A Probabilistic Learning Approach to UWB Ranging Error Mitigation

Chengzhi Mao, Kangbo Lin, Tiancheng Yu, and Yuan Shen

Beijing National Research Center for Information Science and Technology
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
Email: {mcz13, linkb17, yutc14}@mails.tsinghua.edu.cn, shenyuan_ee@tsinghua.edu.cn

***Abstract*—Ultra-Wide Band (UWB) radio is capable of providing sufficient information for high accuracy localization. However, its actual performance is degraded due to the non-line-of-sight (NLOS) propagation. This paper introduces a probabilistic learning approach to mitigate the ranging error and yield uncertainties which correlate with the mitigation results. By combining variational inference with probabilistic neural networks, we propose a new probabilistic deep learning architecture, which can improve the accuracy significantly especially when the training data is limited. Results show that the proposed model can reduce the root mean square error (RMSE) of UWB ranging by 16%$\sim$56% compared with existing support vector machine approach in practical environment.**

***Index Terms*—Ultra-Wide Band (UWB) , indoor localization, ranging likelihood, deep learning, variational inference**

I. Introduction

In recent years, the high demand for accurate indoor localization inspires many methods for ranging error mitigation. Compared with WiFi or GPS, UWB [1] benefits from fine delay resolution and superior obstacle-penetration capabilities. However, UWB localization still suffers from the interference among different devices, the non-line-of-sight (NLOS) propagation, and the multi-path effect, etc. Different methods are conducted to address these challenges. For example, prior knowledge and scheduling strategies are adopted in network navigation for higher accuracy [2], and machine learning methods are explored [3-7] recently for better flexibility.

The conventional machine learning methods for UWB ranging error mitigation, such as support vector machine (SVM) [3] or Gaussian process regression (GPR) [4], relies heavily on sophisticated hand-crafted features. In [3] [5], hand-crafted features are collected to compose the low dimensional features of the channel impulse response (CIR). However, hand-crafted features have to be redesigned according to the specific input data type. For example, [4] uses differently designed features of the received signal strength of Wi-Fi, but the performance is inferior to methods using learned features [6].

Deep learning methods can overcome the above disadvantages by extracting the features automatically with an end-to-end training architecture, and have achieved the state-of-the-art performance in many applications, such as wireless sensing [7]. Nowadays, it has also been explored for localization [6]. However, in applications where the uncertainty is ubiquitous or the training data is insufficient, deep learning methods are prone to overfitting and lacking the ability to handle the uncertainty.

Another type of models called Bayesian networks (BN) can address the above problem via causal inference under uncertainty. However, it cannot handle high-dimensional data such as the CIR waveforms like the deep learning methods. By integrating BN with deep neural networks, the Bayesian deep Learning (BDL) [8] is proposed to combine their strengths [9] [10].

Furthermore, the concept of ranging likelihood [11] and soft range information (SRI) [5] are attracting more attention. Instead of predicting the ranging results as point estimations, they provide distributions with richer information for indoor localization tasks. To leverage the power of SRI, BDL is preferred due to larger model capacity and faster training.

In this paper, a BDL approach is proposed for the UWB ranging error mitigation. The model first uses deep convolutional neural network (CNN) [12] based variational auto-encoder (VAE) [9] to extract low dimensional probabilistic representations rather than scalar representations directly from high dimensional CIR signals, and then utilizes the natural-parameter networks (NPN) [10] to infer the ranging likelihood through the representation extracted. The resulting probabilistic learning approach can know the reliability of each mitigation based on the ranging likelihood, and renders the indoor ranging system more resilient to harsh environment.

II. Problem Statement

We consider the scenario of UWB ranging between one anchor (with known position) and one agent (with unknown positions). The agent measures the CIR waveforms sent by the anchor and estimates the distance. CIR waveforms contain environment specific information and have been widely used in many indoor UWB ranging error mitigation tasks [3]. As shown in Fig. 1, the landscape of CIR waveforms varies a lot in different kinds of environment due to multi-path effect and NLOS propagation. Based on the CIR waveforms, we estimate distance and the corresponding confidence interval by producing a Gaussian ranging likelihood. In the following, we denote the real distance of the measurement by d, the mean and variance of the ranging likelihood by $\hat{d}$ and $\hat{s}$ respectively, and the ranging error by $\epsilon_i = \hat{d} - d$. We measure the results

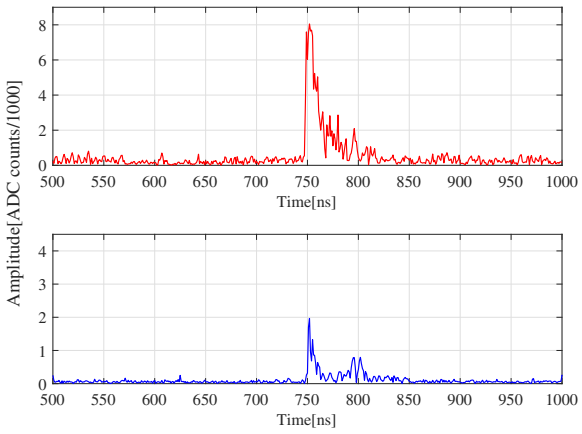

Fig. 1: CIR waveforms examples. The first waveform with sharper peak is collected under the LOS scenario, the second waveform with flatter peak is collected under NLOS scenario.

with root mean square error (RMSE), where for M samples, the average accuracy is defined by

$$L = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(\hat{d}_i - d_i)^2} \tag{1}$$

Furthermore, our explicit uncertainty prediction $\hat{s}_i$ leads to a new evaluation metric WRMSE (weighted RMSE), where each term of the RMSE is weighted by a normalized certainty describing term $\frac{1}{\hat{s}_i s_m}$:

$$s_m = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{\hat{s}_i} \tag{2}$$

$$L_w = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\frac{(\hat{d}_i - d_i)^2}{\hat{s}_i}\frac{1}{s_m}} \tag{3}$$

III. Approach

In this section, we elaborate the ranging error mitigation pipelines, including the deep learning and variational inference (VI) for representation learning, and the Bayesian network for soft range information estimation.

A. Deep Neural Networks

Two kinds of network architecture, Multilayer perceptron (MLP) and CNN, are adopted as building blocks in our models. MLP consists of a sequence of fully connected layers and transformation layers. We denote the input by a vector $\boldsymbol{X}_0$, the weights of the l-th fully connected layer by $\boldsymbol{W}_l$, the bias of the l-th layer by $\boldsymbol{b}_l$, the output of the l-th layer by $\boldsymbol{X}_l$ and the target by $\boldsymbol{Y}$. Learning an L-layer MLP for regression is equivalent to the following optimization problem:

$$\min_{\boldsymbol{W}_l,\boldsymbol{b}_l} ||\boldsymbol{X}_L - \boldsymbol{Y}||^2 + \lambda\sum_l ||\boldsymbol{W}_l||^2 \tag{4}$$

subject to

$$\boldsymbol{X}_l = f(\boldsymbol{X}_{l-1}\boldsymbol{W}_l + \boldsymbol{b}_l), \quad l = 1, ..., L-1 \tag{5}$$

$$\boldsymbol{X}_L = \boldsymbol{X}_{L-1}\boldsymbol{W}_L + \boldsymbol{b}_L \tag{6}$$

where λ controls the proportion of the regularization term, $f(\cdot)$ is a nonlinear transformation, such as sigmoid, $\tanh(x)$, and $\max(0, x)$.

Compared with MLP, the number of parameters in CNN is reduced significantly by the local receptive field and weight sharing to avoid over-fitting. In conventional CNN, the convolution operation is followed by an non-linear transformation $f(\cdot)$:

$$\boldsymbol{a}_{l+1} = f(\boldsymbol{W}_l \circledast \boldsymbol{a}_l + \boldsymbol{b}_l) \tag{7}$$

where $\boldsymbol{a}_l$ is the feature in the l-th layer, $\boldsymbol{W}_l$ is the weight of convolution operation in the l-th layer, $\boldsymbol{b}_l$ is the bias, and $\circledast$ indicates the convolution.

B. Variational Inference and Learning

VI is an efficient tool to approximate the posterior probability, whose potential can be fully excavated if combined with deep neural networks. Here, we use VAE, which is a VI based auto-encoder, to learn the hidden probabilistic representation of the waveforms and enhance our CNN feature extractor.

VAE consists of the inference network and the generative model where the conditional probability is calculated by parameterized neural networks. Let $\boldsymbol{x}$ be a set of variables in our observed dataset $\boldsymbol{X}$, generated by an unobserved continuous random variable $\mathbf{z}$. We assume the prior distribution of the unobserved variable is $p(\mathbf{z})$ and let $p(\mathbf{x}|\mathbf{z})$ be the generative model over the latent variables $\mathbf{z}$.

Upon the given dataset, we perform maximum marginal likelihood learning of the parameters by maximizing

$$\log p(\boldsymbol{X}) = \sum_{i=1}^{N}\log p(\boldsymbol{x}^{(i)}) \tag{8}$$

each term in the sum up can be rewritten as:

$$\log p(\boldsymbol{x}^{(i)}) = D_{KL}(q(\boldsymbol{z}|\boldsymbol{x}^{(i)})||p(\boldsymbol{z}|\boldsymbol{x}^{(i)})) + \mathcal{L}(\boldsymbol{x}^{(i)};\boldsymbol{\theta}) \tag{9}$$

where $\boldsymbol{\theta}$ is the parameter for the parameterized conditional distribution model p and q.

Note that the Kullback-Leibler divergences $D_{KL}(.)$ are non-negative. By introducing the parametric inference model $p(\mathbf{z}|\mathbf{x}^{(i)})$ over the latent variables $\mathbf{z}$, the *variational lower bound* on the marginal log-likelihood of each data $\boldsymbol{x}^{(i)}$ is $\mathcal{L}(\boldsymbol{x}^{(i)};\boldsymbol{\theta})$. For simplicity, we abbreviate $\boldsymbol{x}^{(i)}$ by $\boldsymbol{x}$, here we get:

$$\log p(\boldsymbol{x}) \geq \mathcal{L}(\boldsymbol{x};\boldsymbol{\theta}) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{x},\boldsymbol{z}) - \log q(\boldsymbol{z}|\boldsymbol{x})] \tag{10}$$

which is equivalent to

$$\mathcal{L}(\boldsymbol{x};\boldsymbol{\theta}) = -D_{KL}(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})) + \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log q(\boldsymbol{x}|\boldsymbol{z})] \quad (11)$$

We use the Stochastic Gradient Variational Bayes estimator to approximate the variational lower bound:

$$\widetilde{\mathcal{L}}(\boldsymbol{x};\boldsymbol{\theta}) = -D_{KL}(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})) + \frac{1}{L}\sum_{l=1}^{N}\log p(\boldsymbol{x}|z^{(l)}) \quad (12)$$

We would refer the $q(\mathbf{z}|\mathbf{x})$ as the probabilistic encoder, since by feeding a data point $\boldsymbol{x}$, it produces a distribution $\boldsymbol{z}$ (e.g. a Gaussian) to generate the $\boldsymbol{x}$. Likewise, we would refer $p(\mathbf{x}|\mathbf{z})$ as a probabilistic decoder, since by feeding a code $\boldsymbol{z}$ it produces a distribution of $\boldsymbol{x}$.

The VAE works as follows: The probabilistic encoder takes the waveforms $\mathbf{x}$ as input, and produces the Gaussian latent representation $\mathbf{z}$ by outputting the mean and variance value. The KL-divergence between the produced distribution $\mathbf{z}$ and the prior distribution is calculated and minimized during the training, which can be treated as a regularization term for the learned representation. The probabilistic decoder takes one sample from the latent representation $\mathbf{z}$ and reconstructs the input waveforms $\mathbf{x}$ by minimizing the marginal likelihood. CNN is utilized as the parameterized encoder and decoder here.

C. Bayesian Neural Networks

MLP is incapable of modeling distributions directly, as mentioned in the previous section. To handle this shortcoming, we use NPN, where the input distributions are processed through several linear and non-linear layers before matching the output distributions. By modeling the input, target output, weights, and neurons as distributions from the exponential-family, the sampling-free backpropagation algorithms could be implemented to learn the natural parameters of distributions efficiently.

Specifically, we write a traditional neural network in the form $f_{\boldsymbol{W}}(\boldsymbol{x})$, where $\boldsymbol{W}$ is the deterministic parameter and $\boldsymbol{x}$ is the deterministic input. Correspondingly, a NPN with Gaussian prior gets its $\mathbf{W}$ from a Gaussian distribution $p_{\boldsymbol{\theta}}(\mathbf{W})$ parameterized by $\boldsymbol{\theta}$, and gets its input $\mathbf{x}$ from another Gaussian distribution $N(\boldsymbol{x}_{\mathrm{m}}, \boldsymbol{x}_{\mathrm{s}})$.

Assume the prior distribution is Gaussian with two natural parameters. Let $\boldsymbol{a}_{\mathrm{m}}$, $\boldsymbol{a}_{\mathrm{s}}$, $\boldsymbol{o}_{\mathrm{m}}$, $\boldsymbol{o}_{\mathrm{s}}$ be the mean and variance of the input and output distribution respectively. Let $\boldsymbol{W}_{\mathrm{m}}$, $\boldsymbol{W}_{\mathrm{s}}$, $\boldsymbol{b}_{\mathrm{m}}$, $\boldsymbol{b}_{\mathrm{s}}$ be the mean and variance of parameters $\mathbf{W}$ and $\mathbf{b}$ respectively. The superscript indicates layer index which the variable belongs to, and $\circ$ denotes the element-wise product. Then the linear transformation in NPN have the following formulation:

$$\boldsymbol{o}_{\mathrm{m}}^{(l)} = \boldsymbol{a}_{\mathrm{m}}^{(l-1)}\boldsymbol{W}_{\mathrm{m}}^{(l)} + \boldsymbol{b}_{\mathrm{m}}^{(l)} \quad (13)$$

$$\begin{aligned}\boldsymbol{o}_{\mathrm{s}}^{(l)} = {} & \boldsymbol{a}_{\mathrm{s}}^{(l-1)}\boldsymbol{W}_{\mathrm{s}}^{(l)} + \boldsymbol{a}_{\mathrm{s}}^{(l-1)}(\boldsymbol{W}_{\mathrm{m}}^{(l)} \circ \boldsymbol{W}_{\mathrm{m}}^{(l)}) \\ & + (\boldsymbol{a}_{\mathrm{m}}^{(l-1)} \circ \boldsymbol{a}_{\mathrm{m}}^{(l-1)})\boldsymbol{W}_{\mathrm{s}}^{(l)} + \boldsymbol{b}_{\mathrm{s}}^{(l)}\end{aligned} \quad (14)$$

As for the non-linear transformation, since the exponential-family distribution will no longer be an exponential-family distribution after such transformation, NPN approximate it with an exponential-family distribution by matching the first two moments. The feed-forward calculation is:

$$\boldsymbol{a}_{\mathrm{m}} = \int p_{\boldsymbol{o}}(\boldsymbol{o}|\boldsymbol{o}_{\mathrm{m}}, \boldsymbol{o}_{\mathrm{s}})v(\boldsymbol{o})d\boldsymbol{o} \quad (15)$$

$$\boldsymbol{a}_{\mathrm{s}} = \int p_{\boldsymbol{o}}(\boldsymbol{o}|\boldsymbol{o}_{\mathrm{m}}, \boldsymbol{o}_{\mathrm{s}})v(\boldsymbol{o})^2d\boldsymbol{o} - \boldsymbol{a}_{\mathrm{m}}^2 \quad (16)$$

where v is the non-linear transformation function.

NPN outputs an exponential-family distribution (such as Gaussian distribution in our case). We denote the output distribution of Gaussian NPN with mean $\boldsymbol{o}_{\mathrm{m}}$ and variance $\boldsymbol{o}_{\mathrm{s}}$, the target distribution with mean $\boldsymbol{y}_{\mathrm{m}}$ and variance $\boldsymbol{\epsilon}$. We define the loss function as the *KL divergence* $KL(\mathcal{N}(\boldsymbol{o}_{\mathrm{m}}, diag(\boldsymbol{o}_{\mathrm{s}}))||\mathcal{N}(\boldsymbol{y}_{\mathrm{m}}, diag(\boldsymbol{\epsilon})))$ between the output distribution and target distribution. The loss function $\mathcal{L}_n$ can be calculated as follows:

$$\begin{aligned}\mathcal{L}_n = {} & \frac{1}{2}\sum_i \Big(\frac{\boldsymbol{\epsilon}_i}{[\boldsymbol{o}_{\mathrm{s}}]_i} + \frac{([\boldsymbol{o}_{\mathrm{m}}]_i - [\boldsymbol{y}_{\mathrm{m}}]_i)^2}{[\boldsymbol{o}_{\mathrm{s}}]_i} \\ & + \log[\boldsymbol{o}_{\mathrm{s}}]_i - K\log\boldsymbol{\epsilon}_i - K\Big)\end{aligned} \quad (17)$$

where K and $\boldsymbol{\epsilon}$ are constant and subscript i indicates the i-th element of the vector.

D. Combined Model

The combined model first uses CNN based VAE model to extract probabilistic representation of CIR waveforms, then uses Bayesian inference via NPN to estimate the ranging likelihood. The whole model is trained by jointly optimizing the following loss function:

$$\mathcal{L} = \mathcal{L}_n + \alpha\widetilde{\mathcal{L}}(\boldsymbol{x};\boldsymbol{\theta}) \quad (18)$$

where α define the weight for the VAE loss function term during training.

IV. Measurement Activities

A. Measurement Apparatus

We conduct the experiments with the Decawave1000 UWB radios in an indoor office environment. The central frequency and bandwidth of our UWB radio are 3993.6MHz and 499.2MHz. Each radio is installed on top of a tripod at a height of 1.3m. The antenna is set upright to the ground. The UWB devices calculate the distance based on the two-way ranging method and are capable of capturing the CIR waveforms while conducting the ranging procedure. Each CIR waveform is sampled at $1ns$ intervals with the window length of 1016 sample points. The resolution is $30cm$ for CIR signal and $1cm$ for the ranging distance.

B. Measurement Arrangement

Measurements are conducted under both LOS and NLOS conditions. The NLOS data is collected with obstacles, such as the door, closet, and walls, between the transmitting and receiving antenna. The LOS data is collected with no object on the straight line between the transmitting and receiving

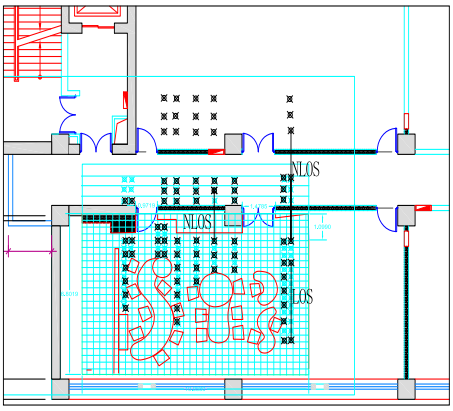

Fig. 2: Map for the measurement environment. The black crosses indicated our measurement points. The black lines are examples of the measurement path. Both LOS and NLOS data are collected here.

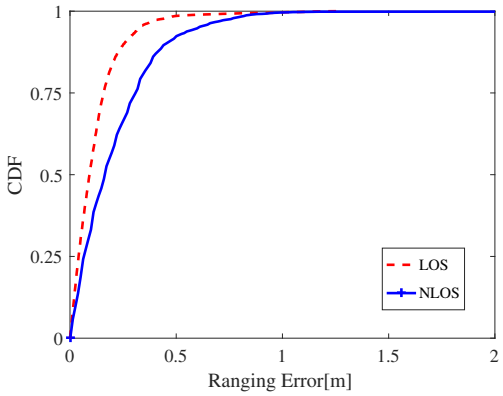

Fig. 3: The CDF for UWB ranging error under LOS (the dotted line) and NLOS (the solid line) scenario.

antenna. A computer-aided design (CAD) map is depicted for our experiment environment to aid our real distance calculation, as shown in Fig. 2. Overall, 700 unique point-to-point measurements were performed. For each pair of points, around 100 distance estimates and received CIR waveforms were recorded. In contrast to limiting the movement of other objects nearby in paper [3], we change the layout of the surroundings during the measurement, just as in real scenarios.

C. Database

Together with the aforementioned measurement apparatus and arrangements, a database was created for validating our approach. The database consists of 700 pairs of links in total, with 456 NLOS measurements and 244 LOS measurements. Each data point in the database consists of three components: the CIR waveform x, the estimated distance $\hat{d}$ obtained by radio-embedded algorithms and the true distance d calculated by CAD software. Only the module of the complex CIR signals are saved. The cumulative distribution function (CDF) of the UWB ranging error of the dataset is shown in Fig.3. During the following experiments, we randomly split the database into the training set and testing set, where waveforms from the same pair of links are guaranteed not to appear in the training and testing set simultaneously.

V. Experimental Results and Performance Evaluation

In this section, we illustrate the implementation details about the models, evaluate our proposed method on our dataset, and validate the robustness of the proposed method under very few training data.

A. Learning Model Setup

As the baseline approach, we adopt the 6 hand-crafted features designed in [3] [5] and feed the features into the **SVM**, **MLP** and **NPN** respectively. We use radial basis function as the kernel for our SVM. For MLP and NPN, we both use 3-layer architecture with 512 hidden neurons. We adopt the function $f(x) = \max(x, 0)$ as our non-linear activation function. We train both MLP and NPN models using stochastic gradient descent (SGD) with the learning rate $lr = 0.01$ and the batch size $b = 128$.

In comparison with the baseline models, we use the deep learning approach to process the raw CIR waveforms directly. Our CNN approach uses 4 ResNeXt [12] blocks which consists of 12 convolution layers since the performance saturates at this model capacity. Two layers of MLP or NPN is applied on the top to construct the **CNN** and **CNPN** (CNN+NPN) respectively. We also combine the VAE and NPN to construct the joint model and refer it as **VNPN** (VAE+NPN). All the models are trained end-to-end by minimizing the loss function using SGD, where $lr = 0.001$ and $b = 128$.

We divide our data into two sets. The training set possesses 20% of the data and the testing set collects 80% of the data. In addition, we select 50% of the training set data as the validation set for our model selection. Thus only 10% of the data is actually trained in our machine learning algorithm. We build the model in Pytorch [13]. We run the model on GTX 1080 GPU with a memory of 12 GB and the accelerator is powered by the NVIDIA Pascal architecture.

B. Experimental Results Analysis

We start by testing the performance of SVM, MLP, and NPN methods on the hand-crafted features. As the RMSE shown in Table 1, NPN outperforms the existing SVM method. In addition, NPN outperforms the MLP constantly under the same model configuration, illustrating that the probabilistic model is better than vanilla neural networks. We further apply vanilla deep learning method of CNN on the raw CIR signal. As we can see, CNN outperforms the MLP steadily, thus the learned CNN feature is superior to the hand-crafted features. In addition, by replacing the MLP with NPN model, the accuracy of resulting model CNPN is higher. During the experiments, we observe the performance of CNN is very sensitive to the model capacity and is prone to overfitting the training set. On the other hand, the VNPN method suffers less from this issue.

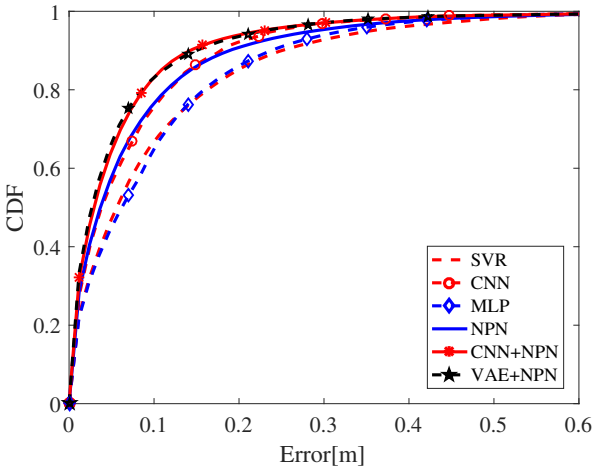

Fig. 4: The CDF of the ranging error on testing data with different trained models.

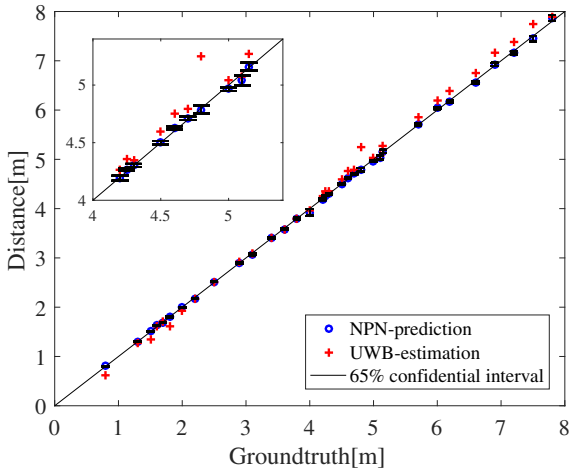

Fig. 5: The 65% confidence interval of LOS ranging prediction on testing data shown by the error bar. The subfigure on the left is the zoomed in of the details.

It demonstrates the best result which reduces the RMSE by **16%** compared with the SVM method.

In addition, we evaluate the performance of NPN, CNPN, VNPN methods with the WRMSE metric, where the certainty of the predictions can be explicitly given. From Table 1, we can see the WRMSE is much lower than the RMSE loss. Further, the VNPN method achieves the least WRMSE loss, and outperforms the SVM method by **36.9%**.

Fig. 4 shows the cumulative distribution function (CDF) of prediction error for the models on the whole dataset. Different from methods of first identifying LOS and NLOS condition then regressing the error separately [3], our method can handle data from different scenarios indiscriminately.

Model	Data Type		
	LOS	NLOS	ALL DATA
SVM	0.213	0.302	0.214
MLP	0.206	0.295	0.226
NPN	0.135 *(0.092)*	0.286 *(0.264)*	0.200 *(0.154)*
CNN	0.141	0.272	0.194
CNPN	0.121 *(0.090)*	0.264 *(0.243)*	0.182 *(0.137)*
VNPN	**0.117** ***(0.088)***	**0.261** ***(0.241)***	**0.179** ***(0.135)***

TABLE I: The RMSE (m) and *WRMSE (m)* of different models testing on 80% of the dataset. The numbers on the left denote the value of RMSE, while the *italic* numbers in the bracket on the right denote the value of WRMSE.

C. Predicting the Ranging Likelihood

As mentioned above, the probabilistic model predicts a distribution of the ranging results. As Fig. 5 shows, the line $d = \hat{d}$, where the prediction equals to the real distance, denotes the ground truth prediction results. We use two lines to demonstrate the 65% confidential interval of the estimation. Fig. 5 shows the ranging likelihood prediction of LOS scenario

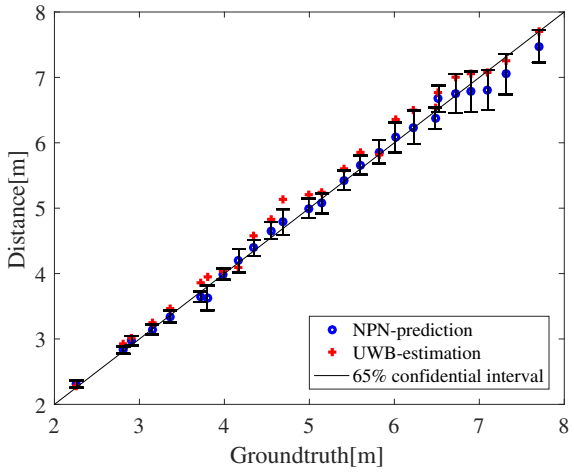

Fig. 6: The 65% confidence interval of NLOS ranging prediction on testing data shown by the error bar.

and Fig. 7 shows that of NLOS scenario. As we can see, the predictions close to the ground-truth line possess high certainty for the prediction. On the contrary, the predictions deviated far from the ground truth line tends to possess large uncertainty for the prediction.

D. Resilience to Insufficient Training Data

In this section, we investigate the sensitivity of our proposed method to limited sample size. We validate our approach by randomly selecting 1%, 2%, 3%, 5%, and 10% pairs of links of the NLOS as training samples and testing the trained model on all the other samples. We exclude the LOS data here to demonstrate the model's performance under harsh NLOS scenarios. Half of the training samples is split to the validation set. The RMSE loss is summarized in Table 2.

As we can see, the performance of the SVM is highly dependent on the number of training data available. The VNPN

Training size	SVM	CNPN	VNPN
1%	0.984	1.43	**0.424**
2%	0.637	0.331	**0.280**
3%	0.388	0.283	**0.266**
5%	0.361	0.305	**0.258**
10%	0.355	0.316	**0.240**

TABLE II: RMSE (m) of ranging estimation on the testing data for different models after training with different size of NLOS data.

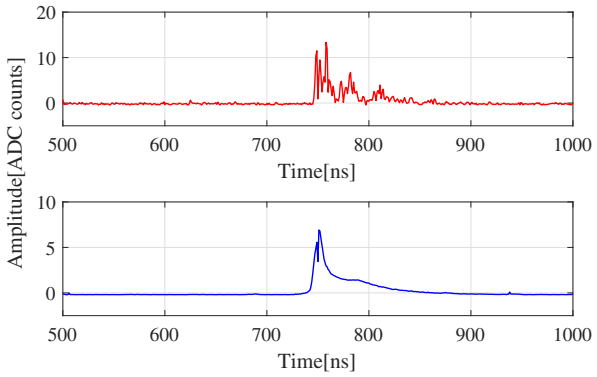

Fig. 7: One input CIR waveform and the reconstruction of the input CIR waveform using VAE.

model can retain the ranging accuracy better than CNPN under insufficient training data. With 1% of training samples, VNPN reduces the RMSE by **56.9%** compared with the SVM method.

We also visualize the reconstruction results of our VAE model to illustrate the role of the encoder-decoder architecture. As we can see in Fig. 7, the peak and the shape of the waveform is captured by the VAE while all the oscillations, which might be noise, are neglected by the model.

E. Real-time Analysis

The running time is tested for one epoch of training and testing of all the models. As summarized in Table 3, the proposed VNPN method, though takes the longest time, is able to conduct more than 4300 estimations in one second, only costs less than 2.5 times of the time of existing SVM method. The robustness gained by the model is worth the extra processing time.

Time/Sample (ms)	Training	Testing
SVM	0.430	0.097
MLP	0.043	0.005
NPN	0.091	0.027
CNN	0.209	0.018
CNPN	0.252	0.021
VNPN	2.55	0.230

TABLE III: The running time for training and testing one sample.

VI. Conclusion and Future Work

In this paper, we introduce a probabilistic learning approach that mitigates the ranging error of UWB radio and achieves 16%∼56% reduction in RMSE over existing SVM methods. We demonstrate that deep CNN can be used to learn the representation of the CIR waveforms and the ranging likelihood can be obtained by a probabilistic neural network through causal inference. We further boost CNN via the VAE architecture and retain the performance under insufficient training data. For future work, the potential of unlabeled data could be excavated by semi-supervised learning method under the VAE architecture. It would be also interesting to apply the ranging likelihood produced by our method to 2D and 3D indoor localization and actively collect data under the guidance of the certainty of the ranging estimation.

Acknowledgment

This research was supported by the National Natural Science Foundation of China under Grant 61501279 and 91638204. We would also thank Y. Liu, Y. Wang, and K. Gu for their valuable suggestions.

References

[1] M. Z. Win, Y. Shen, and W. Dai, "A theoretical foundation of network localization and navigation," *Proc. IEEE*, vol. 106, no. 7, pp. 1136–1165, Jul 2018.

[2] T. Wang, Y. Shen, A. Conti, and M. Z. Win, "Network navigation with scheduling: Error evolution," *IEEE Trans. Inf. Theory*, vol. 63, no. 11, pp. 7509–7534, Nov 2017.

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

[4] Z. Xiao, H. Wen, A. Markham, N. Trigoni, P. Blunsom, and J. Frolik, "Identification and mitigation of non-line-of-sight conditions using received signal strength," in *IEEE Trans. Wireless Commun.*, Lyon, France, Oct. 2013, pp. 667–674.

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

[6] J. S. Choi, W. H. Lee, J. H. Lee, J. H. Lee, and S. C. Kim, "Deep learning based NLOS identification with commodity WLAN devices," *IEEE Trans. Veh. Technol.*, vol. 67, no. 4, pp. 3295–3303, Apr 2017.

[7] M. Zhao, S. Yue, D. Katabi, T. S. Jaakkola, and M. T. Bianchi, "Learning sleep stages from radio signals: A conditional adversarial architecture," in *ICML*, vol. 70, Sydney, Australia, Aug. 2017, pp. 4100–4109.

[8] H. Wang and D. Yeung, "Towards bayesian deep learning: a survey," *ArXiv*, Apr. 2016.

[9] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *ArXiv*, Dec. 2013.

[10] H. Wang, X. Shi, and D.-Y. Yeung, "Natural-parameter networks: a class of probabilistic neural networks," in *NIPS*, Barcelona, Spain, Dec. 2016, pp. 118–126.

[11] H. Lu, S. Mazuelas, and M. Win, "Ranging likelihood for wideband wireless localization," in *Proc. IEEE Int. Conf. Commun.*, Budapest, Hungary, Jun. 2013, pp. 5804–5808.

[12] S. Xie, R. Girshick, P. Dollr, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *CVPR*, Hawaii, United States, Jul. 2017, pp. 5987–5995.

[13] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.