Real-Time Bayesian Neural Networks for 6G Cooperative Positioning and Tracking

Bernardo Camajori Tedeschini, *Graduate Student Member, IEEE*, Girim Kwon, *Member, IEEE*, Monica Nicoli, *Senior Member, IEEE*, and Moe Z. Win, *Fellow, IEEE*

***Abstract*— In the evolving landscape of 5G new radio and related 6G evolution, achieving centimeter-level dynamic positioning is pivotal, especially in cooperative intelligent transportation system frameworks. With the challenges posed by higher path loss and blockages in the new frequency bands (i.e., millimeter waves), machine learning (ML) offers new approaches to draw location information from space-time wide-bandwidth radio signals and enable enhanced location-based services. This paper presents an approach to real-time 6G location tracking in urban settings with frequent signal blockages. We introduce a novel teacher-student Bayesian neural network (BNN) method, called Bayesian bright knowledge (BBK), that predicts both the location estimate and the associated uncertainty in real-time. Moreover, we propose a seamless integration of BNNs into a cellular multi-base station tracking system, where more complex channel measurements are taken into account. Our method employs a deep learning (DL)-based autoencoder structure that leverages the complete channel impulse response to deduce location-specific attributes in both line-of-sight and non-line-of-sight environments. Testing in 3GPP specification-compliant urban micro (UMi) scenario with ray-tracing and traffic simulations confirms the BBK's superiority in estimating uncertainties and handling out-of-distribution testing positions. In dynamic conditions, our BNN-based tracking system surpasses geometric-based tracking techniques and state-of-the-art DL models, localizing a moving target with a median error of 46 cm.**

***Index Terms*— Bayesian neural networks, tracking, deep learning, channel impulse response, intelligent transportation systems.**

Manuscript received 11 November 2023; revised 19 March 2024; accepted 18 April 2024. Date of current version 21 August 2024. The fundamental research described in this paper was supported, in part, by the Roberto Rocca Doctoral Fellowship granted by Massachusetts Institute of Technology and Politecnico di Milano, by the European Union — NextGenerationEU (National Sustainable Mobility Center CN00000023, Italian Ministry of University and Research Decree n. 1033—17/06/2022, spoke 9), by the National Research Foundation of Korea under Grant 2021R1A6A3A14040142, by the National Science Foundation under Grant CNS-2148251, and by the federal agency and industry partners in the RINGS Program. The material in this paper will be presented in part at the IEEE International Conference on Communications, Denver, CO, USA, in June 2024. *(Corresponding author: Moe Z. Win.)*

Bernardo Camajori Tedeschini is with the Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano, 20133 Milan, Italy, and also with the Wireless Information and Network Sciences Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA.

Girim Kwon is with the Wireless Information and Network Sciences Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA.

Monica Nicoli is with the Department of Management, Economics and Industrial Engineering (DIG), Politecnico di Milano, 20156 Milan, Italy.

Moe Z. Win is with the Laboratory for Information and Decision Systems (LIDS), Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: moewin@mit.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSAC.2024.3413950.

Digital Object Identifier 10.1109/JSAC.2024.3413950

I. Introduction

POSITIONING and tracking capabilities have become increasingly crucial in the evolution of cellular networks, as they provide benefits to the 5th generation (5G) use cases [1], [2], [3]. Since the adoption of 3rd Generation Partnership Project (3GPP) Release 15 in 2018, these networks have not only achieved rapid development [4], [5], [6], but have also expanded to introduce new use cases and services [7]. Notably in 3GPP Releases 16 and 17, location awareness systems have been extended beyond from regulatory applications to commercial and roaming functionalities [8], [9], [10]. Still, the major leap forward in positioning performances is expected with the advent of 5G Advanced in 3GPP Release 18 [11], [12], [13], [14], where the primary goal of centimeter-level absolute accuracy will be achieved thanks to transformative key features enablers, i.e., massive multiple-input multiple-output (mMIMO) [15], larger bandwidths and millimeter waves (mmWave) [16]. The main challenges are the higher path loss and frequent blockages, which limit the potential of conventional and global navigation satellite systems (GNSS)-based solutions. Indeed, under the absence of line-of-sight (LOS) link, GNSS becomes challenging to utilize effectively, even with advanced satellite techniques such as real-time kinematic (RTK) [17].

Extensive research in the domain of localization and navigation has explored various aspects of these challenges, focusing on fundamental limits [18], [19], [20], [21], network operation and experimentation [22], [23], [24], [25], and algorithm design [26], [27], [28], [29]. To solve these challenges, 5G Advanced has pushed interest in leveraging artificial intelligence (AI) and machine learning (ML) for assisted or even direct positioning [30]. Indeed, base stations (BSs) often have access to a large number of historical channel state information (CSI) [31], [32], [33], which can be exploited through deep learning (DL) methods (e.g., autoencoder (AE) structures [34]) as location fingerprints [35]. The advantages of direct artificial intelligence (AI)/machine learning (ML) positioning are the ability to perform both LOS and non-LOS (NLOS) positioning, and single-BS localization, enabled by integrated sensing and communication (ISAC) frameworks [36], [37], [38]. Therefore, we foresee these solutions as a promising long-term answer to advanced positioning methods.

Despite the potential of ML in positioning applications, traditional ML approaches have limitations, especially concerning uncertainty quantification. In critical applications, such as tracking of connected automated vehicles (CAVs) [39],

[40], [41], lack of uncertainty quantification can be a major limiting or even blocking factor as integrity and reliability requirements are very stringent. From this point of view, Bayesian neural networks (BNNs) offer a promising solution to these challenges [42], as they not only provide point estimates but also quantify the uncertainty associated with these estimates [43], allowing for more reliable and robust positioning. In particular, in static positioning, BNNs are more resilient to overfitting during training with respect to neural networks (NNs), they can incorporate prior knowledge on the problem at hand and, more importantly, are able to characterize the uncertainty of the model (i.e., whether it is due to the lack of training samples or to the intrinsic characteristics of the data). Furthermore, in dynamic settings, i.e., tracking, the uncertainty can be cleverly combined in Bayesian tracking solutions.

While BNNs address key limitations of conventional ML algorithms, they also present specific challenges. A main drawback is the need for sampling during inference time, which may not be suitable for real-time applications [44]. Another issue is the computational and storage overhead of maintaining multiple NN configurations for Bayesian inference. Given the lack of a complete solution to these problems and the great potential of BNNs in cellular tracking systems, in this paper, we explore and propose the integration of real-time BNN solutions into future 6G systems. These advancements aim to bring together the best of both worlds: the robustness and uncertainty quantification of BNNs along with the speed and efficiency required for next-generation tracking systems.

The rest of this paper is structured as follows. Sec. II presents the related works including next-generation cellular positioning, ML for static and mobile positioning, and BNNs for real-time inference, along with the paper's main contributions. Sec. III describes the channel model and the channel fingerprint used for positioning. In Sec. IV, we discuss the proposed real-time BNN method and its application to a DL model with AE structure. Sec. V presents the integration of BNN methods in next-generation cellular tracking systems. Sec. VI presents a case study, and Sec. VII draws the conclusions.

II. Related Works and Contributions

A. Next Generation Cellular Positioning

3GPP Release 18 is expected to significantly enhance the existing positioning standards by introducing key methodologies. It introduces the support of carrier phase positioning (CPP), a GNSS-based technology known for its centimeter-level precision, though it is traditionally limited to outdoor use where GNSS signals are not blocked [45]. Moreover, Release 18 is set to fulfill low power high accuracy positioning (LPHAP) standards and introduce positioning features for reduced capability (RedCap) user equipments (UEs), such as wearable medical devices, and augmented reality goggles. Finally, it comprises studies on sidelink (SL) positioning, e.g., the design of SL-positioning reference signal (PRS) [46].

However, in case of non-cooperative and non-RedCap UEs moving in frequent blockage environment, e.g., CAVs in urban scenario, geometric-based positioning methodologies struggle in fulfilling the requirements. For this reason, AI-based solutions have been studied, particularly regarding complexity, positioning performances, and generalizations [11].

B. ML for Static Positioning

Significant works on ML-based wireless positioning were developed with the introduction of MIMO-orthogonal frequency division multiplexing (OFDM) systems in the IEEE 802.11a/n protocol, which enabled the extraction of CSI from commercial Wi-Fi devices. The availability of channel information across multiple carriers and antennas allowed for detailed insights into radio signal propagation, enabling learning of the user's position [47], [48]. DL techniques were utilized to learn the best non-linear combination of features for tasks like NLOS classification or position estimation, with many studies adopting convolutional neural networks (CNNs) for feature extraction [49], [50], [51], [52].

Regarding 5G positioning, gathering the knowledge of Wi-Fi works, two main components started to delineate: the usage of full channel impulse response (CIR) data as input features for positioning and the adoption of AE NN structures. Leveraging full CIR data, especially when organized into image-like structures, is gaining a lot of momentum. Authors in [31] adopted the channel frequency response (CFR) matrix obtained both from ray-tracing simulations and real experiments. However, while being effective for positioning, the CFR does not distinctly represent the angle of arrival (AOA) or time of flight (TOF) for each path, potentially complicating feature extraction. A different study employed a 3D angle-delay channel power matrix (ADCPM) and a CNN with inception modules to predict UE position [53], but with double the inference time and four times the computational storage compared to 2D ADCPM. Dealing with image structure permits employing more complex DL models which compress the channel into a compact and efficient representation, namely AE.

C. ML for Mobile Positioning and Tracking

UE tracking by ML-based methods in the context of 5G networks is a relatively unexplored area since the majority of previous works employed conventional Bayesian techniques, e.g., extended Kalman filter (EKF) [54] or message passing algorithm (MPA) [55], in conjunction with mmWave and MIMO enablers. Authors in [56] adopted state-of-the-art temporal convolutional network (TCN) models to perform NLOS outdoor tracking, reaching a mean absolute error (MAE) of 1.8 m. A similar work has been carried out in indoor conditions [57], with long short-term memory (LSTM) and CNN applied to raw CSI fingerprinting. However, LSTMs and TCNs have two main drawbacks. First, they require a set of training trajectories with highly accurate ground truth positions. While this is practical for static positioning, for

dynamic positioning, especially in outdoor conditions, it is difficult to obtain the ground truth target position while moving, unless using high-precision optical laser positioning systems. Second, the conventional LSTMs and TCNs do not provide an uncertainty measure of their predictions, thus limiting the deployment in safety-critical applications.

D. BNN for Real-Time Inference

In the context of DL-based uncertainty estimation, it is essential to distinguish between aleatoric and epistemic uncertainties [58]. These two types of uncertainties are the roots behind the prediction uncertainty and they are generated by two different phenomena. Aleatoric uncertainty refers to the dispersion of the predicted distribution of our target variable, e.g., UE location, based on the given features, which arises from measurement inaccuracies. Therefore, aleatoric uncertainty remains unchanged even with additional data collection under identical experimental conditions. This specific uncertainty, referred to as *data uncertainty*, can be made data-dependent and learned as an additional output from conventional NNs [59]. Thus, these solutions are important for real-time applications, where no Monte Carlo (MC) sampling procedures are required, but fail to generalize in out-of-distribution (OOD) scenarios with sparse training data.

For this reason, BNNs and epistemic uncertainty evaluation have been studied [60]. The epistemic uncertainty, also known as *model uncertainty*, derives from the uncertainty over the BNN model parameters, which are considered random variables. Contrary to the aleatoric uncertainty, the ambiguity on the model parameters, and thus on the output, can be explained away by providing more training samples. Existing BNN methods, both exact methods like Markov chain Monte Carlo (MCMC) [61], [62] and approximations like variational inference (VI) [63], [64], [65], [66], [67], aim at effectively sampling from the posterior distribution of the weights and predicting the posterior predictive distribution over the output. This is done by having multiple NN parameters instances and by predicting multiple times the same input sample.

Many works tried to tackle the problem of simultaneous learning of aleatoric and epistemic uncertainties. The authors in [59] adopted a BNN derived from Monte Carlo Dropout (MCDropout) method which also predicted the aleatoric uncertainty of data through a specific loss function. Despite achieving good results, this method still relies on a sampling inference procedure, thus not being suitable for real-time applications. A solution to this issue can be found in the so-called *teacher-student* techniques, such as Bayesian dark knowledge (BDK) [68], where a student NN, i.e., non-Bayesian, is trained to mimic the output of a teacher BNN, which is on the other hand Bayesian thus learning both the points estimates and the output uncertainties. During the inference phase, the real-time uncertainty estimation is performed by the student NN without requiring time-consuming sampling procedures. The primary challenge in teacher-student methods is the student's inability to differentiate between the two distinct uncertainties in output. Distinguishing between these uncertainties is crucial, as it provides insights into the reasons for a model's uncertainty regarding a specific test sample, such as insufficient training data points or inherent data noise. Moreover, recognizing these uncertainties can guide where additional training points would be most beneficial (i.e., wherever high epistemic uncertainty and low aleatoric uncertainty are present). At the present day and to the authors' knowledge, no real-time solution to both aleatoric and epistemic uncertainty learning is present in the literature (see Table I), in particular regarding safety-critical applications such as automated driving.

TABLE I
COMPARISON OF METHODS FOR REAL-TIME, ALEATORIC AND EPISTEMIC PREDICTION CAPABILITIES

	Real-time prediction	Aleatoric prediction	Epistemic prediction
MCMC/VI [61]–[67]	No	No	Yes
Bayesian & aleatoric regres. [59]	No	Yes	Yes
BDK [68]	Yes	Yes	No
BBK (proposed)	Yes	Yes	Yes

E. Contributions

In this paper, we address the problem of UE tracking in next-generation networks through the usage of the full CIR and DL-based predictions, whose output uncertainty is obtained through real-time BNN techniques and seamlessly integrated into existing tracking systems. The real-time BNN methodology, namely Bayesian bright knowledge (BBK), is built with a teacher-student paradigm as it is the only methodology of BNN which does not require sampling for producing the uncertainty estimation. The localization system is based on a NN trained on offline gathered data, which are limited by spatial density. To account for both the aleatoric and epistemic uncertainties caused by noisy measurements and limited density of the training points, respectively, we propose a Bayesian tracking approach based on a BNN. This BNN constructs the model that links measurements and positions, optimally predicting and weighing uncertainties in the position calculation.

The main contributions of this paper are summarized in the following.

- We design a teacher-student BNN method, i.e., BBK, that predicts both epistemic and aleatoric uncertainties without requiring a sampling procedure during inference phase. This makes it suitable for real-time and safety-critical use cases.
- We propose the integration of BNNs in cellular tracking systems where a set of cooperative BSs aims at tracking a moving target. The integrated system is easy to be implemented and is compatible with any BNN method.
- We develop a DL model based on an AE structure which exploits the complete CIR, i.e., sparse ADCPM matrices, to extract location-specific features and perform positioning in both LOS and NLOS settings.
- We model a realistic cooperative intelligent transportation system (C-ITS) setting in an urban context. Our simulated network aligns with the 5G standard [69] and offers realistic outdoor conditions using Wireless InSite 3D ray-tracing software and MATLAB software. We emulate

various vehicle trajectories, or UEs, designed with the simulation of urban mobility (SUMO) software [70].

1) Notation: A random variable and its realization are denoted by x and x; a random vector and its realization are denoted by $\mathbf{x}$ and $\boldsymbol{x}$; a random matrix and its realization are denoted by $\mathbf{X}$ and $\boldsymbol{X}$, respectively. The function $p_{\mathsf{x}}(x)$, and simply $p(x)$ when there is no ambiguity, denotes the probability density function (PDF) of x. $j = \sqrt{-1}$ denotes the imaginary unit. The notations $\mathbf{X}^\top$, $\mathbf{X}^*$ and $\mathbf{X}^{\mathrm{H}}$ indicate the matrix transposition, conjugation and conjugate transposition. $\det(\cdot)$ and $\mathrm{Tr}(\cdot)$ denote the determinant and the trace of the matrix argument, respectively. The Kronecker and Hadamard products between two matrices are denoted with $\otimes$ and $\odot$, respectively. With the notation $\mathsf{x} \sim \mathcal{N}(\mu, \sigma^2)$ we indicate a Gaussian random variable x with mean μ and standard deviation σ, whose PDF is denoted by $\mathcal{N}(x; \mu, \sigma^2)$. With the notation $\mathsf{y} \sim \mathcal{U}(a, b)$ we indicate a uniform random variable y with support $[a, b]$. We use $\mathbb{E}\{\cdot\}$ and $\mathbb{V}\{\cdot\}$ to denote the expectation and the variance of random variable, respectively. $\mathbb{R}$ and $\mathbb{C}$ stand for the set of real and complex numbers, respectively. $\lfloor x \rfloor$ indicates the largest integer not greater than x. $|\boldsymbol{x}|$ denotes the length of the vector $\boldsymbol{x}$. $\delta(\cdot)$ is the Kronecker delta function.

III. System Model

A. Channel Model

Consider a mmWave OFDM system where a UE communicates with a BS in uplink direction at carrier wavelength λ_{c}. The BS is equipped with a uniform planar array (UPA) composed of $N_{\mathrm{v}} \times N_{\mathrm{h}}$ isotropic antenna elements (i.e., N_{v} and N_{h} elements in the vertical and horizontal directions with the antenna spacings of d_{v} and d_{h}, respectively). On the contrary, the UE holds a single omni-directional antenna. The channel between the BS and the UE (see Fig. 1) is composed of N_{p} distinct paths, each with a TOF τ_p, a zenith AOA $\theta_p \in [0, \pi]$, and an azimuth AOA $\varphi_p \in [0, \pi]$, for path $p = 1, 2, \ldots, N_{\mathrm{p}}$.

We employ an OFDM scheme with a sampling interval of T_{s}, N_{c} sub-carriers, and a symbol duration given by $T_{\mathrm{c}} = N_{\mathrm{c}} T_{\mathrm{s}}$. For the k-th sub-carrier, the frequency is $f_k = \frac{k}{T_{\mathrm{c}}}$, $k = 0, 1, \ldots, N_{\mathrm{c}} - 1$. We assume that the cyclic-prefix duration $T_{\mathrm{g}} = N_{\mathrm{g}} T_{\mathrm{s}}$ surpasses the maximum channel delay, denoted by τ_{MAX}. Here, N_{g} represents the number of sampling intervals constituting a guard interval.

Assuming a sampling rate of $1/T_{\mathrm{s}}$ and treating each path as independent and wide-sense stationary [71], the CFR for the k-th sub-carrier can be expressed as [72], [73]

$$\mathbf{h}_k = \sum_{p=1}^{N_{\mathrm{p}}} \bar{\alpha}_{p,k}\, \mathbf{e}(\theta_p, \varphi_p) \in \mathbb{C}^{N_{\mathrm{h}} N_{\mathrm{v}}} \tag{1}$$

where $\bar{\alpha}_{p,k} = \alpha_p e^{-j2\pi\tau_p f_k}$ is the channel gain in the frequency domain, $\alpha_p = \mathsf{a}_p e^{-j2\pi(\frac{\mathsf{d}_p}{\lambda_{\mathrm{c}}} - \nu_p \tau_p)}$ is the complex path gain of which includes the Doppler frequency shift ν_p and has average power $\sigma_p^2 = \mathbb{E}\{|\mathsf{a}_p|^2\}$ and $\mathsf{d}_p = c\tau_p$ is the traveled distance (where c is the speed of light in air), and $\mathbf{e}(\theta_p, \varphi_p) \in \mathbb{C}^{N_{\mathrm{h}} N_{\mathrm{v}} \times 1}$ is the array response vector [71]. Finally, by considering the different CFRs at every sub-carrier, we get the space-frequency channel response matrix (SFCRM) as:

$$\mathbf{H} = [\mathbf{h}_0\ \mathbf{h}_1\ \cdots\ \mathbf{h}_{N_{\mathrm{c}}-1}] \in \mathbb{C}^{N_{\mathrm{h}} N_{\mathrm{v}} \times N_{\mathrm{c}}}. \tag{2}$$

In the next subsection, we show how to extract the ADCPM fingerprint from (2) obtained at the BS.

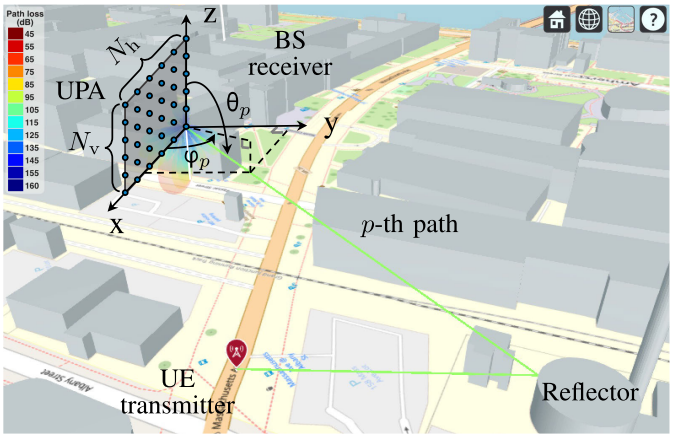


Fig. 1. An uplink scenario with a UE transmitting to a BS where the direction of arrival (DOA) of the p-th path is highlighted with zenith θ_p and azimuth φ_p angles.

B. Location-Dependent Fingerprint

For location estimation, it is advantageous to map the channel response into the angle-delay domain. This transformation simplifies the identification of macro-paths, i.e., clusters for both LOS and NLOS components, which vary with the environmental context, serving as location-specific features or fingerprints. To obtain angle-delay domain features, we convert the SFCRM as defined in (2) by employing phase-shifted discrete Fourier transform (DFT) matrices $\boldsymbol{V}_{N_{\mathrm{h}}} \in \mathbb{C}^{N_{\mathrm{h}} \times N_{\mathrm{h}}}$ and $\boldsymbol{V}_{N_{\mathrm{v}}} \in \mathbb{C}^{N_{\mathrm{v}} \times N_{\mathrm{v}}}$, where $[\boldsymbol{V}_{N_{\mathrm{h}}}]_{\bar{i},\bar{j}} = \frac{1}{\sqrt{N_{\mathrm{h}}}} e^{-j2\pi \frac{\bar{i}\left(\bar{j} - \frac{N_{\mathrm{h}}}{2}\right)}{N_{\mathrm{h}}}}$ and $[\boldsymbol{V}_{N_{\mathrm{v}}}]_{\bar{i},\bar{j}} = \frac{1}{\sqrt{N_{\mathrm{v}}}} e^{-j2\pi \frac{\bar{i}\left(\bar{j} - \frac{N_{\mathrm{v}}}{2}\right)}{N_{\mathrm{v}}}}$. Moreover, we denote with $\boldsymbol{F} \in \mathbb{C}^{N_{\mathrm{c}} \times N_{\mathrm{g}}}$ the matrix formed by the first N_{g} columns of N_{c} dimensional unitary DFT matrix where $[\boldsymbol{F}]_{\bar{i},\bar{j}} = \frac{1}{\sqrt{N_{\mathrm{c}}}} e^{-j2\pi \frac{\bar{i}\bar{j}}{N_{\mathrm{c}}}}$. The angle-delay channel response matrix (ADCRM) can be then computed as [53]

$$\mathbf{G} = \frac{1}{\sqrt{N_{\mathrm{h}} N_{\mathrm{v}} N_{\mathrm{c}}}} (\boldsymbol{V}_{N_{\mathrm{h}}}^{\mathrm{H}} \otimes \boldsymbol{V}_{N_{\mathrm{v}}}^{\mathrm{H}})\, \mathbf{H}\, \boldsymbol{F}^* \in \mathbb{C}^{N_{\mathrm{h}} N_{\mathrm{v}} \times N_{\mathrm{g}}} \tag{3}$$

where $\boldsymbol{V}_{N_{\mathrm{h}}}^{\mathrm{H}} \otimes \boldsymbol{V}_{N_{\mathrm{v}}}^{\mathrm{H}}$ and $\boldsymbol{F}^*$ project the SFCRM into the angle and delay domains, respectively.

From (3), we can obtain the ADCPM as

$$\boldsymbol{P} = \mathbb{E}\{\mathbf{G} \odot \mathbf{G}^*\} \in \mathbb{R}^{N_{\mathrm{h}} N_{\mathrm{v}} \times N_{\mathrm{g}}} \tag{4}$$

where $[\boldsymbol{P}]_{\bar{i},\bar{j}} = \mathbb{E}\{\left|[\mathbf{G}]_{\bar{i},\bar{j}}\right|^2\}$. An important property of the ADCPM is that for N_{v}, N_{h} and $N_{\mathrm{g}} \to \infty$, the ADCPM becomes a sparse matrix whose elements $[\boldsymbol{P}]_{\bar{i},\bar{j}}$ match the average channel power of the $\bar{i}$-th AOA and the $\bar{j}$-th TOF as [53]

$$\lim_{N_{\mathrm{h}}, N_{\mathrm{v}}, N_{\mathrm{g}} \to \infty} [\boldsymbol{P}]_{\bar{i},\bar{j}} = \sum_{p=1}^{N_{\mathrm{p}}} \sigma_p^2\, \delta(\bar{i} - m_p N_{\mathrm{v}} - n_p)\, \delta(\bar{j} - r_p) \tag{5}$$

where $r_p = \lfloor \frac{\tau_p}{T_s} \rfloor$ is the resolvable delay corresponding to the p-th path, $n_p = \frac{N_v}{2} + \frac{N_v d_v}{\lambda_c} \cos\theta_p$ and $m_p = \frac{N_h}{2} + \frac{N_h d_h}{\lambda_c} \sin\theta_p \cos\phi_p$. Hence, the statistical properties of the ADCPM facilitate the DL model's ability to capture location-specific attributes, providing consistent and reliable fingerprints for location estimation.

C. DL Model Input

We propose using the ADCPM in (4) as the measurement basis for estimating the UE location. This sparse matrix effectively serves as a visual snapshot of the multipath environment in the power-angle-delay domain, from which a DL model like a CNN can glean key location-centric features. In addition, the ADCPM encapsulates all essential information, i.e., TOF, AOA, and received signal strength (RSS) for each path, while maintaining low storage and computational requirements due to channel sparsity. To illustrate this aspect, in Fig. 2(b) we show an example ADCPM denoted by $\boldsymbol{P}$, comprising $N_g = 352$ delay samples and $N_h N_v = 64$ angular directions. Notably, even without an extensive array of antennas or high sample resolution, the sparsity of the matrix is evident. Therefore, in the experiment, we employ the ADCPM $\boldsymbol{P}$ as the DL model's input $\mathbf{x}$ for performing tracking.

Despite the ability of the ADCPM to provide consistent and reliable fingerprints for location estimation, we need to face the practical challenges of noisy measurements, i.e., the non-perfect matching between the input ADCPM and the related position due to multipath fading and channel estimate errors, and the limited spatial density of training points. In particular, these two challenges create two types of uncertainties, namely, aleatoric and epistemic uncertainties, respectively, which we propose to assess through the usage of BNN. Therefore, in Sec. IV, we detail how to train a BNN to obtain both the position and its related uncertainties.

IV. Real-Time BNN Methodology

A. Problem Formulation

We consider a supervised regression setting where the UE's position is defined by the target variable $\mathtt{t}$, which is considered as a scalar here to simplify the derivation and modeled as

$$\mathtt{t} = f(\mathbf{x}) + \boldsymbol{\varepsilon}(\mathbf{x}) \tag{6}$$

where $f(\mathbf{x})$ is a non-linear function which takes as input $\mathbf{x}$ (e.g., a channel measurement as the ADCPM) and $\boldsymbol{\varepsilon}(\mathbf{x}) \sim \mathcal{N}\big(0, \sigma_{\boldsymbol{\varepsilon}}(\mathbf{x})^2\big)$ is a random noise. Extension to a vector target variable (i.e., 3D position) is in Sec. IV-E. The objective is to approximate the function $f(\mathbf{x})$ to $y(\mathbf{x}, \boldsymbol{\theta})$ using a NN with parameters $\boldsymbol{\theta}$. The vector $\boldsymbol{\theta}$ is learned using N training points composing a training dataset $\mathcal{D} = \{(t_n, \boldsymbol{x}_n) \mid t_n \in \mathcal{D}_t, \boldsymbol{x}_n \in \mathcal{D}_{\boldsymbol{x}}\}_{n=1}^{N}$, where $\mathcal{D}_t$ and $\mathcal{D}_{\boldsymbol{x}}$ contain the training targets and inputs, respectively. From the training dataset, we define the likelihood function as:

$$p_{\mathsf{D}_t|\mathsf{D}_x,\boldsymbol{\theta}}(\mathcal{D}_t|\mathcal{D}_{\boldsymbol{x}}, \boldsymbol{\theta}) = \prod_{n=1}^{N} \mathcal{N}\big(t_n; y(\boldsymbol{x}_n, \boldsymbol{\theta}), \sigma_{\boldsymbol{\varepsilon}}(\boldsymbol{x}_n)^2\big) \tag{7}$$

assuming independence between target variables.

In non-Bayesian ML settings, a discriminative probabilistic approach is adopted [74], where $\boldsymbol{\theta}$ are obtained via maximum likelihood estimation (MLE) and by directly defining the posterior conditional probability as $p_{\mathsf{t}|\mathbf{x},\mathsf{D}}(t|\boldsymbol{x}, \mathcal{D}) = p_{\mathsf{t}|\mathbf{x},\boldsymbol{\theta}}(t|\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{MLE}})$, where $\boldsymbol{\theta}_{\mathrm{MLE}} = \mathrm{argmin}_{\boldsymbol{\theta}}\{-\log p(\mathcal{D}_t|\mathcal{D}_{\boldsymbol{x}}, \boldsymbol{\theta})\}$ are obtained with gradient descent optimization methods. Usually, the negative log-likelihood $-\log p(\mathcal{D}_t|\mathcal{D}_{\boldsymbol{x}}, \boldsymbol{\theta})$ is called loss or error function. On the contrary, in Bayesian settings, the network is stochastic and is described by random parameters with prior distribution $p_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ accounting for the uncertainty of the model due to the finite size of the training dataset. In Bayesian NNs, a generative approach is employed by computing the so-called *posterior predictive distribution* [43]:

$$p_{\mathsf{t}|\mathbf{x},\mathsf{D}}(t|\boldsymbol{x}, \mathcal{D}) = \int p_{\mathsf{t}|\mathbf{x},\boldsymbol{\theta}}(t|\boldsymbol{x}, \boldsymbol{\theta})\, p_{\boldsymbol{\theta}|\mathsf{D}}(\boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta} \tag{8}$$

where $p_{\boldsymbol{\theta}|\mathsf{D}}(\boldsymbol{\theta}|\mathcal{D})$ is the posterior distribution. It is called *predictive* as it is used to make predictions on new, unseen data, and to differentiate it with respect to the posterior distribution over the model parameters. However, in practice, the posterior $p_{\boldsymbol{\theta}|\mathsf{D}}(\boldsymbol{\theta}|\mathcal{D})$ is highly dimensional and non-convex, leading to computational intractability. Thus, the majority of BNN methods approximate $p_{\boldsymbol{\theta}|\mathsf{D}}(\boldsymbol{\theta}|\mathcal{D})$ with a sampling procedure and estimate the (8) with MC sampling as

$$p_{\mathsf{t}|\mathbf{x},\mathsf{D}}(t|\boldsymbol{x}, \mathcal{D}) \backsimeq \frac{1}{L} \sum_{\ell=1}^{L} p(t|\boldsymbol{x}, \boldsymbol{\theta}_\ell) \tag{9}$$

where L is the number of samples $\boldsymbol{\theta}_\ell$ drawn from $p_{\boldsymbol{\theta}|\mathsf{D}}(\boldsymbol{\theta}|\mathcal{D})$.

B. Predictive Mean and Variance Estimation

Given the posterior predictive distribution in (9), we can obtain the predictive mean of $\mathtt{t}$ as

$$\mathbb{E}\{\mathsf{t}|\mathbf{x}, \mathsf{D}\} \backsimeq \frac{1}{L} \sum_{\ell=1}^{L} \int t\, p(t|\boldsymbol{x}, \boldsymbol{\theta}_\ell) dt \backsimeq \frac{1}{L} \sum_{\ell=1}^{L} y(\boldsymbol{x}, \boldsymbol{\theta}_\ell)\,. \tag{10}$$

For the estimation of the variance, i.e., uncertainty of the prediction of $\boldsymbol{x}$, we need to distinguish between two types of uncertainties, the aleatoric and the epistemic uncertainties. The former derives from the generation of data in (6). Since all the training points in $\mathcal{D}$ contain a realization of the noise $\boldsymbol{\varepsilon}(\mathbf{x})$, this uncertainty is intrinsic within the data and it cannot be reduced by providing more training samples. Still, since it is data-dependent, it can be learned by the NN through a specific loss function with the model $\sigma_{\boldsymbol{\varepsilon}}(\mathbf{x})^2 = y_{\mathrm{al}}(\mathbf{x}, \boldsymbol{\theta}) + \xi_{\mathrm{al}}$ [59], where $y_{\mathrm{al}}(\boldsymbol{x}, \boldsymbol{\theta})$ is an additional NN output which predicts the aleatoric uncertainty of $\boldsymbol{x}$, and $\xi_{\mathrm{al}} \sim \mathcal{N}(0, {\sigma_{\xi_{\mathrm{al}}}}^2)$.

On the contrary, the epistemic uncertainty derives from the uncertainty over the NN parameters, i.e., random variables $\boldsymbol{\theta}$, which contributes to the uncertainty measure in output. In conventional NN, given their point estimate, this uncertainty is zero, while in BNN it can be explained away by providing more training data [42]. In our setting, the total variance predicted by the BNN can

be written as

$$\begin{aligned}\mathbb{V}\{\mathsf{t}|\mathbf{x},\mathsf{D}\} &\backsimeq \frac{1}{L}\sum_{\ell=1}^{L}\int \big(t-\mathbb{E}\{\mathsf{t}|\mathbf{x},\mathsf{D}\}\big)^2 p(t|\boldsymbol{x},\boldsymbol{\theta}_\ell)dt \\ &\backsimeq \frac{1}{L}\sum_{\ell=1}^{L} y(\boldsymbol{x},\boldsymbol{\theta}_\ell)^2 - \left(\frac{1}{L}\sum_{\ell=1}^{L} y(\boldsymbol{x},\boldsymbol{\theta}_\ell)\right)^2 \\ &\quad + \frac{1}{L}\sum_{\ell=1}^{L} y_{\mathrm{al}}(\boldsymbol{x},\boldsymbol{\theta}_\ell) \end{aligned} \tag{11}$$

where the first two terms of (11) represent the epistemic uncertainty prediction, while the last term is the aleatoric uncertainty prediction.

C. Real-Time BNN

While BNNs offer the significant advantage of quantifying uncertainty in NN predictions, which is fundamental in applications for critical scenarios, the requirement to perform multiple inferences for total variance estimation (i.e., sample averaging in (11)) remains a substantial drawback. This can be resolved using non-Bayesian NNs trained to learn the aleatoric uncertainty as described in [59]. However, a second issue, which arises from the usage of NNs for uncertainty estimation, is the inability of predicting the epistemic uncertainty, that is, the incapacity of distinguishing between epistemic and aleatoric uncertainty. This calls for a BNN method that performs real-time inference and, simultaneously it is able to distinguish between epistemic and aleatoric uncertainty.

The first issue is present in all conventional BNN approaches such as VI, e.g., MC-Dropout [67] and Bayes by backpropagation (BBP) [66], which sample the parameters from an approximation of the posterior $p_{\boldsymbol{\theta}|\mathsf{D}}(\boldsymbol{\theta}|\mathcal{D})$, and MCMC methods, e.g., stochastic gradient Langevin dynamics (SGLD) [62], which directly sample from the real posterior. Moreover, the MCMC methods require storing all the sample parameters $\boldsymbol{\theta}_\ell$, which may not be feasible during deployment. A class of BNN which is able to perform real-time inference are the so-called teacher-student methods, e.g., BDK [68], where a conventional NN, i.e., student NN, is trained to approximate the behaviour, i.e., the output, of a teacher BNN. The term *dark knowledge* in BDK was introduced to denote the hidden information within the teacher network that can subsequently be transferred to the student. Considering the BDK teacher (T)-student (S) method, the model between input and output now becomes

$$\begin{aligned} &(\mathrm{T}): \quad \mathsf{t} = y^{(\mathrm{T})}(\mathbf{x},\boldsymbol{\theta}) + \boldsymbol{\varepsilon}^{(\mathrm{T})} \\ &(\mathrm{S}): \quad \begin{cases} \mathsf{t} & = y^{(\mathrm{S})}(\mathbf{x},\boldsymbol{w}) + \boldsymbol{\varepsilon}^{(\mathrm{S})}(\mathbf{x}) \\ \boldsymbol{\sigma}_{\boldsymbol{\varepsilon}^{(\mathrm{S})}}(\mathbf{x})^2 & = y_{\mathrm{al}}^{(\mathrm{S})}(\mathbf{x},\boldsymbol{w}) + \boldsymbol{\xi}_{\mathrm{al}}^{(\mathrm{S})} \end{cases} \end{aligned} \tag{12}$$

where $\boldsymbol{w}$ are the deterministic parameters of the student, $\boldsymbol{\varepsilon}^{(\mathrm{T})} \sim \mathcal{N}(0,{\sigma_{\boldsymbol{\varepsilon}^{(\mathrm{T})}}}^2)$, $\boldsymbol{\varepsilon}^{(\mathrm{S})}(\mathbf{x}) \sim \mathcal{N}(0,\sigma_{\boldsymbol{\varepsilon}^{(\mathrm{S})}}(\mathbf{x})^2)$ and $\boldsymbol{\xi}_{\mathrm{al}}^{(\mathrm{S})} \sim \mathcal{N}(0,{\sigma_{\boldsymbol{\xi}_{\mathrm{al}}^{(\mathrm{S})}}}^2)$. Note that the parameters of the teacher $\boldsymbol{\theta}$ are stochastic, whereas the parameters of the student $\boldsymbol{w}$ are deterministic. During training, the teacher is trained as a regular BNN, while the student learns the output of the teacher using a Kullback-Leibler (KL) divergence loss function as [68]

$$\begin{aligned} J(\boldsymbol{w}|\boldsymbol{x}) &= \mathrm{KL}\big(p(t|\boldsymbol{x},\mathcal{D}) \,\|\, p(t|\boldsymbol{x},\boldsymbol{w})\big) \\ &= \int p(t|\boldsymbol{x},\mathcal{D}) \log \frac{p(t|\boldsymbol{x},\mathcal{D})}{p(t|\boldsymbol{x},\boldsymbol{w})} dt \\ &\backsimeq -\frac{1}{L}\sum_{\ell=1}^{L} \mathbb{E}_{p(t|\boldsymbol{x},\boldsymbol{\theta}_\ell)}\big\{\log p(t|\boldsymbol{x},\boldsymbol{w})\big\}. \end{aligned} \tag{13}$$

On the contrary, during inference, we discard the teacher model and we just keep $y^{(\mathrm{S})}(\boldsymbol{x},\boldsymbol{w})$ as the predictive mean and $y_{\mathrm{al}}^{(\mathrm{S})}(\boldsymbol{x},\boldsymbol{w})$ as the predictive variance. Since the efficacy of this method relies on the teacher training, usually a MCMC method, such as SGLD, is employed [68].

D. Proposed Bayesian Bright Knowledge Method

One drawback of BDK is that the student does not have any knowledge of the epistemic uncertainty of the teacher, since it only outputs the aleatoric uncertainty through $y_{\mathrm{al}}^{(\mathrm{S})}(\boldsymbol{x},\boldsymbol{w})$. Distinguishing between aleatoric and epistemic uncertainties is crucial for several reasons. Firstly, it sheds light on the reasons behind a DL model's uncertainty regarding a particular test sample, which might be due to either insufficient training data or the inherent noise present in the data. Secondly, pinpointing the origin of uncertainty facilitates effective data acquisition by guiding the process towards where collecting additional training samples would be most beneficial. Lastly, considering both kinds of uncertainties allows for a thorough assessment of the total uncertainty associated with a prediction. Another drawback of BDK is that the loss function for regression derived from (13) is such that the student is trained only with one teacher-parameter sample $\boldsymbol{\theta}_\ell$ at the time [68]. In other words, in BDK, the teacher and student are trained sequentially using a stochastic version of (13) where a single parameter sample $\boldsymbol{\theta}_\ell$ is used at each step. On the contrary, we aim at exploiting the full KL loss in (13) obtained by averaging the teacher's output over the L samples.

In order to solve these issues, we propose to employ a student NN that predicts not only the aleatoric uncertainty, but also the epistemic uncertainty. While on the one hand a stand-alone NN is not able to output the uncertainty of the prediction since the weights are deterministic, on the other hand, by approximating the epistemic uncertainty of an existing teacher BNN, we are able to fully capture and distinguish between uncertainties on the data and on the parameters. Since all the information is transferred from the teacher to the student, we call this strategy Bayesian bright knowledge (BBK). The proposed model is

$$\begin{aligned} &(\mathrm{T}): \quad \mathsf{t} = y^{(\mathrm{T})}(\mathbf{x},\boldsymbol{\theta}) + \boldsymbol{\varepsilon}^{(\mathrm{T})} \\ &(\mathrm{S}): \quad \begin{cases} \mathsf{t} & = y^{(\mathrm{S})}(\mathbf{x},\boldsymbol{w}) + \boldsymbol{\varepsilon}^{(\mathrm{S})}(\mathbf{x}) \\ \boldsymbol{\sigma}_{\boldsymbol{\varepsilon}^{(\mathrm{S})}}(\mathbf{x})^2 & = y_{\mathrm{al}}^{(\mathrm{S})}(\mathbf{x},\boldsymbol{w}) + \boldsymbol{\xi}_{\mathrm{al}}^{(\mathrm{S})} \\ \mathbb{V}\big\{\mathsf{t}|\mathbf{x},\mathsf{D},\boldsymbol{\varepsilon}^{(\mathrm{T})}\big\} & = y_{\mathrm{ep}}^{(\mathrm{S})}(\mathbf{x},\boldsymbol{w}) + \boldsymbol{\xi}_{\mathrm{ep}}^{(\mathrm{S})} \end{cases} \end{aligned} \tag{14}$$

where $\boldsymbol{\xi}_{\mathrm{ep}}^{(\mathrm{S})} \sim \mathcal{N}(0,{\sigma_{\boldsymbol{\xi}_{\mathrm{ep}}^{(\mathrm{S})}}}^2)$. Note that the student has three outputs, $y^{(\mathrm{S})}(\mathbf{x},\boldsymbol{w})$ for target location prediction, $y_{\mathrm{al}}^{(\mathrm{S})}(\mathbf{x},\boldsymbol{w})$ for aleatoric uncertainty prediction and $y_{\mathrm{ep}}^{(\mathrm{S})}(\mathbf{x},\boldsymbol{w})$ for epistemic

uncertainty prediction. The total predictive variance is therefore $y_{\text{al}}^{(\text{S})}(\mathbf{x},\boldsymbol{w})+y_{\text{ep}}^{(\text{S})}(\mathbf{x},\boldsymbol{w})$.

Defining with M the number of samples per mini-batch $\mathcal{M}^{(\text{T})}$ and N_{epochs} the number of training epochs, at each step $i=\{1,2,\dots,N_{\text{iter}}=N_{\text{epochs}}\cdot\lfloor N/M\rfloor\}$, the teacher is trained with a SGLD step as

$$\Delta\boldsymbol{\theta}_{i+1}=\frac{\eta_i^{(\text{T})}}{2}\Big(\nabla_{\boldsymbol{\theta}}\log p(\boldsymbol{\theta}_i)+\frac{N}{M}\sum_{m\in\mathcal{M}^{(\text{T})}}\nabla_{\boldsymbol{\theta}}\log p(t_m|\boldsymbol{x}_m,\boldsymbol{\theta}_i)\Big)+\boldsymbol{z}_i \tag{15}$$

where $\Delta\boldsymbol{\theta}_{i+1}=\boldsymbol{\theta}_{i+1}-\boldsymbol{\theta}_i$, $\eta_i^{(\text{T})}$ is the teacher learning rate at step i, and $\boldsymbol{z}_i$ is a noise sample from $\mathcal{N}(\mathbf{0},\eta_i^{(\text{T})}\boldsymbol{I}_{|\boldsymbol{\theta}|})$. Following standard SGLD initialization, the prior $p(\boldsymbol{\theta}_i)$ is chosen to be spherical Gaussian as $p(\boldsymbol{\theta}_i)=\mathcal{N}(\boldsymbol{\theta}_i;\mathbf{0},\lambda^{(\text{T})}\boldsymbol{I}_{|\boldsymbol{\theta}|})$, where $\lambda^{(\text{T})}$ is the L2 regularizer. For the student, we first need to define the loss function and then a practical training procedure. Starting with the loss function, for an input sample $\boldsymbol{x}$, we propose a two blocks function $J(\boldsymbol{w}|\boldsymbol{x})=A(\boldsymbol{w}|\boldsymbol{x})+B(\boldsymbol{w}|\boldsymbol{x})$. The first term $A(\boldsymbol{w}|\boldsymbol{x})$ is the same original loss function (13) in BDK and induces the student to learn the target variable t and the aleatoric uncertainty $\sigma_{\boldsymbol{\epsilon}^{(\text{S})}}(\boldsymbol{x})^2$. It can be shown that for regression, the block $A(\boldsymbol{w}|\boldsymbol{x})$ can be approximated by (see Appendix A)

$$A(\boldsymbol{w}|\boldsymbol{x})\backsimeq\frac{1}{L}\sum_{\ell=1}^{L}\Bigg[\log\Big(y_{\text{al}}^{(\text{S})}(\boldsymbol{x},\boldsymbol{w})\Big)+y_{\text{al}}^{(\text{S})}(\boldsymbol{x},\boldsymbol{w})^{-1}\times\Big({\sigma_{\epsilon^{(\text{T})}}}^2+\Big\|y^{(\text{T})}(\boldsymbol{x},\boldsymbol{\theta}_\ell)-y^{(\text{S})}(\boldsymbol{x},\boldsymbol{w})\Big\|_2^2\Big)\Bigg]. \tag{16}$$

On the contrary, the second term $B(\boldsymbol{w}|\boldsymbol{x})$ forces the student to learn the epistemic uncertainty coming from the teacher and it is obtained via standard MLE as:

$$B(\boldsymbol{w}|\boldsymbol{x})\backsimeq\frac{1}{2{\sigma_{\boldsymbol{\xi}_{\text{ep}}^{(\text{S})}}}^2}\Bigg\|\frac{1}{L}\sum_{\ell=1}^{L}y^{(\text{T})}(\boldsymbol{x},\boldsymbol{\theta}_\ell)^2-\Big(\frac{1}{L}\sum_{\ell=1}^{L}y^{(\text{T})}(\boldsymbol{x},\boldsymbol{\theta}_\ell)\Big)^2-y_{\text{ep}}^{(\text{S})}(\boldsymbol{x},\boldsymbol{w})\Bigg\|_2^2. \tag{17}$$

We refer to Appendix B for a derivation of the block $B(\boldsymbol{w}|\boldsymbol{x})$. It is important to note that calculating $J(\boldsymbol{w}|\boldsymbol{x})$ requires L predictions from the teacher. However, we aim to bypass the storage of L samples of $\boldsymbol{\theta}_\ell$ during training. To address this, we propose a training method that retains the samples $\boldsymbol{x}_m$ and target t_m across L forward-backward steps, subsequently updating the student's parameters as

$$\Delta\boldsymbol{w}_{i+1}=-\frac{\eta_i^{(\text{S})}}{2}\Big(\frac{1}{M}\sum_{m\in\mathcal{M}_{i-L}^{(\text{S})}}\nabla_{\boldsymbol{w}}J(\boldsymbol{w}_i|\boldsymbol{x}_m)+\lambda^{(\text{S})}\boldsymbol{w}_i\Big) \tag{18}$$

where $\Delta\boldsymbol{w}_{i+1}=\boldsymbol{w}_{i+1}-\boldsymbol{w}_i$, $\eta_i^{(\text{T})}$, $\eta_i^{(\text{S})}$ is the student learning rate at step i, $\mathcal{M}_i^{(\text{S})}$ is the mini-batch of the student at step i and $\lambda^{(\text{S})}$ is an L2 regularizer hyper-parameter of the student. Algorithm 1 describes the full training procedure, which takes as input the training teacher and student datasets $\mathcal{D}^{(\text{T})}$ and $\mathcal{D}_{\boldsymbol{x}}^{(\text{S})}$, respectively, and outputs the student parameters $\boldsymbol{w}$. We point out that the student dataset does not contain any target values since the teacher output is adopted as a target. This gives the flexibility to exploit unsupervised, i.e., unlabelled, datasets for training the student in input locations where we are interested in having reliable uncertainty metrics.

Algorithm 1 Training Procedure

Input: Training datasets $\mathcal{D}^{(\text{T})}$ and $\mathcal{D}_{\boldsymbol{x}}^{(\text{S})}$
Output: Student parameters $\boldsymbol{w}$

Initialize Teacher and Student parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{w}_1$
Initialize $\mathcal{D}_{\boldsymbol{x},\text{old}}^{(\text{S})}\leftarrow\emptyset$ and $\mathcal{D}_{t,\text{old}}^{(\text{S})}\leftarrow\emptyset$
Set $N_{\text{iter}}\leftarrow N_{\text{epochs}}\cdot\lfloor N/M\rfloor$
for $i=\{1,\dots,N_{\text{iter}}\}$ **do** ▷ Batch-wise iteration
  Sample minibatch $\mathcal{M}_i^{(\text{T})}$ of size M from $\mathcal{D}^{(\text{T})}$
  Sample $\boldsymbol{z}_i\sim\mathcal{N}(0,\eta_i^{(\text{T})}\boldsymbol{I}_{|\boldsymbol{\theta}|})$
  Update Teacher using (15)
  Sample minibatch $\mathcal{M}_i^{(\text{S})}$ of size M from $\mathcal{D}_{\boldsymbol{x}}^{(\text{S})}$
  $\mathcal{D}_{\boldsymbol{x},\text{old}}^{(\text{S})}\leftarrow\mathcal{D}_{\boldsymbol{x},\text{old}}^{(\text{S})}\cup\{\boldsymbol{x}_m\}_{m\in\mathcal{M}_i^{(\text{S})}}$
  $\mathcal{D}_{t,\text{old}}^{(\text{S})}\leftarrow\mathcal{D}_{t,\text{old}}^{(\text{S})}\cup\{y^{(\text{T})}(\boldsymbol{x},\boldsymbol{\theta}_{i+1})\}_{\boldsymbol{x}\in\mathcal{D}_{\boldsymbol{x},\text{old}}^{(\text{S})}}$
  if $i>L$ **then**
    $\mathcal{D}_{\boldsymbol{x},\text{old}}^{(\text{S})}\leftarrow\mathcal{D}_{\boldsymbol{x},\text{old}}^{(\text{S})}\backslash\{\boldsymbol{x}_m\}_{m\in\mathcal{M}_{i-L}^{(\text{S})}}$
    $\mathcal{D}_{t,\text{old}}^{(\text{S})}\leftarrow\mathcal{D}_{t,\text{old}}^{(\text{S})}\backslash\{y^{(\text{T})}(\boldsymbol{x}_m,\boldsymbol{\theta}_j)\}_{\substack{m\in\mathcal{M}_{i-L}^{(\text{S})},\\ j=\{i-L,\dots,i\}}}$
    Update Student using (18)
  end if
end for

E. Multi-Dimensional Target Variable

In case the target variable is multi-dimensional, such as for 3D location, we have that $\mathbb{E}\{\mathbf{t}|\mathbf{x},\mathsf{D}\}\in\mathbb{R}^{|\mathbf{t}|\times1}$ and $\mathbb{V}\{\mathbf{t}|\mathbf{x},\mathsf{D}\}\in\mathbb{R}^{|\mathbf{t}|\times|\mathbf{t}|}$. Therefore, the new teacher-student model becomes

$$\begin{aligned}&(\text{T}):\quad\mathbf{t}=\boldsymbol{y}^{(\text{T})}(\mathbf{x},\boldsymbol{\theta})+\boldsymbol{\epsilon}^{(\text{T})}\\&(\text{S}):\quad\begin{cases}\mathbf{t}&=\boldsymbol{y}^{(\text{S})}(\mathbf{x},\boldsymbol{w})+\boldsymbol{\epsilon}^{(\text{S})}(\mathbf{x})\\ \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}^{(\text{S})}}(\mathbf{x})&=R_{\substack{|\mathbf{t}|^2\times1\\|\mathbf{t}|\times|\mathbf{t}|}}\big(\boldsymbol{y}_{\text{al}}^{(\text{S})}(\mathbf{x},\boldsymbol{w})\big)+\boldsymbol{\xi}_{\text{al}}^{(\text{S})}\\ \mathbb{V}\Big\{\mathbf{t}|\mathbf{x},\mathsf{D},\boldsymbol{\epsilon}^{(\text{T})}\Big\}&=R_{\substack{|\mathbf{t}|^2\times1\\|\mathbf{t}|\times|\mathbf{t}|}}\big(\boldsymbol{y}_{\text{ep}}^{(\text{S})}(\mathbf{x},\boldsymbol{w})\big)+\boldsymbol{\xi}_{\text{ep}}^{(\text{S})}\end{cases}\end{aligned} \tag{19}$$

where $R_{\substack{p\times q\\n\times m}}:\mathbb{R}^{p\times q}\rightarrow\mathbb{R}^{n\times m}$ indicates the reshape operation, $\boldsymbol{\epsilon}^{(\text{T})}\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}^{(\text{T})}})$, $\boldsymbol{\epsilon}^{(\text{S})}(\mathbf{x})\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}^{(\text{S})}}(\mathbf{x}))$, $\boldsymbol{\xi}_{\text{al}}^{(\text{S})}\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{\text{al}}^{(\text{S})}})$ and $\boldsymbol{\xi}_{\text{ep}}^{(\text{S})}\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{\text{ep}}^{(\text{S})}})$. Consequently, the $A(\boldsymbol{w}|\boldsymbol{x})$ term in (16) in matrix form becomes

$$\begin{aligned}A(\boldsymbol{w}|\boldsymbol{x})\backsimeq\frac{1}{L}\sum_{\ell=1}^{L}\Bigg[&\frac{1}{2}\log\Big(\det R_{\substack{|\mathbf{t}|^2\times1\\|\mathbf{t}|\times|\mathbf{t}|}}\big(\boldsymbol{y}_{\text{al}}^{(\text{S})}(\boldsymbol{x},\boldsymbol{w})\big)\Big)\\&+\frac{1}{2}\big(\boldsymbol{y}^{(\text{T})}(\boldsymbol{x},\boldsymbol{\theta}_\ell)-\boldsymbol{y}^{(\text{S})}(\boldsymbol{x},\boldsymbol{w})\big)^\top\\&\times R_{\substack{|\mathbf{t}|^2\times1\\|\mathbf{t}|\times|\mathbf{t}|}}\big(\boldsymbol{y}_{\text{al}}^{(\text{S})}(\boldsymbol{x},\boldsymbol{w})\big)^{-1}\big(\boldsymbol{y}^{(\text{T})}(\boldsymbol{x},\boldsymbol{\theta}_\ell)-\boldsymbol{y}^{(\text{S})}(\boldsymbol{x},\boldsymbol{w})\big)\\&+\text{Tr}\Big(\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}^{(\text{T})}}R_{\substack{|\mathbf{t}|^2\times1\\|\mathbf{t}|\times|\mathbf{t}|}}\big(\boldsymbol{y}_{\text{al}}^{(\text{S})}(\boldsymbol{x},\boldsymbol{w})\big)^{-1}\Big)\Bigg].\end{aligned} \tag{20}$$

The $B(\boldsymbol{w}|\boldsymbol{x})$ term in (17) becomes

$$B(\boldsymbol{w}|\boldsymbol{x}) \backsimeq \frac{1}{2}\Big(\boldsymbol{y}_{\text{ep}}^{(\text{T})}(\boldsymbol{x}) - R_{\substack{|\mathbf{t}|^2\times 1\\ |\mathbf{t}|\times|\mathbf{t}|}}\big(\boldsymbol{y}_{\text{ep}}^{(\text{S})}(\boldsymbol{x},\boldsymbol{w})\big)\Big)^{\top} \times {\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{\text{ep}}^{(\text{S})}}}^{-1}\Big(\boldsymbol{y}_{\text{ep}}^{(\text{T})}(\boldsymbol{x}) - R_{\substack{|\mathbf{t}|^2\times 1\\ |\mathbf{t}|\times|\mathbf{t}|}}\big(\boldsymbol{y}_{\text{ep}}^{(\text{S})}(\boldsymbol{x},\boldsymbol{w})\big)\Big) \tag{21}$$

where the predictive epistemic uncertainty of the teacher is

$$\boldsymbol{y}_{\text{ep}}^{(\text{T})}(\boldsymbol{x}) = \frac{1}{L}\sum_{\ell=1}^{L}\Big(\boldsymbol{y}^{(\text{T})}(\boldsymbol{x},\boldsymbol{\theta}_\ell)^{\top}\boldsymbol{y}^{(\text{T})}(\boldsymbol{x},\boldsymbol{\theta}_\ell)\Big) - \Big(\frac{1}{L}\sum_{\ell=1}^{L}\boldsymbol{y}^{(\text{T})}(\boldsymbol{x},\boldsymbol{\theta}_\ell)\Big)^{\top}\Big(\frac{1}{L}\sum_{\ell=1}^{L}\boldsymbol{y}^{(\text{T})}(\boldsymbol{x},\boldsymbol{\theta}_\ell)\Big). \tag{22}$$

F. AE-Based DL Model

For the cellular positioning application of the developed BNN method, we propose using an AE stored inside each BS, as depicted in Fig. 2(b), to derive key location features from the sparse ADCPM samples $\boldsymbol{x}$. The encoder function $E(\boldsymbol{x})$ generates the latent features $\boldsymbol{z}$, which encompass location-centric data inherent in the channel. Conversely, the decoder function $D(\boldsymbol{z})$ attempts to recreate the input samples, resulting in $\hat{\boldsymbol{x}}$. The AE aims to minimize the reconstruction error metric $\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|_2^2$ [75], enabling the model to reproduce the input $\boldsymbol{x}$ using the condensed data in $\boldsymbol{z}$. This ensures that $\boldsymbol{z}$ includes all crucial information needed for the single-BS positioning objective. The DL model also contains a multilayer perceptron (MLP) positioning module which takes as input the latent features and predicts the 3D target position. Note that it is not necessary to detect LOS and NLOS conditions since the MLP is combined with other BSs outputs through the posterior predictive distribution. In other words, the positioning module's output is a *soft information* [2], [12], [27], i.e., a distribution, which is coherently combined for tracking applications.

In order to apply the BBK method, we created two identical DL models, one for the teacher and one for the student. In the teacher network, we treated the AE structure as a normal NN, whereas the positioning module is trained as a full BNN with SGLD optimizer. This is mainly done for faster convergence reasons and because unsupervised samples, i.e., CIR samples without position correspondence, can be easily gathered in every position, thereby solving the OOD problem. On the contrary, the student is trained as a conventional NN with Adam optimizer [76] and the loss function

$$J(\boldsymbol{w}|\boldsymbol{x}) = \lambda_{\text{pos}}\big(A(\boldsymbol{w}|\boldsymbol{x}) + B(\boldsymbol{w}|\boldsymbol{x})\big) + \lambda_{\text{rec}}\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|_2^2 \tag{23}$$

where λ_{rec} regulates the sample reconstruction and λ_{pos} controls the position estimation relevance. Generally, the relation between these two hyper-parameters is $\lambda_{\text{rec}} < \lambda_{\text{pos}}$. This is because firstly, the AE model exhibits greater complexity compared to the MLPs used for positioning, resulting in a rapid decrease in reconstruction error. Secondly, the feature count in $\boldsymbol{x}$ surpasses the dimension of $\boldsymbol{t}$. This inherently amplifies the significance of the reconstruction error relative to the positioning error.

V. Location Tracking With BNN Methodology

In this section, we first introduce the general problem of sequential Bayesian tracking, and then we propose a solution that integrates the BNN method.

A. Tracking Problem

We consider the non-linear Bayesian tracking problem of a target whose state evolves according to the motion model [77]

$$\mathbf{t}_n = f_n^{(\text{t})}(\mathbf{t}_{n-1}) + \boldsymbol{\varepsilon}_{n-1}^{(\text{t})} \tag{24}$$

where $f_n^{(\text{t})}(\mathbf{t}_{n-1})$ is a non-linear function of the state $\mathbf{t}_{n-1}$ at time $n-1$ and $\boldsymbol{\varepsilon}_{n-1}^{(\text{t})}$ is a non-independent and identical distributed (non-IID) noise sequence. From (24), we define the corresponding transition PDF $p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1})$. The network positioning system has measurements of the target modeled as

$$\mathbf{x}_n = f_n^{(\text{x})}(\mathbf{t}_n) + \boldsymbol{\varepsilon}_n^{(\text{x})} \tag{25}$$

where $f_n^{(\text{x})}(\mathbf{t}_n)$ is a non-linear function which relates the state and the measurement, and $\boldsymbol{\varepsilon}_n^{(\text{x})}$ is a non-IID noise sequence. Similarly to before, from (25), we can define a likelihood function $p(\boldsymbol{x}_n|\boldsymbol{t}_n)$. Moreover, we define with $\mathbf{x}_{1:n} = \{\mathbf{x}_i, i=1,\dots,n\}$ the set of all available measurements up to time n. Since (24) and (25) are 1st order hidden Markov model (HMM), we can write that $p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1},\boldsymbol{x}_{1:n-1}) = p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1})$ and express the usual prediction phase of the state through the Chapman-Kolmogorov equation

$$p(\boldsymbol{t}_n|\boldsymbol{x}_{1:n-1}) = \int p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1})\, p(\boldsymbol{t}_{n-1}|\boldsymbol{x}_{1:n-1})\, d\boldsymbol{t}_{n-1} \tag{26}$$

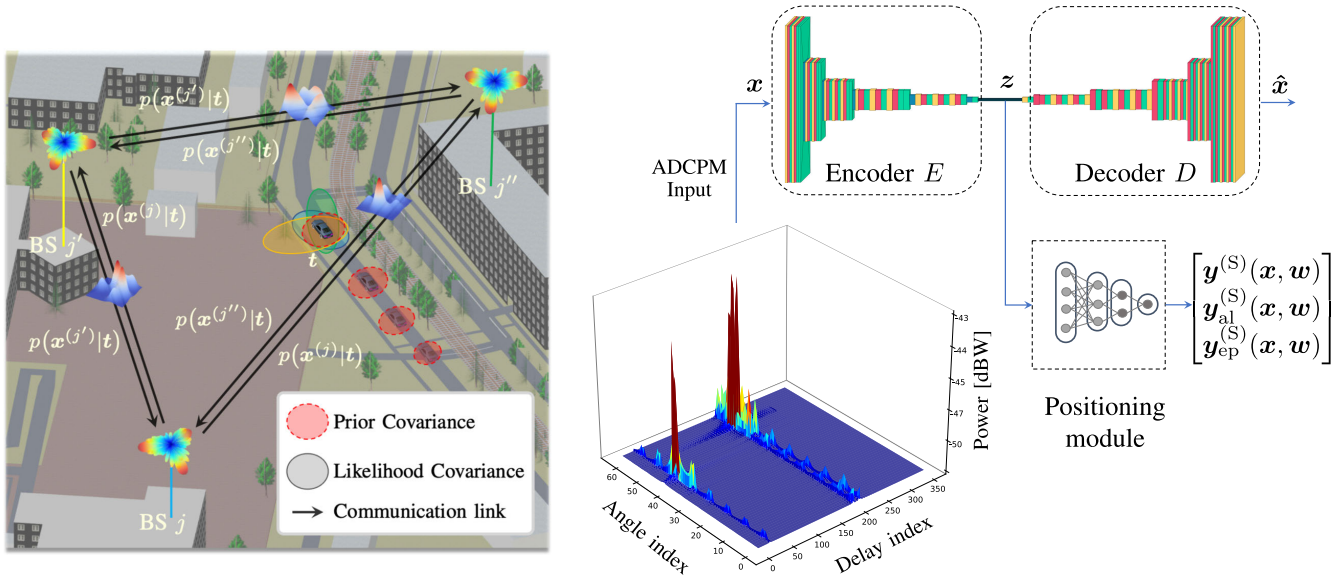
where $p(\boldsymbol{t}_{n-1}|\boldsymbol{x}_{1:n-1})$ is the posterior PDF at time $n-1$ and $p(\boldsymbol{t}_n|\boldsymbol{x}_{1:n-1})$ is the prior PDF at time n. Subsequently, the measurements are taken into account in the update phase which recovers the posterior at time n:

$$p(\boldsymbol{t}_n|\boldsymbol{x}_{1:n}) \propto p(\boldsymbol{x}_n|\boldsymbol{t}_n)\, p(\boldsymbol{t}_n|\boldsymbol{x}_{1:n-1})\,. \tag{27}$$

The EKF, or more complex conventional filters [28], implement the steps (26) and (27) by assuming explicit (and known) parametric models for functions $f_n^{(\text{t})}(\mathbf{t}_{n-1})$ and $f_n^{(\text{x})}(\mathbf{t}_n)$. This is not viable in mixed LOS/NLOS conditions and with complex measurement as ADCPM, which relate to the location through an unknown non-linear and site-dependent function $f_n^{(\text{x})}(\mathbf{t}_n)$. We thus propose to integrate a BNN into the Bayesian filter and learn such function from data, as detailed in the following section.

B. Proposed BNN for Tracking

For the integration of BNN in the tracking system, we consider a set of BSs $\mathcal{S}_{\text{BS}}$. During the offline training phase, each BS j trains its own BNN using a local training dataset $\mathcal{D}^{(j)} = \big\{\boldsymbol{t}_n^{(j)}, \boldsymbol{x}_n^{(j)}\big\}_{n=1}^{N^{(j)}}$. We point out that in this phase, any BNN algorithm and layer structure could be employed and assessed during an ad-hoc testing static positioning. Subsequently, in the online tracking phase, a set of BSs $\mathcal{S}_{\text{BS},n} \subseteq \mathcal{S}_{\text{BS}}$ detects a target at timestep n and obtains a set of samples $\boldsymbol{X}_n = \big\{\boldsymbol{x}_n^{(j)}\big\}_{j\in\mathcal{S}_{\text{BS},n}}$. The main idea is to leave unaltered the



(a) (b)

Fig. 2. (a) Cooperative tracking system composed of three BSs. The timestep index n has been dropped for simplicity of notation. (b) DL model composed of an AE structure and a positioning module. The input is the sparse ADCPM fingerprint, whereas the outputs are the reconstructed $\hat{\boldsymbol{x}}$, the target estimation $\boldsymbol{y}^{(\mathrm{S})}(\boldsymbol{x},\boldsymbol{w})$, the aleatoric uncertainty $\boldsymbol{y}^{(\mathrm{S})}_{\mathrm{al}}(\boldsymbol{x},\boldsymbol{w})$ and the epistemic uncertainty $\boldsymbol{y}^{(\mathrm{S})}_{\mathrm{ep}}(\boldsymbol{x},\boldsymbol{w})$.

prediction phase (where the dynamics function $f_n^{(\mathrm{t})}(\mathbf{t}_{n-1})$ is easier to model) and focus on the tight integration of BNN in the update phase. This is done to facilitate the integration with already existing algorithms since we can just replace or even augment the update part with the BNN as described in the following sections. Moreover, while training of a NN with static measurements is both affordable and precise, replacing the prediction phase with dynamic models (e.g., LSTMs) introduces complexities. One significant challenge is the data collection for target trajectories, which necessitates the gathering of ground truth data, i.e., a reliable benchmark to precisely measure the target's exact trajectory.

After performing the prediction phase and obtaining the prior distribution $p(\boldsymbol{t}_n|\boldsymbol{x}_{1:n-1})$, each BS $j \in \mathcal{S}_{\mathrm{BS},n}$ outputs the posterior predictive distribution $p\big(\boldsymbol{t}_n|\boldsymbol{x}_n^{(j)},\mathcal{D}^{(j)}\big)$, which we indicate with $p\big(\boldsymbol{t}_n|\boldsymbol{x}_n^{(j)}\big)$. Now, we note that the posterior in (27) resembles the posterior predictive distribution in (8), but they are fundamentally different. Indeed, while the former is the result of prior knowledge coming from the tracking, the latter does not have any knowledge on the sequentiality of the target state since the BNN network has just been trained with input-output samples. Therefore, at each timestep n, the BNN does not have any prior knowledge on where the target was at previous time $n-1$, formally, $\mathbf{t}_n \sim \mathcal{U}\big(\boldsymbol{t}_{\min}^{(j)},\boldsymbol{t}_{\max}^{(j)}\big)$, where $\boldsymbol{t}_{\min}^{(j)}$ and $\boldsymbol{t}_{\max}^{(j)}$ are the limits of the coverage area of the j-th BS. This is similar to what happens during the computation of the likelihood function $p(\boldsymbol{x}_n|\boldsymbol{t}_n)$ where the measurement model in (25) is considered without previous time-dependence. By analogy and considering the uniform distribution of $\mathbf{t}_n$, we can write $p\big(\boldsymbol{x}_n^{(j)}|\boldsymbol{t}_n\big) \propto p\big(\boldsymbol{t}_n|\boldsymbol{x}_n^{(j)},\mathcal{D}\big)$. In this way, we combine the predictions of multiple BSs with the prior PDF on the target state and obtain the updated posterior. The full tracking procedure can be found in Algorithm 2.

Algorithm 2 Tracking Procedure

Input: Posterior $p(\boldsymbol{t}_{n-1}|\boldsymbol{x}_{1:n-1})$ at time $n-1$
Output: Posterior $p(\boldsymbol{t}_n|\boldsymbol{x}_{1:n})$ at time n

```
Compute prediction phase in (26)
Measure sample x_n^(j)
Compute p(x_n^(j)|t_n) ∝ p(t_n|x_n^(j), D)
for j' ∈ S_BS,n \ {j} do
    Send p(x_n^(j)|t_n) to j'
    Receive p(x_n^(j')|t_n) from j'
end for
Update p(t_n|x_1:n) ∝ ∏_{j∈S_BS,n} p(x_n^(j)|t_n) p(t_n|x_1:n-1)
```

An example of cooperative tracking performed by three BSs is shown in Fig. 2(a), where the exchange of the likelihood functions permits the reduction of the target position uncertainty (represented by the intersections of covariance areas in the figure).

For the scenario described in Sec. VI, the posterior predictive distribution is described by two parameters, i.e., the predictive mean (10) and the predictive variance (11). Therefore, we propose to approximate the likelihood function obtained by each BS with a multivariate normal distribution as

$$p\big(\boldsymbol{x}_n^{(j)}|\boldsymbol{t}_n\big) \backsimeq \mathcal{N}\Big(\boldsymbol{x}_n^{(j)};\mathbb{E}\big\{\mathbf{t}_n|\mathbf{x}_n^{(j)},\mathsf{D}\big\},\mathbb{V}\big\{\mathbf{t}_n|\mathbf{x}_n^{(j)},\mathsf{D}\big\}\Big) = \mathcal{N}\Big(\boldsymbol{x}_n^{(j)};\boldsymbol{\mu}_n^{(j)},\boldsymbol{\Sigma}_n^{(j)}\Big). \tag{28}$$

Note that for the real-time BBK, the predictive mean and variance are respectively

$$\boldsymbol{\mu}_n^{(j)} = \boldsymbol{y}^{(\mathrm{S})}\big(\boldsymbol{x}_n^{(j)},\boldsymbol{w}^{(j)}\big) \tag{29}$$

$$\boldsymbol{\Sigma}_n^{(j)} = R_{\substack{|\mathbf{t}|^2\times 1\\ |\mathbf{t}|\times|\mathbf{t}|}}\Big(\boldsymbol{y}_{\mathrm{al}}^{(\mathrm{S})}\big(\boldsymbol{x}_n^{(j)},\boldsymbol{w}^{(j)}\big)+\boldsymbol{y}_{\mathrm{ep}}^{(\mathrm{S})}\big(\boldsymbol{x}_n^{(j)},\boldsymbol{w}^{(j)}\big)\Big). \quad (30)$$

This approximation makes it very easy and effective to combine the likelihood functions of the BSs to be used in (27) as [78]

$$p(\boldsymbol{x}_n|\boldsymbol{t}_n) = \prod_{j\in\mathcal{S}_{\mathrm{BS},n}} p\big(\boldsymbol{x}_n^{(j)}|\boldsymbol{t}_n\big) \propto \mathcal{N}\big(\boldsymbol{x}_n;\boldsymbol{\mu}_n,\boldsymbol{\Sigma}_n\big),$$

where

$$\boldsymbol{\mu}_n = \boldsymbol{\Sigma}_n\Big(\sum_{j\in\mathcal{S}_{\mathrm{BS},n}} {\boldsymbol{\Sigma}_n^{(j)}}^{-1}\boldsymbol{\mu}_n^{(j)}\Big)$$

$$\boldsymbol{\Sigma}_n = \Big(\sum_{j\in\mathcal{S}_{\mathrm{BS},n}} {\boldsymbol{\Sigma}_n^{(j)}}^{-1}\Big)^{-1}.$$

VI. Simulation Experiments

A. Simulation Results

To assess the performance of the proposed BNN-based tracking system, due to the unavailability of real-world experimental activities, we simulate a 5G positioning network based on the 5G new radio (NR) MATLAB clustered delay line (CDL) channel model, which can be defined over a bandwidth of 2 GHz in the frequency range from 0.5 GHz to 100 GHz [79]. The radio wave propagation is simulated using a ray-tracing method [80], [81], [82] from the Wireless InSite 3D prediction tool [83], which plots the propagation paths from the UE to the BSs based on the surface geometry from a 3D map file. The ray-based solver can manage up to fifty reflections and three diffractions, ensuring a realistic simulation of the effect of buildings and terrains on the radio signal propagation. In mmWave scenarios, the propagation model integrates atmospheric absorption and allows the inclusion of vegetation within the propagation setting, assessing the impact of diffuse scattering on the channel response and ensuring spatial consistency. The channel is then obtained by using the rays as mean clusters and by including: the Doppler shift, according to the UE mobility, a main LOS cluster (if the UE is in visibility) with K-factor equal to 13.3 dB, a number of sub-cluster per cluster equal to 2, and moving scatterers in the channel. The cluster-wise root mean square (RMS) angle spreads and delay spreads have been set to 3 degrees and 3.90625 ns, respectively.

The 3D map is obtained through Google Maps, RenderDoc, and Blender software with MapsModelImporter plugin. An example of the extracted 3D map can be found in Fig. 3, which represents both the 3D patches with known textures. For our experimental setup, we emulate a 3GPP urban micro (UMi) environment [69] spanning a 1000×1000 m square near the Massachusetts Institute of Technology (MIT) campus, Cambridge, MA 02139, USA. The setting encompasses 19 sites with an inter-site distance (ISD) of 200 m, arranged in a hexagonal pattern. Every site is composed of 3 cells, each at a height of 25 m and spaced 120 degrees apart in azimuth.

Each cellular antenna is equipped with an UPA setup with $N_{\mathrm{h}} = N_{\mathrm{v}} = 8$ antenna elements and a mechanical downtilt of 15 degrees. The antenna element details were derived from [84], ensuring a front-to-back ratio of approximately 30 dB and a peak gain reaching 8 dBi. The UE trajectories are generated by the SUMO software, which simulates realistic vehicular traffic throughout a given road grid, according to the interactions between vehicles, geometry of the map and speed limits. Over 600 s of simulation, we created up to 100 vehicle trajectories and we gathered data points every second. The absolute velocities of the vehicles span in $[0, 34]$ km/h, with a mean and standard deviation of 9.4 and 6.3 km/h, respectively. In total, we obtained 2593 and 702 training and testing positions, respectively, and about $9.3\cdot10^4$ and $2.5\cdot10^4$ training and testing ADCPM samples, respectively. The simulated testing trajectories with their absolute velocities, along with the BSs composing the UMi scenario, are shown in Fig. 4. On the contrary, the different traffic densities can be found in Fig. 8-top. In each position, every UE broadcasts 5G sounding reference signal (SRS) to all neighbor BSs using a carrier frequency $f_{\mathrm{c}} = 28$ GHz and a transmission bandwidth $B = 400$ MHz. Then, each BS performs OFDM demodulation and obtains the SFCRM in (2) through pilot signals and least squares (LS) channel estimation. Finally, the ADCPM is computed using (3) and (4).

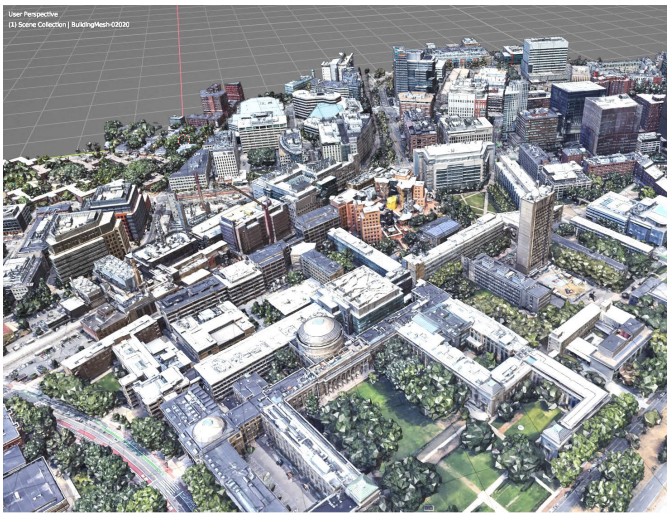


Fig. 3. Digital twin 3D map representation of the urban scenario around MIT campus.

B. Discussion on Practical Implementation

The tracking system is designed to separate the prediction phase in (26) and update phase in (27). Consequently, data acquisition can be performed statically at each position to train the BNN, whereas the motion model can be adjusted according to the dynamics of the UE. However, in case we want to speed up the process, we can adopt a vehicle or platoon of vehicles [85], [86] to record the timestamp and related noisy position with GNSS, which will then be mapped with the channel recording at the BSs. The uncertainty about the ground truth position is automatically learned by the aleatoric uncertainty prediction, whereas the density of the training points is taken into account by the epistemic uncertainty.

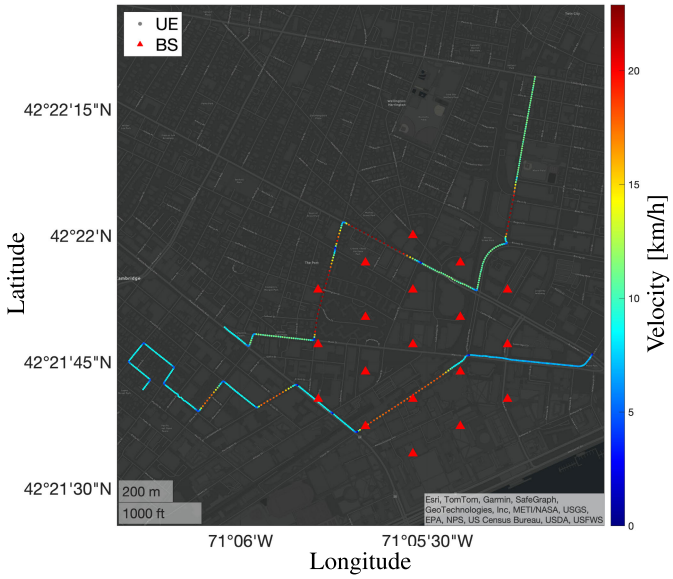


Fig. 4. Test trajectories with two vehicles in the area of Cambridge, MA, USA. The red triangles indicate the BS positions.

For the simulations, we did not consider multi-user interference (MUI) as the focus of the paper is to assess the best-case performances of the proposed BBK method and tracking system. However, when dealing with more than one user, the BSs can perform robust channel estimation methods, e.g., linear minimum mean-square error (LMMSE) or non-linear pre-coding schemes [87], [88], to reduce the impact of MUI. Other possible solutions include the usage of data association (DA) schemes to track and remove the multipath components of the interference [89].

C. DL Model Implementation

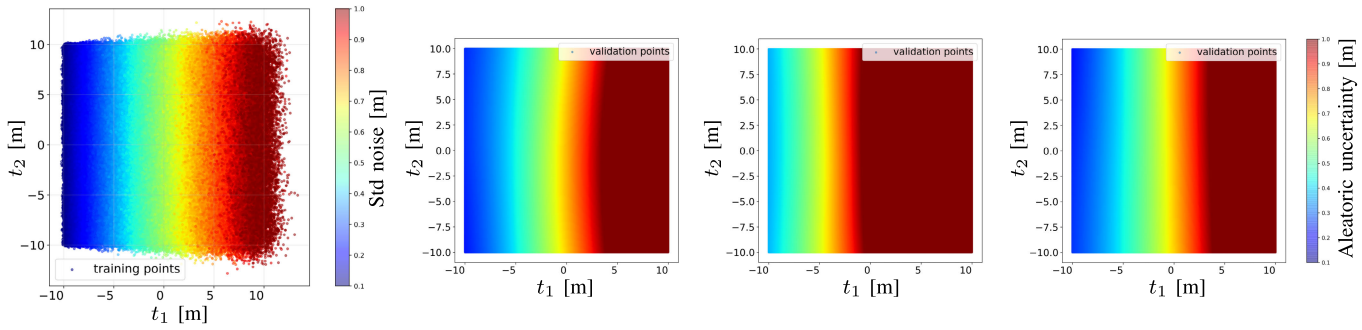
For the specific design of the DL positioning model, we design the AE part with the Segnet architecture [90]. This permits to manage the sparsity of the ADCPM input and perform robust feature extraction, which is pivotal for accurate positioning. Indeed, the upsampling layers utilize encoder pool indices for custom sparse feature mapping. During testing, the decoder segment is discarded, as input reconstruction is only needed for learning latent feature representations during training. Concerning the positioning module, given the natural regularization induced by the BNN teacher model, we inserted Gaussian error linear unit (GELU) activations functions after each linear layer. We performed an architecture search by doubling the number of neurons in each layer, until reaching both the latency requirements of 5 ms for fully autonomous driving vehicles [44] and a low bias in the performances. For the latter, we performed a similar procedure in [34] by testing the position module to perform localization with a synthetic dataset where the input latent features are substituted with geometric measurements (i.e., AOAs and TOFs). After multiple neural architecture searches, we set the number of neurons in each layer to: $[16, 32, 64, 128, 256, 512, 256, 128, 64, 32, 16, 9]$. For the prediction of the uncertainties, we placed softplus activation functions at the outputs of the positioning module which relates to the diagonal elements of $R_{\substack{|\mathbf{t}|^2\times 1\\ |\mathbf{t}|\times|\mathbf{t}|}}\left(\boldsymbol{y}_{\mathrm{al}}^{(\mathrm{S})}(\boldsymbol{x},\boldsymbol{w})\right)$ and $R_{\substack{|\mathbf{t}|^2\times 1\\ |\mathbf{t}|\times|\mathbf{t}|}}\left(\boldsymbol{y}_{\mathrm{ep}}^{(\mathrm{S})}(\boldsymbol{x},\boldsymbol{w})\right)$. This prevents the variances on the diagonal from being negative. Finally, in order to make the aleatoric and epistemic predictions valid covariances, we add a regularization term to the diagonal of each matrix prediction to ensure they are non-singular and enforce symmetry in each matrix by averaging them with their transpose.

To measure the time required by the system to perform inference, we report that the total number of floating point operations (FLOPs) required by the DL model are $27.4 \cdot 10^9$. Simulations were conducted on a workstation boasting an Intel(R) Xeon(R) Silver 4210R CPU @ 2.40 GHz, with 96 GB RAM and a Quadro RTX 6000 24 GB GPU. Since the GPU has a single-precision performance of $16.3 \cdot 10^{12}$ floating point operations per second (FLOPS), the inference time per sample can be estimated as $27.4 \cdot 10^9 / 16.3 \cdot 10^{12}$ s $= 1.6$ ms. Considering the delay required to exchange a packet comprising $\boldsymbol{\mu}_n^{(j)}$ and $\boldsymbol{\Sigma}_n^{(j)}$ between the two farthest BSs, i.e., less than 1 ms with fiber's length of 1 km and up to 80% of traffic load [91], we assume a latency of about 2 ms. In case of multiple targets, we can exploit the tensor operations of the GPU with $130.5 \cdot 10^{12}$ FLOPS.

All the experiments were performed using Pytorch [92] and, unless stated otherwise, models underwent training for 600 epochs with a batch size of $M = 256$ and the number of NN parameter samples $L = 40$. The learning rates of the teacher and student were set to $\eta_0^{(\mathrm{T})} = 10^{-5} < \eta_0^{(\mathrm{S})} = 10^{-4}$, in order to have a better convergence of the teacher (with lower learning rate) and, conversely, a faster convergence of the student. Regarding the hyper-parameters, λ_{rec} and λ_{pos} were empirically set using a grid method in the range $[0.1, 1]$ with a 0.1 step size and following the intuition described in Sec. IV-F, resulting in $\lambda_{\mathrm{rec}} = 0.1$ and $\lambda_{\mathrm{pos}} = 0.9$. For the L2 prior regularizers in (15) and (18), we recommend setting $\lambda^{(\mathrm{T})} \ll \lambda^{(\mathrm{S})}$ since the student is exposed to more data than the teacher, i.e., while the teacher repeatedly processes the same training data, the student encounters new, randomly generated data at every stage. Therefore, we set $\lambda^{(\mathrm{T})} = 0.1$ and $\lambda^{(\mathrm{S})} = 1$.

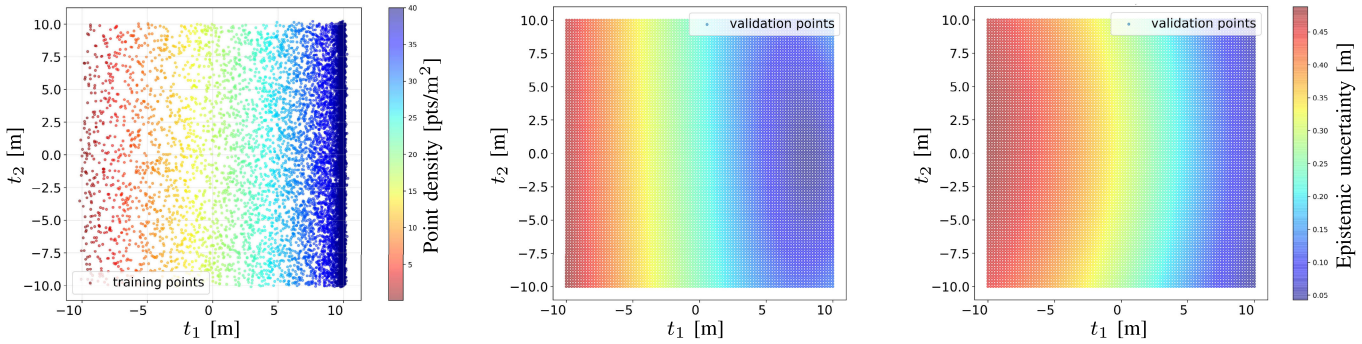
D. Numerical Results

1) Aleatoric and Epistemic Uncertainties: In this first experiment, we assess the capabilities of the proposed BBK method to learn both the aleatoric and epistemic uncertainty of the SGLD-based teacher. To this aim, we created a 2D artificial dataset where the input features are a noisy version of the UE 2D position (i.e., t_1 and t_2). To test the aleatoric uncertainty only, we employ a training dataset with densely sampled positions, so to induce the epistemic uncertainty to zero, and we add a Gaussian noise whose standard deviation varies linearly in $[0.1, 1]$ m along with the t_1 axes. Therefore, the objective is to predict this variation of aleatoric uncertainty which still remains in the dataset. The training dataset is shown in Fig. 5a. Then, we trained the SGLD-based teacher, the BDK-based student, and the BBK-based student, and we plotted the predicted variance measuring the aleatoric



(a) Training dataset (b) SGLD (c) BDK (d) BBK

Fig. 5. (a) Training points of the positioning dataset for aleatoric uncertainty assessment. (b), (c) and (d), predicted aleatoric uncertainty of the whole 2D space for SGLD, BDK and BBK, respectively.



(a) Training dataset (b) SGLD (c) BBK

Fig. 6. (a) Training points of the positioning dataset for epistemic uncertainty assessment. (b) and (c), predicted epistemic uncertainty of the whole 2D space for SGLD, and BBK, respectively.

uncertainty of the three methods on a 2D testing grid for $t_1, t_2 \in [-10, 10]$ m. From the results in Fig. 5, we observe that both the Bayesian teacher SGLD and the BBK are able to accurately reproduce the real additive noise in Fig. 5a. Indeed, the BBK-based student is trained to predict the SGLD-based teacher output, whose performances are boosted by the L consecutive predictions per input sample. On the contrary, the BDK achieves lower fidelity since this method is trained to approximate the point estimate of the teacher at each step, whereas the BBK is trained to predict the average of L predictions, i.e., the sum in (20).

To test the epistemic uncertainty, we fix the standard deviation of the additive Gaussian noise to 0.1 m, while we linearly change the density of the training points in the range $[4, 40]$ pts/m^2. The training dataset and the epistemic uncertainty predictions for SGLD and BBK are reported in Fig. 6. Note that, in this scenario, the BDK cannot be evaluated as it lacks the capability to predict epistemic uncertainty. For comparison of the epistemic uncertainty with the ground truth, we need to confront the predicted epistemic uncertainty with the squared root of the inverse density of the training points. This is because, in a well-calibrated BNN model, whenever a prediction has $P\,\%$ confidence, then the model's forecast aligns with the true occurrence approximately $P\,\%$ of the time. From the results, we can notice that the proposed BBK reconstructs almost completely the epistemic uncertainty, while being at the same time L times faster than the teacher model. Note that, whenever the training points are very dense, i.e., 40 pts/m^2, the epistemic uncertainty almost drops to zero. On the contrary, when we have a low density, such as the extreme case of 4 pts/m^2 (i.e., 1 pt/0.25 m^2), the predicted epistemic uncertainty is very similar to the squared root of the inverse density of the training points.

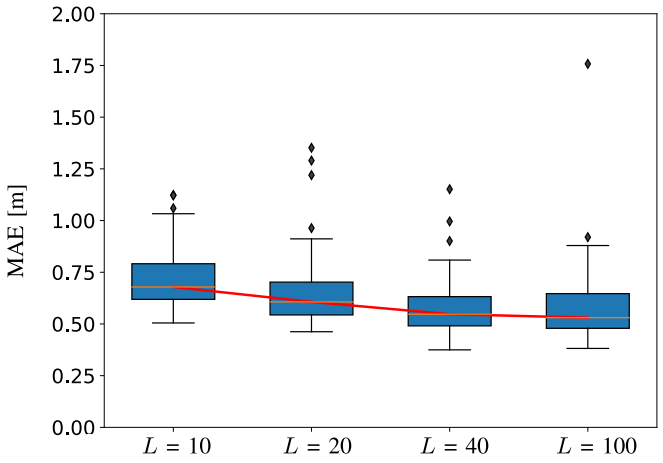


Fig. 7. Boxplot of the MAE per batch for different numbers of samples L realized from the posterior distribution $p_{\boldsymbol{\theta}|\mathrm{D}}(\boldsymbol{\theta}|\mathcal{D})$.

2) Hyper-Parameter Tuning for MC Sampling: This assessment is for tuning the number of samples L adopted in the teacher SGLD, as well as for verifying the maximum static positioning accuracy achieved by the proposed single-BS DL model in the ray-tracing dataset. To this aim, Fig. 7 shows

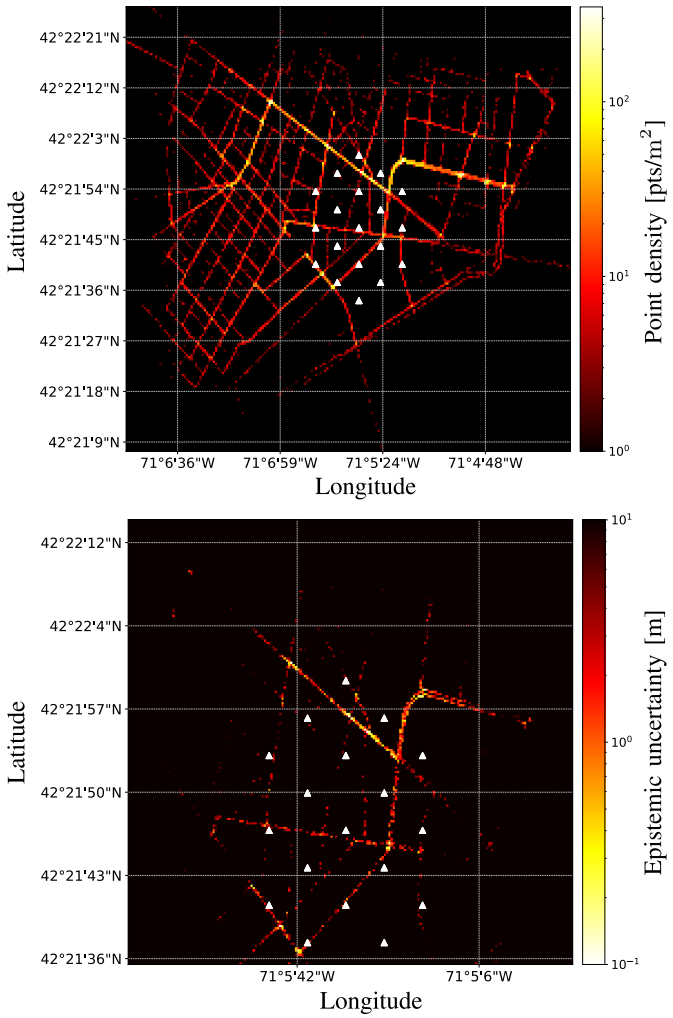


Fig. 8. Density of training ADCPM samples [pts/m^2] (top). Predicted testing epistemic uncertainty [m] (bottom).

the boxplot of the testing MAE over the mini-batches varying $L \in \{10, 20, 40, 100\}$. First, we observe that even with a low L, the achieved median MAE is below 80 cm, confirming the capabilities of the model to infer the position from the full CIR. Second, we notice that generally increasing the number samples, i.e., the number of ensembles employed to estimate the posterior predictive distribution, decreases the positioning error. This is true up to a plateau of $L = 40$ where we can fully represent the real output distribution and achieve about 60 cm of error. In SGLD, a typical choice of L for traversing the posterior is given by N/M [62], which is equal to the number of steps to process the whole dataset. However, in practice, this number could be much smaller and thus, for the rest of the paper, we set $L = 40$. We point out that, considering a 1.6 ms of inference time per sample, using only the teacher with $L = 40$ for position prediction would be unfeasible for real-time applications.

3) Out of Distribution Uncertainty Estimation: This experiment has the goal of assessing the epistemic uncertainty of the DL model and BBK method in positions where no or few training samples are given. To verify this behaviour, in Fig. 8, we show the density of the training points

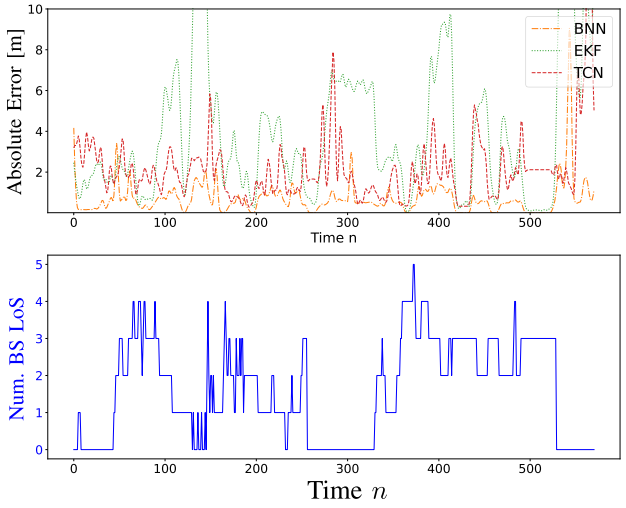


Fig. 9. Tracking performances in terms of absolute error (top). Number of BSs in LOS per timestep (bottom).

(Fig. 8-top) and the predicted epistemic uncertainty of the student over random testing positions around the BS area (Fig. 8-bottom). From the figure, we can clearly notice that the model is much more confident where many training points are provided (around 10 cm for a density 80-90 pts/m^2), confirming the results of previous analysis. On the contrary, in the presence of around 10 pts/m^2, the student correctly predicts an uncertainty of about 50 cm. In the extreme cases of only 1 point and 2 points, the uncertainty greater than 2 m clearly indicates that the model is highly uncertain in those areas. Intuitively, this spatial uncertainty might be linked to the spatial decorrelation distance of the 5G system, but we leave this problem for future research. We point out that, by correctly predicting the epistemic uncertainty, the system automatically generalizes on unseen input samples as the predicted uncertainty is then adopted by the tracking system to weight the importance of the sample by means of the likelihood function.

4) Mobile Positioning in Urban Environment: This final experiment has the objective of comparing the performances of the integrated BNN tracking method with respect to two baselines: an EKF and a state-of-the-art TCN model described in [56]. For a fair comparison, both the BNN-based approach and the EKF adopt the same motion model (i.e., a random walk with 2 m standard deviation on the position), but they differ in the update step. For the EKF, we employ, as in conventional geometric localization, LOS time difference of flight (TDOF) measurements, estimated from the cross-correlation with the SRS according to 3GPP standard, and LOS AOA measurements, obtained through the multiple signal classification (MUSIC) algorithm [93]. Given the high blockage level of 5G signals due to the buildings, the UEs are also equipped with a GNSS receiver from which noisy measurements of the state are gathered. The standard deviation of the Gaussian noise on the GNSS measurements is set to 2 m and it serves as an upper-bound on the accuracy. On the contrary, for BNN-based tracking, we just employ the output of the real-time

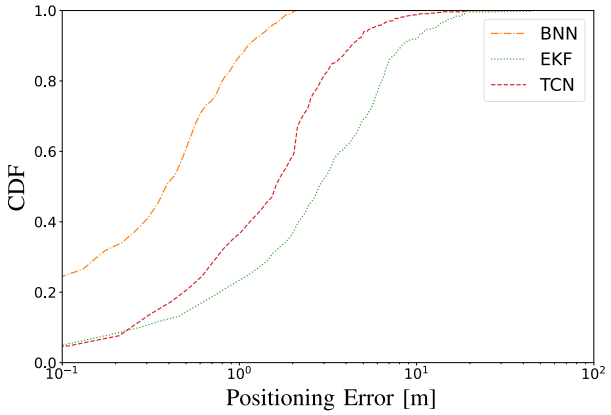


Fig. 10. Positioning performances in terms of cumulative density function (CDF) of the distance error for the proposed BNN-based tracking, EKF and TCN.

student with BBK method obtained from both LOS and NLOS ADCPM measurements. Regarding the TCN model, as suggested in [56], we consider the 1D CIR of the nearest BS by transforming the 2D ADCPM into a single vector.

The results of the tracking, i.e., testing trajectories in Fig. 4, are shown in Fig. 9 where we report the absolute location error per timestep, together with the number of BSs in LOS. Moreover, in Fig. 10 we also report the CDF of the positioning error for all the methods. From Fig. 9 we can clearly notice that the EKF struggles to achieve a location accuracy of 2 m when the number of LOS BSs is less than 3. However, even in a dense UMi scenario with 19 sites (57 BSs), the average number of LOS BSs is 1.6, leading to frequent inaccuracies in positioning. On the contrary, the TCN model achieves slightly higher performances by tracking the UE position even with just one BS measurement, thanks to the fingerprinting approach. However, among all methods, the BNN-based tracking consistently achieves sub-meter accuracy even in the absence of any LOS BS. This is mainly due to its ability to exploit BSs cooperation by fusing multiple NLOS position estimations and to the usage of 2D ADCPM with an AE structure which captures spatial relations in the input. Finally, observing the CDF of the absolute error, we can notice that the BNN-based tracking outperforms both the TCN and EKF by reaching a median error of 46 cm and under 1 m in 87% of the cases.

VII. Conclusion

In this paper, we addressed the problem of real-time 6G tracking in dense urban environments under heavy signal blockage, by presenting a first step toward the development of reliable and trustworthy DL models for precise positioning. We propose a novel teacher-student BNN method, namely BBK, which permits the prediction of real-time location estimates, together with an evaluation of both aleatoric and epistemic uncertainties. Estimation of both terms is of utmost importance for providing reliability indicators in critical applications and for optimizing the positioning process (e.g., augmenting training in uncertain areas). This enables the integration of the BNN method into a proposed tracking system, which seamlessly combines with existing tracking algorithms by substituting or enhancing the measurement-update step. The BNN method is applied to a proposed AE-based DL model, as foreseen by 3GPP standard, which takes as input the whole CIR by means of a 2D ADCPM. This permits to exploit position-linked attributes like TOF, AOA, and RSS of each propagation path, as a channel fingerprint.

The real-time BBK method and tracking integration are tested in a realistic C-ITS setting within a 3GPP-specification compliant UMi scenario, created by means of 3D maps and advanced ray-tracing simulations. The results show that the proposed BBK methodology is able to estimate both the aleatoric uncertainty, outperforming the state-of-the-art real-time BDK method, and the epistemic uncertainty in OOD scenarios from the reference teacher. Regarding mobile positioning performances, the proposed cooperative tracking methodology outperforms geometric-based tracking filters and state-of-the-art TCN models by localizing a moving target with a median absolute error of 46 cm.

Future works include extending our approach to both indoor and outdoor scenarios with next-generation cellular networks. In indoor scenarios, the main challenges include severe multipath and frequent changes of the channel characteristics due to moving objects in the environment. In outdoor scenarios, implementation and assessment in real-world C-ITS is an important research direction, where training procedure can be performed by digital twin simulation.

Appendix A
Proof of (16)

To prove (16), we start by writing the function $A(\boldsymbol{w}|\boldsymbol{x})$ as

$$\begin{aligned} A(\boldsymbol{w}|\boldsymbol{x}) &= \mathrm{KL}\big(p(t|\boldsymbol{x},\mathcal{D})\|p(t|\boldsymbol{x},\boldsymbol{w})\big) \\ &= \int p(t|\boldsymbol{x},\mathcal{D}) \log \frac{p(t|\boldsymbol{x},\mathcal{D})}{p(t|\boldsymbol{x},\boldsymbol{w})} dt \\ &\backsimeq -\mathbb{E}_{p(t|\boldsymbol{x},\mathcal{D})}\big\{\log p(t|\boldsymbol{x},\boldsymbol{w})\big\} \end{aligned} \tag{31}$$

where the last approximation comes from the removal of constant terms in $\boldsymbol{w}$. Therefore, we can write

$$\begin{aligned} A(\boldsymbol{w}|\boldsymbol{x}) &\backsimeq -\int \left(\int p(t|\boldsymbol{x},\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta}\right) \log p(t|\boldsymbol{x},\boldsymbol{w}) dt \\ &= -\int p(\boldsymbol{\theta}|\mathcal{D}) \left(\int p(t|\boldsymbol{x},\boldsymbol{\theta}) \log p(t|\boldsymbol{x},\boldsymbol{w}) dt\right) d\boldsymbol{\theta} \\ &= -\int p(\boldsymbol{\theta}|\mathcal{D}) \left(\mathbb{E}_{p(t|\boldsymbol{x},\boldsymbol{\theta})}\big\{\log p(t|\boldsymbol{x},\boldsymbol{w})\big\}\right) d\boldsymbol{\theta} \\ &\backsimeq -\frac{1}{L}\sum_{\ell=1}^{L} \mathbb{E}\Big\{\log p(t|\boldsymbol{x},\boldsymbol{w})\Big\} = \frac{1}{L}\sum_{\ell=1}^{L} A(\boldsymbol{w}|\boldsymbol{x},\boldsymbol{\theta}_\ell) \end{aligned} \tag{32}$$

where we adopted the MC approximation for calculating the integral using the samples $\boldsymbol{\theta}_\ell$. For regression tasks, $p(t|\boldsymbol{x},\boldsymbol{\theta}_\ell) = \mathcal{N}\left(t; y^{(\mathrm{T})}(\boldsymbol{x},\boldsymbol{\theta}_\ell), {\sigma_{\boldsymbol{\epsilon}^{(\mathrm{T})}}}^2\right)$, whereas $p(t|\boldsymbol{x},\boldsymbol{w}) = \mathcal{N}(t; y^{(\mathrm{S})}(\boldsymbol{x},\boldsymbol{w}), \sigma_{\boldsymbol{\epsilon}^{(\mathrm{S})}}(\boldsymbol{x})^2)$. Thus, we can write

$A(\boldsymbol{w}|\boldsymbol{x},\boldsymbol{\theta}_\ell)$ without considering the constant values in $\boldsymbol{w}$ as

$$A(\boldsymbol{w}|\boldsymbol{x},\boldsymbol{\theta}_\ell) = -\int p(t|\boldsymbol{x},\boldsymbol{\theta}_\ell)\log p(t|\boldsymbol{x},\boldsymbol{w})dt$$
$$\backsimeq \int p(t|\boldsymbol{x},\boldsymbol{\theta}_\ell)\Big(-\frac{\log\left(\sigma_{\boldsymbol{\epsilon}^{(\mathrm{S})}}(\boldsymbol{x})^2\right)}{2}+\frac{\|y^{(\mathrm{S})}(\boldsymbol{x},\boldsymbol{w})-t\|_2^2}{2\sigma_{\boldsymbol{\epsilon}^{(\mathrm{S})}}(\boldsymbol{x})^2}\Big)dt. \tag{33}$$

Finally, by splitting the integral for each of the terms inside the L2 norm and by adopting completion of squares, we obtain

$$\begin{aligned}A(\boldsymbol{w}|\boldsymbol{x},\boldsymbol{\theta}_\ell) &\backsimeq \frac{1}{2}\log\left(\sigma_{\boldsymbol{\epsilon}^{(\mathrm{S})}}(\boldsymbol{x})^2\right)+\frac{1}{2}\sigma_{\boldsymbol{\epsilon}^{(\mathrm{S})}}(\boldsymbol{x})^{-2}\\ &\quad\times\left(\sigma_{\epsilon^{(\mathrm{T})}}{}^2+\left\|y^{(\mathrm{T})}(\boldsymbol{x},\boldsymbol{\theta}_\ell)-y^{(\mathrm{S})}(\boldsymbol{x},\boldsymbol{w})\right\|_2^2\right)\\ &\backsimeq \frac{1}{2}\log\left(y_{\mathrm{al}}^{(\mathrm{S})}(\boldsymbol{x},\boldsymbol{w})\right)+\frac{1}{2}y_{\mathrm{al}}^{(\mathrm{S})}(\boldsymbol{x},\boldsymbol{w})^{-1}\\ &\quad\times\left(\sigma_{\epsilon^{(\mathrm{T})}}{}^2+\left\|y^{(\mathrm{T})}(\boldsymbol{x},\boldsymbol{\theta}_\ell)-y^{(\mathrm{S})}(\boldsymbol{x},\boldsymbol{w})\right\|_2^2\right)\end{aligned} \tag{34}$$

where the final approximation arises from the model (14).

APPENDIX B
PROOF OF (17)

We derive the loss function block $B(\boldsymbol{w}|\boldsymbol{x})$ that permits the student network to learn the epistemic uncertainty of the teacher. By recalling the model in (14), we start by writing the negative log-likelihood according to the MLE approach:

$$\begin{aligned}B(\boldsymbol{w}|\boldsymbol{x}) &= -\log\left(p\left(\mathbb{V}\{\mathsf{t}|\mathbf{x},\mathsf{D},\boldsymbol{\epsilon}^{(\mathrm{T})}\}\middle|\boldsymbol{x},\boldsymbol{w}\right)\right)\\ &= -\log\left(\mathcal{N}\left(\mathbb{V}\{\mathsf{t}|\mathbf{x},\mathsf{D},\boldsymbol{\epsilon}^{(\mathrm{T})}\};\, y_{\mathrm{ep}}^{(\mathrm{S})}(\boldsymbol{x},\boldsymbol{w}),\sigma_{\boldsymbol{\xi}_{\mathrm{ep}}^{(\mathrm{S})}}{}^2\right)\right).\end{aligned} \tag{35}$$

Then, we remove constant values in $\boldsymbol{w}$ and approximate the epistemic uncertainty with the predictive epistemic uncertainty of the teacher as

$$\begin{aligned}B(\boldsymbol{w}|\boldsymbol{x}) &\backsimeq \frac{1}{2\sigma_{\boldsymbol{\xi}_{\mathrm{ep}}^{(\mathrm{S})}}{}^2}\left\|\mathbb{V}\{\mathsf{t}|\mathbf{x},\mathsf{D},\boldsymbol{\epsilon}^{(\mathrm{T})}\}-y_{\mathrm{ep}}^{(\mathrm{S})}(\boldsymbol{x},\boldsymbol{w})\right\|_2^2\\ &\backsimeq \frac{1}{2\sigma_{\boldsymbol{\xi}_{\mathrm{ep}}^{(\mathrm{S})}}{}^2}\left\|\frac{1}{L}\sum_{\ell=1}^{L}y^{(\mathrm{T})}(\boldsymbol{x},\boldsymbol{\theta}_\ell)^2\right.\\ &\quad\left.-\left(\frac{1}{L}\sum_{\ell=1}^{L}y^{(\mathrm{T})}(\boldsymbol{x},\boldsymbol{\theta}_\ell)\right)^2-y_{\mathrm{ep}}^{(\mathrm{S})}(\boldsymbol{x},\boldsymbol{w})\right\|_2^2\end{aligned} \tag{36}$$

concluding the derivation.

REFERENCES

[1] M. Z. Win et al., "Network localization and navigation via cooperation," *IEEE Commun. Mag.*, vol. 49, no. 5, pp. 56–62, May 2011.

[2] A. Conti et al., "Location awareness in beyond 5G networks," *IEEE Commun. Mag.*, vol. 59, no. 11, pp. 22–27, Nov. 2021.

[3] J. Talvitie, M. Säily, and M. Valkama, "Orientation and location tracking of XR devices: 5G carrier phase-based methods," *IEEE J. Sel. Topics Signal Process.*, vol. 17, no. 5, pp. 919–934, Sep. 2023.

[4] A. Ghosh, A. Maeder, M. Baker, and D. Chandramouli, "5G evolution: A view on 5G cellular technology beyond 3GPP release 15," *IEEE Access*, vol. 7, pp. 127639–127651, 2019.

[5] C. Yang, S. Mao, and X. Wang, "An overview of 3GPP positioning standards," *GetMobile, Mobile Comput. Commun.*, vol. 26, no. 1, pp. 9–13, May 2022.

[6] T. Nakamura, "5G evolution and 6G," in *Proc. IEEE Symp. VLSI Technol.*, Sep. 2020, pp. 1–5.

[7] J. Levy, *RAN Workshop on 5G: Chairman Summary*, document RWS-150073, InterDigital Communication Inc., New York, NY, USA, Sep. 2015.

[8] J. A. del Peral-Rosado, R. Raulefs, J. A. López-Salcedo, and G. Seco-Granados, "Survey of cellular mobile radio localization methods: From 1G to 5G," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 1124–1148, 2nd Quart., 2018.

[9] S. Parkvall et al., "5G NR release 16: Start of the 5G evolution," *IEEE Commun. Standards Mag.*, vol. 4, no. 4, pp. 56–63, Dec. 2020.

[10] I. Rahman et al., "5G evolution toward 5G advanced: An overview of 3GPP releases 17 and 18," *Ericsson Technol. Rev.*, vol. 2021, no. 14, pp. 2–12, Oct. 2021.

[11] *Study on Artificial Intelligence (AI)/Machine Learning (ML) for NR Air Interface*, document TR 38.843, Version 18.0.0, 3rd Gener. Partnership Project (3GPP), Sophia Antipolis, France, Jan. 2022.

[12] F. Morselli, S. M. Razavi, M. Z. Win, and A. Conti, "Soft information based localization for 5G networks and beyond," *IEEE Trans. Wireless Commun.*, vol. 22, no. 12, pp. 9923–9938, Dec. 2023.

[13] H. van der Veen, *Summary of RAN Rel-18 Workshop*, document RWS-210659, NEC Laboratories Eur., Heidelberg, Germany, Jul. 2021.

[14] G. Torsoli, M. Z. Win, and A. Conti, "Blockage intelligence in complex environments for beyond 5G localization," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 6, pp. 1688–1701, Jun. 2023.

[15] J. Zheng, J. Zhang, J. Cheng, V. C. M. Leung, D. W. K. Ng, and B. Ai, "Asynchronous cell-free massive MIMO with rate-splitting," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 5, pp. 1366–1382, May 2023.

[16] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May 2020.

[17] N. Zhu, J. Marais, D. Bétaille, and M. Berbineau, "GNSS position integrity in urban environments: A review of literature," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 9, pp. 2762–2778, Sep. 2018.

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

[19] A. De Angelis and C. Fischione, "Mobile node localization via Pareto optimization: Algorithm and fundamental performance limitations," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 7, pp. 1288–1303, Jul. 2015.

[20] Z. Liu, A. Conti, S. K. Mitter, and M. Z. Win, "Communication-efficient distributed learning over networks—Part I: Sufficient conditions for accuracy," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 1081–1101, Apr. 2023.

[21] Z. Liu, A. Conti, S. K. Mitter, and M. Z. Win, "Communication-efficient distributed learning over networks—Part II: Necessary conditions for accuracy," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 1102–1119, Apr. 2023.

[22] M. Z. Win, W. Dai, Y. Shen, G. Chrisikos, and H. V. Poor, "Network operation strategies for efficient localization and navigation," *Proc. IEEE*, vol. 106, no. 7, pp. 1224–1254, Jul. 2018.

[23] A. Maddumabandara, H. Leung, and M. Liu, "Experimental evaluation of indoor localization using wireless sensor networks," *IEEE Sensors J.*, vol. 15, no. 9, pp. 5228–5237, Sep. 2015.

[24] D. Fortin-Simard, J.-S. Bilodeau, K. Bouchard, S. Gaboury, B. Bouchard, and A. Bouzouane, "Exploiting passive RFID technology for activity recognition in smart homes," *IEEE Intell. Syst.*, vol. 30, no. 4, pp. 7–15, Feb. 2015.

[25] A. Conti, M. Guerra, D. Dardari, N. Decarli, and M. Z. Win, "Network experimentation for cooperative localization," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 2, pp. 467–475, Feb. 2012.

[26] U. A. Khan, S. Kar, and J. M. F. Moura, "Distributed sensor localization in random environments using minimal number of anchor nodes," *IEEE Trans. Signal Process.*, vol. 57, no. 5, pp. 2000–2016, May 2009.

[27] A. Conti, S. Mazuelas, S. Bartoletti, W. C. Lindsey, and M. Z. Win, "Soft information for Localization-of-Things," *Proc. IEEE*, vol. 107, no. 11, pp. 2240–2264, Sep. 2019.

[28] F. Meyer et al., "Message passing algorithms for scalable multitarget tracking," *Proc. IEEE*, vol. 106, no. 2, pp. 221–259, Feb. 2018.

[29] S. Bartoletti, A. Giorgetti, M. Z. Win, and A. Conti, "Blind selection of representative observations for sensor radar networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1388–1400, Apr. 2015.

[30] X. Lin, "An overview of 5G advanced evolution in 3GPP release 18," *IEEE Wireless Commun. Stand. Mag.*, vol. 6, no. 3, pp. 77–83, Sep. 2022.

[31] K. Gao, H. Wang, H. Lv, and W. Liu, "Toward 5G NR high-precision indoor positioning via channel frequency response: A new paradigm and dataset generation method," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 7, pp. 2233–2247, Jul. 2022.

[32] A. Conti, G. Torsoli, C. A. Gómez-Vega, A. Vaccari, G. Mazzini, and M. Z. Win, "3GPP-compliant datasets for xG location-aware networks," *IEEE Open J. Veh. Technol.*, vol. 5, pp. 473–484, 2024.

[33] A. Conti, G. Torsoli, C. A. Gómez-Vega, A. Vaccari, and M. Z. Win, 2023, "xG-Loc: 3GPP-compliant datasets for xG location-aware networks," *IEEE Dataport*, Dec. 2023, doi: 10.21227/rper-vc03.

[34] B. C. Tedeschini and M. Nicoli, "Cooperative deep-learning positioning in mmWave 5G-advanced networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 12, pp. 3799–3815, Dec. 2023.

[35] B. C. Tedeschini, M. Nicoli, and M. Z. Win, "On the latent space of mmWave MIMO channels for NLOS identification in 5G-advanced systems," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 6, pp. 1655–1669, May 2023.

[36] F. Liu et al., "Integrated sensing and communications: Toward dual-functional wireless networks for 6G and beyond," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 6, pp. 1728–1767, Jun. 2022.

[37] G. Kwon, Z. Liu, A. Conti, H. Park, and M. Z. Win, "Integrated localization and communication for efficient millimeter wave networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 12, pp. 3925–3941, Dec. 2023.

[38] R. Liu et al., "Integrated sensing and communication based outdoor multi-target detection, tracking, and localization in practical 5G networks," *Intell. Converged Netw.*, vol. 4, no. 3, pp. 261–272, Sep. 2023.

[39] N. Decarli, A. Guerra, C. Giovannetti, F. Guidi, and B. M. Masini, "V2X sidelink localization of connected automated vehicles," *IEEE J. Select. Areas Commun.*, vol. 42, no. 1, pp. 120–133, Jan. 2024.

[40] L. Barbieri, B. C. Tedeschini, M. Brambilla, and M. Nicoli, "Implicit vehicle positioning with cooperative LiDAR sensing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.

[41] N. Piperigkos, A. S. Lalos, and K. Berberidis, "Graph Laplacian diffusion localization of connected and automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 12176–12190, Aug. 2022.

[42] L. V. Jospin, H. Laga, F. Boussaid, W. Buntine, and M. Bennamoun, "Hands-on Bayesian neural networks—A tutorial for deep learning users," *IEEE Comput. Intell. Mag.*, vol. 17, no. 2, pp. 29–48, May 2022.

[43] M. Abdar et al., "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Inf. Fusion*, vol. 76, pp. 243–297, Dec. 2021.

[44] *Study Enhancement 3GPP Support for 5G V2X Services*, document TR 22.886, Version 16.2.0, 3rd Gener. Partnership Project (3GPP), Sophia Antipolis, France, Dec. 2018.

[45] A. Fouda, R. Keating, and H.-S. Cha, "Toward cm-level accuracy: Carrier phase positioning for IIoT in 5G-advanced NR networks," in *Proc. IEEE 33rd Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2022, pp. 782–787.

[46] *Study Expanded Improved NR Positioning*, document TR 38.859, Version 18.0.0, 3rd Gener. Partnership Project (3GPP), Sophia Antipolis, France, Dec. 2023.

[47] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-based fingerprinting for indoor localization: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 763–776, Jan. 2017.

[48] Y. Chapre, A. Ignjatovic, A. Seneviratne, and S. Jha, "CSI-MIMO: An efficient Wi-Fi fingerprinting using channel state information with MIMO," *Pervasive Mobile Comput.*, vol. 23, pp. 89–103, Oct. 2015.

[49] 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. 18066–18074, 2017.

[50] X. Wang, X. Wang, and S. Mao, "ResLoc: Deep residual sharing learning for indoor localization with CSI tensors," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–6.

[51] X. Wang, X. Wang, and S. Mao, "CiFi: Deep convolutional neural networks for indoor localization with 5 GHz Wi-Fi," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[52] A. Foliadis, M. H. C. Garcia, R. A. Stirling-Gallacher, and R. S. Thomä, "CSI-based localization with CNNs exploiting phase information," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2021, pp. 1–6.

[53] C. Wu et al., "Learning to localize: A 3D CNN approach to user positioning in massive MIMO-OFDM systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 7, pp. 4556–4570, Jul. 2021.

[54] A. Shahmansoori, B. Uguen, G. Destino, G. Seco-Granados, and H. Wymeersch, "Tracking position and orientation through millimeter wave lens MIMO in 5G systems," *IEEE Signal Process. Lett.*, vol. 26, no. 8, pp. 1222–1226, Aug. 2019.

[55] H. Kim, H. Wymeersch, N. Garcia, G. Seco-Granados, and S. Kim, "5G mmWave vehicular tracking," in *Proc. 52nd Asilomar Conf. Signals, Syst., Comput.*, Oct. 2018, pp. 541–547.

[56] J. Gante, G. Falc ao, and L. Sousa, "Deep learning architectures for accurate millimeter wave positioning in 5G," *Neural Process. Lett.*, vol. 51, no. 1, pp. 487–514, Feb. 2020.

[57] Y. Ruan, L. Chen, X. Zhou, G. Guo, and R. Chen, "Hi-Loc: Hybrid indoor localization via enhanced 5G NR CSI," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–15, 2023.

[58] E. Hullermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," *Mach. Learn.*, vol. 110, no. 3, pp. 457–506, 2021.

[59] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" 2017, *arXiv:1703.04977*.

[60] J. Postels et al., "On the practicality of deterministic epistemic uncertainty," 2021, *arXiv:2107.00649*.

[61] P. Izmailov, S. Vikram, M. D. Hoffman, and A. Gordon Wilson, "What are Bayesian neural network posteriors really like?" 2021, *arXiv:2104.14421*.

[62] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient Langevin dynamics," in *Proc.28th Int. Conf. Mach. Learn.*, Jun. 5555, pp. 681–688.

[63] E. Daxberger, A. Kristiadi, A. Immer, R. Eschenhagen, M. Bauer, and P. Hennig, "Laplace redux—Effortless Bayesian deep learning," 2021, *arXiv:2106.14806*.

[64] M. Teye, H. Azizpour, and K. Smith, "Bayesian uncertainty estimation for batch normalized deep networks," 2018, *arXiv:1802.06455*.

[65] M. E. Khan, D. Nielsen, V. Tangkaratt, W. Lin, Y. Gal, and A. Srivastava, "Fast and scalable Bayesian deep learning by weight-perturbation in Adam," 2018, *arXiv:1806.04854*.

[66] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in *Proc. 32nd Int. Conf. Mach. Learn.*, Jul. 2015, pp. 1613–1622.

[67] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," 2015, *arXiv:1506.02142*.

[68] A. Korattikara, V. Rathod, K. Murphy, and M. Welling, "Bayesian dark knowledge," 2015, *arXiv:1506.04416*.

[69] *Study NR Positioning Support*, document TR 38.855, Version 16.0.0, 3rd Gener. Partnership Project (3GPP), Sophia Antipolis, France, Sep. 2019.

[70] P. A. Lopez et al., "Microscopic traffic simulation using SUMO," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2575–2582.

[71] H. L. Van Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*, 1st ed. Hoboken, NJ, USA: Wiley, Mar. 2002.

[72] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, Dec. 2005.

[73] X. Sun, X. Gao, G. Y. Li, and W. Han, "Single-site localization based on a new type of fingerprint for massive MIMO-OFDM systems," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6134–6145, Jul. 2018.

[74] C. M. Bishop, *Pattern Recognition and Machine Learning*, vol. 4. New York, NY, USA: Springer, Aug. 2006.

[75] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[76] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[77] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[78] K. B. Petersen and M. S. Pedersen, *The Matrix Cookbook*, document Version 20121115, Nov. 2012. [Online]. Available: http://www2.compute.dtu.dk/pubdb/pubs/3274-full.html

[79] *Study Channel Model for Frequencies From 0.5 to 100 GHz (Rel-16)*, document TR 38.901, Version 16.1.0, 3rd Gener. Partnership Project (3GPP), Sophia Antipolis, France, Nov. 2020.

[80] C.-F. Yang and C.-J. Ko, "A ray tracing method for modeling indoor wave propagation and penetration," in *IEEE Antennas Propag. Soc. Int. Symp. Dig.*, vol. 1, Jul. 1996, pp. 441–444.

[81] H.-J. Li, C.-C. Chen, T.-Y. Liu, and H.-C. Lin, "Applicability of ray-tracing technique for the prediction of outdoor channel characteristics," *IEEE Trans. Veh. Technol.*, vol. 49, no. 6, pp. 2336–2349, Nov. 2000.

[82] A. Hsiao, C. Yang, T. Wang, I. Lin, and W. Liao, "Ray tracing simulations for millimeter wave propagation in 5G wireless communications," in *Proc. IEEE Int. Symp. Antennas Propag., USNC/URSI Nat. Radio Sci. Meeting*, Oct. 2017, pp. 1901–1902.

[83] (2023). *Wireless InSite 3D Wireless Prediction Software*. Accessed: Nov. 2023. [Online]. Available: https://www.remcom.com/wireless-insite-em-propagation-software

[84] *Guidelines for Evaluation of Radio Interface Technologies for IMT-2020*, document ITU-R M.2412-0, Int. Telecommun. Union, Geneva, Switzerland, Oct. 2017.

[85] P. Wang, B. Di, H. Zhang, K. Bian, and L. Song, "Platoon cooperation in cellular V2X networks for 5G and beyond," *IEEE Trans. Wireless Commun.*, vol. 18, no. 8, pp. 3919–3932, Aug. 2019.

[86] S. Roger, M. Brambilla, B. C. Tedeschini, C. Botella-Mascarell, M. Cobos, and M. Nicoli, "Deep-learning-based radio map reconstruction for V2X communications," *IEEE Trans. Veh. Technol.*, vol. 73, no. 3, pp. 3863–3871, Mar. 2024.

[87] L. Gopal, Y. Rong, and Z. Zang, "Tomlinson–Harashima precoding based transceiver design for MIMO relay systems with channel covariance information," *IEEE Trans. Wireless Commun.*, vol. 14, no. 10, pp. 5513–5525, Oct. 2015.

[88] A. Hindy and A. Nosratinia, "Ergodic fading MIMO dirty paper and broadcast channels: Capacity bounds and lattice strategies," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 5525–5536, Aug. 2017.

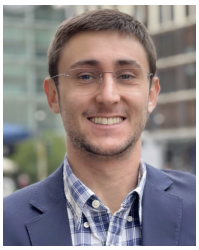
[89] R. Karásek and C. Gentner, "Stochastic data association for multipath assisted positioning using a single transmitter," *IEEE Access*, vol. 8, pp. 46735–46752, 2020.

[90] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[91] G. Kalfas et al., "Next generation fiber-wireless Fronthaul for 5G mmWave networks," *IEEE Commun. Mag.*, vol. 57, no. 3, pp. 138–144, Mar. 2019.

[92] A. Paszke et al., "Automatic differentiation in PyTorch," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, Oct. 2017, pp. 1–4.

[93] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. AP-34, no. 3, pp. 276–280, Mar. 1986.



Bernardo Camajori Tedeschini (Graduate Student Member, IEEE) is pursuing the Ph.D. degree in Information Technology at the Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano, Milan, Italy, since November 2021. He received his M.Sc. (Hons.) degree in Telecommunications Engineering and B.Sc. (Hons.) degree in Computer Science from the Politecnico di Milano, Milan, Italy, in 2021 and 2019, respectively.

He is currently a visiting student with the Wireless Information and Network Sciences Laboratory at the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA. In 2021, he has served as a Visiting Research Scientist at CERN, Geneva, Switzerland, where he worked on the CAFEIN project, focusing on the development and deployment of a Federated network platform. His research interests encompass federated learning, machine learning for signal processing and sensing over networks, and localization methods.

Mr. Camajori Tedeschini is a recipient of a Ph.D. grant from Italy's Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR) and the Roberto Rocca Doctoral Fellowship, which was jointly awarded by MIT and Politecnico di Milano. He earned both his Bachelor's and Master's degrees with highest honors and he was honored with the best freshmen prize from Politecnico di Milano in 2017.

Girim Kwon (Member, IEEE) received the B.S. degree (with the highest honor) in electrical engineering from the University of Seoul, Seoul, South Korea, in 2013, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2014 and 2020, respectively.

He is currently a Postdoctoral Fellow with the Wireless Information and Network Sciences Laboratory at the Massachusetts Institute of Technology, Cambridge, MA, USA. His main areas of research are in statistical inference, information theory, optimization methods, and machine learning with applications to real-world problems, including wireless communications, network localization and navigation, and non-terrestrial networks.

Dr. Kwon received the S-Oil Best Dissertation Award in 2021, the Best Ph.D. Dissertation Award from KAIST in 2020, the ICT Paper Award from the Electronic Times in 2018, and the Global Ph.D. Fellowship from the Korean Government in 2015.

Monica Nicoli (Senior Member, IEEE) received the M.Sc. (Hons.) and Ph.D. degrees in communication engineering from Politecnico di Milano, Milan, Italy, in 1998 and 2002, respectively. She was a Visiting Researcher with ENI Agip, from 1998 to 1999, and Uppsala University, in 2001. In 2002, she joined Politecnico di Milano as a Faculty Member. She is currently an Associate Professor in telecommunications with the Department of Management, Economics and Industrial Engineering.

Her research interests include signal processing, machine learning, and wireless communications, with emphasis on smart mobility and Internet of Things (IoT). She was a recipient of the Marisa Bellisario Award, in 1999, and a co-recipient of the best paper awards of the EuMA Mediterranean Microwave Symposium, in 2022, the IEEE Symposium on Joint Communications and Sensing, in 2021, the IEEE Statistical Signal Processing Workshop, in 2018, and the *IET Intelligent Transport Systems journal*, in 2014. She is an Associate Editor of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS. She has also served as an Associate Editor for the *EURASIP Journal on Wireless Communications and Networking*, from 2010 to 2017, and a Lead Guest Editor for the Special Issue on Localization in Mobile Wireless and Sensor Networks, in 2011.

Moe Z. Win (Fellow, IEEE) is the Robert R. Taylor Professor at the Massachusetts Institute of Technology (MIT) and the founding director of the Wireless Information and Network Sciences Laboratory. Prior to joining MIT, he was with AT&T Research Laboratories and with the NASA Jet Propulsion Laboratory.

His research encompasses fundamental theories, algorithm design, and network experimentation for a broad range of real-world problems. His current research topics include ultra-wideband systems, network localization and navigation, network interference exploitation, and quantum information science. He has served the IEEE Communications Society as an elected Member-at-Large on the Board of Governors, as elected Chair of the Radio Communications Committee, and as an IEEE Distinguished Lecturer. Over the last two decades, he held various editorial positions for IEEE journals and organized numerous international conferences. He has served on the SIAM Diversity Advisory Committee.

Dr. Win is an elected Fellow of the AAAS, the EURASIP, the IEEE, and the IET. He was honored with two IEEE Technical Field Awards: the IEEE Kiyo Tomiyasu Award (2011) and the IEEE Eric E. Sumner Award (2006, jointly with R. A. Scholtz). His publications, co-authored with students and colleagues, have received several awards. Other recognitions include the MIT Frank E. Perkins Award (2024), the MIT Everett Moore Baker Award (2022), the IEEE Vehicular Technology Society James Evans Avant Garde Award (2022), the IEEE Communications Society Edwin H. Armstrong Achievement Award (2016), the Cristoforo Colombo International Prize for Communications (2013), the Copernicus Fellowship (2011) and the *Laurea Honoris Causa* (2008) from the Università degli Studi di Ferrara, and the U.S. Presidential Early Career Award for Scientists and Engineers (2004).