A Semi-Supervised Learning Approach for Ranging Error Mitigation Based on UWB Waveform

Yuxiao Li*, Santiago Mazuelas†, and Yuan Shen*
* Department of Electronic Engineering, Tsinghua University, Beijing, China
† BCAM-Basque Center for Applied Mathematics, and IKERBASQUE-Basque Foundation for Science, Bilbao, Spain
Email: li-yx18@mails.tsinghua.edu.cn, smazuelas@bcamath.org, shenyuan_ee@tsinghua.edu.cn

***Abstract*—Localization systems based on ultra-wide band (UWB) measurements can have unsatisfactory performance in harsh environments due to the presence of non-line-of-sight (NLOS) errors. Learning-based methods for error mitigation have shown great performance improvement via directly exploiting the wideband waveform instead of handcrafted features. However, these methods require data samples fully labeled with actual measurement errors for training, which leads to time-consuming data collection. In this paper, we propose a semi-supervised learning method based on variational Bayes for UWB ranging error mitigation. Combining deep learning techniques and statistic tools, our method can efficiently accumulate knowledge from both labeled and unlabeled data samples. Extensive experiments illustrate the effectiveness of the proposed method under different supervision rates, and the superiority compared to other fully supervised methods even at a low supervision rate.**

***Index Terms*—Variational Bayes, Deep Learning, Semi-Supervised Learning, UWB Radio, ranging error mitigation**

I. Introduction

Wireless network localization is a key enabler for a wide range of emerging applications with the requirement of high-accuracy positional information [1]. Many papers develop localization algorithms based on range-related measurements. Among the measuring approaches for these algorithms, ultra-wideband (UWB) radio emerges to be a dominant trend due to the fine delay resolution and obstacle-penetration [2]. However, its practical performance for range measurements is greatly degraded in harsh environments due to multipath effects [3], and non-line-of-sight (NLOS) conditions [4].

Multiple ranging error mitigation techniques are proposed to improve the distance estimates based on measurements, and subsequently the localization accuracy. Conventional methods are mostly model-based, with a simplified model for signal propagation mechanism [5]. Such modeling leads to a transparent interpretation and generality in different environments [6]. However, their effectiveness is largely limited by oversimplified assumptions and relatively low computational speed of greedy optimization algorithms. Early learning-based methods, such as Support Vector machine (SVM) [7] or Gaussian process regression (GPR) [8], formulate mitigation as a regression problem from statistical characteristics of measurements to ranging errors. These characteristics, however, require a time-consuming extraction phase and may still lose information inherent in the raw waveform [9], [10].

Deep learning methods represent to be a popular trend and have also been applied to wireless communications [11]. Benefiting from a more thorough exploitation of raw data, these methods show great improvements in effectiveness and efficiency compared to conventional methods [12]–[14]. However, these methods require a large amount of fully labeled data to achieve satisfactory results, leading to not-efficient work for data collection.

In this paper, we propose a semi-supervised method based on variational Bayes (VB) for the UWB ranging error mitigation problem, referred to as Semi-VL. Specifically, the generation process of received waveform is assumed to consist of two latent variables: one for range-related features and the other for unrelated environment semantics. The loss function, derived from variational inference, is composed of a supervised term and an unsupervised term. As a result, the proposed method can efficiently exploit information from both labeled and unlabeled data. Extensive experiments illustrate that the proposed Semi-VL achieves efficient error estimation under a range of supervision rates, and provides significant performance improvement compared to conventional fully supervised methods even at a low supervised rate.

The rest of the paper is organized as follows. Section II introduces the generative model for received signal and the variational Bayesian method to extract features for ranging error mitigation. In section III, the semi-supervised learning algorithm is presented, consisting the formulation of supervised and unsupervised loss terms. Section IV presents the UWB dataset for evaluation and implementation of the proposed algorithm. Section V presents the experimental results under different supervision rates and detailed analyses. Finally, the conclusion and our future work is discussed in section VI.

II. Variational Bayesian Method

In this section, we propose a variational Bayesian method to extract specific features from raw received signals for the ranging error mitigation problem. Specifically, we present a generative model for received signal, including a range-related variable estimated by variational Bayes.

A. Generative Model

The generative process for received signal is described by a directed probabilistic model in the presence of two unobserved random variables, as illustrated in Fig.1(a). Specifically, signal data $\mathbf{x}$ is assumed to involve two independent latent variables: $\mathbf{y}$ for range-related feature, and $\mathbf{z}$ for range-unrelated features

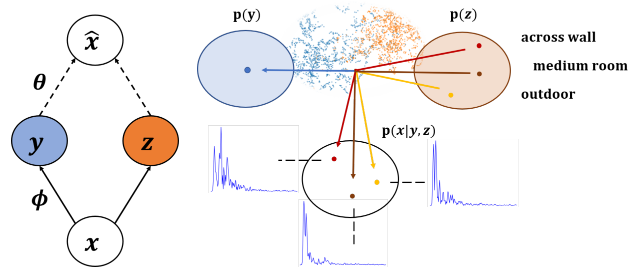


Fig. 1. Illustrations of the proposed Bayesian Model. (a) The graphical model on measured data inference. Dashed lines denote the generative process parameterized by $\boldsymbol{\theta}$, while solid lines denote the embedding from data space to latent space parameterized by $\boldsymbol{\phi}$. (b) waveform samples in data space are generated from according range-related and environment-related features in latent space.

caused by environment variance. The generative process of a signal sample $\mathbf{x}^{(i)}$ can be obtained by first sampling $\mathrm{y}^{(i)}$, $\mathrm{z}^{(i)}$ from $p(\mathbf{y}, \mathbf{z})$, and then generating via the conditional distribution $p(\mathbf{x}|\mathbf{y}^{(i)}, \mathbf{z}^{(i)})$. As illustrated in Fig.1(b), different values environmental variables lead to different output signals.

With the disentanglement of semantics features in the latent space, many measurement-related problems can be solved by mixing the latent variable to incorporate desired semantics, and generate desired results. In particular, the estimation of ranging error Δd can thus be modeled as the estimation of the conditional distribution $p(\Delta d|\mathbf{y})$.

For such downstream tasks, the disentanglement is required to be achieved. Since the conditional distribution $p(\mathbf{y}, \mathbf{z}|\mathbf{x})$ as well as values of latent variables $\mathbf{y}$ and $\mathbf{z}$ are unknown, we introduce a variational distribution $q(\mathbf{y}, \mathbf{z}|\mathbf{x})$ for the intractable true posterior $p(\mathbf{y}, \mathbf{z}|\mathbf{x})$. Inspired by the strategy introduced in [15], we perform maximum likelihood (ML) inference on the observed data and these variables.

B. Variational Lower bound for Data Likelihood

In order to infer two latent variables from data samples, we derive a lower bound over the log-likelihood of data distribution. The variational lower bound of the likelihood $p(\mathbf{x})$ of each sample with respect to the two latent variables $\mathbf{y}$, $\mathbf{z}$ is stated in the following proposition.

Proposition 1. Suppose data $\mathbf{x}$ are generated by a random process, involving two independent latent variables $\mathbf{y}$ and $\mathbf{z}$. The evidence lower bound (ELBO) $\mathcal{L}$ of the marginal likelihood of data $\mathbf{x}$ can be written as,

$$\begin{aligned}\log p(\mathbf{x}) \geq &\mathcal{L}(q; \mathbf{x}) \\ = &\mathbb{E}_{q(\mathbf{y},\mathbf{z}|\mathbf{x})}\left[\log p(\mathbf{x}|\mathbf{y}, \mathbf{z})\right] \\ &- \mathrm{D}_{\mathrm{KL}}\left(q(\mathbf{y}|\mathbf{x})\big|\big|p(\mathbf{y})\right) - \mathrm{D}_{\mathrm{KL}}\left(q(\mathbf{z}|\mathbf{x})\big|\big|p(\mathbf{z})\right)\end{aligned} \tag{1}$$

where where D_{KL} is the Kullback-Leibler divergence, $q(\mathbf{y}|\mathbf{x})$ and $q(\mathbf{z}|\mathbf{x})$ are variational distributions introduced to approximate the true posterior distributions. The inequality is obtained from the Jensen's inequality, achieving equality iff $q(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}|\mathbf{x})$ and $q(\mathbf{z}|\mathbf{x}) = p(\mathbf{z}|\mathbf{x})$.

Such bound can then be used to find a suitable approximated distribution q^* from some distribution family Q that matches the true distribution p in the generative model, i.e.,

$$q^* = \arg\max_{q \in Q} \mathcal{L}(q; \mathbf{x}) \tag{2}$$

Such optimization problem is hard to use directly, due to the complicated nature of the approximated distributions. Instead of assuming them from common distribution families, we construct deep neural networks to accumulate knowledge from data and learn these variational distributions.

III. Semi-Supervised Learning Scheme

In this section, we propose a deep neural network for semi-supervised ranging error mitigation. We first construct parametric forms of the variational distributions for latent variables, which enables efficient learning to solve the optimization in equation (2). Then we estimate the ranging error conditioned on the disentangled range-related feature.

A. Parametric Form of the Bound

The ELBO in (1) is transferred into a parametric form to train a specific variational auto-encoder (VAE), which outputs two probabilistic codes for latent variables in the bottleneck. Suppose the likelihood distribution for $\mathbf{x}$ comes from a parametric family of $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}, \mathbf{z})$ learned by the decoder network $\mathrm{g}_{\mathrm{dec}}(\cdot; \boldsymbol{\theta})$ with parameter $\boldsymbol{\theta}$. The variational posterior distribution for latent variables comes from a parametric family of $q_{\boldsymbol{\phi}}(\mathbf{y}, \mathbf{z}|\mathbf{x})$ learned by the encoder network $\mathrm{g}_{\mathrm{enc}}(\cdot; \boldsymbol{\phi})$ with parameter $\boldsymbol{\phi}$. Due to the independence between two latent variables, the encoder network can further be decomposed to two sub modules for individual posterior distributions for $\boldsymbol{y}$ and $\boldsymbol{z}$, denoted as $\boldsymbol{\phi} = \{\boldsymbol{\phi}_y, \boldsymbol{\phi}_z\}$.

According to the properties of latent variables, we assume the prior distributions as Gaussian to illustrate randomness:

$$\begin{aligned}p(\mathbf{y}) &= \mathcal{N}(\mathbf{y}; \mathbf{0}, \epsilon_y \boldsymbol{I}), \\ p(\mathbf{z}) &= \mathcal{N}(\mathbf{z}; \mathbf{0}, \epsilon_z \boldsymbol{I})\end{aligned} \tag{3}$$

where ϵ_y and ϵ_z are small values arbitrarily given in practice.

The likelihood distribution for $\mathbf{x}$ is learned by the decoder networks $g(\cdot; \boldsymbol{\theta})$ and composes the empirical distribution of $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}, \mathbf{z})$, i.e.,

$$\mathbf{x} = g(\mathbf{y}, \mathbf{z}; \boldsymbol{\theta}) \sim p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}, \mathbf{z}) \tag{4}$$

The approximated posterior distributions are also constructed as Gaussian distributions with mean and variance learned by encoder modules, i.e.,

$$\begin{aligned}q_{\boldsymbol{\phi}_y}(\mathbf{y}|\mathbf{x}) &= \mathcal{N}(\mathbf{y}; \hat{\boldsymbol{\mu}}_y, \hat{\boldsymbol{\sigma}}_y^2 \boldsymbol{I}) \\ q_{\boldsymbol{\phi}_z}(\mathbf{z}|\mathbf{x}) &= \mathcal{N}(\mathbf{z}; \hat{\boldsymbol{\mu}}_z, \hat{\boldsymbol{\sigma}}_z^2 \boldsymbol{I})\end{aligned} \tag{5}$$

where $\hat{\boldsymbol{\mu}}_y = \hat{\boldsymbol{\mu}}(\mathbf{x}; \boldsymbol{\phi}_y), \hat{\boldsymbol{\sigma}}_y^2 = \hat{\boldsymbol{\sigma}}^2(\mathbf{x}; \boldsymbol{\phi}_y)$ denote prediction functions for distribution parameters of ranging-related variable $\mathbf{y}$ learned by neural networks with parameter $\boldsymbol{\phi}$, similar for $\hat{\boldsymbol{\mu}}_z$ and $\hat{\boldsymbol{\sigma}}_z^2$ of environment-related variable $\mathbf{z}$. The prior and posterior distributions assumed here are simple and only

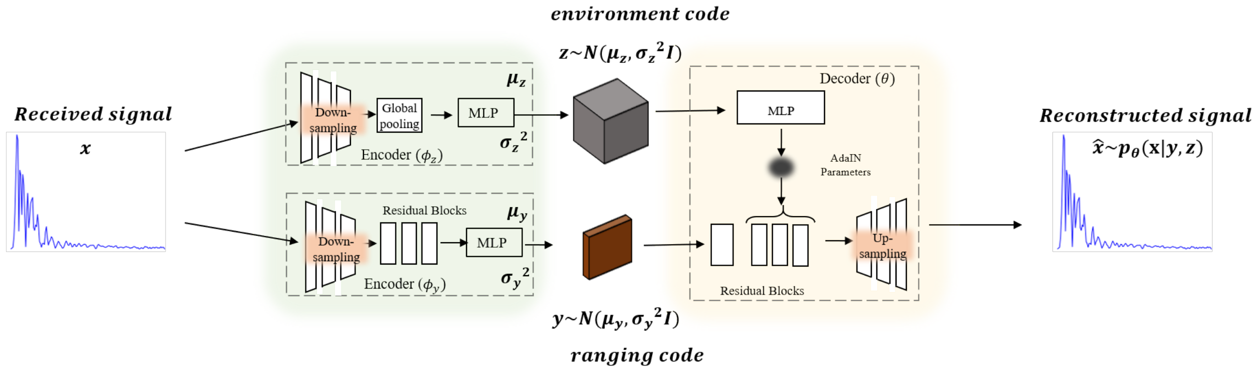


Fig. 2. Our auto-encoder architecture for latent variable disentanglement. The encoder for range-related feature consists of several strided convolutional layers followed by residual blocks. The encoder for range-unrelated features constains several strided layers followed by a global average pooling layer and a fully connected layer. The decoder uses a MLP to produce a set of AdaIN [16] parameters from environment-related features.

to illustrate randomness, and can be further assumed if further information for distribution forms of $\mathbf{y}$ and $\mathbf{z}$ are known.

With the aforementioned expression, we derive the parametric version of ELBO w.r.t. signal data $\mathbf{x}$ and parameters $\{\boldsymbol{\phi}_y, \boldsymbol{\phi}_z, \boldsymbol{\theta}\}$ as follows,

$$\begin{aligned}\mathcal{L}(\mathbf{x};\boldsymbol{\phi},\boldsymbol{\theta}) &= \mathbb{E}_{q_{\boldsymbol{\phi}_y(\mathbf{y}|\mathbf{x})}q_{\boldsymbol{\phi}_z(\mathbf{z}|\mathbf{x})}}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y},\mathbf{z})\right] \\ &- \mathrm{D}_{KL}\left(q_{\boldsymbol{\phi}_y}(\mathbf{y}|\mathbf{x})\big|\big|p(\mathbf{y})\right) - \mathrm{D}_{KL}\left(q_{\boldsymbol{\phi}_z}(\mathbf{z}|\mathbf{x})\big|\big|p(\mathbf{z})\right)\end{aligned} \tag{6}$$

where $\boldsymbol{\phi} = \{\boldsymbol{\phi}_y, \boldsymbol{\phi}_z\}$.

The first term is calculated for efficient error backpropagation, i.e.,

$$\mathbb{E}_{q_{\boldsymbol{\phi}_y(\mathbf{y}|\mathbf{x})}q_{\boldsymbol{\phi}_z(\mathbf{z}|\mathbf{x})}}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y},\mathbf{z})\right] = \frac{1}{L}\sum_{l=1}^{L}\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}^{(l)},\mathbf{z}^{(l)}) \tag{7}$$

where $\boldsymbol{\epsilon}^{(l)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ is an additional Gaussian distribution, with l the index and L the total number of samples from this distribution. Introduced by the so-called reparameterization trick [15], $\mathbf{y}^{(l)}, \mathbf{z}^{(l)}$ are constructed by the learned mean and variance terms combined with sampling the additional distribution to form the estimation, i.e., $\mathbf{y}^{(l)} = \hat{\boldsymbol{\mu}}_y + \hat{\boldsymbol{\sigma}}_y^2 \odot \boldsymbol{\epsilon}^{(l)}$, $\mathbf{z}^{(l)} = \hat{\boldsymbol{\mu}}_z + \hat{\boldsymbol{\sigma}}_z^2 \odot \boldsymbol{\epsilon}^{(l)}$.

The last two terms can be calculated analytically as follows:

$$\begin{aligned}&\mathrm{D}_{KL}\left(q_{\boldsymbol{\phi}_y}(\mathbf{y}|\mathbf{x})\big|\big|p(\mathbf{y})\right) \\ =&\mathrm{D}_{KL}\left(\mathcal{N}(\hat{\boldsymbol{\mu}}_y, \hat{\boldsymbol{\sigma}}_y^2\boldsymbol{I})\big|\big|\mathcal{N}(\mathbf{0}, \epsilon_y\boldsymbol{I})\right) \\ =&1/2\sum_{d=1}^{D_y}\left(\hat{\mu}_y^{2,(d)}/\epsilon_y + \hat{\sigma}_y^{2,(d)}/\epsilon_y - \log\hat{\sigma}_y^{2,(d)}/\epsilon_y - 1\right), \\ &\mathrm{D}_{KL}\left(q_{\boldsymbol{\phi}_z}(\mathbf{z}|\mathbf{x})\big|\big|p(\mathbf{z})\right) \\ =&\mathrm{D}_{KL}\left(\mathcal{N}(\hat{\boldsymbol{\mu}}_z, \hat{\boldsymbol{\sigma}}_z^2\boldsymbol{I})\big|\big|\mathcal{N}(\mathbf{0}, \epsilon_z\boldsymbol{I})\right) \\ =&1/2\sum_{d=1}^{D_z}(\hat{\mu}_z^{2,(d)}/\epsilon_z + \hat{\sigma}_z^{2,(d)}/\epsilon_z - \log\hat{\sigma}_z^{2,(d)}/\epsilon_z - 1)\end{aligned} \tag{8}$$

equations (6)-(8) compose the full parametric form for the ELBO in Proposition 1, with respect to parameters $\{\boldsymbol{\phi}, \boldsymbol{\theta}\}$ for unknown distributions.

B. Unsupervised Loss Term

Given a partially labeled dataset with an unlabeled subset $\mathbb{X} = \{\mathbf{x}^{(i)}\}_{i=1}^{M}$ of M i.i.d. waveform samples, and a fully labeled subset $(\bar{\mathbb{X}}, \bar{\mathbb{L}}) = \{\bar{\mathbf{x}}^{(j)}, \Delta\bar{d}^{(j)}, \bar{k}^{(j)}\}_{j=1}^{N}$ consisting of N i.i.d. samples with paired signal data $\bar{\mathbf{x}}$, actual ranging error $\Delta\bar{d}$, and the actual label $\bar{k}$ for environment.

The unsupervised loss term is given by the negative form of the parametric ELBO according to equations (7)-(8), utilizing only the waveform data $\mathbf{x}$ without any label information. Such term serves to help disentangle data $\mathbf{x}$ into two independent latent variables. In particular, we construct a modified VAE to formulate the disentanglement in its bottleneck. The unsupervised loss term for such VAE is given as:

$$\mathbb{L}_{unsup}(\mathbb{X}, \bar{\mathbb{X}}; \boldsymbol{\phi}, \boldsymbol{\theta}) = -\sum_{i=1}^{M}\mathcal{L}(\mathbf{x}^{(i)}; \boldsymbol{\phi}, \boldsymbol{\theta}) - \sum_{j=1}^{N}\mathcal{L}(\bar{\mathbf{x}}^{(j)}; \boldsymbol{\phi}, \boldsymbol{\theta}) \tag{9}$$

C. Supervised Loss Terms

The two latent variables could hardly express range-related features and range-unrelated features with only the unsupervised loss term. For further guidance of such disentanglement, the labeled subset $(\bar{\mathbb{X}}, \bar{\mathbb{L}}) = \{\bar{\mathbf{x}}^{(j)}, \Delta\bar{d}^{(j)}, \bar{k}^{(j)}\}_{j=1}^{N}$ is used for an additional supervised loss term. In particular, we construct two additional neural modules to formulate the mappings from latent variables $\mathbf{y}$, $\mathbf{z}$ to Δd, k, respectively. The supervised loss term for sub modules is given as:

$$\begin{aligned}\mathbb{L}_{\mathrm{sup}}(\bar{\mathbb{X}}, \bar{\mathbb{L}}; \boldsymbol{\phi}, \boldsymbol{\varphi}) = &\sum_{j=1}^{N}\|\,\mathrm{f}_{\mathrm{est}}(\mathrm{g}_{\mathrm{enc}}(\bar{\mathbf{x}}^{(i)}; \boldsymbol{\phi}); \boldsymbol{\varphi}_e) - \Delta\bar{d}^{(i)}\|^2 \\ &+ \|\,\mathrm{f}_{\mathrm{cls}}(\mathrm{g}_{\mathrm{enc}}(\bar{\mathbf{x}}^{(i)}; \boldsymbol{\phi}); \boldsymbol{\varphi}_c) - \bar{k}^{(i)}\|^2\end{aligned} \tag{10}$$

where $\mathrm{f}_{\mathrm{est}}(\cdot; \boldsymbol{\varphi}_e) : \mathbf{y} \rightarrow \Delta$ maps the range-related features to the real distance label, and $\mathrm{f}_{\mathrm{cls}}(\cdot; \boldsymbol{\varphi}_c) : \mathbf{z} \rightarrow k$ maps the environment-related features to the environment label. Note that labels Δd and k help distinguish the range-related features from range-unrelated feature during the joint training of the neural modules with parameters $\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\varphi}$.

D. Learning Algorithms

The VAE guided by the unsupervised term and the two sub modules guided by the supervised term are learned together for fine-tuning parameters, leading to the efficient disentanglement of the range-related features and range-unrelated features. Given a partially labeled dataset $\mathcal{D} = (\mathbb{X}, \bar{\mathbb{X}}, \bar{\mathbb{L}})$, the total loss function for network learning is thus,

$$\mathbb{L}(\mathcal{D}; \boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\varphi}) = \mathbb{L}_{\text{unsup}}(\mathbb{X}, \bar{\mathbb{X}}; \boldsymbol{\phi}, \boldsymbol{\theta}) + \mathbb{L}_{\text{sup}}(\bar{\mathbb{X}}, \bar{\mathbb{L}}; \boldsymbol{\phi}, \boldsymbol{\varphi}) \quad (11)$$

which can lead to different learning results according to the numbers of labeled and unlabeled samples values, according to M and N.

Algorithms 1-2 describe the training and testing phases for the proposed algorithm.

IV. Dataset and Implementations

This section present the UWB dataset for evaluation and the implementation of our algorithm, including network architecture and optimizer for learning.

A. Dataset

We adopt a public dataset [17] to validate the proposed method. The dataset consists of 100 data samples in total. Each sample includes a waveform of 157 length, an actual ranging error, and two environmental labels for the room setting and blocking materials. The measurements are conducted based on the DWM1000 boards and the actual distances are taken using a Leica AT403 laser tracker to get the ground truth ranging errors. Therefore, the used waveforms and labels for real distance are real world data instead of synthetic data generated by simulation.

The dataset has a great advantage in generality. In particular, measurements are taken in five different room scenarios, including outdoor, big room, medium sized room, small room, and a through-the-wall (TTW) environment. Moreover, obstacles of ten different materials that blocking the LOS path are also taken into account. We assign samples in the medium size room as the testing set (13210 samples), and all the other samples as the training set (36023 samples), as suggested in [12].

B. Network Architecture

Our framework consists of an advanced VAE, a classifier, and an estimator, as claimed in Section III-D. Both the classifier and the estimator are of a simple 3-layer structure. The VAE module, on the other hand, inherits a more delicate structure to combine the environment variable and the range variable generically, as illustrated in Fig.2. In particular, the encoder has parallel structure with similar layer architectures to disentangle environment-related features and range-related features. The layer architecture for range-related features consists of 3 residual blocks and 2 upsampling blocks for range-related features, which is then directly fed into the decoder consisting of 3 residual blocks and 2 upsampling blocks symmetrically for signal reconstruction. The layer architecture for environment-related features, on the other hand consists of 3 residual blocks and a global pooling layer to generate the environment-related features. Unlike the range-related features, the environment-related features is then fed into a MLP block to produce a set of AdaIN [16] parameters for the residual blocks in the decoder.

Aside from the structures of the three sub modules, the detailed choices of their layers (i.e., linear, 1D convolutional, or 2D convolutional) are discussed in the Section V-D.

C. Hyper-Parameters

We use the Adam [18] optimizer with 500 epochs, where the learning rate is 0.0002 and decay half every 100 epochs. The decays of first and second momentum of gradients are $\beta_1 = 0.5$ and $\beta_2 = 0.999$, respectively. Weights for the loss terms in (7) are set as $\lambda_{\text{unsup}} = 10$ and $\lambda_{\text{sup}} = 1$ for all our experiments.

We build the model in Pytorch [19] and conduct learning on a GTX 1080 GPU with a memory of 12 GB and the accelerator powered by the NVIDA Pascal architecture.

V. Experiments

In this section, we evaluate our proposed method on the aforementioned dataset, and validate the robustness of the proposed method under different supervision rate. For convenience of notation, we refer to the proposed semi-supervised method as Semi-VL in the rest of this section. For Section V-B, V-C, the default setting of layer choices is adopted, where the VAE uses 2D convolutional layers and the two sub modules uses linear layers. In Section V-D, different choices of layers for the three sub modules are discussed.

A. Baselines and Evaluation Metrics

We adopt three metrics for performance evaluation: the root mean square error (RMSE), the mean absolute error (MAE), and the inference time.

Two learning-based methods are used as the baseline: SVM method in [7] and REMNet in [12]. Unlike the proposed method, these methods are trained on a fully labeled dataset. Note that results of REMNet [12] are directly adopted from its paper, indicating with asterisks next to the MAE values in table I. Their results are obtained by the same dataset with the same way to assign training set and testing set. The remaining differences are that REMNet treat LOS and NLOS conditions separately, and only evaluate the performance in terms of MAE. Therefore, we present their results in Table I for a rough comparison.

B. Result Analysis

Following the semi-supervised dataset claimed in Section III-B, suppose the dataset consists of N labeled samples and M unlabeled samples, we define the supervision rate η as $\eta = \frac{N}{M+N}$. We start by comparing the baseline approaches and

Algorithm 1 Training Phase

Input: $\mathcal{D}$, the training set, α, the learning rate. m, the batch size.

Input: ϕ_0, initial encoder's parameters. $\boldsymbol{\theta}_0$, initial decoder's parameters. $\boldsymbol{\varphi}_0$, initial sub modules parameters.

while $\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\varphi}$ have not converged **do**
  Sample $\{x^{(i)}\}_{i=1}^{m} \sim \mathcal{D}$ a batch from the dataset.
  $g_{\boldsymbol{\phi}} \leftarrow \nabla_{\boldsymbol{\phi}} \mathbb{L}_{\text{unsup}}(\{x^{(i)}\}_{i=1}^{m}; \boldsymbol{\phi}, \boldsymbol{\theta})$.
  $g_{\boldsymbol{\theta}} \leftarrow \nabla_{\boldsymbol{\theta}} \mathbb{L}_{\text{unsup}}(\{x^{(i)}\}_{i=1}^{m}; \boldsymbol{\phi}, \boldsymbol{\theta})$.
  $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \alpha * Adam(\boldsymbol{\phi}, g_{\boldsymbol{\phi}})$.
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha * Adam(\boldsymbol{\theta}, g_{\boldsymbol{\theta}})$
  if $x^{(i)}$ in the labeled sub set $\mathcal{D}_2$ **then**
    Get $l^{(i)} = \{\Delta d^{(i)}, k^{(i)}\}$ according to $x^{(i)}$.
    $g_{\boldsymbol{\theta}} \leftarrow \nabla_{\boldsymbol{\theta}} \mathbb{L}_{\text{sup}}(x^{(i)}, l^{(i)}; \boldsymbol{\theta}, \boldsymbol{\varphi})$.
    $f_{\boldsymbol{\varphi}} \leftarrow \nabla_{\boldsymbol{\varphi}} \mathbb{L}_{\text{sup}}(x^{(i)}, l^{(i)}; \boldsymbol{\theta}, \boldsymbol{\varphi})$.
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha * Adam(\boldsymbol{\theta}, g_{\boldsymbol{\theta}})$.
    $\boldsymbol{\varphi} \leftarrow \boldsymbol{\varphi} + \alpha * Adam(\boldsymbol{\varphi}, f_{\boldsymbol{\varphi}})$.
  end if
end while
Return Optimized parameters $\boldsymbol{\theta}^*, \boldsymbol{\phi}^*$, and $\boldsymbol{\varphi}^*$.

Algorithm 2 Inference Phase

Input: x_{test}, the observed waveform.

Input: $\boldsymbol{\theta}^*, \boldsymbol{\phi}^*$, encoder's parameters. $\boldsymbol{\varphi}^* = \{\boldsymbol{\varphi}_e^*, \boldsymbol{\varphi}_c^*\}$, parameters for estimator and classifier, respectively.

1: Feed observed data x_{test} to encoder parameterized with $\boldsymbol{\theta}^*$, generate variables $\mathbf{y}, \mathbf{z}$.
2: Feed range-related features $\mathbf{y}$ to estimator parameterized with $\boldsymbol{\varphi}_e^*$, obtain $\Delta\hat{d}_{\text{test}}$.
3: Feed environment-related features $\mathbf{z}$ to estimator parameterized with $\boldsymbol{\varphi}_c^*$, obtain $\hat{k}_{\text{test}}$.
4: **Return** Estimation results $\Delta\hat{d}_{\text{test}}$, $\hat{k}_{\text{test}}$.

the proposed Semi-VL method under 5 different supervision rates, i.e., $\eta = 0.2, 0.4, 0.6, 0.8, 1.0$.

Quantitative results are shown in Table I, including RMSE (m) and MAE (m) for effectiveness and inference time per sample (ms) for efficiency. It can be seen that all the methods successfully mitigate the ranging error to some extend. REMNet outperforms SVM, indicating that learning-based features are superior than hand-crafted features for ranging error mitigation. The proposed Semi-VL achieves significant performance improvement than all the baseline approaches, even at a low rate of supervision. In addition, we compare the results of Semi-VL under different rates of supervision. The results show that the method trained with higher supervision rate tend to have better results, but the performance is rather close. This validate the proposed methodology that the potential of unlabeled data could be excavated by the proposed variational learning method.

C. Convergence of Learning

The learning curves of the proposed Semi-VL under different supervision rates are illustrated in Fig.3. It can be seen that all the learning processes converges at around 300 epochs, while the stability is positively correlated to the supervision rate. In particular, the learning process with higher supervision rate is relatively more steady and can converge faster to a desirable result, in accordance with the observations in Section V-B. Moreover, the differences between learning curves with different supervision rates are far from significant. This implies that the proposed Semi-VL can be optimized efficiently and converge easily even at a low supervision rate.

TABLE I
QUANTITATIVE RESULTS OF COMPETITORS AND THE PROPOSED METHOD (REFERRED TO AS SEMI-VL) UNDER DIFFERENT SUPERVISION RATES.

METHODS	RMSE (M)	MAE (M)	TIME (MS)
UNMITIGATED	0.1244	0.1244	-
SVM [2]	0.1537	0.0889	0.837
REMNET (LOS) [12]	-	0.0445*	-
REMNET (NLOS) [12]	-	0.0687*	-
Semi-VL ($\eta = 0.1$)	0.0663	0.0176	0.242
Semi-VL ($\eta = 0.2$)	0.0603	0.0164	0.252
Semi-VL ($\eta = 0.4$)	0.0603	0.0166	0.311
Semi-VL ($\eta = 0.6$)	0.0580	0.0163	0.210
Semi-VL ($\eta = 0.8$)	0.0567	**0.0151**	0.239
Semi-VL ($\eta = 1.0$)	**0.0558**	0.0157	0.285

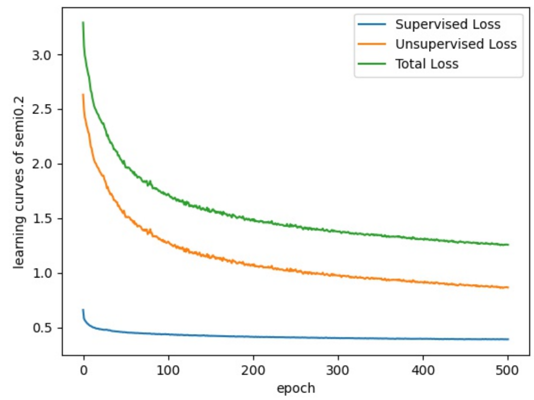
TABLE II
QUANTITATIVE RESULTS OF THE PROPOSED SEMI-VL WITH DIFFERENT NETWORK STRUCTURES WITH SUPERVISION RATE $\eta = 1.0$.

LAYER FORMS		RMSE	MAE	TIME
CONV1D AE	LINEAR EST	0.0740	0.0324	0.129
	CONV1D EST	0.0793	0.0394	0.156
	CONV2D EST	0.0786	0.0391	0.223
CONV2D AE	LINEAR EST	**0.0558**	**0.0157**	**0.285**
	CONV1D EST	0.0693	0.0334	0.350
	CONV2D EST	0.0613	0.0274	0.378

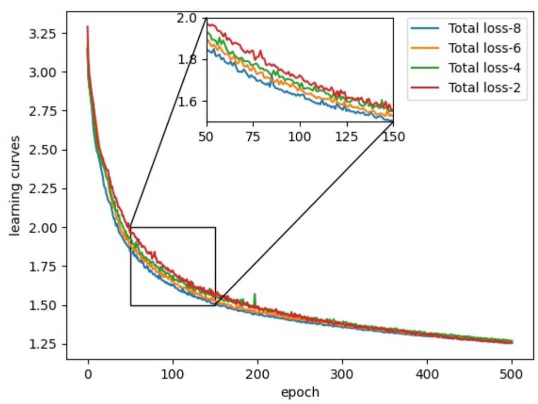
D. Ablation Study

Implemented in the architecture described in Section IV-B, the choice of detailed layers for the three sub modules remain open. We study three different layers for each module, i.e., linear, 1D convolutional, and 2D convolutional layers. The results are obtained under the supervision rate of 1.0, illustrated in Table II. Note that only one module uses different layer kinds each time, while the other two keep the default setting.

The results show that the combination of a 2D convolutional VAE, a 2D convoultional classifier, and a linear estimator achieves the best performance. This may be because the 2D convolutional operation makes use of the information in correlated sample points in signal and integrates more spatial information. However, such benefit come with a cost of longer inference time. In addition, we observe that the performance of mitigation is not very sensitive to the choice of layers,



(a)



(b)

Fig. 3. Learning curves of the proposed Semi-VL. (a) Changes of the total loss, supervised loss, and unsupervised loss terms during the learning process with supervision rate $\eta = 0.2$; (b) Learning curves in terms of total loss under different supervision rates.

implying the generality of Semi-VL and the effectiveness of the proposed variational learning methodology.

VI. Conclusion

We proposed a semi-supervised learning method based on variational Bayes for UWB ranging error mitigation. Specifically, we presented a Bayesian model to interpret the generative process of signal data, with two latent variables for range and environment features. Such methodology can be adapted analogously to multiple signal processing problems. Moreover, the semi-supervised setting introduced here can easily scale to more developed types of supervision (e.g., with different kinds of labels and even mixed labels). Our future work will focus on such scalability on supervision types, which is potential to enable transfer learning and a wider scope of applications in wireless communications.

Acknowledgment

This research is partially supported by National Key R&D Program of China 2020YFC1511803, the Basque Government through the ELKARTEK programme, the Spanish Ministry of Science and Innovation through Ramon y Cajal Grant RYC-2016-19383 and Project PID2019-105058GA-I00, and Tsinghua University - OPPO Joint Institute for Mobile Sensing Technology.

References

[1] Y. Shen, H. Wymeersch, and M. Z. Win, "Fundamental limits of wideband localization – Part II: Cooperative networks," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 4981–5000, Oct. 2010.

[2] H. Wymeersch, S. Maranò, W. M. Gifford, and M. Win, "A machine learning approach to ranging error mitigation for UWB localization," *IEEE Trans. Commun.*, vol. 60, pp. 1719–1728, Apr. 2012.

[3] W. M.Z., C. G., and M. A.F., "Wideband diversity in multipath channels with nonuniform power dispersion profiles," *IEEE Trans. Wireless Commun.*, vol. 5, no. 5, pp. 1014–1022, Jun. 2006.

[4] K. Johan, W. Shurjeel, A. Peter, T. Fredrik, and M. A. F., "A measurement-based statistical model for industrial ultra-wideband channels," *IEEE Trans. Wireless Commun.*, vol. 6, no. 8, pp. 3028–3037, Aug. 2007.

[5] U. A. Khan, S. Kar, and J. M. F. Moura, "Diland: An algorithm for distributed sensor localization with noisy distance measurements," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1940–1947, Mar. 2010.

[6] S. A.H., T. A., and K. N., "Network-based wireless location: challenges faced in developing techniques for accurate wireless location information," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 24–40, Jun. 2005.

[7] S. Maranò, 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, Sep. 2010.

[8] Z. Xiao, H. Wen, A. Markham, A. Trigoni, P. Blunsom, and J. Frolik, "Identification and mitigation of non-line-of-sight conditions using received signal strength," *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 667–674, 2013.

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

[10] A. Conti, S. Mazuelas, S. Bartoletti, W. Lindsey, and M. Win, "Soft information for localization-of-things," *Proc. IEEE*, vol. 107, pp. 2240–2264, Sep. 2019.

[11] M. Zhao, S. Yue, D. Katabi, T. Jaakkola, and M. Bianchi, "Learning sleep stages from radio signals: A conditional adversarial architecture," in *ICML*, 2017.

[12] S. Angarano, V. Mazzia, F. Salvetti, G. Fantin, and M. Chiaberge, "Robust ultra-wideband range error mitigation with deep learning at the edge," *ArXiv*, vol. abs/2011.14684, May 2020.

[13] Y. Li, S. Mazuelas, and Y. Shen, "Deep Generative Model for Simultaneous Range Error Mitigation and Environment Identification," in *Proc. IEEE Global Telecomm. Conf.*, 2022, To Appear.

[14] ——, "A deep learning approach for generating soft range information from RF data," in *Proc. IEEE Global Telecomm. Conf. Workshop*, 2022, To Appear.

[15] K. D. P and W. Max, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[16] H. Xun and B. Serge, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1501–1510.

[17] S. Angarano, F. Salvetti, V. Mazzia, G. Fantin, and M. Chiaberge, "Deep UWB: A dataset for uwb ranging error mitigation in indoor environments." [OL], https://zenodo.org/record/4290069 .X75qYc3-3Dc.

[18] K. D. P and B. Jimmy, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[19] 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," 2017.