LocDreamer: World Model-Based Learning for Joint Indoor Tracking and Anchor Scheduling

Geng Wang[1], Zhouyou Gu[2], Shenghong Li[3], Peng Cheng[4], Jihong Park[2], Branka Vucetic[1], Yonghui Li[1]

[1]The University of Sydney, Australia,
[2]Singapore University of Technology and Design, Singapore,
[3]CSIRO DATA61, Australia,
[4]La Trobe University, Australia
{geng.wang, branka.vucetic, yonghui.li}@sydney.edu.au, {zhouyou_gu, jihong_park}@sutd.edu.sg,
shenghong.li@data61.csiro.au, p.cheng@latrobe.edu.au

Abstract

Accurate, resource-efficient localization and tracking enables numerous location-aware services in next-generation wireless networks. However, existing machine learning-based methods often require large labeled datasets while overlooking spectrum and energy efficiencies. To fill this gap, we propose *LocDreamer*, a world model (WM)-based framework for joint target tracking and scheduling of localization anchors. LocDreamer learns a WM that captures the latent representation of the target motion and localization environment, thereby generating synthetic measurements to *imagine* arbitrary anchor deployments. These measurements enable *imagination-driven* training of both the tracking model and the reinforcement learning (RL)-based anchor scheduler that activates only the most informative anchors, which significantly reduce energy and signaling costs while preserving high tracking accuracy. Experiments on a real-world indoor dataset demonstrate that LocDreamer substantially improves data efficiency and generalization, outperforming conventional Bayesian filter with random scheduling by 37% in tracking accuracy, and achieving 86% of the accuracy of same model trained directly on real data.

Introduction

Location awareness has become a cornerstone of next-generation wireless systems, supporting applications from smart buildings and industrial automation to integrated sensing and communications (ISAC) (Liu et al. 2022; Trevlakis et al. 2023; Yang et al. 2024). Wireless localization estimates a target's position by utilizing geometry-based measurements, e.g., distance, from multiple anchors at a single time instance, and tracking extends this across consecutive timesteps by considering temporal dynamics. Achieving accurate localization and tracking depends on the quality of radio-signal measurements, which often degrade under multipath and non-line-of-sight (NLoS) propagation in dynamic, complex indoor environments. Extensive research has therefore focused on identifying and mitigating these effects from the received signals to improve tracking accuracy (Nkrow et al. 2024; Wang et al. 2024). Nevertheless, they typically assume that all anchors are simultaneously active to provide measurements, leading to increased energy consumption, spectrum usage, and computational overhead. In practical wireless systems, activating all anchors at every timestep is infeasible due to limited measurement opportunities, bandwidth budgets and energy constraints. This motivates the need for *joint tracking and anchor scheduling*[1]–dynamically selecting a subset of anchors to balance resource consumption and tracking accuracy (Win et al. 2018; Wang, Conti, and Win 2019; Hajiakhondi-Meybodi et al. 2022).

To achieve accurate tracking and select informative anchors, model-based frameworks describe the tracking system through mathematical models, e.g., Bayesian filters, and employ heuristic or optimization-based anchor scheduling policies (Albaidhani, Morell, and Vicario 2019; Zhao et al. 2019; Fan et al. 2023; Oh et al. 2023). However, these static models are designed based on handcrafted assumptions that only offer interpretability and generalization under idealized conditions. Consequently, tracking accuracy degrades and the scheduling strategies fail to adapt when the environment deviates from the assumed models. Moreover, their scheduling decisions still require all anchors to be active for measurements, e.g., measurements from anchors predicted to be uninformative will be discarded, and this post-processing scheme does not reduce the signaling overhead.

Alternatively, machine learning-based frameworks can directly learn to infer target position and scheduling policies from data using neural networks (Hajiakhondi-Meybodi et al. 2022; Gómez-Vega, Win, and Conti 2023b,a; Kim et al. 2025), without relying on fixed parametric mathematical models. Reinforcement learning (RL), in particular, has emerged as a promising approach that can flexibly learn strategies from collected interaction data without explicit modeling of complex environments (Hajiakhondi-Meybodi et al. 2022; Kim et al. 2025). While these methods achieve high accuracy in dynamic and complex environments, they rely on extensive labeled training data to capture environment-specific features. Consequently, they exhibit poor generalization when anchor configurations or environmental conditions change and new measurements are unavailable (Kim et al. 2025). How to learn the tracking and scheduling models that generalize to unseen dynamic and complex environments without additional measurements re-

[1]Also referred to as node selection/activation in the literature.

mains an open challenge.

Recent advances in *world models* (WMs) offer a powerful solution by learning compact latent representations of the environment as well as its dynamics, enabling an agent to imagine future trajectories and optimize strategies without real-world interactions (Ha and Schmidhuber 2018; Hafner et al. 2025). Comparing to model-based RL that learns the dynamics in the observable states (Mohammadi et al. 2018; Li et al. 2020), WMs operate in a compact latent space, making them particularly effective in high-dimensional (Ha and Schmidhuber 2018; Hafner et al. 2025) or complex-structured (Lee et al. 2019) environments. By interacting with latent dynamics rather than real-world environments, WMs significantly improve sample efficiency and have achieved remarkable success in data-efficient control tasks where direct interaction with environments is expensive or unsafe, e.g., autonomous driving (Tu et al. 2025), but its potential remains largely untapped in the field of wireless localization and tracking.

In this paper, we introduce LocDreamer, a WM-based learning framework for joint tracking and anchor scheduling to achieve high tracking accuracy and resource efficiency in dynamic indoor environments, while enabling effective adaptation to new environments without additional measurements. Our contributions are summarized as follows:

- **Joint tracking and scheduling based on WM**: We formulate joint tracking and anchor scheduling as a unified maximum likelihood estimation problem based on WM. Specifically, the problem jointly optimizes 1) a WM that learns the most likely dynamics in localization system to accurately track the target, and 2) an anchor scheduling policy that selects the most informative anchors, both using measurements generated by the WM.
- **Imagination-driven learning using WM**: We design the learning framework, LocDreamer, to train the WM and the scheduling policy for joint tracking and anchor scheduling. The WM is first pre-trained using data from a well-measured source environment to learn the dynamics of the localization system. It then imagines the dynamics and generates synthetic measurements for unseen anchor configurations, enabling self-supervised learning of joint tracking and scheduling without additional data collection.
- **Evaluations using real-world dataset**: Through extensive experiments, we demonstrate that LocDreamer generalizes well to unmeasured environments with new anchor configurations without any additional measurements. The imagined WM and scheduling policy outperform model–based tracking with random scheduling strategy by 37%, while achieving 86% of the tracking accuracy of the same WM trained directly on real-world data.

System Model

Tracking and Scheduling Models

We consider an indoor tracking scenario in which a target moves within a two-dimensional space $\mathcal{M} \subset \mathbb{R}^2$, covered by a set of anchors $\mathcal{A} = \{1, \cdots, A\}$, $A \geq 3$ at fixed known positions $\mathbf{p}^k = \left[x^k, y^k\right] \in \mathcal{M}$, $k \in \mathcal{A}$. At each discrete time step $t \in \{1, \cdots, T\}$, the target initiates ranging requests to a selected subset of anchors $\mathcal{K}_t \subseteq \mathcal{A}$ with $|\mathcal{K}_t| = K_t$, $3 \leq K_t \leq A$, and receives corresponding distance measurements $\mathbf{d}_t = \left[d_t^k\right]_{k \in \mathcal{K}_t} \in \mathbb{R}^{K_t}$ from those K_t anchors. The measured distance d_t^k from each individual anchor is modeled as

$$d_t^k = \left\|\mathbf{p}_t - \mathbf{p}^k\right\| + n_t^k, \tag{1}$$

where $\mathbf{p}_t = [x_t, y_t] \in \mathcal{M}$ is the target position and n_t^k represents measurement noise.

Before each ranging request, the anchor scheduling model determines which set of anchors $\mathcal{K}_t$ should be activated for measurements by outputting a binary scheduling policy vector $\boldsymbol{\alpha}_t = \left[\alpha_t^1, \cdots, \alpha_t^A\right] \in \{0,1\}^A$, where $\alpha_t^k = 1$ if anchor k is activated and $\alpha_t^k = 0$ otherwise. The active set is therefore $\mathcal{K}_t = \left\{k \,|\, \alpha_t^k = 1\right\}$ with $\sum_{k=1}^A \alpha_t^k = K_t$. Since distance measurements $\mathbf{d}_t$ can be noisy and erroneous due to multipath and NLoS propagation conditions (Wang et al. 2024), finding the optimal set of anchors $\mathcal{K}_t$ can improve both tracking accuracy and resource efficiency.

Unified Objective for Joint Tracking and Scheduling

The goal is to maximize tracking accuracy subject to the constraint of scheduling K_t anchors per timestep. In classical supervised-learning based tracking, this is equivalent to minimizing the position estimation error between the predicted and ground-truth positions. Since we do not assume access to labeled ground truth positions, we instead maximize the marginal likelihood of distance measurements as a surrogate for tracking accuracy

$$\begin{aligned} \max_{\theta,\, \boldsymbol{\alpha}_{1:T}} \quad & \log p_\theta(\mathbf{d}_{1:T} \,|\, \boldsymbol{\alpha}_{1:T}) \\ \text{s.t.} \quad & \sum_{k=1}^{A} \alpha_t^k = K_t, \quad \forall t, \end{aligned} \tag{2}$$

which measures how well p_θ explains the observed measurements under the scheduled anchors. Intuitively, the anchor scheduling policy is optimized to maximize this likelihood, thereby favoring anchors that yield more informative and accurate measurements.

The marginal likelihood of the measurements $\mathbf{d}_t$ given anchor selections $\boldsymbol{\alpha}_t$ in (2) is linked to the tracking problem through the latent target state defined as $\mathbf{z}_t \triangleq [\mathbf{p}_t, v_t^x, v_t^y] \in \mathbb{R}^{Z=4}$, which also includes the velocities at x and y directions as kinematic states. The tracking process is conventionally described as a discrete-time continuous-value state space model (SSM) characterized by the following state transition and observation functions

$$\mathbf{z}_t \sim f_{\theta_\mathbf{z}}\left(\mathbf{z}_{t-1}\right), \quad \mathbf{z}_t \in \mathbb{R}^Z, \tag{3a}$$

$$\mathbf{d}_t \sim g_{\theta_\mathbf{d}}\left(\mathbf{z}_t, \boldsymbol{\alpha}_t\right), \quad \mathbf{d}_t \in \mathbb{R}^{K_t}. \tag{3b}$$

where the target state $\mathbf{z}_t$ evolves following the first-order Markovian assumption and current distance measurement $\mathbf{d}_t$

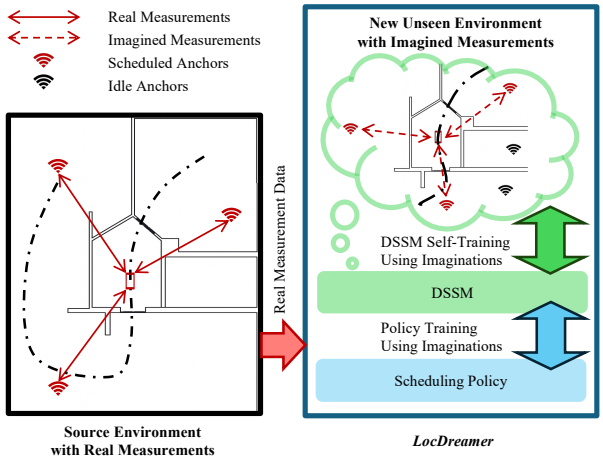

Figure 1: Illustration of the proposed LocDreamer.

is generated from the current latent state $\mathbf{z}_t$ and selected anchors $\boldsymbol{\alpha}_t$. Applying the above, the marginal likelihood in (2) can be factorized as

$$\begin{aligned} &\log p_\theta(\mathbf{d}_{1:T} \,|\, \boldsymbol{\alpha}_{1:T}) \\ &\quad = \log \int p_\theta(\mathbf{d}_{1:T},\, \mathbf{z}_{1:T} \,|\, \boldsymbol{\alpha}_{1:T})\, \mathrm{d}\mathbf{z}_{1:T} \\ &\quad = \log \int \prod_{t=1}^{T} g_{\theta_\mathbf{d}}(\mathbf{d}_t \,|\, \mathbf{z}_t,\, \boldsymbol{\alpha}_t)\, f_{\theta_\mathbf{z}}(\mathbf{z}_t \,|\, \mathbf{z}_{t-1})\, \mathrm{d}\mathbf{z}_{1:T}, \end{aligned} \tag{4}$$

where the scheduling action $\boldsymbol{\alpha}_t$ affects the measurement likelihood, which in turn influences position inference. However, conventional SSMs with simplified first-order assumptions cannot capture the complex hidden dynamics of the tracking system. To address this issue, deep SSMs (DSSMs) (Girin et al. 2021; Gedon et al. 2021; Wang et al. 2025b) have been developed to replace SSM functions with neural networks with an additional recurrence hidden state $\mathbf{h}_t$ for improved expressivity. As a result, we rewrite (4) as[2]

$$\begin{aligned} &\log p_\theta(\mathbf{d}_{1:T} \,|\, \boldsymbol{\alpha}_{1:T}) \\ &= \log \int \prod_{t=1}^{T} g_{\theta_\mathbf{d}}(\mathbf{d}_t \,|\, \mathbf{z}_t,\, \mathbf{h}_t, \boldsymbol{\alpha}_t)\, f_{\theta_\mathbf{z}}(\mathbf{z}_t \,|\, \mathbf{z}_{t-1}, \mathbf{h}_t)\, \mathrm{d}\mathbf{z}_{1:T}. \end{aligned} \tag{5}$$

This architecture uses the latent state $\mathbf{z}_t$ and the deterministic hidden state $\mathbf{h}_t$ to represent the system at each timestep t. Rolling out the generative model yields imagined trajectories, analogous to the imagination process observed in WM-based learning. From imagination, we reconstruct measurements for arbitrary anchor deployments, enabling training of both the DSSM tracker and the scheduling policy in unseen environments without additional real data. An overview of the proposed imagination-based training framework, LocDreamer, is shown in Fig. 1. We detail the DSSM and the imagination-driven training in the next section.

[2]The recurrence hidden state $\mathbf{h}_t$ is a deterministic function of other latent variables and is marginalized for clarity of presentation throughout the paper.

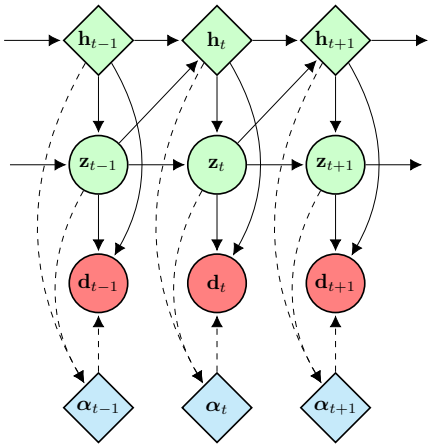

Figure 2: Graphical model of LocDreamer. Diamond and circle represent deterministic and stochastic variables. Arrows indicate conditional dependencies. Solid and dashed lines represent the DSSM modeling the tracking system and the scheduling decision making. Green, red and blue highlight DSSM, observation and anchor scheduler components, respectively.

Proposed LocDreamer

To address the joint tracking and anchor scheduling problem formulated in (2), we propose LocDreamer, a WM-based learning framework that integrates DSSM for probabilistic tracking and imagination-driven RL scheduler training within a single framework. The DSSM learns the latent/hidden dynamics of the target and the environment, while the RL agent learns optimal scheduling policies. This imagination capability allows the DSSM to be first pre-trained in a source environment, and then used to generate synthetic measurements for unseen anchor configurations in new environments. Using the generated measurements, LocDreamer self-supervises the training of both the DSSM and the RL scheduler, without requiring additional real-world measurements in the new environments. Since the overall learning objective is to maximize the marginal likelihood in (2), DSSM approximates the likelihood of the measurements from scheduled anchors, and the RL scheduler learns to select anchors that provide the highest-likelihood measurement in DSSM, thus improving tracking accuracy while reducing resource consumption. The graphical model of LocDreamer is shown in Fig. 2.

Deep State Space Model in LocDreamer

The DSSM backbone (Girin et al. 2021; Gedon et al. 2021; Wang et al. 2025b) of our LocDreamer integrates stochastic recurrent latent dynamics, physics-guided modeling and variational inference to learn the tracking dynamics. It consists of the following components: a sequence model to capture temporal dependencies of the hidden states; a dynamic model to predict the prior latent state in the next step; an encoder model to infer the posterior latent state from measurements; and a decoder model to reconstruct measurements

$$\mathcal{L}^{\text{ELBO}} = \sum_{t=1}^{T} \Bigg\{ \underbrace{\mathbb{E}_{q_\phi}\Bigg[\sum_{k\in\mathcal{K}_t} \log g_{\theta_d}\Big(d_t^k \mid \mathbf{z}_t^{\text{posterior}},\, \mathbf{h}_t,\, \mathbf{p}^k\Big)\Bigg]}_{L_t^{\text{recon}}} - \underbrace{\text{KL}\Bigg(q_\phi\Big(\mathbf{z}_t^{\text{posterior}} \mid \mathbf{z}_{t-1}^{\text{posterior}},\, \mathbf{h}_t,\, \mathbf{o}(\boldsymbol{\alpha}_t)\Big) \Bigg\| f_{\theta_{\mathbf{z}}}\Big(\mathbf{z}_t^{\text{prior}} \mid \mathbf{z}_{t-1}^{\text{posterior}},\, \mathbf{h}_t\Big)\Bigg)}_{L_t^{\text{dyn}}} \Bigg\}. \tag{6}$$

from the latent state. Their structures are detailed as follows.

Sequence Model We use recurrent neural network (RNN) as the sequence model that acts as the memory of DSSM. The RNN deterministic hidden state $\mathbf{h}_t \in \mathbb{R}^H$ evolves as

$$\mathbf{h}_t = r_{\theta_{\mathbf{h}}}\left(\mathbf{z}_{t-1}^{\text{posterior}},\, \mathbf{h}_{t-1}\right), \tag{7}$$

where $r_{\theta_{\mathbf{h}}}(\cdot)$ is a recurrent function parametrized by $\theta_{\mathbf{h}}$, and $\mathbf{h}_t$ depends on both the past posterior estimation of the latent state $\mathbf{z}_{t-1}$ and the past RNN hidden state $\mathbf{h}_{t-1}$.

Dynamic Model We model the target's prior latent state $\mathbf{z}_t^{\text{prior}}$ as a Gaussian-distributed random variable

$$\mathbf{z}_t^{\text{prior}} \sim \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{z},t}^{\text{prior}},\, \text{diag}\left\{\left(\boldsymbol{\sigma}_{\mathbf{z},t}^{\text{prior}}\right)^2\right\}\right), \tag{8}$$

with mean vector $\boldsymbol{\mu}_{\mathbf{z},t}^{\text{prior}}$ and a diagonal covariance matrix $\text{diag}\left\{\left(\boldsymbol{\sigma}_{\mathbf{z},t}^{\text{prior}}\right)^2\right\}$. These parameters are generated by the target dynamic model that consists of a physics-based approximation and a learning-based correction as

$$\begin{aligned}\left[\boldsymbol{\mu}_{\mathbf{z},t}^{\text{prior}},\, \boldsymbol{\sigma}_{\mathbf{z},t}^{\text{prior}}\right] &= f_{\theta_{\mathbf{z}}}\left(\mathbf{z}_{t-1}^{\text{posterior}},\, \mathbf{h}_t\right) \\ &= f^{\text{phy}}\left(\mathbf{z}_{t-1}^{\text{posterior}}\right) + \text{MLP}_{\theta_{\mathbf{z}}}\left(\mathbf{z}_{t-1}^{\text{posterior}},\, \mathbf{h}_t\right),\end{aligned} \tag{9}$$

where $f^{\text{phy}}(\cdot)$ is an arbitrary motion model (Rong Li and Jilkov 2003) and $\text{MLP}_{\theta_{\mathbf{z}}}(\cdot)$ is a multi-layer perceptron (MLP) parametrized by $\theta_{\mathbf{z}}$ that learns a data-driven residual compensating model discrepancies.

Encoder Model At each timestep t, when encoding the measurements from either real or imagined environments, the model receives pairs of distance measurement d_t^k and its corresponding anchor position $\mathbf{p}^k$ as observations $\mathbf{o}(\boldsymbol{\alpha}_t) \triangleq \left\{\left(d_t^k,\, \mathbf{p}^k\right) \mid \alpha_t^k = 1, k = 1, \ldots, A\right\}$ from scheduling action $\boldsymbol{\alpha}_t$. Similarly, we model the target posterior state $\mathbf{z}_t^{\text{posterior}}$ as a Gaussian-distributed random variable

$$\mathbf{z}_t^{\text{posterior}} \sim \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{z},t}^{\text{posterior}},\, \text{diag}\left\{\left(\boldsymbol{\sigma}_{\mathbf{z},t}^{\text{posterior}}\right)^2\right\}\right). \tag{10}$$

And we apply variational Bayesian inference (Kingma and Welling 2022) to approximate the true posterior as

$$\left[\boldsymbol{\mu}_{\mathbf{z},t}^{\text{posterior}},\, \boldsymbol{\sigma}_{\mathbf{z},t}^{\text{posterior}}\right] = q_\phi\left(\mathbf{z}_{t-1}^{\text{posterior}},\, \mathbf{h}_t,\, \mathbf{o}(\boldsymbol{\alpha}_t)\right), \tag{11}$$

where $q_\phi(\cdot)$ is the variational inference function parameterized by ϕ. We implement $q_\phi(\cdot)$ following the set transformer encoder approach (Lee et al. 2019) to ensure permutation invariance with respect to the number and order of active anchors in the observations $\mathbf{o}(\boldsymbol{\alpha}_t)$.

Decoder Model The decoder model reconstructs distance measurements by embedding physics-based distance model in (1) with a learnable noise model. The reconstructed distance measurement $\hat{d}_t^k$ of the k-th anchor is modeled as a Gaussian-distributed random variable as

$$\hat{d}_t^k \sim \mathcal{N}\left(\mu_{d,t}^k,\, \left(\sigma_{d,t}^k\right)^2\right), \quad \forall k \in \mathcal{K}_t, \tag{12}$$

whose parameters are inferred based on the posterior estimation of the latent state as

$$\mu_{d,t}^k = \|\tilde{\mathbf{p}}_t^{\text{posterior}} - \mathbf{p}^k\|, \quad \forall k \in \mathcal{K}_t, \tag{13a}$$

$$\sigma_{d,t}^k = \text{MLP}_{\theta_d}\left(\tilde{\mathbf{z}}_t^{\text{posterior}},\, \mathbf{h}_t,\, \mathbf{p}^k\right), \quad \forall k \in \mathcal{K}_t. \tag{13b}$$

Here, (13a) uses physics-based distance model in (1) to compute the mean of the reconstructed distance and (13b) uses a MLP parameterized by θ_d to learn the measurement noise with reparameterized sample $\tilde{\mathbf{z}}_t^{\text{posterior}}$ from $\mathbf{z}_t^{\text{posterior}}$. We write the above reconstruction process in (12), (13a) and (13b) as $\hat{d}_t^k \sim g_{\theta_d}\left(\mathbf{z}_t^{\text{posterior}},\, \mathbf{h}_t,\, \mathbf{p}^k\right)$.

Loss of DSSM Note that direct maximization of (2) is intractable because it requires the integration over all possible latent state sequences $\mathbf{z}_{1:T}$ in (5). Therefore, to train the DSSM, we maximize the variational *evidence lower bound* (ELBO) of (2) in (6) as an alternative objective that can be computed based on the DSSM components in (7), (9), (11) and (13) (Girin et al. 2021; Wang et al. 2025b). Here, the reconstruction loss L_t^{recon} computes the conditional log-likelihood of the measurements, encouraging the model to reproduce realistic distance distributions, while the dynamic loss L_t^{dyn} regularizes posterior approximation from prior dynamics, enabling coherent imagination.

RL-Based Anchor Scheduling in LocDreamer

With the learned DSSM that models the tracking system dynamics, we can now optimize the anchor scheduling policy to select informative anchors that maximize the measurement likelihood. We use an actor-critic (AC) model to train our scheduling policy entirely based on the imagined measurements generated by the DSSM.

Actor Model The actor is the stochastic scheduling policy that makes the actor selections $\boldsymbol{\alpha}_t$ based on the current latent and hidden state of the DSSM

$$\boldsymbol{\alpha}_t \sim \pi_{\phi_{\boldsymbol{\alpha}}}(\mathbf{s}_t), \quad \forall t, \tag{14}$$

where $\mathbf{s}_t = \left(\mathbf{z}_t^{\text{prior}},\, \mathbf{h}_t\right)$ is the state input of the actor at timestep t and $\phi_{\boldsymbol{\alpha}}$ is the parameter of the actor model.

Critic Model The critic learns to evaluate the expected reward to guide the actor based on the rewards. The reward $R_t = L_t^{\text{recon}}$ in (6) is designed as the reconstruction loss as the policy is rewarded for selecting anchors that yield high measurement likelihood, i.e., anchors that preserve tracking accuracy. The critic predicts the value over the future discounted rewards as

$$V_{\phi_V}(\mathbf{s}_t) = \mathbb{E}\left[\sum_{\tau=t}^{T} \gamma^{\tau-t} R_\tau\right], \tag{15}$$

where ϕ_V is the parameter of the value function, and $\gamma \in (0, 1)$ is the discount factor.

Loss of Actor Critic The actor learns to maximize the rewards with each anchor scheduling strategy

$$L_t^{\text{actor}} = -\mathbb{E}_{\sim\pi_{\phi_{\boldsymbol{\alpha}}},\text{DSSM}}[\log \pi_{\phi_{\boldsymbol{\alpha}}}(\boldsymbol{\alpha}_t \mid \mathbf{s}_t)\, Adv(\mathbf{s}_t, \boldsymbol{\alpha}_t)], \tag{16}$$

where minimizing L_t^{actor} is equivalent to maximizing the expected return under the policy parameterized by $\phi_{\boldsymbol{\alpha}}$ when interacting with the learned DSSM, and $Adv(\mathbf{s}_t, \boldsymbol{\alpha}_t) = \sum_{\tau=t}^{T} \gamma^{\tau-t} R_\tau - V_{\phi_V}(\mathbf{s}_t)$ is the advantage function representing the advantage of taking action $\boldsymbol{\alpha}_t$ in state $\mathbf{s}_t$. The critic learns to evaluate the value by minimizing the error between the predicted value and observed return as

$$L_t^{\text{critic}} = \mathbb{E}_{\sim\pi_{\phi_{\boldsymbol{\alpha}}},\text{DSSM}}\left(V_{\phi_V}(\mathbf{s}_t) - \sum_{\tau=t}^{T} \gamma^{\tau-t} R_\tau\right)^2. \tag{17}$$

The above losses are then used to update the actor and critic networks via gradient descent.

Proposed LocDreamer Learning

The overall training procedure of LocDreamer consists of two stages: DSSM pre-training and WM imagination-based joint training of DSSM and scheduling policy, as illustrated in **Algorithm 1**. In the first stage, the DSSM is pre-trained on a source environment with sufficient real measurements from a set of anchors $\mathcal{R} \nsubseteq \mathcal{A}$, assuming all anchors in $\mathcal{R}$ are always providing measurements, i.e., $\boldsymbol{\alpha}_t = \mathbf{1}, \forall t$. In each pre-training epoch, the model first initalizes the latent and hidden states, $\mathbf{z}_0^{\text{posterior}}$ and $\mathbf{h}_1$, and processes the real measurements $\{d_t^r\}_{r\in\mathcal{R}}$ sequentially over T timesteps. At each timestep t, the model first predicts the prior latent state $\mathbf{z}_t^{\text{prior}}$ using (9), then infers the posterior latent state $\mathbf{z}_t^{\text{posterior}}$ using (11), reconstructs the distance measurements $\{\hat{d}_t^r\}_{r\in\mathcal{R}}$ using (13), and finally updates the RNN hidden state $\mathbf{h}_{t+1}$ using (7). At the end of each timestep, the model computes the reconstruction loss L_t^{recon} and dynamic loss L_t^{dyn} in (6) and accumulates them over T timesteps. After processing all timesteps in the epoch, the model computes the overall ELBO loss $\mathcal{L}^{\text{ELBO}}$ in (6) and uses gradient descent to update the DSSM parameters. The DSSM pre-training repeats for E_{dssm} epochs.

Once trained, the DSSM can model the target dynamics and generate realistic distance measurements for arbitrary anchor deployments. In detail, in each imagination epoch, the DSSM predicts future dynamics considering the imagined anchors in $\mathcal{A}$ without measurements but only with their positions $\{\mathbf{p}^k\}_{k\in\mathcal{A}}$ known. At each timestep t, the DSSM predicts future dynamics by running learned (7) and (9) forward, samples imagined measurements from the learned generative model (13), and select the anchors using (14) accordingly. This closed-loop imagination facilitates training entirely inside its dream, allowing LocDreamer to generalize to unseen scenarios without actual interaction with the

Algorithm 1: Proposed LocDreamer Algorithm

Input : DSSM training epoch E_{dssm}; LocDreamer imagination and training epoch E_{imagine};
Output: Trained LocDreamer for tracking and anchor scheduling;

Initialize LocDreamer;
/* DSSM Training Using Real Data $\mathcal{R}$ */
for $e = 1, \cdots, E_{dssm}$ **do**
  Initialize $\mathbf{z}_0^{\text{posterior}}$ and $\mathbf{h}_1$;
  for $t = 1, \cdots, T$ **do**
    $\mathbf{z}_t^{\text{prior}} \sim f_{\theta_{\mathbf{z}}}\left(\mathbf{z}_{t-1}^{\text{posterior}}, \mathbf{h}_t\right)$;
    $\mathbf{z}_t^{\text{posterior}} \sim q_\phi\left(\mathbf{z}_{t-1}^{\text{posterior}}, \mathbf{h}_t, \{(d_t^k, \mathbf{p}^k) \mid k \in \mathcal{R}\}\right)$;
    $\hat{d}_t^k \sim g_{\theta_d}\left(\mathbf{z}_t^{\text{posterior}}, \mathbf{h}_t, \mathbf{p}^k\right)$ for $k \in \mathcal{R}$;
    $\mathbf{h}_{t+1} = r_{\theta_{\mathbf{h}}}\left(\mathbf{z}_t^{\text{posterior}}, \mathbf{h}_t\right)$;
    Compute L_t^{dyn} from $\mathbf{z}_t^{\text{prior}}$, $\mathbf{z}_t^{\text{posterior}}$;
    Compute L_t^{recon} from d_t^k, $\hat{d}_t^k$ for $k \in \mathcal{R}$;
  Train $\theta_{\mathbf{z}}, \theta_{\mathbf{h}}, \theta_{\mathbf{d}}, \phi$ of DSSM;
/* LocDreamer Imagination and Training for $\mathcal{A}$ */
for $e = 1, \cdots, E_{imagine}$ **do**
  Initialize $\mathbf{z}_0^{\text{posterior}}$ and $\mathbf{h}_1$;
  for $t = 1, \cdots, T$ **do**
    $\mathbf{z}_t^{\text{prior}} \sim f_{\theta_{\mathbf{z}}}\left(\mathbf{z}_{t-1}^{\text{posterior}}, \mathbf{h}_t\right)$;
    $d_t^k \sim g_{\theta_d}\left(\mathbf{z}_t^{\text{prior}}, \mathbf{h}_t, \mathbf{p}^k\right)$ for $k \in \mathcal{A}$;
    $\boldsymbol{\alpha}_t \sim \pi_{\theta_{\boldsymbol{\alpha}}}\left(\mathbf{z}_t^{\text{prior}}, \mathbf{h}_t\right) \in \mathbb{R}^A$;
    $\mathbf{z}_t^{\text{posterior}} \sim q_\phi\left(\mathbf{z}_{t-1}^{\text{posterior}}, \mathbf{h}_t, \mathbf{o}(\boldsymbol{\alpha}_t)\right)$;
    $\hat{d}_t^k \sim g_{\theta_d}\left(\mathbf{z}_t^{\text{posterior}}, \mathbf{h}_t, \mathbf{p}^k\right)$ for $k \in \mathcal{R} \cup \mathcal{K}_t$;
    $\mathbf{h}_{t+1} = r_{\theta_{\mathbf{h}}}\left(\mathbf{z}_t^{\text{posterior}}, \mathbf{h}_t\right)$;
    Compute L_t^{dyn} from $\mathbf{z}_t^{\text{prior}}$, $\mathbf{z}_t^{\text{posterior}}$;
    Compute L_t^{recon} from d_t^k and $\hat{d}_t^k$ for $k \in \mathcal{R} \cup \mathcal{K}_t$;
    Compute AC loss from L_t^{recon};
  Finetune $\theta_{\mathbf{z}}, \theta_{\mathbf{h}}, \theta_{\mathbf{d}}, \phi$ of DSSM and train $\theta_{\boldsymbol{\alpha}}, \phi_V$ of AC for $\mathcal{A}$;

anchors in $\mathcal{A}$. After each epoch, the parameters of the DSSM and AC are updated based on their losses. The training for the imagined anchors is repeated for E_{imagine} epochs.

Experiment Results and Discussions

Experiment Setup

We evaluate the performance of the proposed framework using the indoor Ultra-wideband (UWB) positioning and position tracking dataset (Bregar 2023), which uses commercial DecaWave DW1000 UWB modules as measurement nodes operating across frequencies from 3494.4 MHz to 6489.6 MHz with bandwidths of 499.2 MHz or 900 MHz. Specifically, we use *Environment 0*, a residential house floor (9.18 m $\times$ 12.06 m) with brick inside and outside walls, and a total of 126410 measurements were collected from 8 fixed anchors at 85 positions.

To mimic a data-scarce deployment scenario, we use real measurements from three anchors to bootstrap the model, and then use the trained model to imagine for unseen set of five new anchors $\mathcal{A}$ with $A = 5$. The tracking model and scheduling policy are trained on these imagined measurements, targeting a fixed minimum number of $K_t = 3, \forall t$ anchors. Tracking performance is then evaluated using real measurements from the unseen anchors $\mathcal{A}$, testing the framework's generalization to new anchor deployments when trained using imagined data.

All experiments are conducted on a single NVIDIA RTX 3060 Ti GPU and hyperparameters of the model is summarized in Table 1. For comparison, we assess the tracking performances of the following methods and strategies:

Table 1: Hyperparameter Setting

Parameter	Symbol	Value
Environment		
Total number of anchors	A	5
Scheduled number of anchors	K_t	3
DSSM		
DSSM Training Epoch	E_{dssm}	50
Batch size	B	32
Sequence length	T	32
RNN state dimension	H	50
RNN layers	-	2
Learning rate	-	1e-3
Optimizer	-	Adam
Scheduler	-	Cosine annealing
Weight decay	-	1e-3
AC		
Imagination Epoch	E_{imagine}	300
Learning rate	-	1e-3
Optimizer	-	Adam
Scheduler	-	Cosine annealing
Discount factor	γ	0.99

1. EKF - *random scheduling*: Extended Kalman filter (EKF) with $K_t = 3$ anchors randomly selected at each timestep. A constant velocity transition model (Wang et al. 2025a) is adopted and the standard deviation of acceleration noise for the process noise covariance matrix $\mathbf{Q} \in \mathbb{R}^{4\times4}$ is set to $\sigma_{\text{acc}} = 0.2\text{m/s}^2$. The standard deviation of measurement noise for the measurement noise covariance matrix $\mathbf{R} \in \mathbb{R}^{4\times4}$ is set to $\sigma_n = 1\text{m}$.
2. EKF - *all anchors*: EKF with the same configuration but using all anchors $\mathcal{A}$ at each timestep.
3. DSSM - *all anchors (real)*: Trained with real measurements using all anchors $\mathcal{A}$.
4. LocDreamer - *random scheduling*: Trained with imagined data and it randomly selects K_t anchors instead of using the scheduling policy from learned AC.
5. LocDreamer - *all anchors (imagined)*: Trained with imagined data using all anchors $\mathcal{A}$.
6. LocDreamer - *scheduling*: Trained with imagined data and scheduling from learned AC.

Table 2: Tracking Errors (m) for Different Algorithms

Algorithms	MAE	RMSE	50th	90th
EKF - *random scheduling*	1.05	1.67	0.71	1.83
EKF - *all anchors*	0.92	1.50	0.63	1.41
DSSM - *all anchors (real)*	0.57	0.64	0.54	0.96
LocDreamer - *random scheduling*	1.07	1.43	0.82	2.05
LocDreamer - *all anchors (imagined)*	0.85	1.07	0.68	1.60
LocDreamer - *scheduling*	0.66	0.77	0.60	1.18

Tracking Performance

We summarize the tracking performances in terms of mean absolute error (MAE), root mean squared error (RMSE), and $50^{\text{th}}/90^{\text{th}}$ percentile error for all methods in Table 2. EKF and LocDreamer achieve similar tracking performance with a random scheduling strategy, achieving a MAE around 1m. By using all anchors instead of random scheduling, the tracking performances increase in all metrics for both EKF and LocDreamer, demonstrating the additional robustness with abundant measurements and validating that LocDreamer can generalize to unseen anchors without requiring real measurements. The proposed LocDreamer - *scheduling* further improves accuracy while preserving resource efficiency by actively scheduling most informative anchors with policies learned from imagined data, demonstrating that the imagined measurements from LocDreamer are sufficiently realistic to train both the tracking model and the scheduling policy. And it approaches the performance of DSSM using all anchors trained with real measurements. We also show the estimated trajectories of these methods in Fig. 3. These results demonstrate that LocDreamer can learn latent dynamics and enable imagination-driven learning, leading to efficient and accurate tracking in unseen anchor deployments without real measurements.

We also plot an anchor scheduling heatmap across different locations in Fig. 4. It can be observed that the model consistently prioritizes anchors 1 and 4, which provide favorable geometric diversity and lower geometrical dilution of precision (GDOP) (Horn 2020). This confirms that the learned scheduling policy discovers physically meaningful anchor combinations directly from imagination-driven rewards. The learning curve of tracking performance against

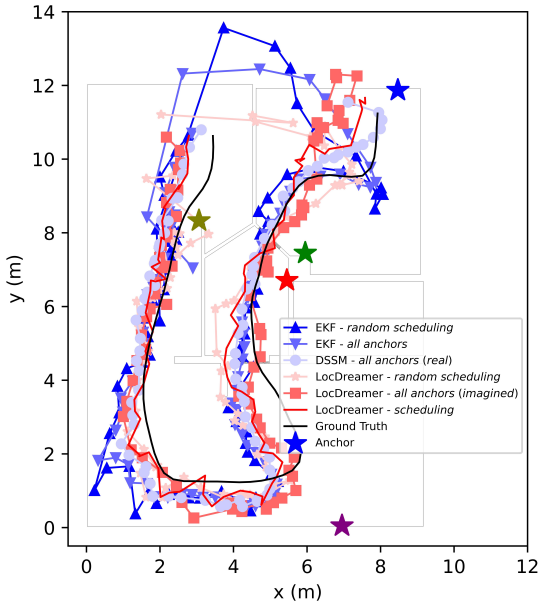

Figure 3: Estimated trajectories with different methods.

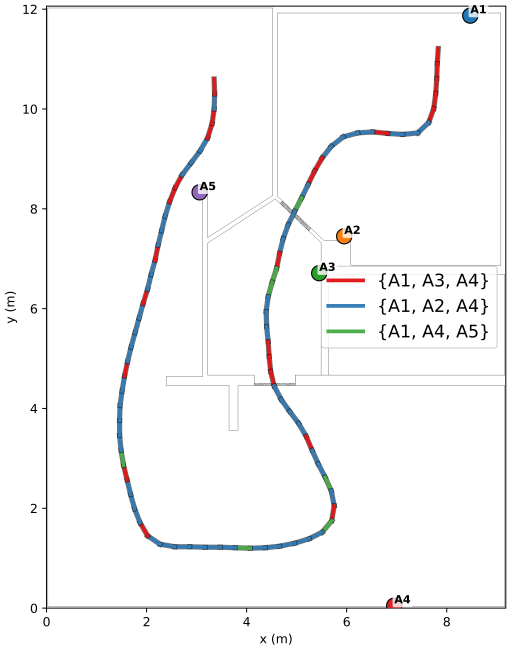

Figure 4: An anchor scheduling heatmap over spatial locations where different color represents different selected anchor sets $\mathcal{K}_t$.

training epochs is shown in Fig. 5. We set $E_{\text{imagine}} = 1000$ to test the LocDreamer's sensitivity to overfitting. The proposed LocDreamer - *scheduling* quickly converges at around 300 epochs, outperforming baseline methods.

Limitations and Future Work

Our current model still assumes real data from a minimal set of 3 anchors to bootstrap training, and the quality of these real data affects the training and imagination performance. Reducing the impact or removing this assumption is an important direction for future work. Moreover, the current number of scheduled anchors is fixed at $K_t = 3$, extending the scheduler to dynamically adapt K_t to channel conditions and desired accuracy-resource tradeoffs is another promising direction. The resource efficiency/usage can be modeled in a more sophisticated way for quantitative evaluation. Finally, further validation on large-scale and more complex environments is a promising next step. In these scenarios, exhaustive search becomes computationally intractable, and the proposed imagination-driven framework could demonstrate clear advantages in scalability, data efficiency and cross-environment generalization.

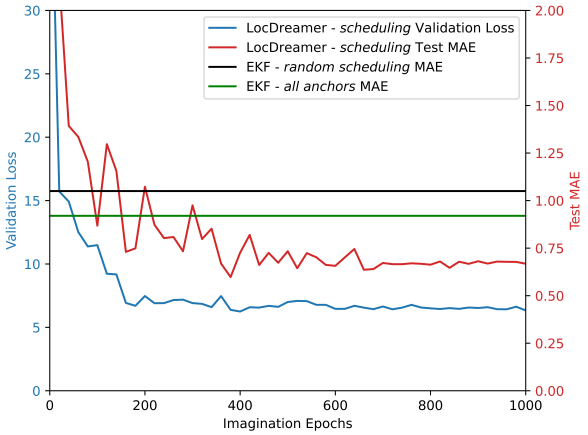

Figure 5: Validation loss and test MAE of LocDreamer - *scheduling* versus imagination epochs E_{imagine}.

Conclusion

This paper introduces LocDreamer, a WM-based learning framework for joint indoor tracking and anchor scheduling. By learning a DSSM that captures target dynamics and environment behavior, the proposed approach can imagine realistic measurements for unseen anchor deployments. These imagined measurements enable efficient training of both the tracking model and a RL-based scheduling policy, without requiring additional data from site surveys. Experiments on a public UWB dataset demonstrate that the proposed framework achieves superior accuracy using only a few anchors for bootstrapping, outperforms random scheduling and approaches the performance of models trained with full real measurements. Future work will explore adaptive anchor selection with variable anchor budgets and self-bootstrapping that requires no real data initialization to further enhance robustness and scalability toward AI-native 6G networks.

Acknowledgments

This work was supported by Australian Research Council (ARC) under Grants DP210103410 and DP220101634, in part by SUTD Kickstarter Initiative (SKI 2021_06_08), and in part by the National Research Foundation, Singapore and Infocomm Media Development Authority under its Communications and Connectivity Bridging Funding Initiative.

References

Albaidhani, A.; Morell, A.; and Vicario, J. L. 2019. Anchor selection for UWB indoor positioning. *Transactions on emerging telecommunications technologies*, 30(6): e3598.

Bregar, K. 2023. Indoor UWB positioning and position tracking data set. *Scientific Data*, 10(1): 744.

Fan, C.; Li, L.; Zhao, M.-M.; Liu, A.; and Zhao, M.-J. 2023. Communication and Energy-Constrained Neighbor Selection for Distributed Cooperative Localization. *IEEE Transactions on Wireless Communications*, 22(6): 4158–4172.

Gedon, D.; Wahlström, N.; Schön, T. B.; and Ljung, L. 2021. Deep state space models for nonlinear system identification. *IFAC-PapersOnLine*, 54(7): 481–486.

Girin, L.; Leglaive, S.; Bie, X.; Diard, J.; Hueber, T.; Alameda-Pineda, X.; et al. 2021. Dynamical variational autoencoders: A comprehensive review. *Foundations and Trends® in Machine Learning*, 15(1-2): 1–175.

Gómez-Vega, C. A.; Win, M. Z.; and Conti, A. 2023a. Machine Learning Based Node Selection for UWB Network Localization. In *MILCOM 2023 - 2023 IEEE Military Communications Conference (MILCOM)*, 735–740.

Gómez-Vega, C. A.; Win, M. Z.; and Conti, A. 2023b. Neural Network Based Node Prioritization for Efficient Localization. In *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, 1–5.

Ha, D.; and Schmidhuber, J. 2018. Recurrent World Models Facilitate Policy Evolution. In *Advances in Neural Information Processing Systems 31*, 2451–2463. Curran Associates, Inc. https://worldmodels.github.io.

Hafner, D.; Pasukonis, J.; Ba, J.; and Lillicrap, T. 2025. Mastering diverse control tasks through world models. *Nature*, 1–7.

Hajiakhondi-Meybodi, Z.; Mohammadi, A.; Hou, M.; and Plataniotis, K. N. 2022. DQLEL: Deep Q-Learning for Energy-Optimized LoS/NLoS UWB Node Selection. *IEEE Transactions on Signal Processing*, 70: 2532–2547.

Horn, B. K. 2020. Doubling the Accuracy of Indoor Positioning: Frequency Diversity. *Sensors*, 20(5).

Kim, D. H.; Park, J.; Ko, Y.-B.; and Choi, J. 2025. Deep Q-Network Based UWB Anchor and Strategy Selection for Accurate Pedestrian Localization in Vehicular Environments. In *2025 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*, 1–6.

Kingma, D. P.; and Welling, M. 2022. Auto-Encoding Variational Bayes. arXiv:1312.6114.

Lee, J.; Lee, Y.; Kim, J.; Kosiorek, A.; Choi, S.; and Teh, Y. W. 2019. Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 3744–3753. PMLR.

Li, Y.; Hu, X.; Zhuang, Y.; Gao, Z.; Zhang, P.; and El-Sheimy, N. 2020. Deep Reinforcement Learning (DRL): Another Perspective for Unsupervised Wireless Localization. *IEEE Internet of Things Journal*, 7(7): 6279–6287.

Liu, F.; Cui, Y.; Masouros, C.; Xu, J.; Han, T. X.; Eldar, Y. C.; and Buzzi, S. 2022. Integrated Sensing and Communications: Toward Dual-Functional Wireless Networks for 6G and Beyond. *IEEE Journal on Selected Areas in Communications*, 40(6): 1728–1767.

Mohammadi, M.; Al-Fuqaha, A.; Guizani, M.; and Oh, J.-S. 2018. Semisupervised Deep Reinforcement Learning in Support of IoT and Smart City Services. *IEEE Internet of Things Journal*, 5(2): 624–635.

Nkrow, R. E.; Silva, B.; Boshoff, D.; Hancke, G.; Gidlund, M.; and Abu-Mahfouz, A. 2024. NLOS Identification and Mitigation for Time-based Indoor Localization Systems: Survey and Future Research Directions. *ACM Comput. Surv.*, 56(12).

Oh, M. S.; Hosseinalipour, S.; Kim, T.; Love, D. J.; Krogmeier, J. V.; and Brinton, C. G. 2023. Dynamic and Robust Sensor Selection Strategies for Wireless Positioning With TOA/RSS Measurement. *IEEE Transactions on Vehicular Technology*, 72(11): 14656–14672.

Rong Li, X.; and Jilkov, V. 2003. Survey of maneuvering target tracking. Part I. Dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4): 1333–1364.

Trevlakis, S. E.; Boulogeorgos, A.-A. A.; Pliatsios, D.; Querol, J.; Ntontin, K.; Sarigiannidis, P.; Chatzinotas, S.; and Di Renzo, M. 2023. Localization as a Key Enabler of 6G Wireless Systems: A Comprehensive Survey and an Outlook. *IEEE Open Journal of the Communications Society*, 4: 2733–2801.

Tu, S.; Zhou, X.; Liang, D.; Jiang, X.; Zhang, Y.; Li, X.; and Bai, X. 2025. The role of world models in shaping autonomous driving: A comprehensive survey. *arXiv preprint arXiv:2502.10498*.

Wang, G.; Cheng, P.; Li, S.; Chen, Y.; Vucetic, B.; and Li, Y. 2025a. Self-Supervised Deep State Space Model for Enhanced Indoor Tracking. In *ICC 2025 - IEEE International Conference on Communications*, 692–697.

Wang, G.; Cheng, P.; Li, S.; Xiang, W.; Vucetic, B.; and Li, Y. 2025b. Label-Free Range-Based Indoor Tracking With Physics-Guided Deep State Space Model. Submitted to IEEE, in major revision.

Wang, G.; Li, S.; Cheng, P.; Vucetic, B.; and Li, Y. 2024. ToF-Based NLoS Indoor Tracking With Adaptive Ranging Error Mitigation. *IEEE Transactions on Signal Processing*, 72: 4855–4870.

Wang, T.; Conti, A.; and Win, M. Z. 2019. Network Navigation With Scheduling: Distributed Algorithms. *IEEE/ACM Transactions on Networking*, 27(4): 1319–1329.

Win, M. Z.; Dai, W.; Shen, Y.; Chrisikos, G.; and Vincent Poor, H. 2018. Network Operation Strategies for Efficient Localization and Navigation. *Proceedings of the IEEE*, 106(7): 1224–1254.

Yang, Y.; Chen, M.; Blankenship, Y.; Lee, J.; Ghassemlooy, Z.; Cheng, J.; and Mao, S. 2024. Positioning Using Wireless Networks: Applications, Recent Progress, and Future Challenges. *IEEE Journal on Selected Areas in Communications*, 42(9): 2149–2178.

Zhao, Y.; Li, Z.; Hao, B.; and Shi, J. 2019. Sensor Selection for TDOA-Based Localization in Wireless Sensor Networks With Non-Line-of-Sight Condition. *IEEE Transactions on Vehicular Technology*, 68(10): 9935–9950.