1 Object Trajectory Estimation with Continuous-Time Neural Dynamic Learning of Millimeter-Wave Wi-F

Vaca-Rubio, Cristian; Wang, Pu; Koike-Akino, Toshiaki; Wang, Ye; Boufounos, Petros T.; Popovski, Petar

TR2024-044 April 27, 2024

Abstract

In this paper, we leverage standard-compliant beam training measurements from commercial millimeter-wave (mmWave) Wi-Fi communication devices for object localization and, specifically, continuous trajectory estimation and prediction. The main challenge is that the sampling of beam training measurements is intermittent, due to the beam scanning overhead and the uncertainty of the transmission instant caused by the contention over the wireless channel. In order to cope with this intermittency, we devise a method to assist the localization by exploiting the underlying object dynamics. The method consists of a dual-decoder neural dynamic learning framework that reconstructs Wi-Fi beam training measurements at irregular time intervals and learns the unknown latent dynamics in a continuous-time fashion powered by the use of an ordinary differential equation (ODE). Utilizing the variational autoencoder (VAE) framework, we have derived a modified evidence lower bound (ELBO) loss function for the dual-decoder architecture that balances the unsupervised waveform reconstruction and supervised coordinate estimation tasks. To evaluate the proposed method, we build an in-house testbed consisting of commercial 802.11ad routers, with a TurtleBot as a mobile user, and collect a real-world mmWave Wi-Fi beam training dataset. Our results demonstrate substantial performance improvements over a list of baseline methods, further validated through an extensive ablation study.

IEEE Journal of Selected Topics in Signal Processing 2024

Object Trajectory Estimation with Continuous-Time Neural Dynamic Learning of Millimeter-Wave Wi-Fi

Cristian J. Vaca-Rubio, Pu (Perry) Wang, Toshiaki Koike-Akino, Ye Wang, Petros Boufounos, Petar Popovski

***Abstract*—In this paper, we leverage standard-compliant beam training measurements from commercial millimeter-wave (mmWave) Wi-Fi communication devices for object localization and, specifically, continuous trajectory estimation and prediction. The main challenge is that the sampling of beam training measurements is intermittent, due to the beam scanning overhead and the uncertainty of the transmission instant caused by the contention over the wireless channel. In order to cope with this intermittency, we devise a method to assist the localization by exploiting the underlying object dynamics. The method consists of a dual-decoder neural dynamic learning framework that reconstructs Wi-Fi beam training measurements at irregular time intervals and learns the unknown latent dynamics in a continuous-time fashion powered by the use of an ordinary differential equation (ODE). Utilizing the variational autoencoder (VAE) framework, we have derived a modified evidence lower bound (ELBO) loss function for the dual-decoder architecture that balances the unsupervised waveform reconstruction and supervised coordinate estimation tasks. To evaluate the proposed method, we build an in-house testbed consisting of commercial 802.11ad routers, with a TurtleBot as a mobile user, and collect a real-world mmWave Wi-Fi beam training dataset. Our results demonstrate substantial performance improvements over a list of baseline methods, further validated through an extensive ablation study.**

***Index Terms*—WLAN sensing, Wi-Fi, 802.11ad/ay, 802.11bf, localization, fingerprinting, beam training, dynamic learning.**

I. Introduction

Wi-Fi sensing has been an integral part of emerging integrated sensing and communications (ISAC), as corroborated by the establishment of a new 802.11bf WLAN Sensing task group for robust and reliable sensing in September 2020. It aims to make greater use of 802.11 Wi-Fi signals for reliable and secure wireless sensing towards new industrial and commercial applications in home security, entertainment, energy management (HVAC, light, device power savings), elderly care, and assisted living.

The scope of 802.11bf covers both sub-7 GHz and millimeter (mmWave) Wi-Fi sensing above 45 GHz, built on 802.11g/n/ac/ax/be and, respectively, 802.11ad/ay standards. For the mmWave Wi-Fi sensing or Directional Multi-Gigabit (DMG) sensing, the new 802.11bf opens up possibilities of reusing beam training measurements, e.g., beam signal-to-noise ratios (beam SNRs), for sensing applications. Our work is to leverage such sensing-supported beam training measurement for indoor localization of moving objects such as a robot or a mobile user; see an illustration in Fig. 1.

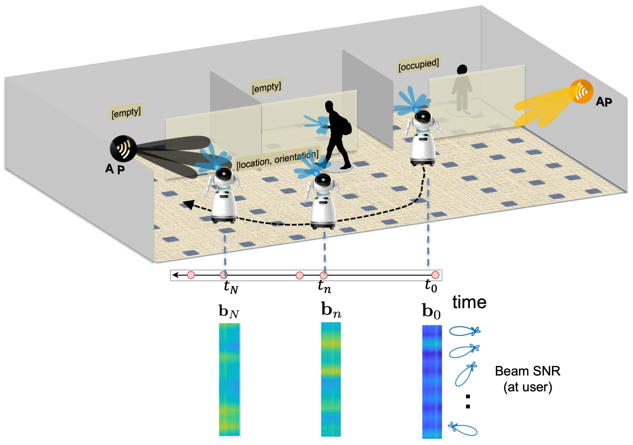


Fig. 1: Trajectory estimation of moving objects using mmWave Wi-Fi beam training measurements.

Along with coarse-grained received signal strength indicator (RSSI) [2]–[5] and fine-grained channel state information (CSI) at sub-7 GHz [6]–[14], mid-grained mmWave beam training measurements have been previously explored in [15]–[23][1]. Most of existing approaches are frame-based. That is, the object location is inferred from the current Wi-Fi frame, without integration of past measurements or previous trajectory history. For the frame-based setting, traditional machine learning and advanced deep learning methods have been applied to all Wi-Fi fingerprinted measurements [2], [5], [24]–[27]. For instance, the k-nearest neighbor (kNN), support vector machine (SVM), and decision trees (DT) were applied to the RSSI-based fingerprinting method [2], [3]. DeepFi exploits 90 CSI amplitudes from all the subcarriers at three antennas for feature extraction using an autoencoder architecture [6], [7]. More recently, a pretrained fusion network between the CSI at sub-7 GHz and the beam training measurements at 60 GHz was proposed for both localization and device-free sensing tasks [21].

Part of this paper was presented in ICASSP 2023 [1].

The work of C.J. Vaca-Rubio was done during his visit and internship at MERL. He was also supported by European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant agreement No. 813999, and in part by the European Union's Horizon EUROPE research and innovation program under grant agreement No. 101037090 - project CENTRIC.

PW, TK, YW, and PB are with Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139, USA.

CVR was with the Department of Electronic Systems, Aalborg University, Aalborg, Denmark, and now is with Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), Barcelona, Spain.

PP is with the Department of Electronic Systems, Aalborg University, Aalborg, Denmark.

*Corresponding author: pwang@merl.com

[1]Please refer to [21, Section II] for detailed discussions on all three types of Wi-Fi channel measurements.

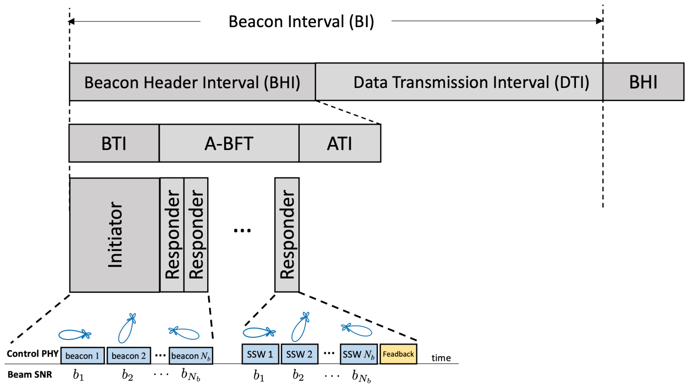


Fig. 2: MmWave Wi-Fi beam training protocol during the mandatory beacon transmission interval (BTI) and association beamforming training (A-BFT) in 802.11ad/ay standards.

On the other hand, *sequence-based* approaches take consecutive Wi-Fi frames as the input, and state estimation (e.g, Kalman filter-like approaches [28], [29]) and recurrent neural networks (e.g., GRU and LSTM [30]) can be applied for trajectory estimation with the RSSI and CSI [5], [9]–[12] at sub-7 GHz; More detailed discussion on sequence-based solutions in Section. II. However, the sequence-based formulation has NOT been applied to mmWave Wi-Fi beam training measurements such as beam SNRs due to the following fundamental technical challenges:

1) **Low beam training rate**: mmWave Wi-Fi such as 802.11ad/ay organizes access to the medium in beacon intervals (BIs), usually in the order of 100ms. A unique feature of mmWave Wi-Fi at 60 GHz is the use of directional beamforming to compensate for path loss and identify unassociated stations at a further distance. As shown in Fig. 2, a BI consists of two main access periods: beacon header interval (BHI) and data transmission interval (DTI). While DTI is mainly used for data transmission, mmWave Wi-Fi (802.11ad/ay) devices are required to perform beam training during the BHI to initiate the data transmission. The BHI is further subdivided into three sub-intervals: beacon transmission interval (BTI), association beamforming training (A-BFT), and announcement transmission interval (ATI) for management frame exchange between the AP and beam-trained stations.

During the BTI sub-interval, an access point (AP) sends directional beacon frames to train its transmission sector-level beampatterns, also referred to as the downlink BTI beam training. Multiple users can simultaneously compute their own received beam SNRs corresponding to each of the transmitted beampatterns using a quasi-omnidirectional receiving beampattern. This mandatory beam training results in significant overhead to the Wi-Fi network and it is desired to limit the number of directional beampatterns within a beacon and the total number of beacons, resulting in sparsely sampled beam measurements than Wi-Fi at sub-7 GHz.

2) **Irregular sample intervals**: At the A-BFT subinterval, the users or responders (e.g., mobile devices) can train its (transmitting or receiving) beampatterns by sending a sequence of (short) sector sweep (SSW) frames to the AP, as shown in Fig. 2. Compared with the downlink BTI beam training for multiple users simultaneously, the uplink A-BFT beam training is reserved for one responder at a time. Specifically, A-BFT is slotted up to 8 slots in 802.11ad and 40 in 802.11ay. Multiple responders randomly choose one of the slots for transmitting SSW frames. Consequently, when multiple responders exist, each responder needs to contend the channel time, and one responder is randomly selected. As a result, such contention-based channel access results in irregularly sampled beam SNR measurements at AP for a given user.

To address the above challenges and inspired by recent advances in neural ordinary differential equation (ODE) [31]–[37], this paper proposes a dual-decoder neural dynamic learning framework that learns consistent latent dynamics described by a unified ODE for both waveform reconstruction and coordinate estimation. Compared with the original neural ODE, our dual-decoder structure enforces that the learned latent dynamics not only recover the input sequences in the waveform domain but also map to the object trajectory, thus grounding the latent dynamic learning into the physical (coordinate) space. This dual-decoder structure is further enhanced by the introduction of a modified evidence lower bound (ELBO) loss function to couple the losses from the dual decoder. It is worth noting that a conference version of the proposed method was published in [1] with limited derivation and performance evaluation. This paper significantly expands [1] with the following contributions:

1) We propose the first-of-its-kind sequence-to-sequence object trajectory estimation workflow that can handle a set of consecutive Wi-Fi measurements, e.g., beam SNRs, at irregular sample instances and directly output the whole trajectory over the same time interval.
2) We present a dual-decoder neural network that learns the underlying latent dynamics in a continuous-time fashion by exploring the neural ODE framework. The dual-decoder architecture regularizes the learnable neural ODE function in the latent space by grounding it to the coordinate-level measurable space.
3) We derive a customized cost function by extending the ELBO for the sequence-to-sequence formulation and the dual decoder structure.
4) We build an in-house mmWave Wi-Fi testbed consisting of commercial 802.11ad-compliant Wi-Fi routers and a moving robot with coordinates labels. The mmWave Wi-Fi trajectory estimation dataset was collected continuously over several hours in a span of days[2].
5) We benchmark our proposed method against a list of baseline methods including frame-based and sequence-based approaches using classic machine learning and state-of-art deep learning pipelines.
6) We shed more light on the representation capacity of the proposed method via comprehensive ablation studies on 1) sequence length, 2) supervision intensity (regular versus dense) at the coordinate decoder, 3) day-to-day

[2]Our mmWave Wi-Fi trajectory estimation dataset will be available at https://www.merl.com/.

generalization, 4) varying tasks (trajectory estimation versus extrapolation), and 5) inspection of latent space.

The remainder of this paper is organized as follows. The problem formulation is described in Section II where existing solutions are also briefly reviewed. Section III introduces our dual-decoder neural dynamic learning framework, the derivation of a customized cost function, and an analysis of computational complexity. Section IV describes the data collection system and the mmWave Wi-Fi trajectory estimation dataset collected over multiple days. Simulation and experimental results are given in Section V, followed by conclusions in Section VI.

II. Problem Formulation and Existing Solutions

A. Problem Formulation

We formulate the indoor localization as a sequence-to-sequence regression problem using the mmWave Wi-Fi beam training measurements over a period of ΔT_w seconds. Specifically, stacking a set of N_b beam SNRs computed at AP over an A-BFT slot t_n as $\mathbf{b}_n = [b_1, b_2, ..., b_{N_b}]^T \in \mathbb{R}^{N_b \times 1}$, the problem of interest is to utilize beam SNR measurements $\{\mathbf{b}_n\}_{n=0}^N$ at time steps $\{t_n\}_{n=0}^N$ with irregular sample intervals to localize the object,

$$\{\mathbf{b}_n, t_n\}_{n=0}^N \rightarrow \{\mathbf{c}_n\}_{n=0}^N, \quad s.t. \quad \Delta t_n = t_n - t_{n-1} \neq \Delta t_{n+1} \tag{1}$$

where $\mathbf{c}_n = [x_n, y_n]^T$ consists of corresponding two-dimensional coordinates (x_n, y_n) at t_n. This is illustrated in Fig. 3 where the trajectory estimation is to convert the set of beam SNRs $\{\mathbf{b}_n\}_{n=0}^N$ at intermittently-sampled steps $\{t_n\}_{n=0}^N$ (shown in the top left part) to the set of $\{\mathbf{c}_n\}_{n=0}^N$ over a continuous trajectory (shown in the right bottom part).

Like fingerprinting-based Wi-Fi localization methods, we collect beam SNR measurements and corresponding two-dimensional coordinate labels. As detailed in Section IV, the beam SNR data is continuously collected over several hours each day and spanning over multiple days. We then divide the beam SNRs into non-overlapping sequences and split the sequences into training and test datasets without any data leakage. The learning-based approach is to extract time-dependent features from the sequence of beam SNR measurements $\{\mathbf{b}_n, t_n\}_{n=0}^N$ and regress these features to a trajectory or a sequence of coordinate $\{\mathbf{c}_n\}_{n=0}^N$ from the training dataset. Once the model is trained, the trajectory estimation performance is evaluated in the test dataset.

B. Existing Solutions

The frame-based approach takes the beam SNR measurement $\mathbf{b}_n$ at a time step t_n and directly estimates the corresponding coordinate $\hat{\mathbf{c}}_n$. Specifically, we have

$$\{\mathbf{b}_n\} \rightarrow \mathbf{c}_n = \{x_n, y_n\}. \tag{2}$$

As we mentioned earlier, classic machine learning methods such as support vector regression (SVR) [38] and Gaussian processing (GP) [39] and deep learning pipelines such as multi-layer perceptron (MLP) [40], [41] and convolution neural network (CNNs) [42], [43] can be applied.

Similar to our formulation, sequence-based approaches take multiple consecutive beam training measurement and estimate a trajectory

$$\{\mathbf{b}_n\}_{n=0}^N \rightarrow \{\mathbf{c}_n\}_{n=0}^N, \tag{3}$$

with or without time sampling instances t_n. In the case of regularly sampled data, $\Delta t_1 = \Delta t_2 = \cdots = \Delta t_N$ is constant in both the training and test datasets such that t_n becomes irrelevant. In this case, standard RNN can be used to utilize historic Wi-Fi data [5], [10], [12] and learn time-dependent features for trajectory estimation. Particularly, an LSTM is to estimate the conditional probability [30]

$$p(\mathbf{c}_n | \{\mathbf{b}_i\}_{i=n-N+1}^n). \tag{4}$$

where each LSTM unit is trained to sequentially update time-dependent latent (hidden) variables $\mathbf{h}_n$ using previous latent variable $\mathbf{h}_{n-1}$ and the current mmWave beam training measurement $\mathbf{b}_n$

$$\begin{aligned} \mathbf{h}'_n &= \mathbf{h}_{n-1}, \\ \mathbf{h}_n &= \mathcal{R}(\mathbf{h}'_n, \mathbf{b}_n; \boldsymbol{\theta}), \quad n = 0, 1, \cdots, N, \end{aligned} \tag{5}$$

where we introduce an auxiliary variable $\mathbf{h}'_n$ for a unified interpretation and $\mathcal{R}$ represents an LSTM unit with trainable parameters $\boldsymbol{\theta}$. The detailed implementation of $\mathcal{R}$ is shown in Appendix A. The updated latent variable $\mathbf{h}_n$ can be used to infer the coordinate sequentially and, hence, estimate the trajectory.

When the input signal is irregularly sampled, one can augment the beam SNR with corresponding sampling interval Δt_n and feed the augmented beam SNR to the LSTM unit as

$$\begin{aligned} \mathbf{h}'_n &= \mathbf{h}_{n-1}, \quad \tilde{\mathbf{b}}_n = \left[\mathbf{b}_n^T, \Delta t_n\right]^T, \\ \mathbf{h}_n &= \mathcal{R}(\mathbf{h}'_n, \tilde{\mathbf{b}}_n; \boldsymbol{\theta}), \quad n = 0, 1, \cdots, N. \end{aligned} \tag{6}$$

We refer to the above as *RNN-*Δ for baseline comparison in Section V.

Furthermore, one can further damp the latent variable with an exponential decaying factor to decide the amount of hidden state should be kept from the previous time step to the current time step:

$$\begin{aligned} \mathbf{h}'_n &= \mathbf{h}_{n-1} e^{-\Delta t_{n-1}}, \\ \mathbf{h}_n &= \mathcal{R}(\mathbf{h}'_n, \mathbf{b}_n; \boldsymbol{\theta}), \quad n = 0, 1, \cdots, N, \end{aligned} \tag{7}$$

which is referred to as *RNN-Decay*.

Besides the learning-based solutions, sparsity-based and/or optimization-based solutions were considered in [15], [44], [45]. However, they require the knowledge of scanning beampatterns as one needs to express the observed beam SNR values as a weighted sum of beam gains and reflection strength at (LOS/NLOS) propagation directions. In [45], each beampattern of the considered commercial $802.11ad$ router was measured at an anechoic chamber; see Fig. 5 of [45]. These measured beampatterns were then utilized to jointly reconstruct the propagation angles at multiple APs and subsequently locate the user.

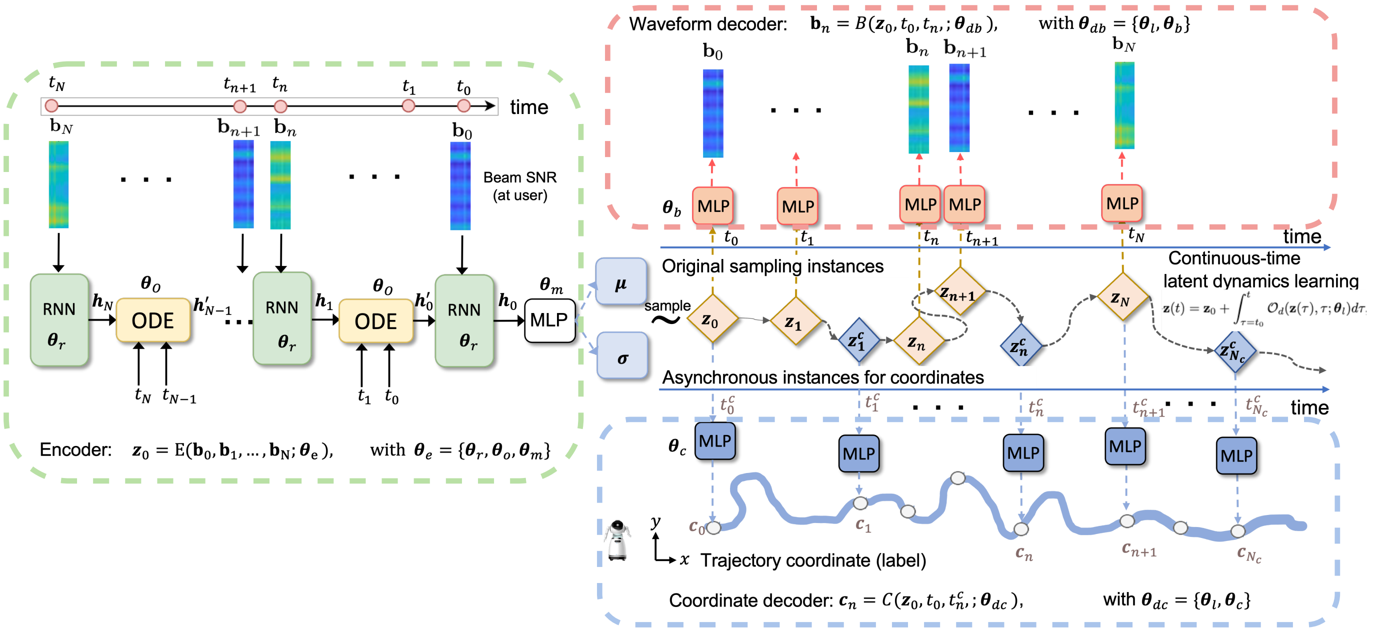

Waveform decoder: $\mathbf{b}_n = B(\mathbf{z}_0, t_0, t_n; \boldsymbol{\theta}_{db})$, with $\boldsymbol{\theta}_{db} = \{\boldsymbol{\theta}_l, \boldsymbol{\theta}_b\}$
time
Beam SNR (at user)
RNN
ODE
MLP
μ
σ
sample
Original sampling instances
Continuous-time latent dynamics learning
$\mathbf{z}(t) = \mathbf{z}_0 + \int_{\tau=t_0}^{t} \mathcal{O}_d(\mathbf{z}(\tau), \tau; \boldsymbol{\theta}_l) d\tau$
Asynchronous instances for coordinates
Trajectory coordinate (label)
Encoder: $\mathbf{z}_0 = \mathrm{E}(\mathbf{b}_0, \mathbf{b}_1, \ldots, \mathbf{b}_N; \boldsymbol{\theta}_e)$, with $\boldsymbol{\theta}_e = \{\boldsymbol{\theta}_r, \boldsymbol{\theta}_o, \boldsymbol{\theta}_m\}$
Coordinate decoder: $\boldsymbol{c}_n = C(\mathbf{z}_0, t_0, t_n^c; \boldsymbol{\theta}_{dc})$, with $\boldsymbol{\theta}_{dc} = \{\boldsymbol{\theta}_l, \boldsymbol{\theta}_c\}$

Fig. 3: Object trajectory estimation using mmWave Wi-Fi beam training measurements with neural dynamic learning. Left: Encoder maps a sequence of irregularly sampled Wi-Fi beam SNRs into a continuous-time latent space and infers its initial condition $\mathbf{z}_0$. Top right: Waveform Decoder maps latent dynamic states in the same sampling time instances back to the original measurement space for waveform reconstruction. Middle right: Numerical ODE solvers compute latent states at any queried time instances (e.g., original sampling time instances t_n or time instances with coordinate labels t_n^c) with a trainable ODE function and sampled initial condition $\mathbf{z}_0$. Bottom right: Coordinate Decoder maps latent dynamic states into observable coordinate space at asynchronous/new queried time instances to regularize the learning of latent dynamic ODE function.

III. Dual-Decoder Neural Dynamic Learning

As opposed to the above existing solutions, we propose a dual-decoder neural dynamic (DDND) framework that explicitly utilizes the continuous-time ODE function and its numerical solver to propagate the time-dependent latent feature from one time instance to the next time instance with an irregular time interval and regularizes the learning of latent ODE by grounding it to both waveform and observable coordinate spaces.

More specifically, Fig. 3 shows our DDND framwork that takes a sequence of beam SNR measurements $\{\mathbf{b}_n\}_{n=0}^N$ to the encoder (Left) and reconstructs the beam SNR measurement in one of the decoders (Top right) and outputs a sequence of coordinate estimates in the other (Bottom right). On the encoder side, the latent dynamic learning is achieved by solving a shared ODE function over the union of two sets of time instances (Center right): one for waveform reconstruction and the other for trajectory estimation. In the following, we introduce the main blocks in order.

A. Encoder

Our encoder follows the unrolled RNN architecture but in a reverse-time order from t_N to t_0; The motivation for inputting time-reversed sequences is related to improving the expressiveness and capturing richer dynamics of the underlying process. The main reasons are:

- Bidirectional modeling: By using time-reversed sequences, the model effectively considers information from both past and future time points simultaneously. The encoder learns dynamics from the future, while the decoder learns from the past. This bidirectional modeling can be beneficial for capturing dependencies that exist in both directions in time. It allows the latent variable to encode information not only from the past but also from the future, potentially improving the overall representation.
- Handling irregular sampling: Our data is intermittently sampled. Time-reversed sequences can provide an alternative perspective for the model to learn from the available beam SNR measurements, potentially enhancing its ability to handle irregularities in the time series.

Furthermore, at a time instance n, the beam SNR measurement $\mathbf{b}_n$ and a time-dependent latent feature $\mathbf{h}'_n$, properly propagated from the previous time step $\mathbf{h}_{n+1}$, are fed into a standard LSTM unit to an updated latent feature $\mathbf{h}_n$ of dimension L_h as

$$\mathbf{h}_n = \mathcal{R}_e(\mathbf{h}'_n, \mathbf{b}_n; \boldsymbol{\theta}_r), \quad n = N, N-1, \cdots, 0, \tag{8}$$

where $\mathcal{R}_e$ is a standard LSTM update step with associated learnable parameters $\boldsymbol{\theta}_r$ defined in Appendix A. Note that $\boldsymbol{\theta}_r$ is shared over all time steps in the encoder.

Different from RNN-Δ and RNN-Decay, we adopt a continuous-time ODE function to explicitly describe the evolving of $\mathbf{h}_{n+1}$ at time t_{n+1} to the auxiliary latent $\mathbf{h}'_n$ at time t_n.

Mathematically, the continuous-time ODE function is given as [31], [32]

$$\frac{d\mathbf{h}(t)}{dt} = \mathcal{O}_e(\mathbf{h}(t), t; \boldsymbol{\theta}_o), \tag{9}$$

where $\mathcal{O}_e$ is represented by a neural network, e.g., an MLP, parameterized by $\boldsymbol{\theta}_o$. Given the latent variable $\mathbf{h}_{n+1}$ at time t_{n+1} and $\mathcal{O}_e$, one can numerically solve the propagated auxiliary variable $\mathbf{h}'_n$ at time t_n as

$$\mathbf{h}'_n = \mathbf{h}_{n+1} + \int_{\tau=t_{n+1}}^{t_n} \mathcal{O}_e(\mathbf{h}(\tau), \tau; \boldsymbol{\theta}_o) d\tau, \tag{10}$$

In most cases, the above integration is implemented using a numerical ODE solver, e.g., Euler and Runge-Kutta solvers:

$$\mathbf{h}'_n = \mathcal{S}(\mathcal{O}_e, \mathbf{h}_{n+1}, (t_{n+1}, t_n)), \tag{11}$$

where $\mathcal{S}$ represents a specific ODE solver.

By comparing (10) and (7), it is clear that the ODE-based latent propagation from t_{n+1} to t_n can be more representative for different modes in the latent space and different time intervals Δt_n, while the decay-based propagation is only exponentially monotonic from the starting point $\mathbf{h}_{n+1}$.

By iterating the latent propagation step (10) and the RNN unit (8), the input sequence $\{\mathbf{b}_n\}_{n=0}^N$ can be represented by the latent variable $\mathbf{h}_0$ at time t_0, as shown in the left side of Fig. 3. We then use $\mathbf{h}_0$ of dimension L_h to generate $\mathbf{z}_0$ of dimension L to represent the initial state of the latent trajectory of the input sequence via an approximate posterior. More specifically,

$$q_{\boldsymbol{\theta}_m}(\mathbf{z}_0|\mathbf{h}_0) = q_{\boldsymbol{\theta}_e}(\mathbf{z}_0|\mathbf{b}_0, \cdots, \mathbf{b}_N) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2), \tag{12}$$

where $\boldsymbol{\theta}_e = \{\boldsymbol{\theta}_r, \boldsymbol{\theta}_o, \boldsymbol{\theta}_m\}$ groups all learnable parameters in the encoder, and the mean and standard deviation are mapped from $\mathbf{h}_0$ as

$$\boldsymbol{\mu}, \boldsymbol{\sigma} = \mathcal{M}_e(\mathbf{h}_0; \boldsymbol{\theta}_m), \tag{13}$$

where $\mathcal{M}_e$ is a neural network, e.g., MLP, with learnable parameters $\boldsymbol{\theta}_m$ that converts the last hidden state of the encoder into an output of dimension $2L$ for the mean and standard deviation vectors.

B. Latent Dynamics Learning

Following the variational autoencoder (VAE) [46], we first sample $\mathbf{z}_0 \sim q_{\boldsymbol{\theta}_e}(\mathbf{z}_0|\mathbf{b}_0, \cdots, \mathbf{b}_N)$ according to the approximate posterior in (12) via the reparameterization trick,

$$\mathbf{z}_0 = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{14}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are given in (13) from the encoder output and $\odot$ denotes an element-wise product[3].

Given the sampled initial value in the latent space $\mathbf{z}_0$, we would like to learn a unified continuous-time latent dynamic trajectory $\mathbf{z}(t)$ that can be mapped into the original sampled waveform, i.e., beam SNRs, and the labeled trajectory coordinates. The latent dynamics learning of $\mathbf{z}(t)$ is achieved by using another continuous-time ODE function $\mathcal{O}_d$ modeled by a neural network with parameters $\boldsymbol{\theta}_l$; see the middle right portion of Fig. 3,

$$\begin{aligned} \frac{d\mathbf{z}(t)}{dt} &= \mathcal{O}_d(\mathbf{z}(t), t; \boldsymbol{\theta}_l) \\ \rightarrow \mathbf{z}(t) &= \mathbf{z}_0 + \int_{t_0}^{t} \mathcal{O}_d(\mathbf{z}(t), t; \boldsymbol{\theta}_l) dt, \end{aligned} \tag{15}$$

where $\mathbf{z}_0 = \mathbf{z}(t)|_{t=t_0}$. Consequently, we can resort to the numerical ODE solver to compute the latent variables at the original beam SNR sampling instances $\{t_n\}_{n=0}^N$ and at new (potentially asynchronous) queried time instances $\{t_n^c\}_{n=0}^{N_c}$ for grounding the latent dynamic learning into the physical coordinate space. For the original set of beam SNR sampling instances $\mathcal{T} = \{t_0, t_1, \cdots, t_N\}$, the latent variables at t_n can be numerically solved as

$$\begin{aligned} \mathbf{z}_n = \mathbf{z}(t)|_{t=t_n} &= \mathbf{z}_0 + \int_{t_0}^{t_n} \mathcal{O}_d(\mathbf{z}(t), t; \boldsymbol{\theta}_l) dt \\ &= \mathcal{S}(\mathcal{O}_d, \mathbf{z}_0, (t_0, t_n); \boldsymbol{\theta}_l), \quad n = 1, \cdots, N, \end{aligned} \tag{16}$$

where $\mathcal{S}$ is a specific numerical ODE solver like the one used in the encoder. For the new, potentially asynchronous time instances $\mathcal{T}_c = \{t_0^c, t_1^c, \cdots, t_{N_c}^c\}$, we have

$$\begin{aligned} \mathbf{z}_n^c = \mathbf{z}(t)|_{t=t_n^c} &= \mathbf{z}_0 + \int_{t_0^c}^{t_n^c} \mathcal{O}_d(\mathbf{z}(t), t; \boldsymbol{\theta}_l) dt \\ &= \mathcal{S}(\mathcal{O}_d, \mathbf{z}_0, (t_0^c, t_n^c); \boldsymbol{\theta}_l), \quad n = 1, \cdots, N_c, \end{aligned} \tag{17}$$

where $t_0 = t_0^c$ as the same initial time instance for both encoders. Note that (16) and (17) share the same ODE function parameters $\boldsymbol{\theta}_l$ and the same initial condition $\mathbf{z}_0$ for a unified latent dynamic representation.

C. Dual Decoder

For the decoder, we propose to use a mixture of two decoding branches: one is for unsupervised waveform reconstruction $\mathcal{M}_w$ and the other for supervised trajectory estimation $\mathcal{M}_c$. Our presented framework exploits the fact that, once $\mathcal{O}_d(\mathbf{z}(t), t; \boldsymbol{\theta}_l)$ is learned, the latent dynamic learning can be used to query any arbitrary time instant (i.e., t_n for waveform reconstruction and t_n^c for supervised trajectory estimation) within the trained time horizon. Then, the unified latent dynamic representation $\mathcal{O}_d(\mathbf{z}(t), t; \boldsymbol{\theta}_l)$ is enforced not only to recover the waveform in an unsupervised fashion but also to be grounded to a two-dimensional trajectory dynamics via labelled object coordinates at asynchronous time instances. Specifically, the waveform decoder takes the computed latent variables at t_n as input and output the sequence of original beam SNRs at t_n,

$$\begin{aligned} \hat{\mathbf{b}}_n &= \mathcal{M}_b(\mathbf{z}_n; \boldsymbol{\theta}_b), \quad n = 1, 2, \cdots, N, \\ &= \mathbf{W}_{b_2} \text{ReLU}(\mathbf{W}_{b_1} \mathbf{z}_n + \mathbf{v}_{b_1}) + \mathbf{v}_{b_2}, \end{aligned} \tag{18}$$

where $\boldsymbol{\theta}_b = \{\mathbf{W}_{b_1/b_2}, \mathbf{v}_{b_1/b_2}\}$ groups the weight matrices and bias terms and $\text{ReLU}(\mathbf{x}) = \max(0, \mathbf{x})$ is the rectified linear unit (ReLU) activation function. Similarly, the coordinate decoder

$$\hat{\mathbf{c}}_n = \mathcal{M}_c(\mathbf{z}_n^c; \boldsymbol{\theta}_c), \quad n = 1, 2, \cdots, N_c,$$

[3]An alternative reparameterization is to generate $\log \boldsymbol{\sigma}^2$, the logarithm of the variance, at the output of MLP $\mathcal{M}_e$ and sample the initial condition as $\mathbf{z}_0 = \boldsymbol{\mu} + e^{0.5 \log \boldsymbol{\sigma}^2} \odot \boldsymbol{\epsilon}$.

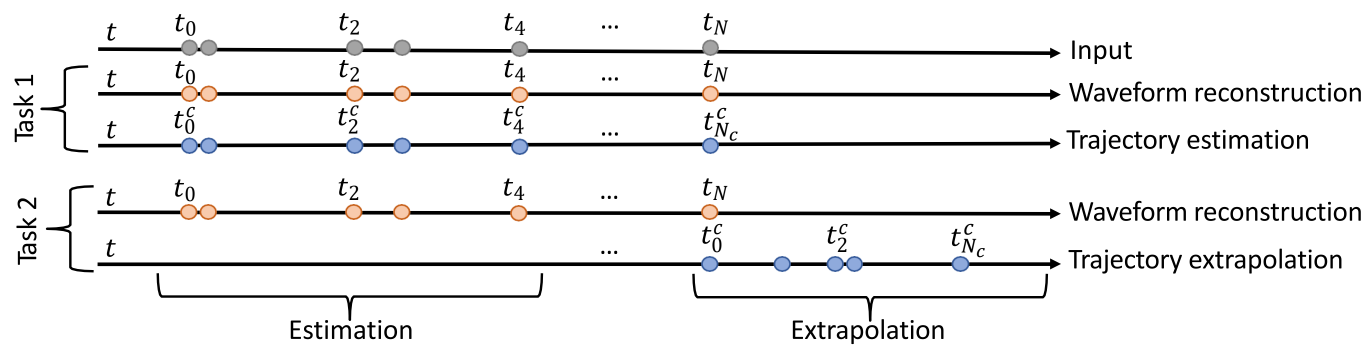


Fig. 4: Illustration of different supervision tasks (trajectory estimation and trajectory extrapolation).

$$= \mathbf{W}_{c_2}\mathrm{ReLU}(\mathbf{W}_{c_1}\mathbf{z}_n^c + \mathbf{v}_{c_1}) + \mathbf{v}_{c_2}, \tag{19}$$

where $\boldsymbol{\theta}_c = \{\mathbf{W}_{c_1/c_2}, \mathbf{v}_{c_1/c_2}\}$ groups the weight matrices and bias terms. For both decoders, the parameters $\boldsymbol{\theta}_w$ and $\boldsymbol{\theta}_c$ are shared over time instances t_n and, respectively, t_n^c.

Combining (18) and (16), we can directly decode $\hat{\mathbf{b}}_n$ from the sampled latent variable $\mathbf{z}_0$ as

$$\begin{aligned}\hat{\mathbf{b}}_n &= \mathcal{M}_w(\mathcal{S}(\mathcal{O}_d, \mathbf{z}_0, (t_0, t_n); \boldsymbol{\theta}_l); \boldsymbol{\theta}_w)\\ &= \mathcal{B}(\mathbf{z}_0, t_0, t_n; \boldsymbol{\theta}_{db}),\end{aligned} \tag{20}$$

where $\mathcal{B}$ stands for the integrated waveform decoder (the latent dynamic learning and waveform decoder) for beam SNRs and $\boldsymbol{\theta}_{db} = \{\boldsymbol{\theta}_l, \boldsymbol{\theta}_b\}$ groups all trainable parameters to decode $\hat{\mathbf{b}}_n$. Similarly, we have

$$\begin{aligned}\hat{\mathbf{c}}_n &= \mathcal{M}_c(\mathcal{S}(\mathcal{O}_d, \mathbf{z}_0, (t_0, t_n^c); \boldsymbol{\theta}_l); \boldsymbol{\theta}_c)\\ &= \mathcal{C}(\mathbf{z}_0, t_0, t_n^c; \boldsymbol{\theta}_{dc}),\end{aligned} \tag{21}$$

where $\mathcal{C}$ stands for the integrated coordinate decoder (the latent dynamic learning and coordinate decoder) for **c**oordinates and $\boldsymbol{\theta}_{dc} = \{\boldsymbol{\theta}_l, \boldsymbol{\theta}_c\}$. In this way, we are imposing strong supervision for every time instant in the latent trajectory by using the real trajectory and the variation of the signal as conditions to modify the learning dynamics of the latent trajectory. This leads to an enhancement in learning the continuous dynamics of the trajectory from the latent space.

For the supervised coordinate decoder, we can have two levels of supervision intensity,

- **Regular Supervision**: the coordinate decoder maps the continuous-time latent dynamics onto the exact same time instants of the input beam SNR sequences. i.e., $t_n = t_n^c$ and $N = N_c$.
- **Dense Supervision**: the coordinate decoder maps the latent dynamics onto more densely queried time instances with respect to those instances in the waveform decoder, i.e., $t_0 = t_0^c, t_N = t_{N_c}^c$, and $N \ll N_c$. In other words, we enforce the latent dynamic learning to be consistent over more queried time instances within the same time window.

In terms of the time horizon of latent dynamics learning, we can have two supervision tasks,

- **Trajectory Estimation**: In the trajectory estimation task, we condition the encoder on the subset of points $(t_0, ..., t_N)$ and reconstruct the same set of points in the same time interval in the decoder side for both the waveform reconstruction and the trajectory regression, i.e., $(t_0, ..., t_N) = (t_0^c, ..., t_{N_c}^c)$ and $N = N_c$. This is illustrated in Task 1 of Fig. 4.
- **Trajectory Extrapolation**: For the trajectory extrapolation, we predict the object trajectory over the time window ΔT_w immediately after the time window ΔT_w of the input beam SNR sequences; see Task 2 of Fig. 4. In other words, we extend the time horizon of the latent dynamics twice as that of Task 1. Particularly, we have $(t_0, ..., t_N) \neq (t_0^c, ..., t_{N_c}^c)$ and $t_N = t_0^c$.

D. Customized Loss Function

In the following, we propose a customized loss function that is modified from the VAE-based loss function for the dual decoder architecture. Grouping the input sequence of beam SNRs $\mathbf{b} = \{\mathbf{b}_n\}_{n=0}^N$ (and similarly $\mathbf{c} = \{\mathbf{c}_n\}_{n=0}^{N_c}$), we can simplify the approximate posterior distribution of (12) as $q_{\boldsymbol{\theta}_e}(\mathbf{z}_0|\mathbf{b})$. The original loss function is to maximize the marginal likelihood function $p(\mathbf{b})$. However, due to its intractability, one can instead maximize an evidence lower bound (ELBO) [46] modified for the dual decoder

$$\begin{aligned}\mathrm{ELBO} =& \mathbb{E}[\log p(\mathbf{b}|\mathbf{z}_0)] + \lambda \mathbb{E}[\log p(\mathbf{c}|\mathbf{z}_0)]\\ &- \mathrm{KL}(q_{\boldsymbol{\theta}_e}(\mathbf{z}_0|\mathbf{b})||p(\mathbf{z}_0)),\\ \overset{(a)}{\approx}& \frac{1}{M}\sum_{m=1}^{M} \log p(\mathbf{b}|\mathbf{z}_0^{(m)}) + \frac{\lambda}{M}\sum_{m=1}^{M} \log p(\mathbf{c}|\mathbf{z}_0^{(m)})\\ &+ 0.5\sum_{l}^{L}\left(1 + \log\sigma_l^2 - \mu_l^2 - \sigma_l^2\right),\end{aligned} \tag{22}$$

where λ is the regularization parameter on the coordinate likelihood function in addition to the original KL divergence, $p(\mathbf{z}_0)$ is the prior of $\mathbf{z}_0$ which is assumed to be [46]

$$p(\mathbf{z}_0) = \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{23}$$

the expectation is with respect to the posterior distribution $q_{\boldsymbol{\theta}_e}(\mathbf{z}_0|\mathbf{b})$ and (a) holds since we replaces the posterior mean by the its sample mean over M samples $\mathbf{z}_0^{(m)}$ according to (12) and due to the fact that the KL divergence can be analytically derived between the Gaussian prior $p(\mathbf{z}_0)$ and the approximate

Gaussian posterior distribution in (12). In addition, μ_l and σ_l denote the posterior mean and, respectively, standard deviation of the l-th element of $\mathbf{z}_0$ given the input sequence $\mathbf{b}$.

To compute the likelihood functions $\log p(\mathbf{c}|\mathbf{z}_0)$ for the coordinate decoder (similarly $\log p(\mathbf{b}|\mathbf{z}_0)$ for the waveform decoder), we invoke an independent assumption over the elements of the decoder output $\mathbf{c} = \{\mathbf{c}_0, \mathbf{c}_1, \cdots, \mathbf{c}_N\}$. This is similar to the pixel-wise independence at the decoder output used in the VAE. This implies that

$$\begin{aligned}\log p(\mathbf{c}|\mathbf{z}_0^{(m)}) &= \log p(\mathbf{c}_0, \cdots, \mathbf{c}_N|\mathbf{z}_0^{(m)}) \\ &= \sum_n \log p(\mathbf{c}_n|\mathbf{z}_0^{(m)}) \\ &= \sum_n \sum_d \log p(c_{n,d}|\mathbf{z}_0^{(m)}), \end{aligned} \tag{24}$$

where $d = \{1, 2\}$ denotes the 2D coordinate and $c_{n,d}$ denotes the x- and y-coordinate at time n. The element-wise likelihood function $p(c_{n,d}|\mathbf{z}_0)$ of the (x- or y-) coordinate follows a Laplace distribution[4] as

$$p(c_{n,d}|\mathbf{z}_0^{(m)}) = \frac{1}{2a_c} e^{-\frac{\left|c_{n,d} - \mathcal{C}_d(\mathbf{z}_0^{(m)}, t_0, t_n^c; \boldsymbol{\theta}_{dc})\right|}{a_c}}, \tag{25}$$

where a_c is a scaling parameter, and $\mathcal{C}_d(\mathbf{z}_0^{(m)}, t_0, t_n^c; \boldsymbol{\theta}_{dc})$ is the d-th element of the output at the integrated coordinate decoder of (21) at time t_n^c. Then, it is easy to see that

$$\mathbb{E}[\log p(\mathbf{c}|\mathbf{z}_0^{(m)})] \propto -\frac{1}{a_c} \sum_{n=0}^{N_c} \left\| \mathbf{c}_n - \mathcal{C}(\mathbf{z}_0^{(m)}, t_0, t_n^c; \boldsymbol{\theta}_{dc}) \right\|_1, \tag{26}$$

where $\|\cdot\|_1$ denotes the ℓ_1 norm. Similarly, we have

$$\mathbb{E}[\log p(\mathbf{b}|\mathbf{z}_0^{(m)})] \propto -\frac{1}{a_b} \sum_{n=0}^{N} \left\| \mathbf{b}_n - \mathcal{B}(\mathbf{z}_0^{(m)}, t_0, t_n; \boldsymbol{\theta}_{db}) \right\|_1, \tag{27}$$

where a_b is a scaling parameter for the Laplace distribution of the beam SNR $\mathbf{b}_n$. As a result, considering $a_b = a_c = 1$, we aim to minimize the negative customized ELBO of (22) for the dual decoder as

$$\begin{aligned}-\text{ELBO} \propto & \sum_{n=0}^{N} \|\hat{\mathbf{b}}_n - \mathbf{b}_n\|_1 + \lambda \sum_{n=0}^{N_c} \|\hat{\mathbf{c}}_n - \mathbf{c}_n\|_1 \\ & - 0.5 \sum_{l}^{L} \left(1 + \log \sigma_l^2 - \mu_l^2 - \sigma_l^2\right). \end{aligned} \tag{28}$$

In implementation, the following cost function is used

$$\begin{aligned}\mathcal{L} &= \sum_{n=0}^{N} \|\hat{\mathbf{b}}_n - \mathbf{b}_n\|_1 + \lambda \sum_{n=0}^{N_c} \|\hat{\mathbf{c}}_n - \mathbf{c}_n\|_1 + (\eta - \lambda)\|\hat{\mathbf{c}}_0 - \mathbf{c}_0\|_1 \\ &= \sum_{n=0}^{N} \|\hat{\mathbf{b}}_n - \mathbf{b}_n\|_1 + \lambda \sum_{n=1}^{N_c} \|\hat{\mathbf{c}}_n - \mathbf{c}_n\|_1 + \eta \|\hat{\mathbf{c}}_0 - \mathbf{c}_0\|_1 \end{aligned} \tag{29}$$

where we replace the explicit KL divergence term in (28) by an implicit regularization term of $(\eta - \lambda)\|\hat{\mathbf{c}}_0 - \mathbf{c}_0\|_1$ with $\eta > \lambda$. Our motivation is to amplify the significance of the initial latent dynamic state $\mathbf{z}_0$ and its mapping to the physical coordinate space $\mathbf{c}_0$ in the loss function, rather than through the KL divergence term. The two hyperparameters λ and η play the tradeoff roles from the physical coordinate reconstruction $\sum_{n=1}^{N_c} \|\hat{\mathbf{c}}_n - \mathbf{c}_n\|_1$ and, respectively, the implicit regularization term $\sum_{n=0}^{N} \|\hat{\mathbf{c}}_0 - \mathbf{c}_0\|_1$ to the waveform (beam SNR) reconstruction error of $\sum_{n=0}^{N} \|\hat{\mathbf{b}}_n - \mathbf{b}_n\|_1$.

E. Complexity Analysis

We describe the time complexity of the proposed method by following Fig. 3. Assuming the time complexity for the numerical ODE solvers is k at both encoder and decoder sides, the time complexity for the forward pass of the encoder (at the left side of Fig. 3) can be approximated as $\mathcal{O}(N(k + L_h^2 + N_b L_h))$, where N is the number of time steps in the input sequence of beam SNRs and $(L_h^2 + N_b L_h)$ is from matrix products used in the LSTM update step in Appendix A with N_b is the input dimension and L_h the hidden state dimension.

Then for the initial latent state z_0 of (13) using an MLP with one hidden layer of dimension M_z, the time complexity is approximately $\mathcal{O}(M_z(L_h + 2L))$, where $2L$ is the MLP output dimension. Following that, for the neural dynamic learning block on the middle right portion of Fig. 3, the time complexity is approximately $\mathcal{O}(kN)$ and $\mathcal{O}(kN_c)$, respectively, for the two sets of queried time instances $\{t_n\}_{n=0}^{N}$ and $\{t_n^c\}_{n=0}^{N_c}$.

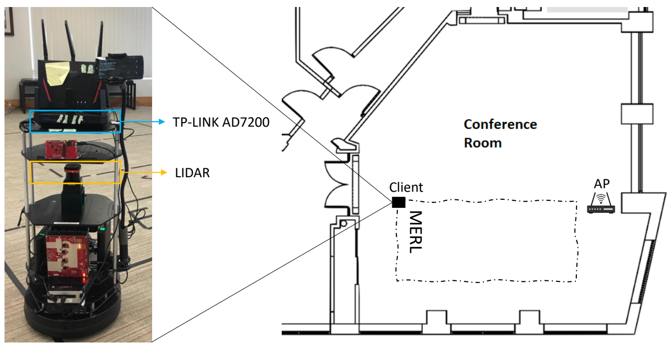
Finally, for the waveform decoder $\mathcal{M}_b$ of (18) using an MLP with one hidden layer of dimension M_b, the time complexity is $\mathcal{O}(N(LM_b + 2M_b + M_b N_b + N_b)) \approx \mathcal{O}(NM_b(L + N_b))$, where L is the latent dimension in the decoder and N_b is the output dimension of beam SNRs. Similarly for the coordinate decoders M_c of (19) using an MLP with one hidden layer of dimension M_c, its complexity is approximated as $\mathcal{O}(N_c M_c(L + 2))$ where 2 is the dimension of the coordinate output. With all combined, the overall time complexity of the proposed method is approximately

$$\begin{aligned}&\mathcal{O}(k(N + N_c) + M_z(L_h + 2L) + N_c M_c(L + 2) \\ &\quad + N(L_h^2 + N_b L_h + M_b(L + N_b))). \end{aligned} \tag{30}$$

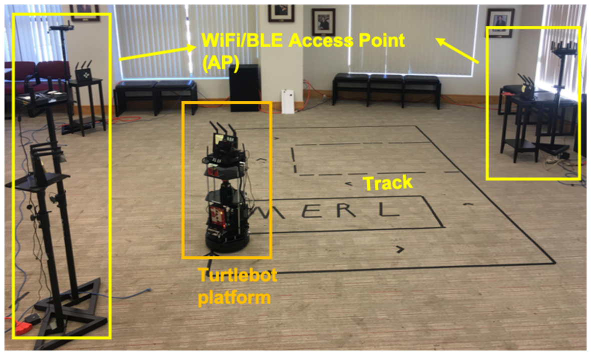
IV. mmWave Wi-Fi Testbed and Data Collection

To evaluate the proposed DDND framework and baseline comparison, we built a mmWave Wi-Fi testbed consisting of multiple commercial-of-the-shelf (COTS) 802.11ad devices to collect real-world mmWave Wi-Fi beam SNR data. Particularly, we used a pair of TP-Link AD7200 routers to acting as an AP fixed at a standing post and a mobile user; see the TurtleBot photo in the left plot of Fig. 5 (a). The TurtleBot is equipped with a 2D scanning LiDAR sensor and a wheel encoder to map the environment and localize itself in a 2D floorplan with an accuracy of less than 1 cm. The 2D localization results are considered as the labels for training and groundtruth for test. The AP sequentially underwent beam training, and during this process, beam SNR measurements were collected at the AP and transmitted to a workstation via

[4]This assumption is adopted as we prefer the mean absolute error as the loss function. This can lead to more robust localization performance in the case of outliers or unseen locations, compared with the mean squared error that can be derived from a standard Gaussian assumption on the coordinate.

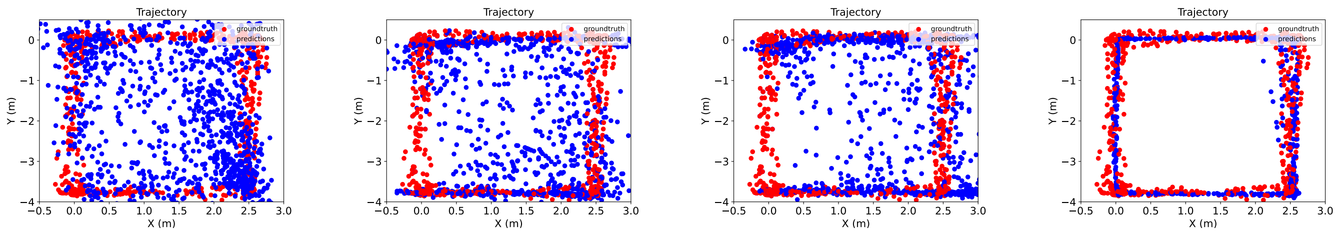


(a) A TurtleBot mobile user, trajectory configuration, and floorplan.



(b) A photo of data collection campaign.

Fig. 5: An in-house mmWave Wi-Fi testbed with TP-Link AD7200 routers and a TurtleBot as a mobile user and a campaign of multi-day data collection in a conference room.



(a) (b) (c) (d)

Fig. 6: Visualization of trajectory estimation over selected test sequences: (a) SVR (b) RNN-Decay (c) RNN-Δ (d) DDND-DS

Ethernet cables. For a given pair of transmitting and receiving beampatterns, the corresponding beam SNR can be defined as

$$b_n = \text{BeamSNR}_n = \frac{1}{\sigma^2} \sum_{i=1}^{I} \gamma_n(\phi_i)\zeta_n(\psi_i)P_i, \quad (31)$$

where $n = 1, \cdots, N_b$ is the index of scanned beampatterns, I is the total number of paths, ϕ_i and ψ_i are the transmitting and receiving azimuth/elevation angles for the i-th path, respectively, P_i is the signal power at the i-th path, $\gamma_n(\phi_i)$ and $\zeta_n(\psi_i)$ are the transmitting and receiving beampattern gains at the i-th path for the n-th beampattern, respectively, and σ^2 is the noise variance. Due to antenna housing, the beampatterns can be highly irregular and contain significant sidelobes; see Fig. 5 of [45] for measured beampatterns in an anechoic chamber.

To access the raw beam SNR measurements from the COTS routers, we followed the methods described in [15] and utilized an open-source software package introduced in [47]. Specifically, we employed the Nexmon firmware patching framework [48], which allows the development of binary firmware extensions using the C programming language. By analyzing the patterns of IEEE 802.11ad beam training frames stored in the chip's memory, we were able to identify the firmware components responsible for handling these frames. Consequently, we extracted the beam SNR measurements from the corresponding memory addresses. For the TP-Link AD7200 router, an analog phased array of 32 antenna elements is used to sequentially scan over $N_b = 36$ predefined directional beampatterns for one air time for a given responder in Fig. 2.

For data collection, we place the pair of TP-Link AD7200 routers in a corner conference room as shown in Fig. 5 (b). The AP router is fixed at a standing post during the data collection, while the TurtleBot with the other router moves along a rectangular trajectory marked in a dash-dotted line in Fig. 5 (a). It is noted that the TurtleBot experienced different motion patterns such as acceleration, deceleration, constant-velocity motion, and rotation along the rectangular trajectory. To be specific, we need to slow down the Turtlebot when it moves to the turning point, rotate it about $90°$, and accelerate it towards the next turning point. In total, we collected two separate data sessions over two separate days with each data session lasting for multiple consecutive hours, resulting in:

- Day 1: $11.5K$ samples of beam SNR
- Day 2: $10K$ samples of beam SNR

with each beam SNR vector of dimension $N_b = 36$. The frame rate of the coordinate labels is 10 Hz and about $1 \sim 2$ Hz on average for the beam SNR measurements.

To preprocess the raw beam SNRs, we employ a sequence time window of $\Delta T_w = [2, 5, 8]$ seconds. This allows us to group the dataset into non-overlapping sequences, accommodating the irregular sampling of the beam SNR. The sequences are divided into training (0.8), validation (0.1), and test (0.1) sets, respectively. We standardize each entry $b_n, n = 1, \cdots, 36$ of the beam SNR by subtracting its mean and normalizing it with the corresponding standard deviation. Furthermore, we normalize the time vectors $\{t_n\}_{n=0}^{N}$ and $\{t_n^c\}_{n=0}^{N_c}$ within each ΔT_w seconds to the range $[0, 1]$.

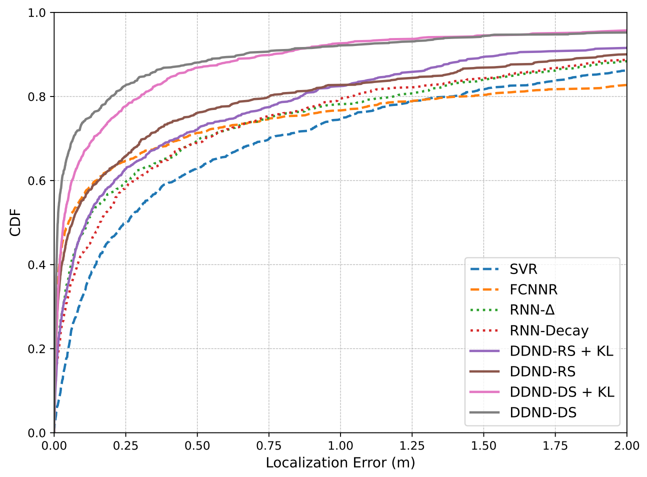


Fig. 7: Cumulative distribution function (CDF) of localization errors.

TABLE I: Localization errors (m) on mmWave Wi-Fi dataset

	Mean	Median	CDF@0.9
SVR	0.82	0.25	2.64
FCNNR	0.92	0.05	3.04
RNN-Decay	0.74	0.16	2.19
RNN-Δ	0.76	0.11	2.36
DDND-RS + KL (ours)	0.58	0.11	1.57
DDND-RS (ours)	0.52	0.03	1.98
DDND-DS + KL (ours)	0.30	0.03	0.76
DDND-DS (ours)	**0.28**	**0.01**	**0.66**

V. Performance Evaluation

In the following, we evaluate the localization performance using the above datasets. We first quantify the performance against a list of baseline methods including both frame-based and sequence-based methods. Later on, we aim to provide a comprehensive ablation study of the proposed method against sequence length, supervision intensity (regular $N = N_c$ versus dense ($N < N_c$)) at the coordinate decoder, day-to-day performance generalization, and varying tasks (estimation versus extrapolation). Finally, we provide an interpretation of the learned latent dynamics.

A. Model Training

Our implementation of the proposed method is built using the PyTorch framework. The model incorporates a hidden state dimension of $L_h = 20$ and a latent dimension of $L = 20$. For the encoder and decoder, we utilize the 5th-order Runge-Kutta ODE solver. During training, we employ the Adamax optimizer with a learning rate of 0.01 and no weight decay. To form mini-batches, the model is trained using a batch size of 32 sequences. The proposed loss function performs an adaptive weighting strategy, to ensure stability during the training with respect to the amount of time samples. In this way, $\lambda = \frac{(N+1)}{\beta(N_c+1)}$, and $\eta = \alpha\lambda(N_c + 1)$ being $\alpha = 0.5$ and $\beta = 0.1$. For an exemplary case of $N = N_c = 4$, $\lambda = 10$ and $\eta = 25$.

B. Baseline Comparison

We here provide a performance comparison for the object trajectory estimation. We evaluate the following baseline methods and variants of our framework:

- Frame-based methods:
  1) Support Vector Regressor (SVR)
  2) Fully Connected Neural Network Regressor (FC-NNR)
- Sequence-based methods (Section II)
  1) RNN-Δ
  2) RNN-Decay
- Our DDND framework with variants:
  1) DDND-Regular Supervision + KL (DDND-RS + KL) of (28)
  2) DDND-Regular Supervision (DDND-RS) of (29)
  3) DDND-Dense Supervision + KL (DDND-DS + KL) of (28)
  4) DDND-Dense Supervision (DDND-DS) of (29)

For this baseline comparison, we set $\Delta T_w = 8$ seconds for all considered sequence-based methods.

Fig. 6 illustrates the estimated trajectories over test sequences for the selected methods. In the frame-based SVR method (Fig. 6 (a)), the coordinate estimates appear scattered within the square trajectory. However, there is a slight improvement observed in the sequence-based methods. In Fig. 6 (b) and (c), which correspond to the RNN expdecay and RNN Δt methods, respectively, more trajectory estimates are pushing towards the square trajectory. A noticeable difference can be observed when comparing Fig. 6 (d) to (a)-(c). It is evident that the proposed DDND-DS method is capable of learning the underlying dynamics, leading to more clustered trajectory estimates around the true square trajectory. The DDND-DS method demonstrates superior performance in capturing the trajectory patterns, indicating its effectiveness in trajectory estimation tasks.

Fig. 7 presents a comparison of the baseline methods with the proposed solution in terms of the Cumulative Distribution Function (CDF) of root mean squared errors (RMSEs). The results highlight the impact of proper learning and modeling of latent dynamics and trajectory motion from irregularly-sampled Wi-Fi data on improving localization performance compared to frame-based and traditional sequence-based methods. The proposed method and its variants exhibit a notable reduction in large localization errors. These findings underscore the effectiveness of the proposed method in enhancing localization accuracy by capturing the underlying dynamics and motion patterns of the moving object based on irregularly-sampled Wi-Fi data. Table I further quantifies the performance.

C. Impact of Sequence Length

In the following, we evaluate the impact of the sequence length ΔT_w on the trajectory estimation performance. Specifically, we convert the original training data into three non-overlapping sequence datasets with different time window sizes $\Delta T_w = [2, 5, 8]$ seconds. We use the three sequence

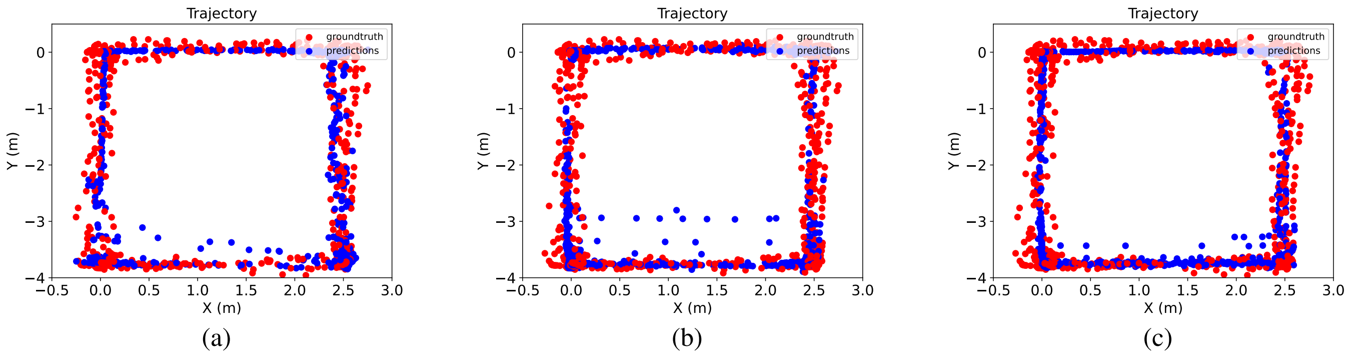


Fig. 8: Visualization of trajectory estimation over selected test data: (a) $\Delta T_w = 2$s, (b) $\Delta T_w = 5$s, (c) $\Delta T_w = 8$s.

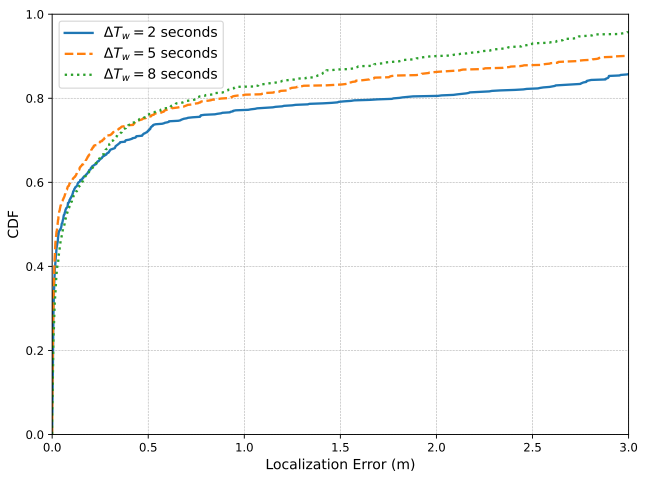


Fig. 9: CDF curves of localization errors for different sequence lengths ΔT_w.

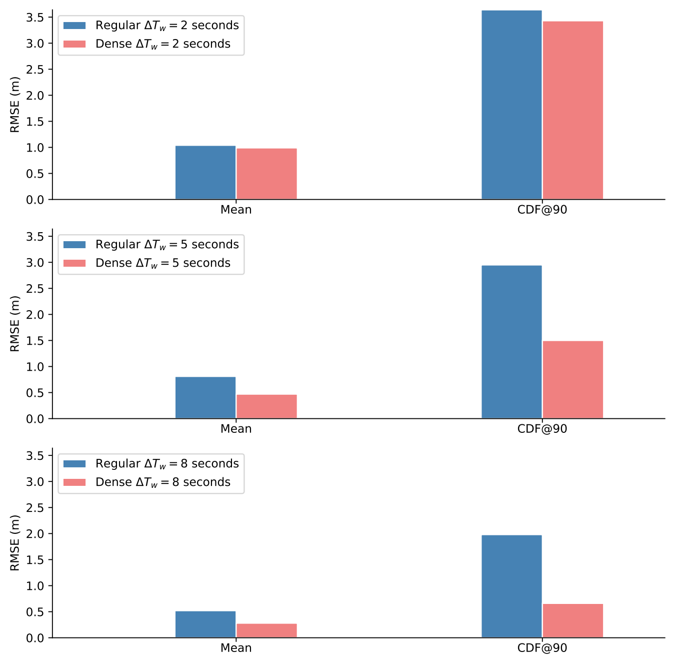


Fig. 10: Regular vs dense supervision: quantitative performance comparison for different sequence lengths ΔT_w.

datasets to train three separate DDND models and test these trained models on the test dataset sequenced according to the corresponding time window size.

Fig. 8 shows a visual representation of the estimated trajectories over the test data for different configurations. It is seen that, longer time windows $\Delta T_w = 8$s lead to the inclusion of more coordinates $\mathbf{c}_n$ in a sequence and a stronger supervision in the loss function of (29). With a larger time window, our DDND framework shows a stronger capability of modeling and learning latent dynamics of various motion patterns (e.g., turning, acceleration, deceleration) from both the beam SNR measurements and the coordinate supervision. This is seen from Fig. 8 as the predicted coordinates (blue dots) are closer to the ground truth (red dots) when the time window size grows from 2s to 8s.

Fig. 9 show the CDF curves of the localization errors in terms of RMSE for different sequence lengths. It shows that although they exhibit similar performance in terms of median values or up to the 70-th percentile, the longer sequences are more suitable for learning the dynamics and present a lower variance in the results. Providing a longer sequence, allows learning a more dense supervision of the beam SNR variations translating into a better latent trajectory learning. Concretely, the sequence length of $\Delta T_w = 8$ seconds gives the best performance. Table II further lists the quantitative localization performance for different sequence lengths ΔT_w.

In summary, the presented results show the clear advantage of long-sequence learning for the trajectory estimation task. When using longer sequences, more information can be obtained leading to more complex dynamics that can be captured and learned in the latent trajectory. This enables the model to better understand and predict future dynamics based on

TABLE II: Localization errors (m) for different sequence lengths ΔT_w.

	Mean	Median	CDF@0.9
$\Delta T_w = 2$ sec	1.04	0.05	3.64
$\Delta T_w = 5$ sec	0.81	0.03	2.95
$\Delta T_w = 8$ sec	**0.52**	**0.03**	**1.98**

past information and translate into a significant increase in performance.

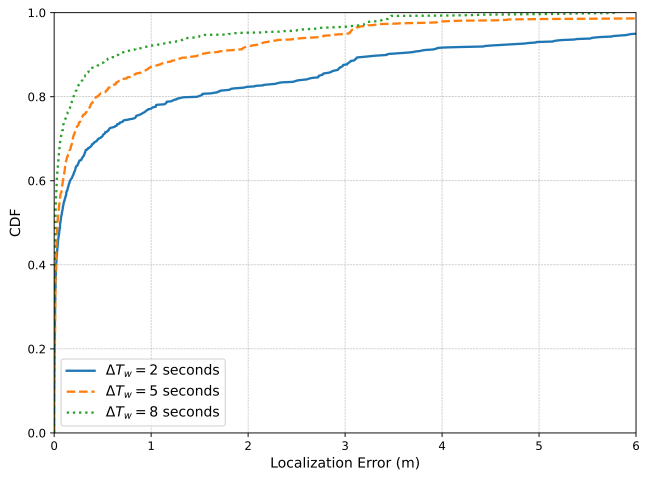


Fig. 11: CDF curves of localization errors with dense supervision training.

D. *Comparison between Regular and Dense Supervision*

We leverage the flexibility of the design of our framework to condition the learning dynamics in the latent space and we evaluate the trajectory estimation. Due to the intermittent sampling of the beam SNR measurements, result in irregular samples within a ΔT_w. However, we can perform stronger supervision to enhance the latent ODE dynamics, as we have the flexibility of conditioning the learning with whatever physical information we have. In our setup, the LiDAR has a fixed sampling frequency of 10 Hz that gives us way more coordinate points within a ΔT_w. We perform this study again on $\Delta T_w = [2, 5, 8]$ seconds to assess the dense supervision enhancement with respect to the sequence length.

Fig. 10 shows the localization errors for regular and dense supervision training. The results show when doing dense supervision the errors are decreased, incrementing the difference the longer the sequences due to the enhanced learning. In this way, we can see how dense supervision is beneficial. We further quantify this improvement by looking at Fig. 11. It shows the enhancement in sequence length is bigger than in the previous case. This further justifies the fact that having a denser supervision is not only beneficial in terms of sequence length but in terms of amount of coordinates due to the LIDAR sampling rate.

Fundamentally, dense supervision provides more accurate learning. Having more points within the sequence can provide a more accurate and detailed picture of the system dynamics. The model can thus learn more accurately from the temporal dependencies, patterns, and nuances that might not be apparent or might be lost with fewer points. Also, more densely supervised sequences can enable the model to handle more complex systems and patterns.

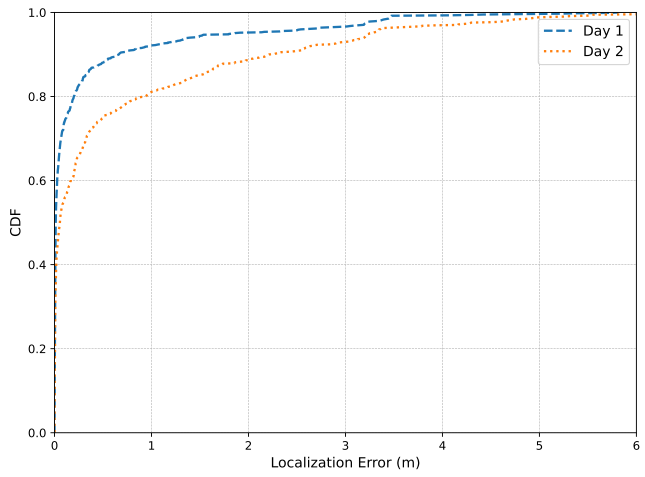


Fig. 12: CDF curves of localization errors over separate days.

E. *Day-to-Day Generalization*

We assess the generalization capabilities of the dense-trained model with a sequence length of $\Delta T_w = 8$ seconds by evaluating its performance on a different day of data collection. Specifically, we aim to test the model's ability to make accurate predictions when exposed to a new set of data collected on a different day. This ablation study is considered as we expect the beam SNR measurements experience a certain level of fluctuation from one day to another even under the same room environment due to small-scale and large-scale channel fading and channel instability. Such measurement fluctuations from one day to another were observed for RSSI and CSI in [13], [14], [49]. Therefore, our day-to-day generalization study is designed to test the model's generalization capability under such measurement fluctuation.

To accomplish this, we maintain the same model that was trained on the original dataset and evaluate its predictive performance on a new day's data. This allows us to analyze how well the model generalizes to unseen instances and adapts to changes in the environment. By examining the model's performance in this context, we can assess its robustness and effectiveness in real-world scenarios beyond the training data.

Fig. 12 shows the CDF of localization error between the testing on the same day (day 1) and testing on a different day (day 2). Table IV further quantifies this performance. By looking at the results, we can see an acceptable performance regardless of the change in the environment. Also, if we

TABLE III: Localization errors (m) for different ΔT_w seconds with dense supervision training.

	Mean	Median	CDF@0.9
Regular $\Delta T_w = 2$ sec	1.04	0.05	3.64
Regular $\Delta T_w = 5$ sec	0.81	0.03	2.95
Regular $\Delta T_w = 8$ sec	0.52	0.03	1.98
Dense $\Delta T_w = 2$ sec	0.99	0.06	3.43
Dense $\Delta T_w = 5$ sec	0.47	0.03	1.50
Dense $\Delta T_w = 8$ sec	**0.28**	**0.01**	**0.66**

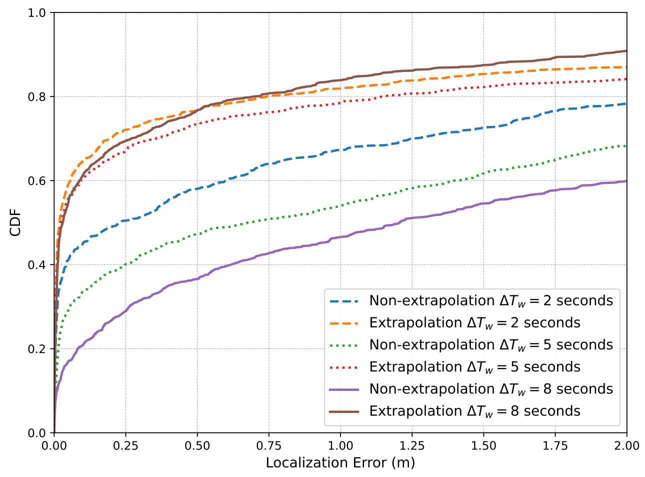


Fig. 13: CDF curves of localization errors between trajectory estimation and extrapolation tasks.

compare to the results in Table III we show the longer sequences are also beneficial for generalization, as even with the worsening in performance it does better than $\Delta T_w = 2$ or $\Delta T_w = 5$ seconds on the same day for the regular cases.

F. Trajectory Generalization

We evaluate the generalization capabilities of our method with a sequence length of $\Delta T_w = 8$ seconds by assessing its performance in a trajectory not seen during training. To this purpose, we train our model by cropping out one of the four corners from the training data: top-left, top-right, bottom-left, and bottom-right, resulting in four different scenarios as shown in Fig. 15. The training set comprises points and trajectories outside of the cropped corner, while the test sets consist of points from the cropped area.

Fig. 16 compares the trajectory estimation performance over baseline methods in Scenario (a) of Fig. 15 where the top-left corner is cropped out from the training dataset. In Table VI, we also report the trajectory estimation errors averaged over the four cropped-out scenarios. From Fig. 16 and Table VI, we have the following observations:

- Frame-based methods: The results underscore the inadequacy of frame-based methods in accurately capturing the system dynamics.
- RNNs: While demonstrating partial understanding of dynamics, RNNs struggle with spatial placement, indicating limitations in generalization.
- Our model with regular supervision: Despite higher localization errors compared to training with the full square, the model exhibits stronger generalization capability at unseen locations, indicating the learned latent dynamics can potentially lead to better generalization capability.

TABLE IV: Localization errors (m) for $\Delta T_w = 8$ seconds with dense supervision training for two different days.

	Mean	Median	CDF@0.9
Day 1	**0.28**	**0.01**	**0.66**
Day 2	0.61	0.05	2.21

TABLE V: Localization errors (m) for extrapolation task.

	Mean	Median	CDF@0.9
Non-extrapolation $\Delta T_w = 2$ seconds	1.36	0.23	4.89
Extrapolation $\Delta T_w = 2$ seconds	0.82	0.02	3.02
Non-extrapolation $\Delta T_w = 5$ seconds	1.51	0.68	3.96
Extrapolation $\Delta T_w = 5$ seconds	0.72	0.02	2.84
Non-extrapolation $\Delta T_w = 8$ seconds	2.15	1.20	6.63
Extrapolation $\Delta T_w = 8$ seconds	**0.62**	**0.03**	**1.89**

Table VI shows the localization error averaged over the four cropped-out scenarios. Although the localization errors are larger than those in Table I, the results clearly show stronger generalization capability of our trained model at unseen locations than the baseline methods.

TABLE VI: Average localization errors (m) over four cropped-out scenarios in Fig. 15.

	Mean	Median	CDF@0.9
SVR	4.64	4.42	6.89
FCNNR	4.92	4.58	8.08
RNN-Decay	3.73	3.95	5.56
RNN-Δ	3.68	3.87	5.48
DDND-RS (ours)	**2.47**	**2.31**	**3.46**

G. Extrapolation Performance

We aim to assess the flexibility of our decoder by evaluating its performance on an extrapolation task. To measure its effectiveness, we compare the performance of the model trained specifically for prediction and test it on extrapolation, against training the model directly for extrapolation. Additionally, we investigate the influence of different sequence lengths, specifically $\Delta T_w = [2, 5, 8]$ seconds, to examine their impact on the extrapolation results. By conducting this evaluation, we can gain insights into the decoder's ability to generalize beyond the observed data and generate accurate extrapolations. The comparison between prediction-trained and extrapolation-trained models, along with the analysis of different sequence lengths, will provide valuable information on the decoder's flexibility and performance in extrapolation tasks.

Fig. 13 shows the CDF of the localization error for the different configurations. It shows when training prediction models and testing them for extrapolation they become worse the longer the sequence length. This shows it is difficult to perform extrapolation if just training in a prediction fashion, especially when the time window increases because the extrapolation task becomes harder. However, when we train for extrapolation, we have a huge improvement in the performance. This highlights the flexibility of our framework that can be trained for different specific tasks, thanks to the querying freedom due to the use of neural ODE on the decoder side. Table V further quantifies this analysis. It shows our extrapolation method for the longest sequence exhibits quite good performance, comparable to dense prediction tasks.

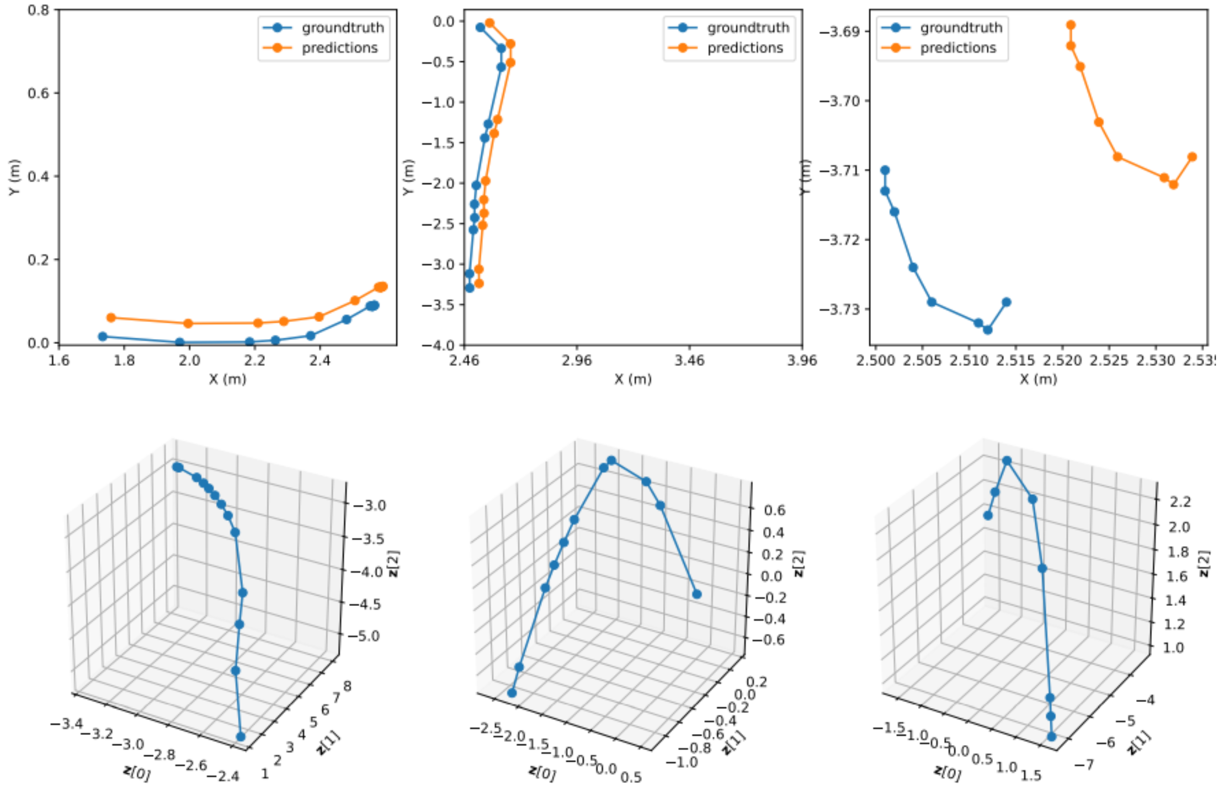
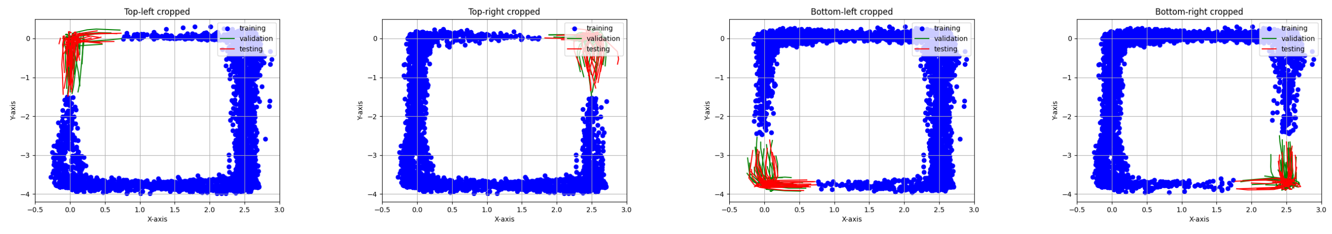
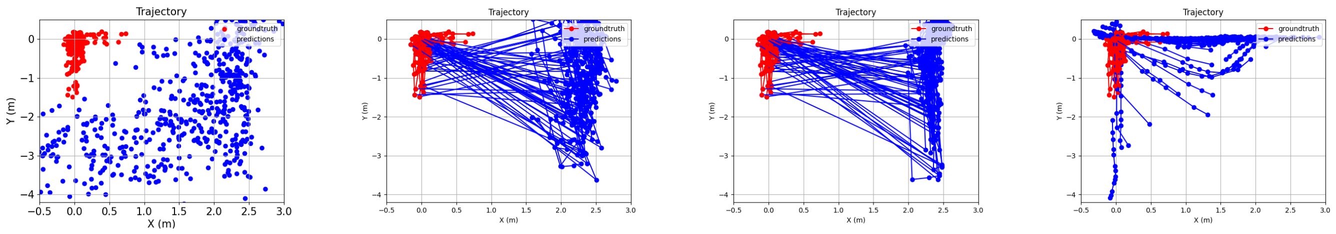


Fig. 14: Visualization of the learned latent dynamics for the proposed method.



(a) (b) (c) (d)

Fig. 15: Visualization of four cropped-out scenarios for trajectory generalization at unseen locations: (a) top-left cropped, (b) top-right cropped, (c) bottom-left cropped, (d) bottom-right cropped. Blue dots denote coordinates used for training; red lines denote trajectories for testing.



(a) (b) (c) (d)

Fig. 16: Comparison of trajectory estimation in Scenario (a) of Fig. 15: (a) SVR (b) RNN-Decay (c) RNN-Δ (d) DDND-RS.

Our framework models the latent dynamics for the input sequence of beam SNRs in a continuous-time fashion. This provides a more robust and accurate representation of the system dynamics, leading to an enhancement in extrapolation. Besides, the presented method continuity is well-suited to capture long-term dependencies in the data. This is particularly beneficial for extrapolation tasks, where understanding long-term trends and dynamics is often crucial.

H. Interpretation of Learned Latent Dynamics

Here we inspect the learned latent trajectories by plotting the first three dimensions of the latent space. In this case, we fix the sequence length as $\Delta T_w = 8$ seconds. The top row of Fig. 14 shows estimated and ground truth trajectories for three sequences from the test data. It is seen that the estimation results follow relatively well the object coordinates in various motion patterns. On the other hand, the bottom row of Fig. 14 shows corresponding learned dynamics in the first three dimensions of the latent space at the decoder side, i.e., $\mathbf{z}(t) = \mathbf{z}_0 + \int_{t_0}^{t} \mathcal{O}_d(\mathbf{z}(t), t; \boldsymbol{\theta}_l) dt$, of (15). Specifically, we sample an initial value for the latent dynamics $\mathbf{z}_0$ for each input sequence, and the learned latent dynamics are computed using the sampled $\mathbf{z}_0$ and the same learned latent ODE function $\mathcal{O}_d(\mathbf{z}(t), t; \boldsymbol{\theta}_l)$ in (15). From the visualized results, it appears that the learned dynamics show sufficient capacity to model distinct patterns using the same continuous latent ODE functions.

VI. Conclusion

This paper tackled the challenging task of localizing objects using intermittently sampled mmWave Wi-Fi beam training measurements. We proposed a novel solution called the dual-decoder neural dynamic framework. Through extensive performance comparisons, we demonstrated notable performance gains against baseline methods and provided a comprehensive study of the proposed method. Our method is directly compatible with upcoming Wi-Fi sensing standards, allowing seamless integration between communication and sensing of Wi-Fi devices in practical deployments.

References

[1] C. J. Vaca-Rubio, P. Wang, T. Koike-Akino, Y. Wang, P. Boufounos, and P. Popovski, "Mmwave Wi-Fi trajectory estimation with continuous-time neural dynamic learning," in *ICASSP 2023*, 2023, pp. 1–5.

[2] M. Youssef and A. Agrawala, "The horus location determination system," *Wirel. Netw.*, vol. 14, no. 3, pp. 357–374, Jun. 2008.

[3] Z.-L. Wu, C.-H. Li, J. K.-Y. Ng, and K. Leung, "Location estimation via support vector regression," *IEEE Transactions on Mobile Computing*, vol. 6, no. 3, pp. 311–321, 2007.

[4] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, and L. M. Ni, "CSI-based indoor localization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1300–1309, 2012.

[5] M. T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy, "Recurrent neural networks for accurate RSSI indoor localization," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 639–10 651, Dec 2019.

[6] X. Wang, L. Gao, S. Mao, and S. Pandey, "DeepFi: Deep learning for indoor fingerprinting using channel state information," in *WCNC*, March 2015, pp. 1666–1671.

[7] ——, "CSI-based fingerprinting for indoor localization: A deep learning approach," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 763–776, Jan 2017.

[8] H. Chen, Y. Zhang, W. Li, X. Tao, and P. Zhang, "ConFi: Convolutional neural networks based indoor Wi-Fi localization using channel state information," *IEEE Access*, vol. 5, pp. 18 066–18 074, 2017.

[9] J. Ding and Y. Wang, "WiFi CSI-based human activity recognition using deep recurrent neural network," *IEEE Access*, vol. 7, pp. 174 257–174 269, 2019.

[10] M. T. Hoang, B. Yuen, K. Ren, X. Dong, T. Lu, R. Westendorp, and K. Reddy, "A CNN-LSTM quantifier for single access point CSI indoor localization," *arXiv preprint arXiv:2005.06394*, 2020.

[11] H. Sun, X. Zhu, Y. Liu, and W. Liu, "WiFi based fingerprinting positioning based on seq2seq model," *Sensors*, vol. 20, no. 13, p. 3767, 2020.

[12] J. Yu, H. M. Saad, and R. M. Buehrer, "Centimeter-level indoor localization using channel state information with recurrent neural networks," in *IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2020, pp. 1317–1323.

[13] H. Xia, P. Wang, T. Koike-Akino, Y. Wang, P. V. Orlik, and Z. Ding, "Adversarial bi-regressor network for domain adaptive regression," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2022, pp. 3608–3614.

[14] J. Yu, P. Wang, T. Koike-Akino, and P. V. Orlik, "Multi-modal recurrent fusion for indoor localization," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2022.

[15] G. Bielsa, J. Palacios, A. Loch, D. Steinmetzer, P. Casari, and J. Widmer, "Indoor localization using commercial off-the-shelf 60 GHz access points," in *IEEE INFOCOM*, 2018, pp. 2384–2392.

[16] M. Pajovic, P. Wang, T. Koike-Akino, H. Sun, and P. V. Orlik, "Fingerprinting-based indoor localization with commercial MMWave WiFi–Part I: RSS and Beam Indices," in *GLOBECOM*, Dec. 2019.

[17] P. Wang, M. Pajovic, T. Koike-Akino, H. Sun, and P. Orlik, "Fingerprinting-based indoor localization with commercial mmwave WiFi–Part II: Spatial beam SNRs," in *GLOBECOM*, Dec 2019.

[18] T. Koike-Akino, P. Wang, M. Pajovic, H. Sun, and P. V. Orlik, "Fingerprinting-based indoor localization with commercial mmwave WiFi: A deep learning approach," *IEEE Access*, vol. 8, pp. 84 879–84 892, 2020.

[19] P. Wang, T. Koike-Akino, and P. V. Orlik, "Fingerprinting-based indoor localization with commercial mmwave WiFi: NLOS propagation," in *GLOBECOM*, December 2020.

[20] J. Yu, P. Wang, T. Koike-Akino, and P. V. Orlik, "Human pose and seat occupancy classification with commercial mmwave WiFi," in *GLOBECOM Workshop on Integrated Sensing and Communication (ISAC)*, December 2020.

[21] J. Yu, P. Wang, T. Koike-Akino, Y. Wang, P. V. Orlik, and R. M. Buehrer, "Multi-band Wi-Fi sensing with matched feature granularity," *IEEE Internet of Things Journal*, pp. 1–1, 2022.

[22] A. Blanco, P. J. Mateo, F. Gringoli, and J. Widmer, "Augmenting mmwave localization accuracy through sub-6 GHz on off-the-shelf devices," in *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*, 2022, pp. 477–490.

[23] T. Koike-Akino, P. Wang, and Y. Wang, "Quantum transfer learning for Wi-Fi sensing," in *IEEE International Conference on Communications (ICC)*, May 2022.

[24] S. Mazuelas, A. Bahillo, R. M. Lorenzo, P. Fernandez, F. A. Lago, E. Garcia, J. Blas, and E. J. Abril, "Robust indoor positioning provided by real-time RSSI values in unmodified WLAN networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 5, pp. 821–831, 2009.

[25] X. Wang, L. Gao, and S. Mao, "CSI phase fingerprinting for indoor localization with a deep learning approach," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1113–1123, Dec 2016.

[26] ——, "Biloc: Bi-modal deep learning for indoor localization with commodity 5GHz WiFi," *IEEE Access*, vol. 5, pp. 4209–4220, 2017.

[27] C. Hsieh, J. Chen, and B. Nien, "Deep learning-based indoor localization using received signal strength and channel state information," *IEEE Access*, vol. 7, pp. 33 256–33 267, 2019.

[28] A. S. Paul and E. A. Wan, "Wi-Fi based indoor localization and tracking using sigma-point Kalman filtering methods," in *PLANS*, 2008, pp. 646–659.

[29] J. Wang and J. G. Park, "An enhanced indoor ranging method using CSI measurements with extended Kalman filter," in *PLANS*, 2020, pp. 1342–1348.

[30] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 11 1997.

[31] R. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018.

[32] Y. Rubanova, R. Chen, and D. K. Duvenaud, "Latent ordinary differential equations for irregularly-sampled time series," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

[33] J. Kelly, J. Bettencourt, M. J. Johnson, and D. K. Duvenaud, "Learning differential equations that are easy to solve," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 4370–4380, 2020.

[34] C. Finlay, J.-H. Jacobsen, L. Nurbekyan, and A. Oberman, "How to train your neural ODE: the world of Jacobian and kinetic regularization," in

International Conference on Machine Learning (ICML), 2020, pp. 3154–3164.

[35] A. Zhu, P. Jin, B. Zhu, and Y. Tang, "On numerical integration in neural ordinary differential equations," in *International Conference on Machine Learning (ICML)*, 2022, pp. 27 527–27 547.

[36] H. H. N. Nguyen, T. Nguyen, H. Vo, S. Osher, and T. Vo, "Improving neural ordinary differential equations with Nesterov's accelerated gradient method," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 7712–7726, 2022.

[37] C. Challu, K. G. Olivares, B. N. Oreshkin, F. G. Ramirez, M. M. Canseco, and A. Dubrawski, "NHITS: Neural hierarchical interpolation for time series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 6, 2023, pp. 6989–6997.

[38] G. M. Mendoza-Silva, A. C. Costa, J. Torres-Sospedra, M. Painho, and J. Huerta, "Environment-aware regression for indoor localization based on WiFi fingerprinting," *IEEE Sensors Journal*, vol. 22, no. 6, pp. 4978–4988, 2021.

[39] W. Sun, M. Xue, H. Yu, H. Tang, and A. Lin, "Augmentation of fingerprints for indoor WiFi localization based on gaussian process regression," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10 896–10 905, 2018.

[40] M. Nowicki and J. Wietrzykowski, "Low-effort place recognition with WiFi fingerprints using deep learning," in *Automation 2017: Innovations in Automation, Robotics and Measurement Techniques*, 2017, pp. 575–584.

[41] R. Zhou, M. Hao, X. Lu, M. Tang, and Y. Fu, "Device-free localization based on CSI fingerprints and deep neural networks," in *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2018, pp. 1–9.

[42] X. Song, X. Fan, C. Xiang, Q. Ye, L. Liu, Z. Wang, X. He, N. Yang, and G. Fang, "A novel convolutional neural network based indoor localization framework with WiFi fingerprinting," *IEEE Access*, vol. 7, pp. 110 698–110 709, 2019.

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

[44] J. Chen, D. Steinmetzer, J. Classen, E. Knightly, and M. Hollick, "Pseudo lateration: Millimeter-wave localization using a single RF chain," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, March 2017, pp. 1–6.

[45] D. Steinmetzer, D. Wegemer, M. Schulz, J. Widmer, and M. Hollick, "Compressive millimeter-wave sector selection in off-the-shelf IEEE 802.11ad devices," in *CoNEXT 2017*, December 2017.

[46] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[47] D. Steinmetzer, D. Wegemer, and M. Hollick. (2018) Talon tools: The framework for practical ieee 802.11ad research. [Online]. Available: https://seemoo.de/talon-tools/

[48] M. Schulz, D. Wegemer, and M. Hollick, "Nexmon: The C-based firmware patching framework," *Res. Gate*, 2017.

[49] M. Arnold and F. Schaich, "Indoor positioning systems: Smart fusion of a variety of sensor readings," *arXiv:2105.05438*, 2021.

Appendix A
LSTM Update Step

Given the beam SNR $\mathbf{b}_n$ at time step n and the auxiliary variable $\mathbf{h}'_n$, one can use a standard LSTM unit to update the latent variable

$$\mathbf{h}_n = \mathcal{R}(\mathbf{h}'_n, \mathbf{b}_n; \boldsymbol{\theta}), \quad n = 0, 1, \cdots, N, \tag{32}$$

where $\mathcal{R}(\cdot, \cdot|\boldsymbol{\theta})$ is implemented with the following process (with abuse of notation)

$$\tilde{\mathbf{c}}_n = \tanh\left(\mathbf{W}_{rc}\mathbf{b}_n + \mathbf{W}_{hc}\mathbf{h}'_n + \mathbf{b}_c\right), \tag{33}$$

$$\mathbf{f}_n = \sigma\left(\mathbf{W}_{rf}\mathbf{b}_n + \mathbf{W}_{hf}\mathbf{h}'_n + \mathbf{b}_f\right), \tag{34}$$

$$\mathbf{i}_n = \sigma\left(\mathbf{W}_{ri}\mathbf{b}_n + \mathbf{W}_{hi}\mathbf{h}'_n + \mathbf{b}_i\right). \tag{35}$$

The above process consists of three *gates*:

- a memory gate of (33) uses the tanh function to combine the auxiliary hidden state $\mathbf{h}'_n$ and the current input $\mathbf{b}_n$ into a value range of $(-1, 1)$.
- a forget gate of (34) also acts on $(\mathbf{h}'_n, \mathbf{b}_n)$ but compresses the value into $(0, 1)$ with the sigmoid function $\sigma(\cdot)$ to determine how much of the old memory should retain.
- an input gate of (35) compresses $(\mathbf{h}'_n, \mathbf{b}_n)$ into another value in between 0 and 1 and decides how much information we should take from the new input $\mathbf{b}_n$,

along with weight matrices $\mathbf{W}_{rc/rf/ri/hc/hf/hi}$ and bias terms $\mathbf{b}_{c/f/i}$. Then new hidden state $\mathbf{h}_n$ in (32) is updated as

$$\mathbf{h}_n = \tanh\left(\hat{\mathbf{c}}_n\right) \odot \mathbf{o}_n, \tag{36}$$

where the new memory variable $\hat{\mathbf{c}}_n$ updates its "old" memory $\hat{\mathbf{c}}_{n-1}$ passing through the "current" forget gate output $\mathbf{f}_n$ and adds new memory cell $\tilde{\mathbf{c}}_n$ weighted by the "current" input gate output $\mathbf{i}_n$:

$$\hat{\mathbf{c}}_n = \mathbf{f}_n \odot \hat{\mathbf{c}}_{n-1} + \mathbf{i}_n \odot \tilde{\mathbf{c}}_n, \tag{37}$$

and the output gate $\mathbf{o}_n$ is computed as

$$\mathbf{o}_n = \sigma\left(\mathbf{W}_{ro}\mathbf{b}_n + \mathbf{W}_{ho}\mathbf{h}'_n + \mathbf{W}_{co} \odot \hat{\mathbf{c}}_n + \mathbf{b}_o\right). \tag{38}$$

It is seen that the parameters $\boldsymbol{\theta}$ in the LSTM update step is given as $\boldsymbol{\theta} = \{\mathbf{W}_{rc/rf/ri/hc/hf/hi/ro/ho/co}, \mathbf{b}_{c/f/i/o}\}$.

Cristian J. Vaca-Rubio (Member, IEEE) is currently a Researcher at the Centre Tecnològic de Telecommunicacions de Catalunya (CTTC) in Barcelona, Spain. He received the Telematics Engineering degree, and the master's degree in Telematics and Telecommunication Networks from University of Malaga (UMA), Malaga, Spain, in 2018, and 2019, respectively. He received the Ph.D. degree in Wireless Communications from Aalborg University, Aalborg, Denmark in 2023.

In parallel to his telematics and master studies, he worked as a research assistant with the Communication Engineering Department at University of Malaga from February 2018 until July 2019 in a joint project with DEKRA Gmbh. He pursued the Ph.D. degree with the Connectivity Section, Electronic Systems Department, Aalborg University, under a Marie Curie Fellowship as an Early Stage Researcher (ESR) in the H2020 ITN WindMill Project, where he joined in September 2019. During his Ph.D., he was a visiting researcher at Mitsubishi Electric Research Laboratories (MERL), Cambridge, USA for a period of 9 months. His main research activities are in wireless communications and sensing, machine learning , massive MIMO, mobile networks, telematics, and applied statistics in video streaming.

He received the IEEE International Conference on Acoustics, Speech and Signal Processing top 3% paper award in 2023. He is also the author of a submitted patent along with MERL co-authors.

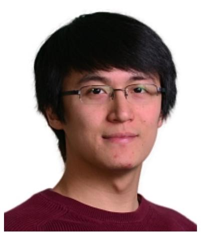
Pu (Perry) Wang (Senior Member, IEEE) received the Ph.D. degree in Electrical Engineering from the Stevens Institute of Technology, Hoboken, NJ, USA, in 2011.

He is a Senior Principal Research Scientist at the Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, where he was an intern in the summer of 2010. Before returning to MERL in 2016, he was a Research Scientist at the Schlumberger-Doll Research, Cambridge, MA, contributing to developments of next-generation logging-while-drilling Acoustics/NMR products and commercialization. His current research focuses on signal processing, Bayesian inference, deep learning, and their applications in wireless sensing, radar perception, and multi-modal understanding.

Dr. Wang received the IEEE Jack Neubauer Memorial Award from the IEEE Vehicular Technology Society in 2013 and was recognized as a Distinguished Speaker by the Society of Petrophysicists and Well Log Analysts (SPWLA) in 2017. He also received the Outstanding Doctoral Thesis in Electrical Engineering Award in 2011, the Edward Peskin Award in 2011, and the Francis T. Boesch Award in 2008 from the Stevens Institute of Technology. He has served as an Associate Editor and then a Senior Area Editor (SAE) for *IEEE Signal Processing Letters*, a Guest Editor for *IEEE Signal Processing Magazine* and *IEEE Sensors Journal*, a Member of the IEEE SPS Signal Processing Theory and Methods (SPTM) Technical Committee, a Member of the IEEE SPS Industry Technical Working Group (ITWG), and a Voting Member of the IEEE 802.11 Standards Association. He has organized special sessions and satellite workshops at ICASSP and SAM and given tutorials at the IEEE Radar Conference and GLOBECOM.

PLACE
PHOTO
HERE

Toshiaki Koike-Akino (Senior Member, IEEE) received the Ph.D. degree from Kyoto University, Kyoto, Japan, in 2005. During 2006-2010, he was a Postdoctoral Researcher at Harvard University, Cambridge, MA, USA, and he is currently Distinguished Research Scientist at MERL, Cambridge, MA, USA. He was the recipient of the 2008 Ericsson Young Scientist Award, the IEEE GLOBECOM'08 Best Paper Award, the 24th TELECOM System Technology Encouragement Award, and the IEEE GLOBECOM'09 Best Paper Award. He is a Fellow of Optica (formerly OSA).



Ye Wang (Senior Member, IEEE) received the B.S. degree in electrical and computer engineering from Worcester Polytechnic Institute, Worcester, MA, USA, in 2005, and the M.S. and Ph.D. degrees in electrical and computer engineering from Boston University, Boston, MA, USA, in 2009 and 2011, respectively.

In 2012, he joined Mitsubishi Electric Research Laboratories, Cambridge, MA, USA, where he had also previously completed an internship in 2010. His research interests are information theory, machine learning, signal processing, communications, and data privacy/security.

Petros T. Boufounos (Fellow, IEEE) is a Distinguished Research Scientist, a Deputy Director and the Computational Sensing Senior Team Leader at Mitsubishi Electric Research Laboratories (MERL). Dr. Boufounos completed his undergraduate and graduate studies at MIT. He received the S.B. degree in Economics in 2000, the S.B. and M.Eng. degrees in Electrical Engineering and Computer Science (EECS) in 2002, and the Sc.D. degree in EECS in 2006. Between September 2006 and December 2008, he was a postdoctoral associate with the Digital Signal Processing Group at Rice University. Dr. Boufounos joined MERL in January 2009, where he has been heading the Computational Sensing Team since 2016.

Dr. Boufounos' immediate research focus includes signal acquisition and processing, computational sensing, inverse problems, quantization, and data representations. He is also interested in how signal acquisition interacts with other fields that use sensing extensively, such as machine learning, robotics, and dynamical system theory. He has over 40 patents granted and more than 10 pending, and more that 100 peer reviewed journal and conference publications in these topics. Dr. Boufounos was the general co-chair of the ICASSP 2023 organizing committee and is currently a regional director-at-large in the IEEE Signal Processing Society's Board of Governors. He has also served as an Area Editor and a Senior Area Editor for the IEEE Signal Processing Letters, an AE for IEEE Transactions on Computational Imaging, and as a member of the SigPort editorial board and the IEEE Signal Processing Society Theory and Methods technical committee. Dr. Boufounos is an IEEE Fellow and an IEEE SPS Distinguished Lecturer for 2019-2020.

Petar Popovski (Fellow, IEEE) is a Professor at Aalborg University, where he heads the section on Connectivity and a Visiting Excellence Chair at the University of Bremen. He received his Dipl.-Ing and M. Sc. degrees in communication engineering from the University of Sts. Cyril and Methodius in Skopje and the Ph.D. degree from Aalborg University in 2005. He received an ERC Consolidator Grant (2015), the Danish Elite Researcher award (2016), IEEE Fred W. Ellersick prize (2016), IEEE Stephen O. Rice prize (2018), Technical Achievement Award from the IEEE Technical Committee on Smart Grid Communications (2019), the Danish Telecommunication Prize (2020) and Villum Investigator Grant (2021). He was a Member at Large at the Board of Governors in IEEE Communication Society 2019-2021. He is currently an Editor-in-Chief of IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS and a Chair of the IEEE Communication Theory Technical Committee. Prof. Popovski was the General Chair for IEEE SmartGridComm 2018 and IEEE Communication Theory Workshop 2019. His research interests are in the area of wireless communication and communication theory. He authored the book "Wireless Connectivity: An Intuitive and Fundamental Guide".