of t_r, the x, y, and z coordinates are not available to the UWB systems, while the x, y, and z coordinates of each anchor a_q are known. In UWB systems, $\Delta a_q t_r$ is determined by measuring the time of flight between anchor a_q and tag t_r. Specifically, in this paper, we focus on two way ranging (TWR), more specific DS-TWR (double sided - two way ranging) which calculates the time of flight with three packets (there exists also variants from TWR with only 2 packets). Two packets are send from tag t_r to a_q and one vice versa, which eliminates any processing time differences due to the influence of crystal inaccuracies, timings offsets or processing delays. More details on TWR can be found in [47]. The result is a time of flight which can be used to calculate $\Delta a_q t_r$ in LOS conditions as follows

(a) ML for UWB improvements

(b) ML solutions in multiple environments

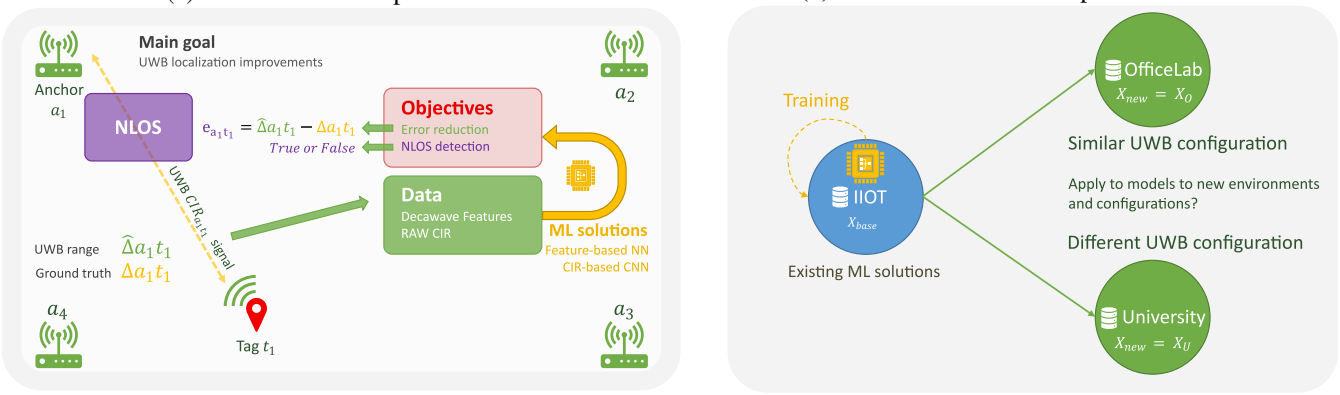


Fig. 1: Illustration of the mathematical system description. (a) ML when applied to localization systems for error correction and/or (N)LOS detection. (b) Using the ML models for different UWB radio configurations and/or environments.

$$ToF_{a_q t_r} = \frac{\Delta a_q t_r}{c}, \tag{6}$$

where c is the speed of light of value 3 x 10^8 m/s.

In NLOS conditions, the time of flight might be longer due to the presence of multipath components in the environment [48], [49]. Before calculating this error, the received UWB signal needs to be determined. The CIR, logged by the UWB receiver, represents information about the first- and multipath signal propagation between a_q and t_r and can be represented as follows [50].

$$CIR_{a_q t_r}(t) = \sum_{s=1}^{S} \alpha_s \delta(t - \tau_s) + n(t), \tag{7}$$

where t indicates the timestamp for each value within the CIR (there are 1016 values, each corresponding to 10^{-9} s); δ is the Dirac delta function; S is the number of multipath components; α_s and τ_s are the amplitude and the time delay of the s-th multipath components, respectively; and n represents the additive white Gaussian noise present in the channel. In LOS, the first path (FP) component in the CIR corresponds to $ToF_{a_q t_r}$ and is typically estimated by a leading edge algorithm, as used by the popular Qorvo DW1000 UWB chip [51]. This leading edge algorithm detects the position in time where the arriving signal, from the accumulated UWB pulses, first rises above the noise floor (the position of the FP). However, when the true first path, denoted by fp, is severely attenuated in NLOS conditions, the detected first path (fp') by the UWB receiver is degraded. As a result in NLOS conditions, the calculated $\widehat{ToF}_{a_q t_r}$ can be inaccurate,

$$\widehat{ToF}_{a_q t_r} = ToF_{a_q t_r} + \tau_{fp'-fp}, \tag{8}$$

where fp' is the first detected path above the noise floor $n(fp-1)$ where earlier multipaths are not detected and $\tau_{fp'-fp}$ is the time difference between the detected first path (fp') and the true first path (fp). The amplitude of $CIR_{a_q t_r}$ at $\tau_{fp'}$ is thus:

$$CIR_{a_q t_r}(fp') > n(fp-1). \tag{9}$$

The distance error (in meters) between a_q and t_r thus becomes

$$e_{a_q t_r} = \tau_{fp'-fp} \times c. \tag{10}$$

where c is the speed of light and the calculated range between a_q and t_r is

$$\widehat{\Delta} a_q t_r = \Delta a_q t_r + e_{a_q t_r}. \tag{11}$$

The goal of UWB error correction ML model is to predict the distance $e_{a_q t_r}$ accurately. To collect a dataset for the training process, ground truth of $\Delta a_q t_r$ is required, together with the UWB estimated ranges $\widehat{\Delta} a_q t_r$. Alternatively, ML models can also be used to predict whether the received $CIR_{a_q t_r}$ was LOS or NLOS and avoid $\widehat{\Delta} a_q t_r$ in the UWB IPS. The ground truth of (N)LOS is typically based on a topology of the environment or based on visual indicators during the data measurement campaign.

B. UWB ML in Different Environments and Configurations

ML models make their predictions based on the first- and multipath components in the received CIR signals. Such signals are collected in dataset X_{base}, where a ML model ML_{base} is trained using P samples from X_{base}:

$$ML_{base} \xleftarrow{\text{training}} X_{base} = \{1, 2, \cdots, P\}. \tag{12}$$

In different environments these multipath components s change severely, and depend on various factors including walls, objects, Δat, presence of humans, etc. The model ML_{base} trained in thus specifically trained for N environments collected in X_{base}. Moreover, we distinguish various UWB hardware-level configurations: (i) different UWB platforms with different transceivers and antennas (which can be (omni)directional and have different form factors, gain levels and radiation patterns), which can impact the received signal strength of the first and multipath components, (ii) different UWB configurable settings that can include TX power, UWB

channels, the UWB pulse repetition frequency, the UWB preamble length, etc. As these configurations impact the UWB signals, they will influence the UWB packet propagation, and therefore the CIR of the deployed UWB IPS system. These factors can consist of different noise levels $n(t)$, first- and multipath amplitudes α_s and timings τ_s. A model ML_{base} trained in thus specifically trained for L configurations collected in X_{base}.

Thus, building a rich base dataset that targets N environments and L configurations requires the dataset collection X_{base}.

$$X_{base} = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1N} \\ X_{21} & X_{22} & \ddots & X_{2N} \\ \vdots & \ddots & \ddots & \vdots \\ X_{L1} & X_{L2} & \cdots & X_{LN} \end{bmatrix}. \tag{13}$$

These environments and UWB configurations reliant influences make it difficult for a model ML_{base} trained on X_{base} to generalize to a new and unseen environment and UWB configuration. As a result, a model ML_{base} can show poor performance on an entirely new dataset X_{new}. To capture M samples in any new environment and UWB configuration will result in creating a new large dataset X_{new}:

$$X_{new,m} = \{1,2,\cdots,M\}. \tag{14}$$

Training ML_{new} from scratch and collecting X_{new} is time consuming and often not feasible. It is therefore desirable to train a model $\acute{M}L_{new}$ using a $\acute{X}_{new}$ dataset with K samples where

$$\acute{X}_{new,k} = \{1,2,\cdots,K\} \in X_{new}, \tag{15}$$

so that,

$$K \ll M, \tag{16}$$

resulting in a selection of K samples, which is much smaller than collecting a new large dataset with M samples.

IV. Data description

In this section, the datasets are described, which are collected from three different environments (industrial, office, and university). The datasets from the industrial and office environments are in-house measurements, while the dataset from university measurements is from [34]. As part of our contribution, we make our used dataset publicly available[1] to researchers, which allows the comparison of future algorithms and solutions with the proposed solution in this paper.

[1]Dataset available at: https://github.com/JaronFontaine/UWB-dataset-from-an-office-industrial-and-university-environment

A. Dataset environments

*1) **Large Industrial Environment** X_{base}:* The first environment is part of the IIoT testbed [52], which mimics industrial (warehouse) locations. In total, 18 measurement points were used in this environment, where an UWB tag performed TWR with all reachable 21 anchors. The IIoT testbed, illustrated in Figure 2a, contains a variation of open space and metal racks, which act as LOS and NLOS signal conditions. Out of the 18 measurement positions, four were located relatively close to the wall (10 cm) which can influence the first path of the CIR, but represents a realistic setup. The total size of this environment is approximately 30 $m \times 10$ m. A fixed set with small variation of UWB settings was used in X_{base}, i.e. UWB channel 1, 2 and 3, a data rate of 110 kb/s, a pulse repetition frequency (PRF) of 64 MHz and a preamble length of 1024 and 1536 symbols. The presence of multiple setting combinations act as a variety factor in the dataset, while mutually not diverging too much from a signal perspective. Although for the TWR method the CIR data is available on both anchor and tag, for this dataset we recorded the CIR locally at the tag. The outcome of this data capture is a total of 21,085 ranging measurements, of which 8,947 are LOS and 12,138 NLOS. Ground truth measurements were achieved using a laser Leica Disto D2 measuring tool, a precise floor plan to measure the distance to the walls and known central zero-point to determine the coordinates in a Cartesian system. In addition, we visually noted (N)LOS labels at each position and between each anchor-tag range on a best effort basis. The mean absolute error (MAE) without error correction in this dataset is 203 mm, while the MAE for LOS and NLOS measurements are 109 mm and 273 mm, respectively.

*2) **Office Environment** X_O:* The second environment characterizes an office environment, which is significantly different than the environments in X_{base}. Specifically, the data was captured in the OfficeLab testbed [53]. This testbed contains 15 anchors across different rooms and corridors, which contain less metal but more concrete and drywall than in the industrial environment. Because of this, the dataset contains signals with a lot of multipath components and NLOS. All anchors were configured to range with each other (for the combinations that can reach each other) and ground truth was obtained similar to the X_{base} dataset, using a laser measurement tool and precise floor plan. The UWB devices were configured with the same settings as the X_{base} dataset. In total, 44,894 UWB signals were captured in the office environment, of which 9946 were LOS and 35448 were NLOS. The (N)LOS label for each anchor combination was determined using the floor-plan of the office environment, as illustrated in figure 2b. The resulting MAE of the dataset without error correction is 306 mm, while the MAE for LOS and NLOS measurements are 155 mm and 346 mm, respectively.

*3) **University Environment** X_U:* The university datasets originates from a publicly available dataset [34], which was collected in multiple (sub)environments in a university building and consists of ranges between different types of environments such as laboratories, hallways and offices. In total, 733 unique ranging positions consist of 15,208 UWB

(a) The industrial environment (X_{base})

(b) The office environment X_O

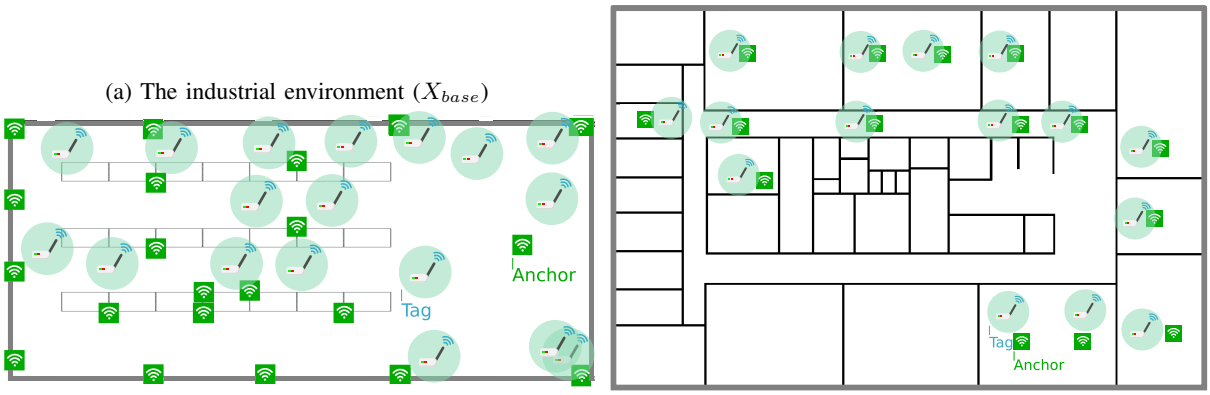


Fig. 2: (a) The industrial environment consists of a large industrial environment with metal racks as obstacles where anchors are deployed both in open and NLOS locations. (b) The office environment consists of multiple rooms divided by different wall types (brick, plywood and reinforced concrete).

TABLE III: Overview of datasets from different environments showing heterogeneous characteristics.

Dataset	**Positions**	**Samples**			**MAE (mm)**			**UWB Transceiver**	**Samples / CIR**
		Total	**LOS**	**NLOS**	**Total**	**LOS**	**NLOS**		
Industrial (X_{base})	18	21085	8947	12138	203	109	273	Wi-PoS [54] (Qorvo DW1000)	300
Office (X_O)	15	44894	9946	35448	306	155	346	Wi-PoS [54] (Qorvo DW1000)	300
University (X_U) [34]	733	15208	8735	6473	419	71	961	DWM1001-DEV (Qorvo DW1000)	1015

ranges, of which 8,735 are LOS and 6,473 are NLOS. In different sub-environments, both LOS and NLOS signals were present (generated by large walls), as well as weak LOS measurements, due to obstructions from smaller office devices and equipment. Not only this dataset acts as an extra evaluation measure for TL towards different environments, it also serves as an assessment of TL towards different UWB configurations. Indeed, both the hardware and UWB settings were different from our in-house captured dataset. The DWM1001-DEV UWB platform configuration uses channel 5, which has a higher center frequency of 6489.6 MHz compared to the previous datasets at channel 1, 2 and 3 with a center frequency of 3494.4 MHz, 3993.6 MHz and 4492.8 MHz, respectively. All these channels use the same bandwidth of 500 MHz. A short preamble length of 128 symbols and a high data rate of 6.8 Mbps was used. Together with the different nature of the environments and UWB configurations, the resulting ranging errors have a wider range. The total MAE of the dataset is 419 mm, while the MAE for LOS and NLOS measurements are 71 mm and 961 mm, respectively.

B. Dataset Measurement Campaigns

A general overview of the collected datasets can be found in Table III. Multiple datasets were captured across three heterogeneous environments, each with different characteristics. Moreover, the datasets were captured using different hardware types and configurations. The in-house captured datasets X_{base} and X_O in this research were collected with Wi-PoS devices [54] which include UWB Decawave (now Qorvo) DW1000 transceivers with a sub-GHz backbone for MAC scheduling and omnidirectional antennas. CIR information together with other useful metadata, e.g. first path index, peak path index, rx power, first path power, etc., was collected and stored for further data pre-processing. The CIR is a complex-valued array of 1016 samples, which is the result of the used PRF setting at 64 MHz. [51] The majority of the reported first paths is between index 740 and 750. By limiting the CIR collection to 300 samples (from 650 untill 950), at least 90 ns before the detected first path was collected and around 200 ns after the detected path, the complexity is reduced, while the ranging rate and throughput is increased. This way, the device collects all signals from paths up to 27 m longer than the direct path. Samples before 650 can usually contain noise and can be ignored. The publicly available dataset X_U [34] was captured using the DWM1001-DEV UWB platform, which consists of the same DW1000 UWB transceiver with a different omnidirectional antenna compared to the Wi-PoS devices, with a smaller form factor, different gain levels and radiation pattern, impacting the CIR. The dataset accommodates all CIR samples, although we use only a small portion (see Section IV-C). Although most research focuses on Qorvo's DW1000 or DW3000 chips, the proposed solution should still be useful for performance improvements on other UWB chips, as long as sufficient CIR samples around the first path index, or similar features are logged. As investigated in [55], some CIRs from different UWB chips can have higher amplitude multipath components and or shorter CIR lengths. In these conditions, transfer learning or training from scratch with different CIR segment lengths may be the only viable method.

TABLE IV: List of features extracted from the UWB transceiver

Features	Description
F_1	The signal amplitude of third sample after the FP
F_2	The signal amplitude of second sample after the FP
F_3	The signal amplitude of first sample after the FP
CIR_pwr	The CIR power (scaled with 2^{17} to fit 16-bit register [51])
Std_noise	σ of the noise reported in the CIR accumulator
$RXPACC$	The number of preamble symbols accumulated in the receiver
$fpindex$	The index of the detected FP, reported in the register
FP_pwr	The estimated FP power level and can be calculated as FP_pwr (in dBm) $= 10\log_{10}\left(\frac{F_1^2+F_2^2+F_3^2}{N^2}\right) - A$ where A is 113.77 for PRF = 16 MHz and 121.74 for PRF = 64 MHz, N is the preamble accumulation count
RX_pwr	The estimated RX power level and can be calculated as RX_pwr (in dBm) = $10\log_{10}\left(\frac{CIR_{pwr}\times 2^{17}}{N^2}\right) - A$
PD	The power difference between the RX_pwr and FP_pwr and can be calculated as PD $= RX\_pwr - FP\_pwr$
PR	The power ratio between the RX_pwr and FP_pwr and can be calculated as PR $= RX\_pwr/FP\_pwr$
$\widehat{\Delta}a_q t_r$	The estimated range calculated as $\widehat{\Delta}a_q t_r = \tau \times c$ where c represents the speed of light in m/s and τ is the signal propagation time from tag t_i to anchor a_q.

C. Data Pre-processing

1) Feature calculation: In addition to CIR feature based approaches for UWB error correction and NLOS detection, which are widely used [23], [36]–[38], we also propose a TL framework for UWB error correction and (N)LOS detection using UWB features. For signal features, the conventional features used mainly rely on the statistical condition of the received signals (e.g., maximum amplitude, mean excess delay, and rise time). However, in most cases, the FP is the strongest under LOS. Therefore, the power difference method have attracted larger body of work in different current research for featured based UWB error correction and NLOS detection. This feature is based on taking the Power Difference (PD) between the estimated Received signal (RX) power and FP power which can be used to decide whether the signal is LOS or NLOS. Specifically, if the PD between RX power and FP power is less than 6 dB, then the channel is likely to be LOS, while if the PD is greater than 10 dB the channel is likely to be NLOS [51] (page-46). In total 12 features, as considered in [23], are extracted from UWB sensors for our evaluation. The brief description of each feature is shown in Table IV.

2) Raw CIR data: Before we feed the CIR data into the NN, proposed in Section V, we first process the raw CIR data in the pre-processing phase. The raw CIR pre-processing step is three-fold. (i) Firstly, we convert the IQ-sampled array into a RSSI-sampled array ($RSSI = \sqrt{I^2 + Q^2}$). The real and imaginary components of the complex IQ-samples can be represented by the Cartesian coordinate system. Next, using the Euclidean distance from the origin in the complex plane and the found Cartesian coordinates gives us the absolute value of the complex IQ sample, corresponding to the amplitude of the signal (RSSI). (ii) Secondly, the RSSI-sampled array is trimmed further to 150 samples, with 50 samples before the FP index and 100 afterwards (including the first path). (iii) Lastly, min-max normalization is applied to each CIR of the dataset. Min-max normalization scales each CIR, so that the highest value is 1 and the lowest is 0 ($CIR_{norm} = \frac{(CIR - min(CIR))}{(max(CIR) - min(CIR))}$). Data normalization resulting in smaller numerical values can be beneficial for the training optimization of NNs and their generalization capabilities. It also ensures that the model learns more SNR and peak features of the signals instead of absolute signal strength features, which can vary by a significant factor for different settings and environments (average distance between tag and anchor) and do not necessarily indicate larger errors or (N)LOS signal propagation.

In contrast to solutions that include CIR compression in the raw data pre-processing [56], in this paper we focused on reducing the length and rescaling the CIR samples. This ensures exploring all available information around the first path, as it is used by the leading edge algorithm to estimate the time of flight and where most errors occur.

V. Machine- and transfer learning approaches

In this section, first we compare two ML-based approaches: feature-based and (raw) CIR-based ML for UWB positioning improvements. We expect both approaches to have a clear trade-off between model (and data) complexity, accuracy and TL capabilities. In the results section, we will analyze and try to quantify this trade-off for both error correction and (N)LOS detection. Next, we propose TL steps to adapt the UWB models towards new environments using new, but limited information. Finally, in the experimental evaluation methodology we present the implementation details of the evaluation results.

A. UWB ML models

For each ML approach, we investigate two methods for UWB improvements: error correction and (N)LOS detection. The goal of error correction is to estimate the error on the UWB ranges. This error is a direct result of the leading edge algorithm implemented on-chip failing to correctly and precisely identify the first path. This occurs due to number of factors including the received signal strength and used bandwidth (impacting the width of the first peak) [57], low SNR ($<$ 6 dB) [58] and due to the presence of humans or other obstructions (NLOS) [59]. With ML, the goal is to identify this error (output data) using precise ground truth and the associated CIR data (input data). Maximizing this correction performance will have a positive impact on the localization accuracy, which uses multiple ranges to triangulate the position as illustrated in Figure 1. N(LOS) detection permits further UWB localization improvements with intelligent anchor selection. More specifically, given at least 4 anchors, an anchor selection algorithm can mitigate one or more ranges based on their (N)LOS information before providing the ranges to the localization algorithm. The following describes the model architecture for feature-based and raw CIR-based models for error prediction and (N)LOS classification.

1) Feature-based machine learning: Feature-based ML for error correction and (N)LOS detection typically uses DNNs to make predictions based on previously extracted features. In this paper, we employ a DNN architecture, as presented in Table V. The first layer of the NN is the input layer and expects an array of 12 values (features). Next, four dense layers are

TABLE V: NN architecture for error correction and (N)LOS detection. Both UWB model architectures (feature-based and CIR-based) use the same Dense layers (blue), while the CIR-based CNN consists of 7 additional layers (purple) on top of this to automatically discover and learn features from the raw CIR data.

Layer name	**Activation**	**Output size**	**Layer name**	**Activation**	**Output size**
CIR-based (454,259 parameters and 4,443,552 FLOPs)					
Input		(150x1x1)			
Conv2D (128x(16x1))	ReLu	150x1x128			
MaxPoo2D (2x1)		75x1x128			
Conv2D (64x(8x1))	ReLu	75x1x64			
Conv2D (32x(2x1))	ReLu	75x1x32			
BatchNorm		75x1x32			
Dropout 25%		75x1x32	**Feature-based (23,401 parameters and 23,081 FLOPs)**		
Flatten		2400	Input		12
Dense (150)	ReLu	150	Dense (150)	ReLu	150
BatchNorm		150	BatchNorm		150
Dropout 20%		150	Dropout 20%		150
Dense (100)	ReLu	100	Dense (100)	ReLu	100
Dropout 20%		100	Dropout 20%		100
Dense (50)	ReLu	50	Dense (50)	ReLu	50
Dropout 10%		50	Dropout 10%		50
Dense (25)	Sigmoid	25	Dense (25)	Sigmoid	25
Dense (1) (Error) or	Linear or	1 or	Dense (1) (Error) or	Linear or	1 or
Dense (2) ((N)LOS)	Softmax	2	Dense (2) ((N)LOS)	Softmax	2

attached followed by the final dense output layer. In between, *BatchNormalization* and *Dropout* layers are added, which both try to mitigate overfitting of the training dataset. Each dense layer uses a *ReLu* layer as a non-linear activation function to introduce non-linearity into the training and enables more effective deep learning. The output layer has a linear activation function for the UWB error prediction task (with one output neuron, corresponding to the floating point precision error) or a *Softmax* activation function for the (N)LOS classification task (with two output neurons, one for LOS and one for NLOS classification).

2) Raw CIR-based machine learning: The RAW CIR-based model uses a CNN architecture, as presented in Table V. The architecture was inspired from the CNN published in [31], with the DNN portion further fine-tuned for optimal performance for the feature-based approach. As the input here considers an even smaller input size (150 in this research vs 500 samples in [31]), the convolutional layers were further optimized, resulting in a slightly different overall architecture. At the input layer, the model expects 150 signal strength samples, with 50 samples before the first path index, and 100 after (including the first path index). Next, 3 convolutional layers are trained to extract local time series features from each CIR. These layers include *kernel regularization*, to reduce the size of the weights, *ReLU* activation functions, zero padding, and one max-pooling layer in between. Afterwards, a flatten layer reshapes the activations from the convolutional layer to 1D, and lets the data flow towards the fully connected layers. These last layers have the same architecture as the feature-based ML model to have a symmetric comparison between the two ML approaches.

B. Transfer Learning Optimization Strategy

As generalization towards multiple environments remains challenging, training models in new environments is important to boost accuracy. TL is a ML approach that can help with this task, by transferring the knowledge already learned on X_{base}, e.g. time series features of the CIR, to data captured in X_O and X_U. In this subsection, we describe the (automatic) TL optimization strategy, which acts as a general framework for a broader scope of use-cases with unseen environments.

Step 1 ***base model training:*** The first step combines existing datasets and trains a base model NN. Best practices in training such as dropout, (batch)normalization, large train-test splits, etc., ensure steps towards best-effort generalization. In this paper, the base models architecture will differ for the feature-based and non-feature based models. Convolutional layers in the non-feature based model learn high-level features, e.g. temporal features, from the raw CIR data, while the feature-based model assumes high-level features are already extracted. Once the convolutional layers extract high-level features from the raw data, fully connected layers follow in the NN. To increase model capability fairness, both models share the same fully connected layer architecture. Table V presents more details of the NNs for both feature and non-feature based methods.

Step 2 ***training with frozen layers:*** With the ML_{base} trained, the next step makes some trainable layers (H) (i.e. convolutional layers and dense layers) untrainable and creates ($configure$) a partially frozen model. The remaining trainable layers can be either randomly initialized or they can preserve the weights from the base model. In this paper, we utilized weight preservation which can enhance model convergence significantly by providing a good starting point for the optimization process and preventing overfitting during training. The decision on which layers to freeze is made by based on a trial-and-error or consider these layer options as hyperparameters and optimize them in an iterative manner. The main expected benefit from this step is reduced training time and a decreased required training samples to meet an expected accuracy. DNN models working with manual extracted feature can freeze fully connected layers, while CNN models trained using raw data can additionally freeze convolutional layers.

Step 3 ***Fine-tuning with unfrozen layers:*** This step of the TL framework is optional and follows either after the base model training step or after the frozen layer training step.

Although (latent) features represented in the frozen layers should ideally be transferable, it is, however, possible for a model to achieve higher accuracy when these features still need to be slightly adapted to the new environment. Reducing the learning rate (lr) and epochs in this step ensures earlier learned features are not neglected completely.

Deciding how to finetune the aforementioned hyperparameters of steps 2 and 3 can be challenging and requires expertise of the developer. Furthermore, blindly trying all different combinations (in this paper 256) is computationally expensive and becomes infeasible quickly. In our TL framework, we include automatic hyperparameter finetuning with Bayesian optimization using the Gaussian process to mitigate manual hyperparameter tuning and reduce the computational overhead. In this paper, the hyperparameters are the optimal combination of which frozen layers (H) to use (step 2) and whether to perform step 3, configurable with Boolean parameter $\delta_{finetuning}$. The finetuning process is configured as an extra set of (50) epochs with a low learning rate (lr) of 0.0001, after the first round of 500 TL epochs have been executed at a higher lr of 0.001. The input of this optimizer is the current state (H and $\delta_{finetuning}$) together with the model accuracy using these hyperparameters. We configured the optimizer to start with 15 random initialization hyperparameters configurations and further explore and exploit more accurate configurations in additional 15 steps, which are experimentally derived values based on an accuracy and training time trade-off. More details of this optimizer can be found in [60]. The output of the optimizer is the hyperparameters H' and $\delta'_{finetuning}$ during the automatic discovery phase which are estimated to increase accuracy. We propose the integration of the Bayesian optimizer for UWB TL models in Algorithm 1.

Algorithm 1 Transfer learning optimization algorithm

Input:
X_{base} *(Features or CIR)* ▷ Base-dataset
$[ML_{base}]$ ▷ Optional pre-trained model
$X_{new}^{'}$ ▷ Dataset new environment
n_b ▷ Number of Bayesian optimization steps
Output: ML_{new}

```
if ML_base then
    while ML_base not converged do                    ▷ Step 1
        loss ← ML_base(X_base)
        optimize(ML_base, loss)
    end while
end if
ML_new ← ML_base
H ← ML_new.get_trainable_layers()
δ_finetuning ← True
i ← 0
while i ≠ n_b do
    H', δ'_finetuning =
    Bayesian_opt(ML_new, H, δ_finetuning)             ▷ 4)
    configure(ML_new, H', lr = 0.001)
    while ML_new not converged do                     ▷ Step 2
        loss ← ML_new(Ẋ_e1)
        optimize(ML_new, loss)
    end while
    if δ'_finetuning == True then
        configure(ML_new, [True, ...], lr = 0.0001)
        while ML_new not converged do                 ▷ Step 3
            loss ← ML_new(Ẋ_e1)
            optimize(ML_new, loss)
        end while
    end if
    keep_best_model(ML_new)
    i ← i + 1
end while
return get_best_model()
```

C. Experimental Evaluation Methodology

In this paper we intentionally focus on the performance improvements for (ranging) error correction and (N)LOS detection, which are the most predominant parameters impacting the positioning accuracy [31].

The TL steps outlined in the optimization strategy can be performed with a small number of samples from a new unseen environment. As such, in this paper, we assume it is possible to collect CIRs from a few known positions. It is important to analyze the number of samples we need to capture to allow effective model TL. Obviously, the best results are achievable when the model uses a large and varied (spread across the environment) amount of samples. However, this is often not a realistic assumption and diminishes the potential of TL. In this paper, we evaluate sample selection from random positions spread out in the unseen environment, obtaining a high spatial diversity. Although this can require many measurement points (depending on the size of the environment), we will only investigate a range of few samples $K << M$. Specifically, the evaluation analyzes TL with K samples ranging between 25 and 400.

To mitigate overfitting, we applied early stopping when the validation loss didn't decrease for more than 30 epochs. Additionally, the learning rate automatically reduced when the loss didn't decrease for more than 10 epochs. To aim for the highest accuracy in the unseen environment, the TL samples were split into a training (70%) and validation sets (30%). The remaining data of the unseen environment is then considered as a test set, of which the results are shown in Section VI. The *Adam* optimizer was used to train the models for 300 epochs with a dynamic learning rate starting from 0.001, together with a MAE loss function. To construct the MAE loss each training step, a batch size of 1024 was used, while a smaller batch size of 16 can help the NN effectively learn the variation during TL with limited samples. All models were trained using the TensorFlow framework on a high-end NVIDIA $V100$ GPU, available in our in-house Virtual Wall [61].

VI. Analysis and Experimental Results

In this section, we evaluate the performance of error correction and (N)LOS detection ML models with feature-based and raw CIR-based input data. First, we discuss the optimal

TABLE VI: This table presents optimal found configuration by the Bayesian optimizer for each dataset, ML task and data type. The '+' symbol indicates that the layer was selected trainable (unlocked), while the '-' symbol indicates a locked layer. There are no clear TL hyperparameter optimizations that can be derived for a specific use case. Manually determining the optimal configurations requires trial-and-error which can be challenging and time-consuming. Some patterns can be found, such as unlocked (+) and locked (-) convolutional layers for error correction and NLOS detection, respectively. Nonetheless, the optimal found hyperparameter configurations indicate the need for an automatic optimization strategy.

Dataset	**ML task**	**Data type**	**Fine-tuning**	**Trainable Conv.**			**Trainable FC**			
				I	II	III	I	II	III	IV
X_O	Error	Features	✓				-	+	+	+
		CIR	✗	+	+	-	+	+	+	+
	(N)LOS	Features	✓				-	+	+	-
		CIR	✓	-	-	+	+	-	-	+
X_U	Error	Features	✓				-	+	+	+
Hallway		CIR	✗	+	+	-	+	+	-	+
	(N)LOS	Features	✗				-	+	+	+
		CIR	✓	-	-	+	+	-	+	+
Hallway	Error	Features	✗				+	+	-	+
1st floor		CIR	✗	+	-	-	+	+	+	+
	(N)LOS	Features	✗				+	+	+	+
		CIR	✓	-	-	+	+	+	+	+
Lab room	Error	Features	✓				+	+	+	+
		CIR	✗	+	+	-	+	+	-	+
	(N)LOS	Features	✓				+	-	-	+
		CIR	✓	-	-	+	+	-	+	+

hyperparameter configurations found for each experiment using the Bayesian optimizer. Next, we analyze the performance of the ML models on datasets X_O and X_U. We compare the (i) UWB - no correction base ranging accuracy (with no error mitigation algorithms applied), (ii) ML generalization (using ML_{base}), (iii) ML from scratch (ML_{new}) with $k = 50$ and (iv) TL performance ($\acute{M}L_{new}$) in considered new environment (X_O or X_U). In the rest of the paper, we use 'ML from scratch' and 'training from scratch' interchangeably. Next, we perform an identical evaluation, but on a publicly available dataset, captured in different environments with different UWB hardware and UWB configurations. After that, the impact on the TL accuracy with the number of chosen TL samples is illustrated, and finally followed by a complexity analysis between feature and raw CIR-based solutions.

A. Automatic Hyperparameter Optimization

Various ways to perform TL can be seen as hyperparameters of the knowledge transferring process. In this paper, we explore trainable (or locked) layers of the models as hyperparameters, together with the finetuning process ($\delta finetuning$). Trainable layers (H) are the collection of (three) convolutional layers and (four) fully connected layers, which contain many trainable parameters. For the experiments using feature-based data, only the fully connected layers are chosen, as the convolutional layers are typically used for feature extraction of raw data. Table VI presents the optimal configurations found by the Bayesian optimizer for each dataset, ML task and data type. These configurations are then further used throughout the remainder of the evaluation in this paper. Some patterns can be found, such as unlocked (+) and locked (-) convolutional layers for error correction and NLOS detection, respectively. Nonetheless, the optimal found hyperparameter configurations indicate the need for an automatic optimization strategy. Generally, we noticed that locking the convolutional layers is a good idea, as this consistently provides good results, even if they are not the global optimal solution. For most configurations the accuracy diverged only a few percent, using an automatic hyperparameter optimizer like a Bayesian optimization approach is required for reaching optimal accuracy.

B. Generalization Towards Multiple Environments

1) Same UWB configuration: To assess the TL performance of the proposed approach, we first analyze error correction and (N)LOS detection in X_O, which is a different environment compared to X_{base}, but uses the same radio configurations. In Figure 3a, we compare the UWB - no correction, which is the MAE of UWB ranging in X_O, without any optimization algorithms applied. Next, *ML generalization* shows the performance of models trained on X_{base}, validated on X_O. *ML from scratch* and *TL* both uses 50 samples of X_O to improve the ML performance. The *ML from scratch* approach does not use any previously acquired knowledge, and starts training a model from scratch with only these 50 samples, while the proposed TL approach uses ML_{base} and further applies TL techniques as described in Section V-B. Figure 3a illustrates the shortcomings of the *ML generalization*, where the error after correction is worse than the UWB - no correction performance using the feature-based approach. The CIR-based approach, where neural networks have the ability to extract more (generalizable) features, does improve the accuracy. However, it does not reach performance gains when it would be applied in a similar environment. A model trained from scratch with 50 samples measured in the new environment does offer significant improvements for both the feature-based and CIR-based approaches compared to the UWB - no correction MAE. However, most performance gains can be seen when applying the proposed TL strategy. We see more than $2\times$ the performance increase for feature-based error correction and almost $3\times$ for the CIR-based approach. Compared to generalization, TL with only 50 samples offers an increase of 56.9% and 54.0% over the *ML generalization* performance.

Figure 4 presents a CDF to further analyze error correction at different percentiles. At the 95th percentile, both feature-based and CIR-based approaches have a large relative improvement of 57.8% and 48.5% when applying TL compared to the *ML generalization* performance of the base model. For 95% of the samples, with TL we achieved an error of less than 300 mm, which is a significant accuracy improvement over the UWB - no correction 800 mm error. Together with a median error of only a few centimeters, a reliable target accuracy of a few decimeters can be achieved, even in harsh NLOS environments.

Figure 3b shows similar improvements for the (N)LOS classification results. Both feature- and CIR-based approaches using *ML generalization* fail to provide satisfactory results with detection accuracies of 64.9% and 78.1%, respectively. Again, training the model from scratch does offer higher

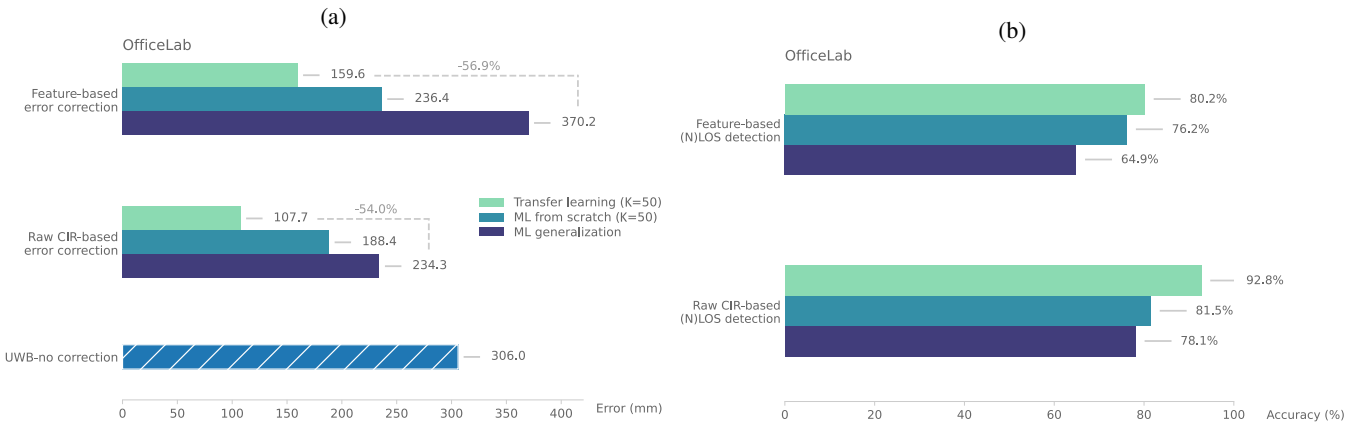


Fig. 3: Performance of the feature- and raw-based ML models in X_O for (a) error prediction models and (b) (N)LOS detection. The following data volume parameters are used: K=50, M=44894, K/M=0.11%.

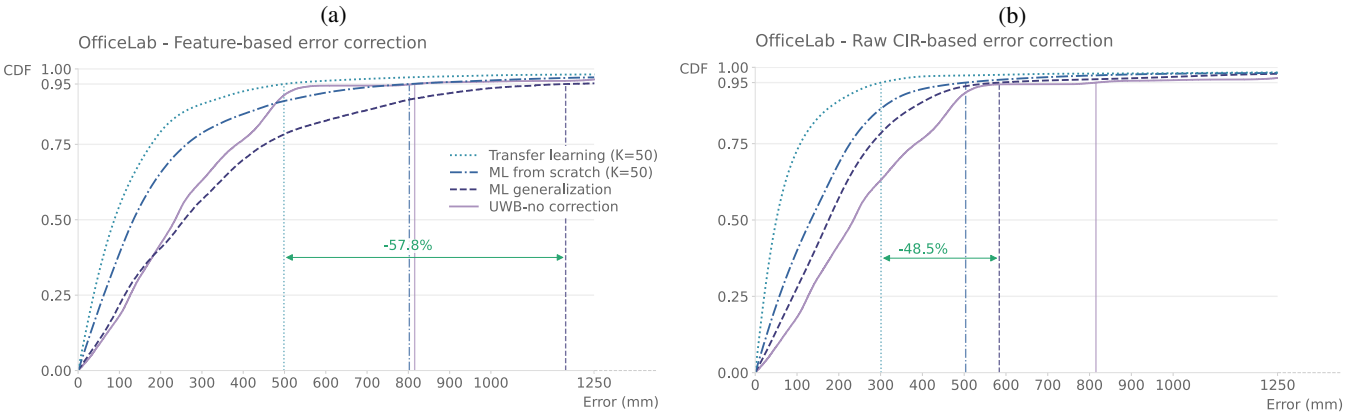


Fig. 4: The cumulative distribution function (CDF) for the office dataset X_O shows a large precision improvement. The base (generalization) model ML_{base} is trained on the industrial datasets X_{base} and further transferred using 50 samples of X_O.

accuracy than the generalization model, but is outperformed by applying TL. Here, the accuracy improved significantly, where the CIR-based approach reaches with 92.8% a relatively high accuracy. Although this NLOS accuracy of 92.8% is not perfect, a significant improvement can be found by applying transfer learning versus generalization. As shown in [10], NLOS can effectively improve the accuracy of the UWB positioning by applying anchor selection, even when the accuracy is not 100%. For both error correction and (N)LOS detection, it is clear that the CIR-based approach outperforms the feature-based approach. Although the CIR-based generalization appears more robust in new environment, the relative performance gains ($\pm 50\%$) compared to the feature-based approach is similar and results in an overall higher accuracy.

2) Different UWB radio configurations: Not every UWB localization deployment will be similar in terms of settings, UWB chip, antenna's used, etc. To this end, the first three figures in Figure 5 shows the TL performance on the publicly available dataset of [34]. In this dataset, we selected three environments with sufficient samples for evaluation. Although the error correction improvements using TL are still significant, the relative improvement is much smaller. Interestingly, we also see that when training from scratch the model offers similar performance, and even leads to higher improvements for some feature-based cases. The base model that is used for TL is trained on the IIoT dataset with a different configuration. As a result, transferring the learned knowledge in the from the base model towards a different environment with also a different configuration is less ideal than only changing the environment. As such, the features learned in one dataset, which contains signals subject to different channels and antennas from one type of device, are not ideal to use on different devices. Moreover, the error made by different UWB systems can vary a lot depending on the settings, while this is not always reflected in the CIR and available and extracted features. In contrast, the features learned in one environment do seem useful knowledge when only the environment changes. The 90th percentile results of error correction in Figure 6 further confirms this conclusion with relative improvements between 2.7% and 27.4% between model generalization and TL, which is far smaller than the results on X_O. In the lab environment, the feature-based approach achieves a higher performance improvement than

the CIR-based approach. Despite achieving only small error correction improvements in different configurations, the last three figures in Figure 5 show large a large improvement in (N)LOS detection accuracy. The difference between feature- and CIR-based models is small for the TL approach, however, the generalization performance of the CIR-based model is higher. The (N)LOS label does not depend on the UWB configuration, but rather on the topology and physical layout of the environment. To this end, regardless of different configurations, TL improvements for (N)LOS detection can be expected in new environments.

C. Impact of Changing the Number of Samples for TL

Up to now we have considered using $K = 50$ samples from the new environment to enable TL. Ideally, we want to keep this number small to limit the overhead of manual data collection and labeling. In Figure 7, we investigate the impact of using $K = 25$ to $K = 400$ samples. A downwards trend can be observed for error correction of both feature-based and raw CIR-based models. Compared to collecting only 25 samples, an increase up to 51.6 mm and 100.4 mm can be seen when collecting $K = 400$ samples for the feature-based and raw CIR-based models, respectively. Similarly, an upward trend exists for (N)LOS detection of both feature- and CIR-based models. Both types of models benefit from more TL samples and achieve an additional 7.8% and 9.5% for feature- and CIR-based models, respectively, as compared to no TL. Additionally, building on top of the best results from the previous subsections (CIR-based ML), Figure 7 shows the performance of traditional ML using CIR data, where a model is trained from scratch using the data available in a new environment. For a small K, CIR-based TL outperforms the ML from scratch method. To reach the same error prediction accuracy of the traditional ML ML from scratch method using K=200 samples, with transfer learning only K=25 samples are needed, a difference of 8x. For a larger K, this difference becomes smaller as training from scratch becomes more effective. Nonetheless, depending on the required effort, choosing only a few samples can still be the desired choice, and offers significant performance gains compared to no TL ($K = 0$ samples) or not applying any ML approaches (UWB - no correction performance).

D. Impact of ranging error correction on localization accuracy

Finally, we analyze the impact of improved ranging errors on the UWB localization accuracy, calculated using the least square method. Table VII shows the UWB localization accuracy in X_O when no correction has been applied, as well as the corrected ranges using ML generalization, ML from scratch ($K = 50$) and the proposed transfer learning method ($K = 50$). The improvements to the localization accuracy follow a similar trend as for the ranging error improvements. Clearly, the proposed raw transfer learning method outperforms the feature-based methods, generalization and ML from scratch. As such, it can effectively improve the localization accuracy in challenging NLOS environments from 643 mm to 245 mm.

Method	**Localization accuracy (X_O)**	
UWB - no correction	642.9 mm	
	Feature-based	**Raw**
ML generalization	638.3 mm	482.5 mm
ML from scratch ($K = 50$)	424.3 mm	392.6 mm
Transfer learning ($K = 50$)	294.7 mm	245.1 mm

TABLE VII: Localization accuracy of X_O with ranging error correction shows large gains using TL

E. Complexity Analysis

Not only is it interesting to compare the accuracy of two different ML approaches (feature- and CIR-based), they also have different characteristics regarding the complexity. Additionally, TL can add complexity on top of training ML_{base}, but it can also reduce the convergence time when accuracy in a new environment is sub-optimal. In this subsection we further analyze the complexity of the proposed solution.

1) Model complexity: As shown in Table V, the complexity of feature-based models used in this paper are lower compared to the CIR-based models due to two reasons: (i) the dimensionality of the input layer is smaller and (ii) the amount of fully connected neurons is smaller. The feature-based model has less trainable parameters ($23{,}401$ vs $454{,}259$) and FLOPs ($23{,}081$ vs $4{,}443{,}552$) as compared to the CIR-based model. On the other side, the calculation of some features requires additional calculations, which is not the case for the raw CIR data. Typically, these features are calculated from raw data by applying (expert) rule-based algorithms. However, the Qorvo DW1000 UWB chip already provides most of the diagnostics features (see Table IV), which can directly be read from the radio chip registers for each UWB range, thus limiting the calculation overhead to a negligible amount.

2) Transfer learning complexity: The TL process first requires the training of a base model. This model can either be provided, or trained with a existing dataset, where this complexity was discussed above. Next, the model is transferred with a small dataset $\acute{X}_{new}$ with K samples, which already will significantly reduce the complexity as compared to training a model from scratch with X_{new} with M samples, where $K << M$ (a varying number of K was illustrated in Figure 7). However, searching for the optimal TL methodology can require $O(H \times K)$ operations, with H the number of Bayesian optimization search steps as compared to only $O(M)$ with a fixed TL configuration. Depending on the dataset reduction factor $\frac{M}{K}$, TL can result in a lower computational complexity than training a model with X_{new}. Moreover, TL only happens once in each new environment, where the computational effort of the trained model remains unaltered. Requiring significantly less training samples in each new environment will reduce manual data collection and labeling effort.

VII. Conclusions and Future Work

To enable high precision indoor positioning systems ($1-10$ cm) using UWB in both (N)LOS environments, ML algorithms have been proposed to improve the errors of UWB localization systems. However, these models lack generalization capabilities towards new environments and UWB configurations.

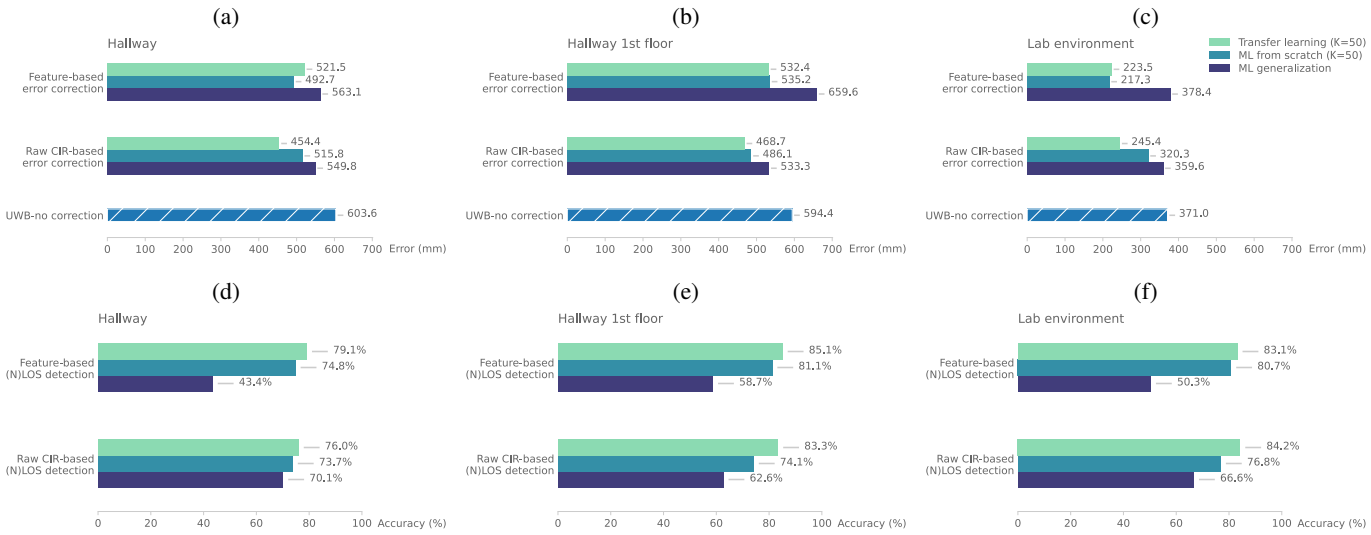


Fig. 5: Error prediction for both feature-based and raw-based TL in X_U in (a) the hallway environment, (b) the hallway environment on another floor and (c) the lab room environment. Additionally, the accuracy of the (N)LOS detection performance is shown in the same respective environments in (d), (e) and (f). The following data volume parameters are used for (a) and (d): K=50, M=5345, K/M=0.94%, for (b) and (e): K=50, M=5509, K/M=0.91% and for (c) and (f): K=50, M=2871, K/M=1.74%.

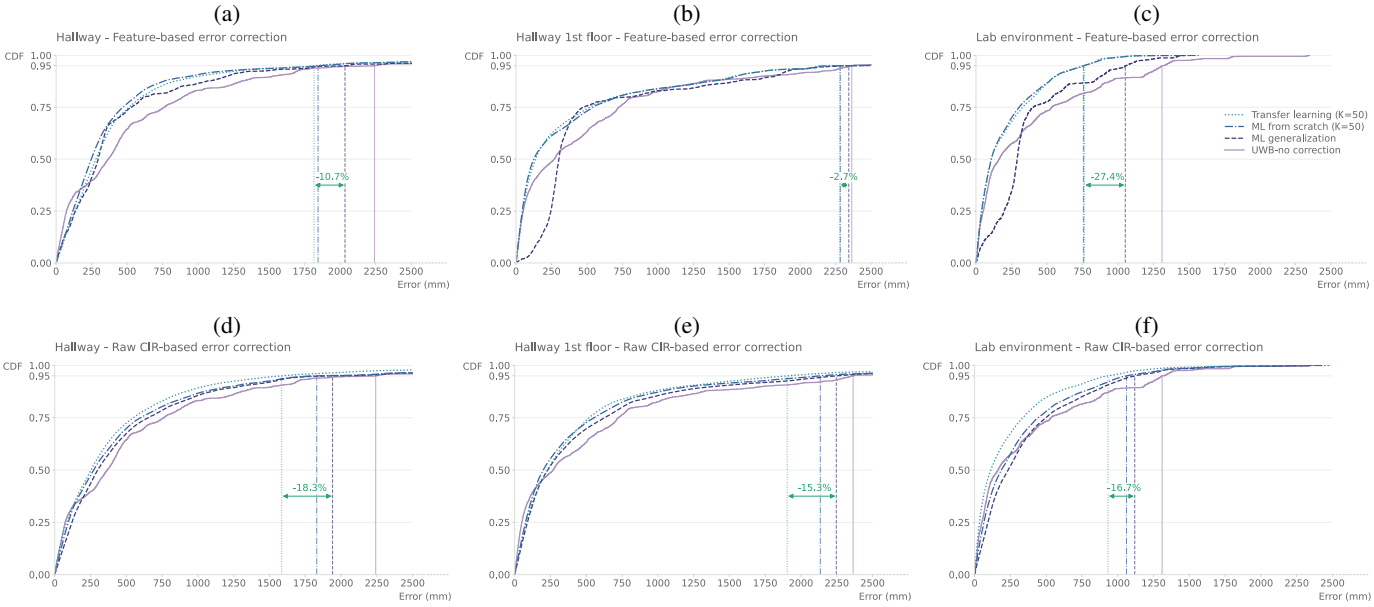
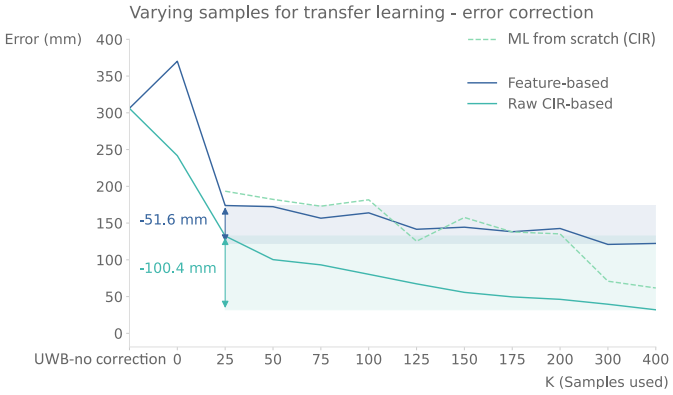


Fig. 6: Cumulative distribution function shows a smaller increase of the UWB ranging precision after applying TL on dataset X_U, which was captured in a different environment with a different configuration.

As such, enabling these models in new environments and UWB configurations is a challenging and expensive manual process requiring the collection of large training datasets for each environment. In this paper, we proposed automatic optimizations for TL DNNs towards new environments and UWB configurations. While keeping the complexity small due to a limited number of training samples, the optimization steps include an Bayesian optimizer which selects which layers to freeze during TL and whether to apply a final finetuning step with all layers unfrozen and a smaller learning rate. Our results demonstrated 50% error improvements and 15% (N)LOS classification accuracy improvements (for both feature- and CIR-based approaches) over a model which was trained only in a different environment. We also showed the expected gains for both approaches from using 25 until 400 transfer learning samples. The highest absolute accuracy was typically achieved by the CIR-based approach, where with as few as 50 samples from the new NLOS environment, we showed ± 10 cm precision after error correction with 93% (N)LOS detection. The presented results demonstrate high precision UWB localization (from 643 mm to 245 mm) through ML with minimal data collection effort in challenging

(a) UWB error correction.



(b) UWB (N)LOS detection.

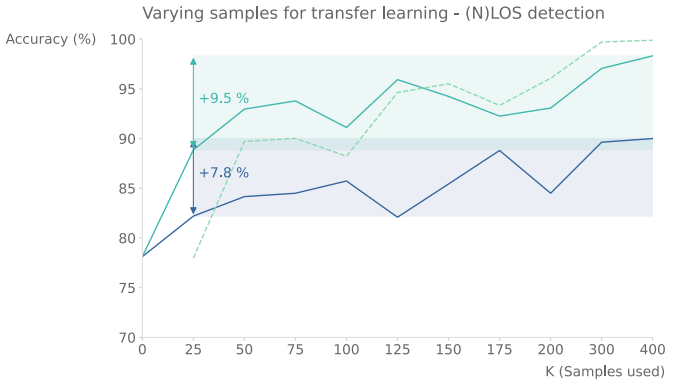


Fig. 7: Impact of using different number of samples from the new (unseen) environment X_O used for TL.

NLOS environments. In future work, we foresee sampling selection strategies in new environments and configurations (e.g. selecting only LOS samples or samples with a low or high standard deviation at a position) to further limit the overhead of manual data collection. Additionally, manual labeling effort can be alleviated with the use of unsupervised transfer learning, self-labeling and online learning techniques to quickly adapt a pre-trained NN to any new condition.

Acknowledgments

This work was supported by the Fund for Scientific Research Flanders, Belgium, FWO-Vlaanderen, FWO-SB, under Grant 1SB7619N.

References

[1] L. Cheng, A. Zhao, K. Wang, H. Li, Y. Wang, and R. Chang, "Activity recognition and localization based on uwb indoor positioning system and machine learning," in *2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, 2020, pp. 0528–0533.

[2] H.-B. Li, R. Miura, H. Nishikawa, T. Kagawa, and F. Kojima, "Tracking of warehouse forklifts using an indoor positioning system based on ir-uwb," in *Proceedings of the International Conference on indoor positioning and navigation (IPIN 2016), Madrid, Spain*, 2016, pp. 4–7.

[3] R. Bazo, C. A. da Costa, L. A. Seewald, L. G. da Silveira, R. S. Antunes, R. d. R. Righi, and V. F. Rodrigues, "A survey about real-time location systems in healthcare environments," *Journal of Medical Systems*, vol. 45, no. 3, pp. 1–13, 2021.

[4] S. Sun, J. Hu, J. Li, R. Liu, M. Shu, and Y. Yang, "An ins-uwb based collision avoidance system for agv," *Algorithms*, vol. 12, no. 2, p. 40, 2019.

[5] S. Krishnan, R. X. M. Santos, E. R. Yap, and M. T. Zin, "Improving uwb based indoor positioning in industrial environments through machine learning," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2018, pp. 1484–1488.

[6] M. Stahlke, S. Kram, T. Mumme, and J. Seitz, "Discrete positioning using uwb channel impulse responses and machine learning," in *2019 International Conference on Localization and GNSS (ICL-GNSS)*. IEEE, 2019, pp. 1–6.

[7] M. Stahlke, S. Kram, C. Mutschler, and T. Mahr, "NLOS detection using UWB channel impulse responses and convolutional neural networks," in *2020 International Conference on Localization and GNSS (ICL-GNSS)*. IEEE, 2020, pp. 1–6.

[8] Y. Xianjia, L. Qingqing, J. P. Queralta, J. Heikkonen, and T. Westerlund, "Applications of uwb networks and positioning to autonomous robots and industrial systems," in *2021 10th Mediterranean Conference on Embedded Computing (MECO)*. IEEE, 2021, pp. 1–6.

[9] C. L. Sang, M. Adams, T. Korthals, T. Hörmann, M. Hesse, and U. Rückert, "A bidirectional object tracking and navigation system using a true-range multilateration method," in *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2019, pp. 1–8.

[10] K. Yu, K. Wen, Y. Li, S. Zhang, and K. Zhang, "A novel NLOS mitigation algorithm for uwb localization in harsh indoor environments," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 686–699, 2018.

[11] W. Wang, D. Marelli, and M. Fu, "Multiple-vehicle localization using maximum likelihood kalman filtering and ultra-wideband signals," *IEEE Sensors Journal*, vol. 21, no. 4, pp. 4949–4956, 2020.

[12] A. A. Momtaz, F. Behnia, R. Amiri, and F. Marvasti, "NLOS identification in range-based source localization: Statistical approach," *IEEE Sensors Journal*, vol. 18, no. 9, pp. 3745–3751, 2018.

[13] Q. Z. Ahmed, K.-H. Park, and M.-S. Alouini, "Ultrawide bandwidth receiver based on a multivariate generalized gaussian distribution," *IEEE Transactions on Wireless Communications*, vol. 14, no. 4, pp. 1800–1810, 2015.

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

[15] F. Che, A. Ahmed, Q. Z. Ahmed, S. A. R. Zaidi, and M. Z. Shakir, "Machine learning based approach for indoor localization using ultra-wide bandwidth (UWB) system for industrial internet of things (IIoT)," in *2020 International Conference on UK-China Emerging Technologies (UCET)*. IEEE, 2020, pp. 1–4.

[16] W. M. Gifford, D. Dardari, and M. Z. Win, "The impact of multipath information on time-of-arrival estimation," *IEEE Transactions on Signal Processing*, vol. 70, pp. 31–46, 2022.

[17] B. Cao, S. Wang, S. Ge, and W. Liu, "Improving the positioning accuracy of UWB system for complicated underground nlos environments," *IEEE Systems Journal*, vol. 16, no. 2, pp. 1808–1819, 2022.

[18] L. Nosrati, M. S. Fazel, and M. Ghavami, "Improving indoor localization using mobile UWB sensor and deep neural networks," *IEEE Access*, vol. 10, pp. 20 420–20 431, 2022.

[19] Q. Tian, I. Kevin, K. Wang, and Z. Salcic, "An INS and UWB fusion approach with adaptive ranging error mitigation for pedestrian tracking," *IEEE Sensors Journal*, vol. 20, no. 8, pp. 4372–4381, 2020.

[20] L. Flueratoru, S. Wehrli, M. Magno, E. S. Lohan, and D. Niculescu, "High-accuracy ranging and localization with ultrawideband communications for energy-constrained devices," *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7463–7480, 2022.

[21] J. Jia, K. Guo, W. Li, X. Yu, and L. Guo, "Composite filtering for UWB-based localization of quadrotor UAV with skewed measurements and uncertain dynamics," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2022.

[22] K. Zhao, M. Zhu, B. Xiao, X. Yang, C. Gong, and J. Wu, "Joint rfid and uwb technologies in intelligent warehousing management system," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11 640–11 655, 2020.

[23] A. G. Ferreira, D. Fernandes, S. Branco, A. P. Catarino, and J. L. Monteiro, "Feature selection for real-time NLOS identification and mitigation for body-mounted UWB transceivers," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–10, 2021.

[24] J. B. Kristensen, M. M. Ginard, O. K. Jensen, and M. Shen, "Non-line-of-sight identification for UWB indoor positioning systems using

support vector machines," in *2019 IEEE MTT-S International Wireless Symposium (IWS)*. IEEE, 2019, pp. 1–3.
[25] A. Musa, G. D. Nugraha, H. Han, D. Choi, S. Seo, and J. Kim, "A decision tree-based NLOS detection method for the UWB indoor location tracking accuracy improvement," *International Journal of Communication Systems*, vol. 32, no. 13, p. e3997, 2019.
[26] B. Li, K. Zhao, and E. B. Sandoval, "A UWB-based indoor positioning system employing neural networks," *Journal of Geovisualization and Spatial Analysis*, vol. 4, no. 2, pp. 1–9, 2020.
[27] P. A. ShirinAbadi and A. Abbasi, "UWB channel classification using convolutional neural networks," in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2019, pp. 1064–1068.
[28] Z. Sun, K. Wang, R. Sun, and Z. Chen, "Channel state identification in complex indoor environments with st-cnn and transfer learning," *IEEE Communications Letters*, pp. 1–1, 2022.
[29] B. Yang, J. Li, Z. Shao, and H. Zhang, "Robust UWB indoor localization for NLOS scenes via learning spatial-temporal features," *IEEE Sensors Journal*, vol. 22, no. 8, pp. 7990–8000, 2022.
[30] L. Schmid, D. Salido-Monzú, and A. Wieser, "Accuracy assessment and learned error mitigation of UWB ToF ranging," in *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2019, pp. 1–8.
[31] J. Fontaine, M. Ridolfi, B. Van Herbruggen, A. Shahid, and E. De Poorter, "Edge inference for UWB ranging error correction using autoencoders," *IEEE access*, vol. 8, pp. 139 143–139 155, 2020.
[32] V. Rayavarapu and A. Mahapatro, "NLOS identification and mitigation in uwb positioning with bagging-based ensembled classifiers," *Annals of Telecommunications*, pp. 1–14, 2021.
[33] A. Albaidhani and A. Alsudani, "Fuzzy logic control for NLOS identification method in an indoor environment using UWB technology," *Int. J. Intell. Syst.*, vol. 13, no. 1, pp. 270–281, 2020.
[34] M. Stocker, M. Gallacher, C. A. Boano, and K. Römer, "Performance of support vector regression in correcting UWB ranging measurements under LOS/NLOS conditions," in *Proceedings of the Workshop on Benchmarking Cyber-Physical Systems and Internet of Things*, ser. CPS-IoTBench '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 6–11. [Online]. Available: https://doi.org/10.1145/3458473.3458820
[35] B. Van Herbruggen, J. Fontaine, and E. De Poorter, "Anchor pair selection for error correction in time difference of arrival (TDoA) ultra wideband (UWB) positioning systems," in *2021 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2021, pp. 1–8.
[36] C. L. Sang, B. Steinhagen, J. D. Homburg, M. Adams, M. Hesse, and U. Rückert, "Identification of NLOS and multi-path conditions in UWB localization using machine learning methods," *Applied Sciences*, vol. 10, no. 11, p. 3980, 2020.
[37] F. Che, Q. Z. Ahmed, F. A. Khan, and P. I. Lazaridis, "Anomaly detection based on generalized gaussian distribution approach for ultra-wideband (UWB) indoor positioning system," in *2021 26th International Conference on Automation and Computing (ICAC)*. IEEE, 2021, pp. 1–5.
[38] Z. Zeng, S. Liu, and L. Wang, "UWB NLOS identification with feature combination selection based on genetic algorithm," in *2019 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2019, pp. 1–5.
[39] C. Jiang, S. Chen, Y. Chen, D. Liu, and Y. Bo, "An UWB channel impulse response de-noising method for NLOS/LOS classification boosting," *IEEE Communications Letters*, vol. 24, no. 11, pp. 2513–2517, 2020.
[40] Q. Liu, Z. Yin, Y. Zhao, Z. Wu, and M. Wu, "UWB LOS/NLOS identification in multiple indoor environments using deep learning methods," *Physical Communication*, vol. 52, p. 101695, 2022.
[41] J. Fan and A. S. Awan, "Non-line-of-sight identification based on unsupervised machine learning in ultra wideband systems," *IEEE Access*, vol. 7, pp. 32 464–32 471, 2019.
[42] C. Jiang, J. Shen, S. Chen, Y. Chen, D. Liu, and Y. Bo, "UWB NLOS/LOS classification using deep learning method," *IEEE Communications Letters*, vol. 24, no. 10, pp. 2226–2230, 2020.
[43] J. Park, S. Nam, H. Choi, Y. Ko, and Y.-B. Ko, "Improving deep learning-based UWB LOS/NLOS identification with transfer learning: An empirical approach," *Electronics*, vol. 9, no. 10, p. 1714, 2020.
[44] K. Adamkiewicz, P. Koch, B. Morawska, P. Lipiński, K. Lichy, and M. Leplawy, "Improving UWB indoor localization accuracy using sparse fingerprinting and transfer learning," in *International Conference on Computational Science*. Springer, 2021, pp. 291–302.
[45] R. Klus, L. Klus, J. Talvitie, J. Pihlajasalo, J. Torres-Sospedra, and M. Valkama, "Transfer learning for convolutional indoor positioning systems," in *2021 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2022, pp. 1–8.
[46] L. H. Yong and M. Zhao, "Indoor positioning based on hybrid domain transfer learning," *IEEE Access*, vol. 8, pp. 130 527–130 539, 2020.
[47] Y. Jiang and V. C. Leung, "An asymmetric double sided two-way ranging for crystal offset," in *2007 International Symposium on Signals, Systems and Electronics*, 2007, pp. 525–528.
[48] J. Karedal, S. Wyne, P. Almers, F. Tufvesson, and A. F. Molisch, "A measurement-based statistical model for industrial ultra-wideband channels," *IEEE Transactions on Wireless Communications*, vol. 6, no. 8, pp. 3028–3037, 2007.
[49] Q. Z. Ahmed and L.-L. Yang, "Reduced-rank adaptive multiuser detection in hybrid direct-sequence time-hopping ultrawide bandwidth systems," *IEEE Transactions on Wireless Communications*, vol. 9, no. 1, pp. 156–167, 2010.
[50] B. Großwindhager, C. A. Boano, M. Rath, and K. Römer, "Concurrent ranging with ultra-wideband radios: From experimental evidence to a practical solution," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, 2018, pp. 1460–1467.
[51] "Dw1000 user manual," https://www.qorvo.com/products/d/da007967, (accessed on 03/02/2023).
[52] "Iiot testbed," https://www.ugent.be/ea/idlab/en/research/research-infrastructure/industrial-iot-lab.htm/, (Accessed on 2022-02-04).
[53] "Officelab testbed," https://www.ugent.be/ea/idlab/en/research/research-infrastructure/officelab.htm, (Accessed on 2022-02-04).
[54] B. Van Herbruggen, B. Jooris, J. Rossey, M. Ridolfi, N. Macoir, Q. Van den Brande, S. Lemey, and E. De Poorter, "Wi-pos: A low-cost, open source ultra-wideband (UWB) hardware platform with long range sub-ghz backbone," *Sensors*, vol. 19, no. 7, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/7/1548
[55] L. Flueratoru, E. S. Lohan, and D. Niculescu, "Challenges in platform-independent uwb ranging and localization systems," in *Proceedings of the 16th ACM Workshop on Wireless Network Testbeds, Experimental evaluation & CHaracterization*, 2022, pp. 9–15.
[56] V. B. Vales, T. Domínguez-Bolaño, C. J. Escudero, and J. A. Garcia-Naya, "Using the power delay profile to accelerate the training of neural network-based classifiers for the identification of los and nlos uwb propagation conditions," *IEEE Access*, vol. 8, pp. 220 205–220 214, 2020.
[57] J. Sidorenko, V. Schatz, N. Scherer-Negenborn, M. Arens, and U. Hugentobler, "Decawave ultra-wideband warm-up error correction," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 1, pp. 751–760, 2020.
[58] S. Pala, S. Jayan, and D. G. Kurup, "An accurate uwb based localization system using modified leading edge detection algorithm," *Ad Hoc Networks*, vol. 97, p. 102017, 2020.
[59] S. Coene, C. De Cock, E. Tanghe, D. Plets, L. Martens, and W. Joseph, "Using sage on cots uwb signals for toa estimation and body shadowing effect quantification," in *2021 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2021, pp. 1–8.
[60] F. Nogueira, "Bayesian Optimization: Open source constrained global optimization tool for Python," 2014–. [Online]. Available: https://github.com/fmfn/BayesianOptimization
[61] "imec virtualwall," https://doc.ilabt.imec.be/ilabt/virtualwall/, (Accessed on 2022-02-04).