Data-Induced Intelligent Kalman Filtering Approach and Application in Beam Prediction and Tracking

Jianjun Zhang, *Member, IEEE*, Yongming Huang, *Senior Member, IEEE*
Christos Masouros, *Senior Member, IEEE*, Xiaohu You, *Fellow, IEEE*
and Björn Ottersten, *Fellow, IEEE*

***Abstract*—Beam prediction and tracking (BPT) are key technology for high-frequency communications. Typical techniques include Kalman filtering and Gaussian process regression (GPR). However, Kalman filter requires explicit system dynamics equation, which is challenging to obtain, especially for complicated environments. In contrast, as a data-driven approach, there is no need to derive the system dynamics equation for GPR. However, the computational complexity of GPR is often prohibitive, which makes real-time application challenging. To tackle this issue, we propose a novel hybrid model and data driven approach in this paper, which can exploit simultaneously the advantages of the two techniques while overcoming their drawbacks. In particular, the required system dynamics can be obtained via the data-driven manner. In view that the system dynamics has been available, we further investigate the long-term behavior and propose two more efficient algorithms - long-term prediction and beam width optimization. Our BPT approach enjoys two advantages. First, the computational complexity is low due to the inherent Kalman filter, which facilitates real-time implementation. Second, system performance can be significantly improved thanks to the long-term prediction and beam width optimization.**

***Index Terms*—Beam prediction, beam tracking, hybrid model and data driven, stochastic differential equation, Kalman filtering, Gaussian process, millimeter wave communications.**

I. Introduction

Millimeter wave (mmwave) communications, occupying 30-300 GHz spectrum resources and offering significant under-utilized bandwidth, have been considered as one of the most promising solutions to meet high-speed wireless data demands in the era of 5G and beyond [1]. However, the high frequencies of mmwave signals lead to a large path-loss, which poses a severe challenge to mmwave communications. To combat the large path-loss, an effective solution is beamforming with high gain, which is realized via large-scale antenna arrays thanks to the short wave-length. Nevertheless, high-gain but narrow beams make beam alignment challenging, especially in mobile applications or dynamic environments [2], [3].

To obtain channel state information in mmwave communications, different design methodologies have been investigated, among which the most widely accepted one is beam training and tracking [4]–[9]. This scheme consists of two stages, i.e., initial beam alignment and subsequent beam tracking. In the first stage, the optimal beam or beam pair is found via adaptive or hierarchical search [4], [6], [7]. In general, a large training overhead is involved in this stage. To enable efficient search, a key task in this stage is to design a codebook that has desired properties [6], [7], [10], [11]. To avoid frequent search and thus reduce the training overhead, beam tracking is invoked in the second stage. Compared to the initial beam align-ment, the number of beams used for tracking is often very small, e.g., maybe even only one beam. If beam tracking fails, the (initial) beam alignment operation should be invoked again. In industry, beam training or tracking involving beam measurements is also referred to as beam sounding.

The key of beam tracking is beam prediction, i.e., to predict a beam subspace that contains the real beam. Apparently, two challenging issues are closely related to beam prediction. The first one is success rate and prediction efficiency, i.e., the beam subspace predicted should contain the real optimal beam, and meanwhile, the beam subspace should be as small as possible. The second one is the complexity of prediction, including both sample complexity and inference complexity. To tackle the two issues, various temporal and spatial channel correlations have been excavated and exploited. The existing beam prediction algorithms roughly fall into two categories, i.e., the classical Kalman filtering based methods [12]–[15] and recent machine learning (ML) based methods [16]–[22].

The most important step toward beam prediction is to construct an appropriate prediction model. The Kalman filtering technique addresses this issue by building a dynamical model for the underlying physical system. Specifically, two stochastic differential equations (SDEs), often referred to as state-space and measurement equations, are established. As long as the two SDEs are available, the well-known Kalman filter can be invoked. An appealing advantage of the Kalman filtering based methods is that they have low computational complexity. In particular, the scaling of computational complexity for Kalman filter is linear $\mathcal{O}(N)$ (where N is the number of samples), as opposed to the cubic scaling $\mathcal{O}(N^3)$ for GPR. This advantage is attributed to the fact that the underlying system dynamics is used explicitly. But the dynamics model is obtained via manual derivation, which fails for complicated environments.

In contrast to Kalman filtering, ML addresses the issue of prediction modeling by employing the data-driven mode. In fact, one powerful ability of ML is that it can automatically

J. Zhang is with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China. (Email: jianjun.zhang@nuaa.edu.cn).

Y. Huang and and X. You are with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China. (Email: huangym@seu.edu.cn; xhyu@seu.edu.cn).

C. Masouros is with the Department of Electronic & Electrical Engineering, University College London, London, U.K. (E-mail: c.masouros@ucl.ac.uk).

B. Ottersten is with the Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, L-1359 Luxembourg, Luxembourg (Email: bjorn.ottersten@uni.lu).

extract meaningful patterns and derive an appropriate model from observed data directly. The ML-based beam prediction methods fall into two categories, i.e., reinforcement learning (RL) based algorithms [19]–[23] and supervised learning (SL) based algorithms. A salient advantage of the RL-based solutions is that they can collect training samples via interacting with environments, which enables to implement the algorithms online [24]. However, since the fundamental of RL is Markov decision process, it leads to low convergence rate and fails to achieve good performance on the short term.

Another major category of the ML-based beam prediction solutions is the SL-based algorithms [9], [16]–[18], [25]–[29]. The SL-based solutions circumvent the difficulty of manually building models, due to the data-driven design paradigm. But one drawback of the SL-based solutions is that the number of training samples required is often very large, so as to achieve good performance. Moreover, the rapidly fluctuating wireless environments can also invalidate the deterministic prediction models established. Among the many SL-based solutions, the GPR-based algorithms have attracted considerable attentions [26]–[29], thanks to the abilities of uncertainty calibration and non-parametric modeling. Unfortunately, their computational complexity is typically $\mathcal{O}(N^3)$. Since wireless channels vary quickly in practice, N can be very large. The large complexity becomes an obstacle to real-time applications.

To tackle the aforementioned issues, in this paper we propose a novel hybrid model and data driven based approach, which is referred to as data-induced intelligent Kalman filtering (DIIKF). Our DIIKF can exploit the advantages of both Kalman filtering and GPR techniques while overcoming their drawbacks. In contrast to the existing GPR-based algorithms, whose scaling is often cubic, the scaling of the computational complexity of DIIKF is only linear. Compared to the Kalman filtering methods, there is no need to derive the state-space and measurement equations manually. Instead, the SDEs required can be obtained via the data-driven manner, which makes our approach applicable to more complicated scenarios. Since the state-space equation is available, we propose to predict long-term behavior of the underlying beam process, which can significantly lower the frequency of beam sounding. The main contributions of this paper are summarized as follows:

- We establish the equivalence (in the sense of probability law) between two different system modeling methodologies, namely, the classical SDE-based system dynamics modeling approach and the recent ML-based modeling approach (via the neural network and GPR). The equivalence constitutes the theoretical foundation of our DIIKF approach, which, in fact, addresses the challenging issue of lacking interpretability in ML methods.
- Based on the equivalence, we propose the DIIKF approach, which can exploit the powerful modeling ability from ML and enjoy efficient inference from Kalman filter. Specifically, we model complicated system dynamics via GPR (and the deep neural network) while implementing efficient online inference via Kalman filtering.
- The DIIKF approach consists of two components, i.e., implicit system dynamics modeling and explicit dynamics representation via SDE. We first employ Bayesian multi-task learning to implicitly extract underlying system dynamics. To enable efficient inference, we then convert the implicit representation into explicit SDE representation, based on which the Kalman filter can be invoked.
- In view that the SDE characterizing the system evolution has been available, we further investigate the long-term behavior of the underlying beam process. In particular, we propose to lower the frequency of beam sounding by incorporating long-term prediction via SDE, which can notably improve system performance.
- We apply our DIIKF approach to the BPT problem and propose efficient BPT algorithms. Simulation results are provided to demonstrate the advantages of our algorithms, which are two-fold. First, due to the powerful modeling ability, they can be applicable to complicated scenarios. Second, they have low computational complexity. As a result, they achieve the state-of-the-art performance. Due to the Bayesian design methodology, the desired small sample performance can also be achieved.

The remainder of this paper is organized as follows. System model of BPT is described in Section II. In Section III, the principle of the DIIKF approach is elaborated, and efficient BPT algorithm is also presented. The principle of long-term prediction and the derived algorithm are proposed in Section IV. Simulation results and conclusions are given in Section V and Section VI, respectively. For completeness, the GPR is introduced in the appendix. Proofs of propositions are also deferred to the appendix to improve readability.

Notations: Bold uppercase $\mathbf{A}$ and bold lowercase $\mathbf{a}$ denote matrices and column vectors, respectively. To distinguish, the bold italic uppercase and bold italic lowercase represent matrix function and vector function, respectively. Without particular specification, non-bold letters A, a denote scalars. Caligraphic letters $\mathcal{A}$ stand for sets. $\mathbb{E}(\cdot)$ and $(\cdot)^{\mathrm{H}}$ denote the mathematical expectation and Hermitian operators, respectively. $\mathbb{I}\{\cdot\}$ and $\mathrm{card}(\mathcal{A})$ stand for the indicator function and the cardinality of set $\mathcal{A}$, respectively. $(\cdot)^{\star}$ represents an optimal quantity, e.g., an optimal solution of an optimization problem. $\mathcal{CN}(\mathbf{m}, \mathbf{\Sigma})$ stands for the complex Gaussian random vector with mean $\mathbf{m}$ and covariance matrix $\mathbf{\Sigma}$. $\mathbf{I}$ denotes the identity matrix.

II. System Model

Consider the mmwave point-to-point communication system, which consists of one base station (BS) equipped with N transmit antennas and a single-antenna user (UE). Typical scenarios include the high-speed train, outdoor communication (e.g., a pedestrian walks along a street), vehicle-to-everything (V2X), and so on. Without loss of generality, we take the V2X scenario as an example. To facilitate practical system implementation, we consider the codebook-based analog beamforming, i.e., each beam is chosen from a predefined codebook $\mathcal{C} = \{\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_M\}$, where M is the size of codebook $\mathcal{C}$.

Due to the sparsity of mmwave channels, an extended Saleh-Valenzuela geometric model is considered here. The channel vector between the BS and the UE is given by

$$\mathbf{h} = \sqrt{N/\beta} \sum_{l=1}^{L} \alpha_l \mathbf{a}(\phi_l, \psi_l), \tag{1}$$

where β is the average path-loss, L is the number of paths, and α_l is the complex path gain of the l-th path. In Eq.(1), ϕ_l and ψ_l represent the elevation angle and azimuth angle of the l-th path, respectively. If the i-th beam, i.e., $\mathbf{f}_i$, is chosen by the BS, the signal received at the UE is given by

$$y_i = \sqrt{P}\mathbf{h}^{\mathrm{H}}\mathbf{f}_i s + w_i, \tag{2}$$

where P denotes the transmit power, s with $|s| = 1$ denotes the pilot symbol, and $w_i \sim \mathcal{CN}(0, 1)$ is the received noise.

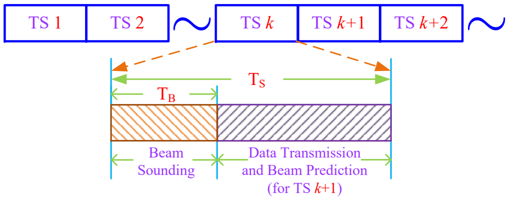

Fig. 1. Frame structure of the typical prediction-and-sweeping based scheme.

The frame structure of the typical prediction-and-sweeping based beam sounding scheme is shown in Fig. 1. For the beam sounding scheme, each time-slot consists of three phases, i.e., beam prediction, beam sweeping and data transmission. First, beam prediction module predicts a small beam subspace, i.e., a small subset of $\mathcal{C}$. Then, the beam sweeping module finds out an optimal beam within the subspace for the subsequent phase of data transmission. The effective achievable rate, introduced to measure the throughput performance, is defined as [9]

$$R_{\text{eff}} = (1 - T_{\mathrm{B}}/T_{\mathrm{S}}) \log\left(1 + P|\mathbf{h}^{\mathrm{H}}\mathbf{f}_i|^2\right), \tag{3}$$

where T_{B} and T_{S} denote the duration of beam sounding within a time-slot and the duration of entire time-slot, respectively.

It can be observed from (3) that to achieve a high throughput, the time allocated for beam sounding T_{B} should be as little as possible, so as to reserve more time for data transmission. To reduce T_{B}, the core is to develop an efficient prediction model, which can predict a small but correct beam subspace. However, it is a nontrivial task in practice. In what follows, we will propose an efficient approach to take this issue.

III. Hybrid Model and Data Driven Approach

It is well-known that Kalman filtering is inference-efficient but modeling-inefficient, while GPR is modeling-efficient but inference-inefficient. It is natural to raise the question, namely, whether we can incorporate the two methods into a novel one, which can exploit the advantages of them while overcoming their drawbacks. In this section, we will tackle this issue by proposing the DIIKF approach. For completeness, a brief introduction of GPR is provided in Appendix A.

A. Outline of the DIIKF Approach

As shown in Fig. 2, to fully exploit the advantages of both Kalman filtering and GPR, the DIIKF approach models the system dynamics via Gaussian process and perform inference or prediction via Kalman filter, i.e., Gaussian process for modeling and Kalman filtering for inference. The key components of DIIKF include historical data collection, prior extraction and implicit representation of system dynamics, model conversion, explicit SDE representation of system dynamics. The main role of these components is as follows.

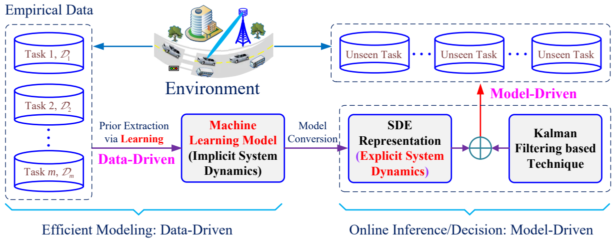
Empirical Data
Task 1, $\mathcal{D}_1$
Task 2, $\mathcal{D}_2$
Task m, $\mathcal{D}_m$
Environment
Unseen Task
Unseen Task
Unseen Task
Model-Driven
Prior Extraction via Learning
Data-Driven
Machine Learning Model (Implicit System Dynamics)
Model Conversion
SDE Representation (Explicit System Dynamics)
Kalman Filtering based Technique
Efficient Modeling: Data-Driven
Online Inference/Decision: Model-Driven

Fig. 2. The principle of data-induced intelligent Kalman filtering approach.

To train or optimize a learning system or prediction model, we first collect the training samples. Fortunately, the required dataset can be collected online. Then, we choose an appropriate learning model, and train the learning model to obtain an implicit representation of system dynamics. With the implicit representation available, we can already make decisions when faced with new tasks, but at the cost of large complexity. To tackle this issue, we further convert the implicit representation of system dynamics into an explicit one, namely, the SDE that describes system evolution, so as to use the Kalman filter.

During this process, we mainly encounter two key challenging problems. The first one is how to choose an appropriate learning model and, more difficultly, how to train the learning model, which makes the learned prior or model applicable to future/new tasks. The second one is how to convert the implicit representation of system dynamics into an explicit one. Next, we proceed to tackle the two challenging issues.

B. Probabilistic ML Modeling for System Dynamics

In practice, the system dynamics is almost always characterized by a system of SDEs, which can be written as

$$d\boldsymbol{x}(t) = \boldsymbol{f}(\boldsymbol{x}, t)dt + \boldsymbol{\sigma}(t)d\boldsymbol{\xi}(t), \quad \boldsymbol{x}(0) = \mathbf{0}, \tag{4}$$

where $\boldsymbol{x}(t)$ and $\boldsymbol{\xi}(t)$ represent the system state and Brownian motion, respectively. $\boldsymbol{f}(\boldsymbol{x}, t)$ and $\boldsymbol{\xi}(t)$ (modulated by the term $\boldsymbol{\sigma}(t)$ referred to as volatility) characterize the deterministic and stochastic part of system evolution, respectively. It seems that we can obtain the system dynamics via estimating $\boldsymbol{f}(\boldsymbol{x}, t)$ and $\boldsymbol{\sigma}(t)$. However, it is a difficult task, because the data samples required for $\boldsymbol{f}(\boldsymbol{x}, t)$ and $\boldsymbol{\sigma}(t)$ are unavailable.

To tackle the above challenging issue, we first investigate a closely related but simpler SDE, where $\boldsymbol{f}(\boldsymbol{x}, t)$ is absent:

$$d\boldsymbol{y}(t) = \boldsymbol{\sigma}(t)d\boldsymbol{\xi}(t), \quad \boldsymbol{y}(0) = \mathbf{0}. \tag{5}$$

To avoid confusion, we shall explicitly point out the underlying probability space for the above SDEs, which is denoted by $(\Omega, \mathcal{F}, P)$. For SDE (5), we have the following lemma.

Lemma 1. *The solution of SDE* (5) *is a Gaussian process with mean function* $\mathbf{0}$ *and kernel function given by* [1]

$$\mathbb{E}_P\left(\boldsymbol{x}(t)\boldsymbol{x}^T(t')\right) = \int_0^{\min(t,t')} \boldsymbol{\sigma}(\tau)\boldsymbol{\sigma}^T(\tau)d\tau. \tag{6}$$

[1] The notation $\mathbb{E}_P$ is introduced to emphasize that the expectation is taken with respect to probability measure P.

Proof: See Appendix B. ■

The following theorem further establishes the equivalence between the SDEs in (4) and (5).

Theorem 1. *There exists a Borel measurable function $\boldsymbol{F}$ and a probability measure Q (still defined on $(\Omega, \mathcal{F})$) such that under the measure Q, $\boldsymbol{x}(t)$ satisfies the following SDE:*

$$d\boldsymbol{x}(t) = \boldsymbol{\sigma}(t)d\tilde{\boldsymbol{\xi}}(t), \quad \boldsymbol{x}(0) = \boldsymbol{0}, \tag{7}$$

where $\tilde{\boldsymbol{\xi}}(t)$ is a standard Brownian motion under the measure Q. In particular, the Q-law of $\boldsymbol{y}(t)$ is the same as the P-law of $\boldsymbol{x}(t)$, where $\boldsymbol{y}(t)$ is the solution of SDE (5).

Proof: See Appendix C. ■

Because the Q-law of $\boldsymbol{y}(t)$ and the P-law of $\boldsymbol{x}(t)$ coincide, they have the same finite-dimensional distributions. Moreover, for arbitrary bounded functions $\{f_k\}$, we have

$$\begin{aligned}&\mathbb{E}_P\big(f_1(\boldsymbol{x}(t_1))f_2(\boldsymbol{x}(t_2))\cdots f_k(\boldsymbol{x}(t_k))\big)\\ =&\mathbb{E}_Q\big(f_1(\boldsymbol{y}(t_1))f_2(\boldsymbol{y}(t_2))\cdots f_k(\boldsymbol{y}(t_k))\big).\end{aligned} \tag{8}$$

Note that Q can be regarded as a probability measure defined on the latent or transformed space $\boldsymbol{F}(\Omega)$. Theorem 1 implies that conditioned on measure Q and transformed space $\boldsymbol{F}(\Omega)$, we can equivalently characterize the probabilistic behavior of SDE (4) by investigating the properties of SDE (5). In contrast to SDE (4), whose probabilistic characteristics are unknown, the solution of SDE (5) is a tractable Gaussian process.

Remark 3.1 Theorem 1, in fact, constitutes the theoretical foundation of the DIIKF approach. First, as a bridge, it links two different branches of system modeling methodologies. In particular, it provides an efficient ML model or structure for a dynamical system, which avoids heuristic (and even) random selection of the prediction model. Moreover, it guarantees that the ML model does not cause a systematic error.

The discussion above indicates that a Gaussian process that incorporates the deep neural networks (DNNs) can completely characterize the probabilistic behaviors of dynamical system (4). In fact, the measurable function can be approximated by a DNN parameterized by Θ_{T}. For simplicity, we consider only the basic kernel (e.g., the squared exponential kernel) whose parameters are collected into Θ_{B}. If necessary, the kernel can also be parameterized and enhanced by a DNN. The overall prediction model incorporated DNN can be denoted by

$$\boldsymbol{x}(t) \sim \mathcal{GP}(t, t'|\Theta_{\mathrm{T}}, \Theta_{\mathrm{B}}). \tag{9}$$

Next, we derive an efficient learning method for the prediction model in (9) in the data-driven manner.

C. Implicit Dynamics Learning via Multi-task Learning

To elaborate on the key idea, we first shed light on single-task GPR, which mainly consists of two steps. The first one is to choose an appropriate model, which can characterize and encode important prior (e.g., domain knowledge). The prior is updated into a posterior when a dataset is available, which constitutes the second step. The details are as follows. The dataset is denoted by $\boldsymbol{S} = \{(x_1, y_1), \cdots, (x_n, y_n)\}$, with x_i and y_i denoting input and output, respectively [2]. Given $\boldsymbol{S}$, the predictive (or conditional) distribution for an unseen input x_{u} can be calculated explicitly (See (32) - (34)).

In practice, we often have multiple datasets $\mathfrak{D} = \boldsymbol{S}_1, \cdots, \boldsymbol{S}_m$ from multiple related tasks, where $\boldsymbol{S}_i$ corresponds to task i. As an example, for the BPT problem, $\boldsymbol{S}_i$ takes the form

$$\boldsymbol{S}_i = \{(t_1, a_1), (t_2, a_2), \cdots, (t_{n_i}, a_{n_i})\}, \tag{10}$$

where t_i and a_i represent sampling time and beam direction, respectively. For intuitive understanding, an example is shown in Fig. 3, where each $\boldsymbol{S}_i$ corresponds to the beam trajectory of a vehicle. As shown in Fig. 3, the distributions of different datasets are different because of different system parameters (e.g., the vehicle speed). Hence, it is problematic to extend the single-task method directly to the multi-task case, i.e., to train a single and shared model with $\mathfrak{D}$. But note that the datasets really share many features, e.g., a similar trend, which can be exploited to improve system performance, typically, the small sample performance of a learning algorithm [3].

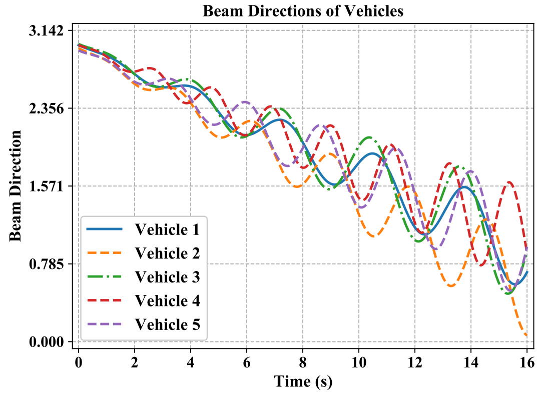

Fig. 3. An illustration of beam direction trajectories of different vehicles.

We tackle these issues from the perspective of hierarchical modeling and stochastic process. Specifically, datasets $\{\boldsymbol{S}_i\}$ are regarded as different realizations of a stochastic process, which is characterized by a set of parameters. Mathematically, each $\boldsymbol{S}_i$ is obtained by discretizing a continuous-time function $x_i(t)$, while $x_i(t)$ is sampled from a stochastic process, e.g., $\mathcal{GP}(t, t'|\Theta_{\mathrm{T}}, \Theta_{\mathrm{B}})$. Let $\Theta = \{\Theta_{\mathrm{T}}, \Theta_{\mathrm{B}}\}$. Then, we shall optimize Θ based on $\mathfrak{D}$. Note that the prior shared across different tasks, which is applicable to future and unseen tasks, should be extracted from $\mathfrak{D}$. To this end, the Bayesian perspective is considered and a prior distribution, denoted by $\mathcal{P}(\Theta)$, is imposed on Θ. Without loss of generality, Θ is assumed to be distributed as $\mathcal{N}(\boldsymbol{0}, \sigma_\Theta^2\mathbf{I})$, i.e., $\mathcal{P} = \mathcal{N}(\boldsymbol{0}, \sigma_\Theta^2\mathbf{I})$. Given $\mathfrak{D}$, it is sufficient to update the prior $\mathcal{P}(\Theta)$ into a posterior $\mathcal{Q}(\Theta)$.

The method to obtain the posterior $\mathcal{Q}(\Theta)$ is closely related to the optimization criterion. Note that our design goal is that the obtained posterior can improve prediction performance on unseen tasks, more specifically, to maximize the generalization performance, e.g., to minimize the transfer-error. An effective posterior $\mathcal{Q}^\star$ has been derived in the following theorem.

[2] For simplicity, we assume x_i and y_i are scalars. But, they can be vectors.

[3] Typically, the size of each dataset $\boldsymbol{S}_i$ is very small. We will encounter this situation in the next section, where long-term prediction is considered.

Theorem 2. *With the aim of minimizing the generalization or transfer error, an efficient posterior is given by*

$$\mathcal{Q}^\star(\Theta) \\ = \frac{\mathcal{P}(\Theta)\exp\left(\left(1+\sum_{i=1}^m n_i^{-1}\right)^{-1}\sum_{i=1}^m \frac{1}{n_i} Z(\boldsymbol{S}_i,\Theta)\right)}{\mathbb{E}_{\Theta\sim\mathcal{P}}\left[\exp\left(\left(1+\sum_{i=1}^m n_i^{-1}\right)^{-1}\sum_{i=1}^m \frac{1}{n_i} Z(\boldsymbol{S}_i,\Theta)\right)\right]},$$

where $Z(\boldsymbol{S}_i,\Theta)$ represents the marginal log-likelihood for $\boldsymbol{S}_i$ and Θ. In particular, $Z(\boldsymbol{S}_i,\Theta)$ for GPR is calculated as [4]

$$Z(\boldsymbol{S}_i,\Theta) = -\frac{1}{2}\mathbf{a}_i^{\mathrm{T}}(\mathbf{C}_i+\sigma_i^2\mathbf{I})^{-1}\mathbf{a}_i - \\ \frac{1}{2}\log\det(\mathbf{C}_i+\sigma_i^2\mathbf{I}) - \frac{n_i}{2}\log 2\pi. \quad (11)$$

Proof: The basic idea to obtain the posterior $\mathcal{Q}^\star$ is to first derive an upper bound of the transfer-error and then minimize the upper bound. See Appendix D for more details. ■

To facilitate the use of Kalman filter and reduce complexity, a point estimate of Θ is preferable. The optimal point estimate is the MAP (maximum a posterior) estimate, i.e.,

$$\Theta^\star = \arg\max_{\Theta} \mathcal{Q}(\Theta). \quad (12)$$

The optimal point estimate $\Theta^\star$ can be equivalently obtained by maximizing the numerator of $\mathcal{Q}^\star(\Theta)$, i.e.,

$$\Theta^\star \\ = \arg\max_{\Theta} \ln\mathcal{P}(\Theta) + \left(1+\sum_{i=1}^m \frac{1}{n_i}\right)^{-1}\sum_{i=1}^m \frac{Z(\boldsymbol{S}_i,\Theta)}{n_i}. \quad (13)$$

The complete procedure to obtain the implicit system dynamics is summarized in Algorithm 1.

```
Algorithm 1: Implicit System Dynamics Modeling
1: input: empirical datasets $\mathfrak{D} = \{\boldsymbol{S}_1, \boldsymbol{S}_2, \cdots, \boldsymbol{S}_m\}$
2: construct and initialize mean function network and
     GP kernel
3: repeat
     (a) sample a small batch of datasets from $\mathfrak{D}$
     (b) compute loss as per (13) with sampled data
     (c) update parameters of both mean and kernel
   until current time-frame is terminated
4: output: $\Theta^\star$ - parameters of system dynamics
```

Once $\Theta^\star$ is available, the implicit system dynamics model has been obtained, based on which an effective Bayesian inference can be made when faced with a new task. However, because it is still based on GPR, it inevitably leads to a large computational complexity. Next, we tackle this issue.

D. SDE Representation and Efficient Inference

To enable efficient inference, explicit system dynamics, i.e., the SDEs that describe system state evolution and measurement are required. To this end, we shall convert the model in (9) with $\Theta^\star$ obtained based on (13) into an SDE representation. Without loss of generality, the fully-connected neural network is chosen to parameterize the transform, and the input-output relationship of the neural network can be written as

$$z = h(t, \Theta_{\mathrm{T}}). \quad (14)$$

Then, the output of the fully-connected DNN is further fed to the basic kernel (e.g., the squared exponential kernel), which is denoted by $k_{\mathrm{B}}(\cdot,\cdot \mid \Theta_{\mathrm{B}})$ with parameters Θ_{B}. Hence, the overall GP prediction model can be expressed as

$$k(t,t') = k_{\mathrm{B}}\big(h(t,\Theta_{\mathrm{T}}), h(t',\Theta_{\mathrm{T}}) \mid \Theta_{\mathrm{B}}\big). \quad (15)$$

In view that many basic kernels are stationary, it is assumed here that the basic kernel $k_{\mathrm{B}}(z,z'|\Theta_{\mathrm{B}})$ is stationary. Then, the basic kernel can be equivalently written as

$$k_{\mathrm{B}}(z,z' \mid \Theta_{\mathrm{B}}) = k_{\mathrm{B}}(z-z' \mid \Theta_{\mathrm{B}}). \quad (16)$$

To obtain the SDE representation, the spectral factorization method can be used [30]. The procedure to convert a stationary kernel into an equivalent SDE representation whose kernel function coincides with the given kernel is as follows:

1) Step 1: The corresponding spectral density $S(\omega)$ can be obtained by computing the Fourier transform of $k_{\mathrm{B}}(\cdot \mid \Theta_{\mathrm{B}})$.

2) Step 2: In many cases, $S(\omega)$ can be written as a rational function taking the form

$$S(\omega) = \frac{1}{p_n(\omega^2)}, \quad (17)$$

where $p_n(\cdot)$ is a polynomial of nth order. Otherwise, we can approximate $S(\omega)$ with such a function, e.g., via the Taylor series expansion or Padé approximate.

3) Step 3: A stable rational transfer function $H(i\omega)$ can be found, which takes the following form

$$H(i\omega) = \frac{\sigma_0}{(i\omega)^n + a_{n-1}(i\omega)^{n-1} + \cdots + a_1(i\omega) + a_0}, \quad (18)$$

and meanwhile, the spectral density $S(\omega)$ can be decomposed as $S(\omega) = \sigma_0^2 H(i\omega)H(-i\omega)$. [5]

With the transfer function $H(j\omega)$ available, we can obtain an SDE of nth order, which can be written as

$$\frac{d^n u(z)}{dz^n} + \cdots + a_1\frac{du(z)}{dz} + a_0 u(z) = \sigma_0\frac{d\xi(z)}{dz},$$

where $\xi(z)$ denotes the standard Brownian motion. Let $\boldsymbol{x}(z) = (u(z), du(z)/dz, \cdots, d^{n-1}u(z)/dz^{n-1})$ and $\mathbf{V} = \mathrm{diag}(0, \cdots, 0, \sigma_0)$. The SDE can be compactly rewritten as

$$d\boldsymbol{x}(z) = \mathbf{C}\boldsymbol{x}(z)dz + \mathbf{V}d\boldsymbol{\xi}(z). \quad (19)$$

where $\boldsymbol{\xi}(z)$ denotes the standard Brownian motion of dimension n and matrix $\mathbf{C}$ is given by

$$\mathbf{C} = \begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & 0 & 1 \\ -a_0 & -a_1 & \cdots & -a_{n-1} \end{pmatrix}.$$

[4] For a one-dimension dataset, $\boldsymbol{S}_i = \{(t_1,a_1),(t_2,a_2),\cdots,(t_{n_i},a_{n_i})\}$, where $a_j = f_i(t_j) + \varepsilon_i$ with $\varepsilon_i \sim \mathcal{N}(0,\sigma_i^2)$. Then, $\mathbf{a}_i$ takes the form $\mathbf{a}_i = (a_1, a_2, \cdots, a_{n_i})^{\mathrm{T}}$ and $\mathbf{C}_i$ is calculated as $[\mathbf{C}_i]_{uv} = k(t_u, t_v)$.

[5] The procedure to find the transfer function is called spectral factorization, which consists of two steps. First, the roots of the denominator are computed. Note that the roots always appear in pairs, where one member of the pair is the complex conjugate of the other one. Then, the denominator polynomial of $H(i\omega)$ can be constructed from the positive-imaginary-part roots only.

We proceed to derive the measurement equation. We define vector $\mathbf{b} = [1, 0, \cdots, 0]^{\mathrm{T}}$, whose dimension coincides with $\boldsymbol{x}(z)$. The measurement equation can be expressed as

$$b(z) = \mathbf{b}^{\mathrm{T}} \boldsymbol{x}(z) + \eta(z), \tag{20}$$

where $\eta(z)$ represents the measurement noise. With (19) and (20) available, the continuous-time Kalman filtering algorithm can be applied, which is, however, computation-unfriendly. It is also problematic to use the discrete Kalman filter based on the simple discretization method, which will explained later. Next, we propose an efficient discretization method.

IV. Low-Frequency Beam Sounding via Long-Term Prediction

In the classical BPT scheme, beam sounding, e.g., the local beam sweeping or training, is performed in each time-slot, as shown in Fig. 4-(1). The high frequency of beam sounding inevitably degrades system performance of interest, e.g., the effective achievable rate. It is well-known that a (stochastic) differential equation can characterize the variation tendency of the underlying physical system. Now, since the SDE that describes the beam variation has been obtained, we can predict its long-term variation behavior. As shown in Fig. 4-(2), low frequency of beam sounding is involved in the long-term BPT scheme, which can further improve system performance.

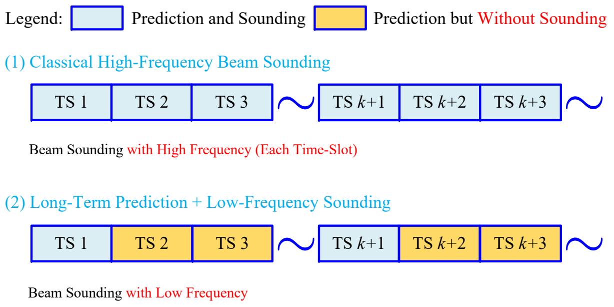
Legend:
Prediction and Sounding
Prediction but Without Sounding
(1) Classical High-Frequency Beam Sounding
TS 1
TS 2
TS 3
TS k+1
TS k+2
TS k+3
Beam Sounding with High Frequency (Each Time-Slot)
(2) Long-Term Prediction + Low-Frequency Sounding
TS 1
TS 2
TS 3
TS k+1
TS k+2
TS k+3
Beam Sounding with Low Frequency

Fig. 4. The comparison between the classical beam sounding scheme and the proposed novel long-time beam sounding scheme.

To employ the discrete Kalman filter, the standard assumption used in literature is that typical system parameters (e.g., velocity) keep constant within each time-slot, based on which the discrete-time model takes the following form

$$\boldsymbol{x}(z_{k+1}) = (\mathbf{I} + \mathbf{C}\Delta z_k)\boldsymbol{x}(z_k) + \mathbf{q}_k, \tag{21}$$

where $\mathbf{q}_k$ denotes the noise random vector. It seems that given an initial condition of $\boldsymbol{x}(z_0)$, the discrete Kalman filter can be invoked. Unfortunately, the discretization format in (21) does not hold true in general for our problem concerned. In fact, due to the nonlinear transform $z = h(t, \Theta_{\mathrm{T}})$, Δz_k is non-uniform and can be very large even for a fixed Δt_k. As a result, the conventional discretization results in a large error.

To tackle this issue, we propose an equivalent discretization for the SDE in (19) by investigating the statistics (e.g., the differential equations that characterize the mean and variance variations) of the corresponding stochastic process. Note that the equivalence means that the probability distributions of the continuous-time SDE and the discrete-time stochastic system coincide at sampling points $\{z_k\}$. The equivalent discretization is formally stated in the following theorem.

Theorem 3. *The SDE in* (19) *is in distribution equivalent to the following discrete-time stochastic system*

$$\boldsymbol{x}(z_{k+1}) = \mathbf{T}_k \boldsymbol{x}(z_k) + \mathbf{q}_k, \qquad \mathbf{q}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{\Pi}_k), \tag{22}$$

where $\mathbf{T}_k$ *and* $\mathbf{\Pi}_k$ *are respectively given by*

$$\begin{aligned}\mathbf{T}_k &= \exp(\mathbf{C}(z_{k+1} - z_k))\\ \mathbf{\Pi}_k &= \int_{z_k}^{z_{k+1}} \exp(\mathbf{C}(z_{k+1} - s))\mathbf{V}\mathbf{V}^{\mathrm{T}} \exp(\mathbf{C}(z_{k+1} - s))^{\mathrm{T}} ds.\end{aligned}$$

Proof: The explicit solution to the SDE in (19), which is a stochastic process (i.e., Gaussian process), is given by

$$\boldsymbol{x}(z) = \exp(\mathbf{C}(z - z_0))\boldsymbol{x}(z_0) + \int_{z_0}^{z} \exp(\mathbf{C}(z - s))\mathbf{V} d\boldsymbol{\xi}(s),$$

where $\exp(\cdot)$ denotes the matrix exponential function. It can be verified that the mean and covariance functions, denoted respectively by $\boldsymbol{m}(z)$ and $\boldsymbol{P}(z)$, can be calculated as

$$\boldsymbol{m}(z) = \exp(\mathbf{C}(z - z_0))\boldsymbol{m}(z_0) \tag{23}$$

$$\begin{aligned}\boldsymbol{P}(z) =& \exp(\mathbf{C}(z - z_0))\boldsymbol{P}(z_0) \exp(\mathbf{C}(z - z_0))^{\mathrm{T}} +\\ & \int_{z_0}^{z} \exp(\mathbf{C}(z - s))\mathbf{V}\mathbf{V}^{\mathrm{T}} \exp(\mathbf{C}(z - s))^{\mathrm{T}} ds.\end{aligned} \tag{24}$$

Let z_k denote the current "time". The initial condition meets $\boldsymbol{m}(z_k) = \boldsymbol{x}(z_k)$ and $\boldsymbol{P}(z_k) = \mathbf{0}$. The transition density (from z_k to z), which is a Gaussian distribution, is given by

$$p(\boldsymbol{x}(z)|\boldsymbol{x}(z_k)) = \mathcal{N}(\boldsymbol{m}(z|z_k), \boldsymbol{P}(z|z_k)), \tag{25}$$

where $\boldsymbol{m}(z|z_k)$ and $\boldsymbol{P}(z|z_k)$ are respectively given by

$$\begin{aligned}\boldsymbol{m}(z|z_k) &= \exp(\mathbf{C}(z - z_k))\boldsymbol{x}(z_k)\\ \boldsymbol{P}(z|z_k) &= \int_{z_k}^{z} \exp(\mathbf{C}(z - s))\mathbf{V}\mathbf{V}^{\mathrm{T}} \exp(\mathbf{C}(z - s))^{\mathrm{T}} ds.\end{aligned}$$

Hence, the equivalent discrete-time stochastic system takes the form in (22), which completes the proof. ■

To invoke the discrete Kalman filter, we still need an initial distribution for $\boldsymbol{x}(t_0)$, which is assumed to be Gaussian. To find an appropriate distribution, we assume the system starts from $t = -\infty$ and it is stable at $t = 0$. Hence, an appropriate choice is the stable distribution of SDE (19). Equs (23) and (24) satisfy the following ordinary differential equations

$$\begin{aligned}d\boldsymbol{m}(z)/dz =& \mathbf{C}\boldsymbol{m}(z)\\ d\boldsymbol{P}(z)/dz =& \mathbf{C}\boldsymbol{P}(z) + \boldsymbol{P}(z)\mathbf{C}^{\mathrm{T}} + \mathbf{V}\mathbf{V}^{\mathrm{T}}.\end{aligned}$$

By letting $d\boldsymbol{m}(z)/dz = \mathbf{0}$ and $d\boldsymbol{P}(z)/dz = \mathbf{0}$, we can obtain the steady-state solution $\mathcal{N}(\mathbf{0}, \mathbf{P}_0)$, where $\mathbf{P}_0$ is calculated as follows. Let $\mathrm{vec}(\cdot)$ denote the vectorization operator. By vectoring both sides of matrix equation $\mathbf{C}\mathbf{P}_0 + \mathbf{P}_0\mathbf{C}^{\mathrm{T}} + \mathbf{V}\mathbf{V}^{\mathrm{T}} = \mathbf{0}$, we can obtain $(\mathbf{C} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{C})\mathrm{vec}(\mathbf{P}_0) = -\mathrm{vec}(\mathbf{V}\mathbf{V}^{\mathrm{T}})$. If $\mathbf{C} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{C}$ is invertible, $\mathrm{vec}(\mathbf{P}_0)$ is calculated as

$$\mathrm{vec}(\mathbf{P}_0) = -(\mathbf{C} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{C})^{-1}\mathrm{vec}(\mathbf{V}\mathbf{V}^{\mathrm{T}}). \tag{26}$$

In contrast to the conventional BPT scheme, where Δt_k is a constant and is almost always equal to T_{S} (i.e., $\Delta t_k = T_{\mathrm{S}}$), in the long-term BPT scheme, $\Delta t_k = t_{k+1} - t_k \gg T_{\mathrm{S}}$ often spans across multiple time-slots. Hence, we need the predictive distribution of beam direction $y(t)$ at any time t. Without loss of generality, we focus on an arbitrary but fixed time t_k and calculate the distribution of beam $b(t)$ for $t > t_k$.

Corollary 1. *The predictive distribution of $b(t)$ is given by*

$$b(t) \sim \mathcal{N}(m(t), a(t)), \quad (t > t_k), \tag{27}$$

where $m(t) = \mathbf{b}^{\mathrm{T}}\mathbf{T}(h(t_k), h(t))\boldsymbol{x}(h(t_k))$ and $a(t) = \sigma_m^2 + \mathbf{b}^{\mathrm{T}}\mathbf{\Pi}(h(t_k), h(t))\mathbf{b}$, with $\mathbf{T}(z_k, z)$ and $\mathbf{\Pi}(z_k, z)$ given by

$$\begin{aligned}\mathbf{T}(z_k, z) &= \exp(\mathbf{C}(z - z_k))\\ \mathbf{\Pi}(z_k, z) &= \int_{z_k}^{z} \exp(\mathbf{C}(z-s))\mathbf{V}\mathbf{V}^{\mathrm{T}} \exp(\mathbf{C}(z-s))^{\mathrm{T}} ds.\end{aligned}$$

σ_m^2 denotes the variance of the noise caused by $\eta(t)$ in (20).

Proof: Note that since $b(t)$ is a linear function of $\boldsymbol{x}(z)$, we can obtain (27) from Theorem 3 immediately. ■

It is observed that $a(t)$ is monotonously increasing in terms of t, which indicates that the predicted beam becomes more inaccurate as t increases. Apparently, this coincides with our intuition. If $a(t)$ is greater than a predefined threshold value, there are two methods to handle this situation. The first is to perform beam sounding to find the optimal beam, e.g., via the hierarchical search [7] or adaptive local search [19]. The second is beam width optimization, i.e., to adjust the beam width in real time, which will be discussed next.

In general, wide beams incur low array gains. We introduce a function $g(w)$, which characterizes the array gain of a beam with width w. We further introduce an adjustable parameter $c > 0$ to control the beam width. The main-lobe of the beam to be chosen is denoted by $\mathcal{I}$, which is given by

$$\mathcal{I} = (m(t) - c\sqrt{a(t)}, m(t) + c\sqrt{a(t)}). \tag{28}$$

Under some mild assumptions (similar to [31]), the posterior effective achievable rate can be expressed as

$$\begin{aligned}R(c) &= \int_{\mathcal{I}} \frac{e^{-(x-m(t))^2/(2a(t))}}{\sqrt{2\pi a(t)}} \log\left(1 + pHg(2c\sqrt{a(t)})\right) dx\\ &= \int_{-c}^{c} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \log\left(1 + pHg(2c\sqrt{a(t)})\right) dx, \end{aligned} \tag{29}$$

where p denotes the transmit power and H is determined by the channel condition concerned, e.g., the average path gain. Since $R(c)$ is a unimodal function and has a unique maximal point [31], it can be efficiently found via classical derivative-free search methods, e.g., the gold-search method.

For clarity, the complete DIIKF approach is summarized in Algorithm 2. With the historical datasets $\mathfrak{D}$ available, a ML model is first constructed in Step 2 as per Theorem 1, based on which the implicit system dynamics can be learned in Step 3. To facilitate efficient inference, the implicit system dynamics model is converted to the SDE representation in Step 4 and meanwhile the initial distribution is calculated as per (26) in Step 5. Then, for a new task, the discrete Kalman filter can be applied to predict the beams in Step 6.

Algorithm 2: Data-Induced Intelligent Kalman Algorithm

```
1: input: historical dataset $\mathfrak{D} = \{\boldsymbol{S}_1, \cdots, \boldsymbol{S}_m\}$
2: construct machine learning model (as per Theorem 1)
3: learn implicit system dynamics (invoke Algorithm 1)
4: convert implicit model to explicit SDE representation
5: compute initial or prior distribution according to (26)
6: repeat for each time-slot (for a new task)
     (a) construct equivalent discrete-time stochastic
         system (according to Theorem 3)
     (b) predict beam by invoking discrete Kalman filter
   until current task (e.g., time-frame) is terminated
```

V. Numerical Results

In this section, simulation results are provided to demonstrate the performance of the proposed algorithms. For clarity, we first describe the experiment environment [24], [26].

A. Simulation Environment

Without loss of generality, the uniform linear array is chosen in this section. Two cases of antenna array, i.e., $N = 64$ and $N = 128$, are chosen to evaluate different algorithms. The size of codebook $\mathcal{C}$ satisfies $M = N$ (with estimation accuracy $1/N$). For all experiments, the channel model in (1) includes one LOS path and three NLOS paths. The AoDs of the NLOS paths are distributed uniformly in $[0, 2\pi)$. The average power ratio of the LOS path gain α_{L} and each NLOS path gain α_{N} is 10dB. The path gain of a NLOS path is distributed as $\mathcal{CN}(0, \sigma_{\mathrm{N}}^2)$, where σ_{N}^2 is calculated as per the path gain of the LOS path. Let a uniform distribution taking values in $[a, b]$ be denoted by $\mathrm{U}(a, b)$. The simulation environment and relevant system parameters are described below.

Without loss of generality, we focus on the V2X communication scenario. But compared to the straight road [12], a more complex road condition, i.e., the typical flyover, chosen from the real environment, is considered here. As shown in Fig. 5, three roads, along with the lane directions, are chosen to evaluated different algorithms. The length of each time-slot, i.e., transmission time interval, is set to 20 milliseconds. The widths of Road 1 and Road 2 are both 12m, while the width of Road 3 is 20m. Moreover, it is allowed to change lane in Road 3, with probability 1/10. The initial velocity, i.e., the velocity when the vehicle enters the coverage area, for Road 1 and Road 2 is distributed uniformly as $\mathrm{U}(54, 80)$ (km/h), while the initial velocity for Road 3 is distributed uniformly as $\mathrm{U}(54, 108)$ (km/h). To simulate the scenario more practical, with probability 0.1, the driver speeds up (or slows down) the velocity to the maximal (or minimal) speed in each road.

To confirm the effectiveness of our approach, the state-of-the-art benchmarks are chosen to evaluate our BPT algorithms, including the conventional hierarchical search (HS) method, two learning-based solutions proposed recently (i.e., the s-tochastic bandit learning (SBL) based algorithm [19] and the classical GPR based (GPR-Only) algorithm [28] [6]) and the

[6] Compared to Algorithm 1 proposed in this paper (named as GPM-GPI in this section), the classical GPR method fails to learn the system dynamics.

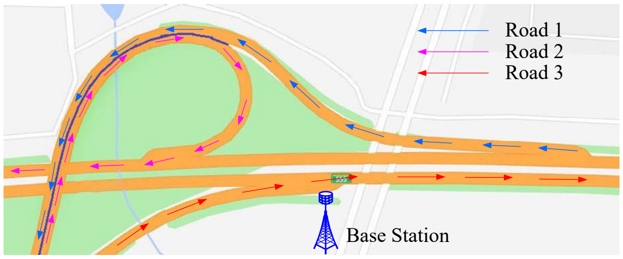

Fig. 5. The road condition of typical flyover chosen from real environment.

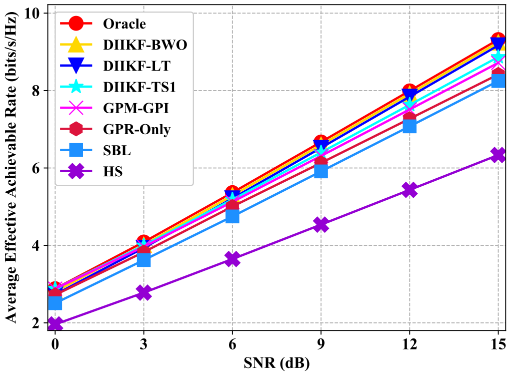

Fig. 6. The average EAR performance of different algorithms: $N = 128$ and Road 1.

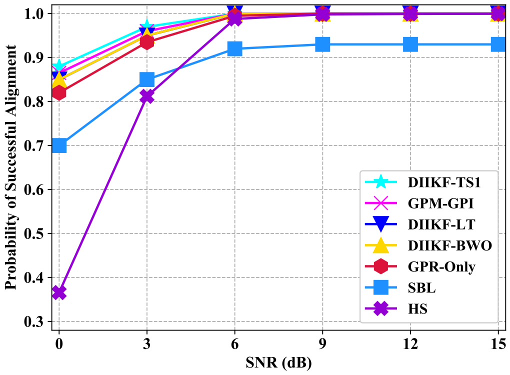

Fig. 7. The PSA performance of different algorithms: $N = 128$ and Road 1.

oracle aided BPT algorithm. [7] To demonstrate the advantages of DIIKF, the following algorithms are evaluated:

- GPM-GPI (i.e., Gaussian process modeling and Gaussian process inference): The implicit system dynamics learning based on Algorithm 1 is incorporated into the BPT solution. But GPR is still used to perform prediction.
- DIIKF-TS1: The BPT solution is designed based on the DIIKF approach, but without the long-term prediction and beam width optimization techniques. As a result, the beam sounding is still operated in each time-slot.
- DIIKF-LT: Compared to the DIIKF-TS1 algorithm, the long-term prediction technique is incorporated. But the beam width optimization is not considered.
- DIIKF-BWO: In contrast to DIIKF-LT, where the beam sounding is executed if necessary (e.g., wide beam confidence interval), DIIKF-BWO adjusts the beam width.

For comparison, probability of successful alignment (PSA), average effective achievable rate (EAR) and running time are chosen as the performance metrics to evaluate different BPT algorithms. In view of the definition of EAR, i.e., $R_{\mathrm{EAR}} = (1-T_{\mathrm{B}}/T_{\mathrm{S}}) \log\left(1+P|\mathbf{h}^{\mathrm{H}}\mathbf{f}_i|^2\right)$, where $P|\mathbf{h}^{\mathrm{H}}\mathbf{f}_i|^2$ for different algorithms (except for DIIKF-BWO) is almost the same, EAR also characterizes equivalently the performance in terms of beam training overhead. The path gain of the LOS path is assumed to be distributed as $\alpha_{\mathrm{LOS}} \sim \mathcal{CN}(\bar{m}, 0.1)$ with $|\bar{m}| = 1$.

B. Performance of Beam Prediction and Tracking

First, we evaluate the average EAR performance of different BPT algorithms, as shown in Fig. 6. It is observed that the algorithms proposed in this paper (i.e., GPM-GPI, DIIKF-TS1, DIIKF-LT and DIIKF-BWO) outperform the benchmarks (i.e., SBL, GPR-Only and HS). The reason for this is that the underlying environment system dynamics can be efficiently characterized by the ML model constructed in this paper and learned by the implicit system dynamics learning method. As a result, the beam training overhead required is very small. In particular, the BPT algorithms incorporating the long-term prediction scheme approach the ideal oracle-aided algorithm, thanks to the negligible beam training overhead.

Fig. 7 shows the PSA performance of different algorithms. It can be also observed that our proposed algorithms achieve better performance than both SBL and HS. In particular, our BPT algorithms outperform SBL, which is also a learning-based algorithm. The reason for this is that the beam index difference technique involved in SBL predicts the difference of successive beams, which therefore fails to capture and exploit the variation tendency of beam direction function. In contrast, our algorithms explicitly and fully exploit the prior information to narrow the beam interval containing the real beam. Due to the noise effect within beam sounding, if less beams are swept, better PSA performance can be achieved [19]. Since the conventional GPR method fails to learn the system dynamics, its performance is worse than the DIIKF-based solutions. It can be seen from the two figures that the performance achieved by DIIKF-TS1 is a bit better than that achieved by GPM-GPI. The reason for this is that the Kalman filtering technique also applies to the non-stationary environments.

To intuitively demonstrate the advantage of the long-term prediction scheme, the PSA performance with respect to the period of beam sounding is shown in Fig. 8. It can be seen that compared to the conventional beam sounding scheme whose sounding period is one, the long-term prediction based beam prediction algorithms can achieve a high prediction accuracy even for a relatively large beam sounding period. The reason for this is that the underlying system dynamics (e.g., the beam variation tendency) can be efficiently learned from the multiple beam trajectories. When facing larger uncertainties, the beam width optimization based method can adjust beam

[7] The direct search algorithm searches the optimal beam within the codebook exhaustively. The oracle aided algorithm is served as a benchmark, which can always find the optimal beam with training overhead zero.

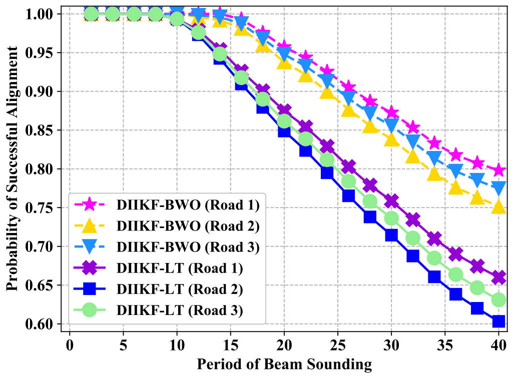

Fig. 8. The PSA performance varying with the beam sounding period for DIIKF-LT and DIIKF-BWO: $N = 64$ and Road 1.

width adaptively and dynamically, it is not surprising that it can also achieve better performance.

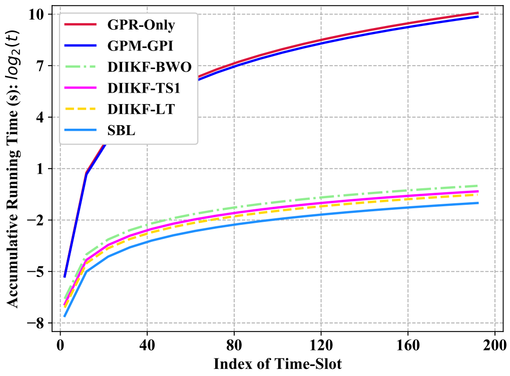

Fig. 9. The average accumulative run-time of different algorithms: $N = 128$ and Road 1.

Another appealing advantage of the DIIKF approach is that it inherits the property of low computational complexity from the classical Kalman filtering technique. In particular, thanks to the linear complexity order, the performance curves in terms of accumulative run-time of DIIKF-TS1, DIIKF-LT and DIIKF-BWO increase very slow, as shown in Fig. 9. In contrast, due to the cubic complexity order, the time-resource and computing resource required by both GPR-Only and GPM-GPI increase remarkably. Moreover, thanks to the inherent Kalman filtering characteristics, DIIKF-TS1, DIIKF-LT and DIIKF-BWO can well adapt to non-stationary environments, which accounts for better performance than GPR-Only and GPM-GPI.

In contrast to the other machine learning applications, e.g., computer vision or natural language processing, the wireless environments fluctuate rapidly, which easily invalidates many machine learning based algorithms. Hence, good small sample performance is desired in wireless communications. Fortunately, the good small sample performance can be achieved by our algorithms, as shown in Fig. 10. It is observed that although the size of available training curves is small, e.g., $m \leq 4$, our algorithms can still obtain satisfactory EAR performance. The reason for this is two-fold. First, the Bayesian methodology is adopted in our algorithms, which has already been shown to help achieve good small performance. More importantly, we explicitly minimize the transfer error when deriving the loss function, which improves the generalization performance.

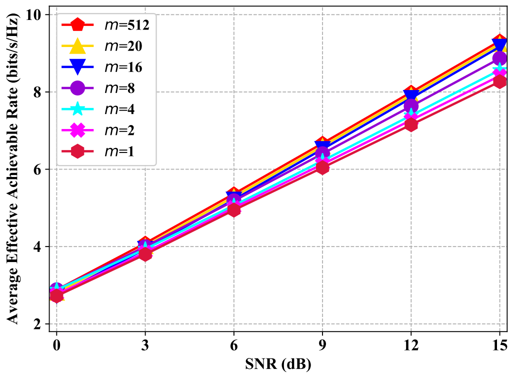

Fig. 10. The average EAR performance achieved by DIIKF-TS1 models trained with varying amounts of beam trajectories: $N = 128$ and Road 1.

VI. Conclusion

To exploit the advantages of both GPR and Kalman filter while overcome their drawbacks, in this paper we proposed a novel hybrid model and data driven approach in this paper. In particular, the system dynamics required can be obtained via the data-driven manner. Since the system dynamics was available, we further studied long-term prediction and proposed a more efficient BPT algorithm along with adaptive beam width optimization, so as to further enhance system performance. Typical advantages of our approach include low computational complexity (due to inherent Kalman filter) and significantly improved system performance (thanks to long-term prediction and beam width optimization). Simulation results confirmed the effectiveness and superiority of our proposal.

Appendix A
Gaussian Process Regression

A stochastic process $f(x)$ is a Gaussian process (GP) if and only if for any finite number of points $x_1, \cdots, x_n$, the joint probability density function $p(f(x_1), \cdots, f(x_n))$ is Gaussian [32]. A GP is completely characterized by the mean function $m(x)$ and covariance function $k(x, x')$, which are similar to the mean and covariance for a Gaussian vector. The mean and covariance functions are respectively defined by

$$\begin{aligned} m(x) =&\mathbb{E}[f(x)], \\ k(x, x') =&\mathbb{E}\big[\big(f(x) - m(x)\big)\big(f(x') - m(x')\big)\big]. \end{aligned} \tag{30}$$

The mean function is assumed to be zero next, i.e., $m(x) = 0$.

GPR is to predict or infer $f(x_{\mathrm{u}})$ for an unseen x_{u} based on a set of observations $\boldsymbol{S} = \{(x_i, y_i) \,|\, y_i = f(x_i) + w_i, w_i \sim \mathcal{N}(0, \sigma^2), i = 1, \cdots, n\}$, where x_i and y_i denote input and output, respectively. In contrast to many parametric regression methods, GPR is based on Bayesian inference, which generates a probability distribution, rather than only a point estimate for the quantity of interest. Given $\boldsymbol{S}$ above, we next derive the conditional or predictive distribution for $f(x_{\mathrm{u}})$ at x_{u} [33].

The observed points are stacked into $\mathbf{y}_\text{o} = (y_1, \cdots, y_n)^\text{T}$. Based on the Gaussian assumption, the joint probability distribution between $\mathbf{y}_\text{o}$ and $y_\text{u} = f(x_\text{u}) + w$ is given by

$$\begin{bmatrix}\mathbf{y}_\text{o}\\ y_\text{u}\end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix}\mathbf{C}_\text{oo}+\sigma^2\mathbf{I} & \mathbf{c}_\text{uo}\\ \mathbf{c}_\text{uo}^\text{T} & c_\text{uu}\end{pmatrix}\right), \tag{31}$$

where the matrix and vector are formed as $[\mathbf{C}_\text{oo}]_{ij} = k(x_i, x_j)$, $[\mathbf{c}_\text{uo}]_j = k(x_\text{u}, x_j)$ and $c_\text{uu} = k(x_\text{u}, x_\text{u})$. Then, the conditional distribution of $f_\text{u} = f(x_\text{u})$ at x_u is given by

$$p(f_\text{u}|\boldsymbol{S}, x_\text{u}) \sim \mathcal{N}\big(\mu(x_\text{u}), c(x_\text{u})\big) \tag{32}$$

$$\mu(x_\text{u}) = \mathbf{c}_\text{uo}^\text{T}(\mathbf{C}_\text{oo} + \sigma^2\mathbf{I})^{-1}\mathbf{y}_\text{o} \tag{33}$$

$$c(x_\text{u}) = c_\text{uu} - \mathbf{c}_\text{uo}^\text{T}(\mathbf{C}_\text{oo} + \sigma^2\mathbf{I})^{-1}\mathbf{c}_\text{uo}. \tag{34}$$

It is referred to [33] for more details about GP and GPR.

Appendix B
Proof of Theorem 1

According to the definition of Itô-integral and the Fubini's theorem, we can assert that $\boldsymbol{x}(t)$ is a Gaussian process with mean function is $\mathbf{0}$. For simplicity, we consider only the one-dimension case, and the high-dimension case can be proved similarly. Without loss of generality, we assume $t < t'$ holds. The covariance function can be calculated as

$$\begin{aligned}\mathbb{E}\big(x(t)x(t')\big) =&\mathbb{E}\Big(x(t)\big(x(t') - x(t) + x(t)\big)\Big)\\ =&\mathbb{E}\Big(x(t)\big(x(t') - x(t)\big)\Big) + \mathbb{E}(x^2(t))\\ \overset{(*)}{=}&\mathbb{E}(x^2(t)).\end{aligned}$$

where $(*)$ is due to the fact that random variables $x(t') - x(t)$ and $x(t)$ are Gaussian and also independent.

As per the isometry property of Itô-integral, the covariance function $\mathbb{E}\big(x(t)x(t')\big)$ can be calculated as

$$\begin{aligned}\mathbb{E}(x^2(t)) =&\mathbb{E}\left(\int_0^t \sigma(s)d\xi(s)\right)^2 = \mathbb{E}\left(\int_0^t \sigma^2(s)ds\right)\\ =&\int_0^t \sigma^2(s)ds = \int_0^{\min(t,t')} \sigma^2(s)ds,\end{aligned}$$

which completes the proof.

Appendix C
Proof of Theorem 1

Without loss of generality, we consider only the scalar case, and then the SDE can be simplified as $dx(t) = f(x,t)dt + \sigma(t)d\xi(t)$. The proof consists of three steps.

1) Step 1: We first prove that there exists another probability space $(\Omega, \mathcal{F}, Q)$ such that Q is a Gaussian measure. In fact, the probability measure desired can be constructed explicitly. By defining $\theta_t = f(x,t)/\sigma(t)$ and $\tilde{\xi}(t) = \xi(t) + \int_0^t \theta_s ds$, the desired probability measure can be, in fact, constructed as $Q(\mathcal{E}) = \mathbb{E}(M_T 1_\mathcal{E})$, $(\forall \mathcal{E} \in \mathcal{F})$, where M_t is given by

$$\begin{aligned}M_t =& \exp\left(-\int_0^t \theta_s d\tilde{\xi}_s + \frac{1}{2}\int_0^t \theta_s^2 ds\right)\\ =& \exp\left(-\int_0^t \theta_s d\xi_s - \frac{1}{2}\int_0^t \theta_s^2 ds\right).\end{aligned} \tag{35}$$

Because $Q(\Omega) = \mathbb{E}(M_T 1_\Omega) = \mathbb{E}(M_T) = 1$ and $M_t > 0$ hold, it can be verified that Q is a probability measure.

2) Step 2: We next prove that Q is Gaussian. It is sufficient to show that for arbitrary $n \in \mathcal{N}$, constants λ_j and $(t_j, j \le n)$ partition of $[0, T]$ with $t_n = T$, the following equation holds

$$\tilde{\mathbb{E}}\bigg[\exp\bigg(\sum_{j=0}^{n-1}\lambda_j(\tilde{\xi}_{t_{j+1}} - \tilde{\xi}_{t_j})\bigg)\bigg] = \exp\bigg(\sum_{j=0}^{n-1}\frac{\lambda_j^2}{2}(t_{j+1} - t_j)\bigg), \tag{36}$$

where $\tilde{\mathbb{E}}$ is taken with respect to P. Note that (36) means that the increments are the ones of standard Brownian motion.

Let $(\mathcal{F}_{t_j}, j \le n)$ be the Brownian filtration at the time of the partition. The basic idea to prove (36) is by successively conditioning from t_{n-1} down to t_1. As a first step, we have

$$\begin{aligned}&\tilde{\mathbb{E}}\left[e^{\sum_{j=0}^{n-1}\lambda_j(\tilde{\xi}_{t_{j+1}} - \tilde{\xi}_{t_j})}\right]\\ =&\mathbb{E}\left[\mathbb{E}\left[e^{\sum_{j=0}^{n-1}\lambda_j(\tilde{\xi}_{t_{j+1}} - \tilde{\xi}_{t_j})}\middle|\mathcal{F}_{t_{n-1}}\right]\right]\\ \overset{(*)}{=}&\mathbb{E}\Bigg[M_{t_{n-1}}\exp\bigg(\sum_{j=0}^{n-2}\lambda_j(\tilde{\xi}_{t_{j+1}} - \tilde{\xi}_{t_j})\bigg)\cdot\\ &\quad\mathbb{E}\left[e^{\int_{t_{n-1}}^{t_n}(-\theta_s+\lambda_{n-1})d\tilde{\xi}_s + 1/2\int_{t_{n-1}}^{t_n}\theta_s^2 ds}\middle|\mathcal{F}_{t_{n-1}}\right]\Bigg].\end{aligned} \tag{37}$$

where $(*)$ is due to the definition of M_{t_n}, i.e., $M_{t_n} = M_{t_{n-1}} \cdot \exp\big(\int_{t_{n-1}}^{t_n}(-\theta_s + \lambda_{n-1})d\tilde{\xi}_s + \frac{1}{2}\int_{t_{n-1}}^{t_n}\theta_s^2 ds\big)$, and the fact that $M_{t_{n-1}}$ is $\mathcal{F}_{t_{n-1}}$-measurable.

Since the integral $\int_{t_{n-1}}^{t_n}(-\theta_s + \lambda_{n-1})d\tilde{\xi}_s$ only depends on the increment $\tilde{\xi}_{t_n} - \tilde{\xi}_{t_{n-1}}$, it is independent of $\mathcal{F}_{t_{n-1}}$. In light of $d\tilde{\xi}_t = d\xi_t + \theta_t d_t$, we can obtain

$$\begin{aligned}&\mathbb{E}\left[e^{\int_{t_{n-1}}^{t_n}(-\theta_s+\lambda_{n-1})d\tilde{\xi}_s + \frac{1}{2}\int_{t_{n-1}}^{t_n}\theta_s^2 ds}\middle|\mathcal{F}_{t_{n-1}}\right]\\ =&\mathbb{E}\left[e^{\int_{t_{n-1}}^{t_n}(-\theta_s+\lambda_{n-1})d\xi_s + \int_{t_{n-1}}^{t_n}(-\theta_s^2/2+\theta_s\lambda_{n-1})ds}\middle|\mathcal{F}_{t_{n-1}}\right]\\ \overset{(*)}{=}&\exp\left(\frac{1}{2}\int_{t_{n-1}}^{t_n}\big((-\theta_s + \lambda_{n-1})^2 + (-\theta_s^2/2 + \theta_s\lambda_{n-1})\big)ds\right.\\ =&\exp\Big(\lambda_{n-1}^2(t_n - t_{n-1})/2\Big).\end{aligned}$$

where $(*)$ is owing to the fact that $\int_{t_{n-1}}^{t_n}(-\theta_s + \lambda_{n-1})d\xi_s$ is Gaussian (with mean 0 and variance $\int_{t_{n-1}}^{t_n}(-\theta_s + \lambda_{n-1})^2 ds$, according to the properties of Itô integral).

Putting this back into Equ.(37), we have shown that

$$\begin{aligned}&\tilde{\mathbb{E}}\left[e^{\sum_{j=0}^{n-1}\lambda_j(\tilde{\xi}_{t_{j+1}} - \tilde{\xi}_{t_j})}\right]\\ =&\exp\left(\frac{1}{2}\lambda_{n-1}^2(t_n - t_{n-1})\right)\mathbb{E}\left[M_{t_{n-1}}e^{\sum_{j=0}^{n-2}\lambda_j(\tilde{\xi}_{t_{j+1}} - \tilde{\xi}_{t_j})}\right]\\ \overset{(*)}{=}&\exp\left(\frac{1}{2}\lambda_{n-1}^2(t_n - t_{n-1})\right)\tilde{\mathbb{E}}\left[e^{\sum_{j=0}^{n-2}\lambda_j(\tilde{\xi}_{t_{j+1}} - \tilde{\xi}_{t_j})}\right],\end{aligned}$$

where $(*)$ is due to the martingale property. By conditioning from $\mathcal{F}_{t_{n-2}}$ down to $\mathcal{F}_{t_1}$ and repeating the above procedure, we can prove (36), which shows that Q is Gaussian.

3) Step 3: Under the assumption that σ has no zero points, the SDE $dx(t) = f(x,t)dt + \sigma(t)d\xi(t)$ can be rewritten as

$$dx(t) = \sigma(t)\left(d\xi(t) + \frac{f(x,t)}{\sigma(t)}dt\right) = \sigma(t)d\tilde{\xi}(t). \tag{38}$$

The solution of SDE in (38) is given by $x(t)=\int_0^t \sigma(s)d\tilde{\xi}(s)$. Because the SDEs in (5) and (7) take the same form, their solutions are, in fact, two (maybe different) solutions of the same SDE. Hence, we can conclude that the probability laws of the solutions of the two SDEs coincide, according to the weak uniqueness property of SDE, although they are defined on different probability spaces. The proof is completed.

Appendix D
Proof of Theorem 2

A. Preliminaries

A learning task is described by an unknown data distribution $\mathcal{D}$ over a domain $\mathcal{Z}$, from which n points $\boldsymbol{S}=\{\boldsymbol{z}_i \,|\, \boldsymbol{z}_i \sim \mathcal{D}, i=1,\cdots,n\}$ are sampled. Let $\boldsymbol{S}\sim\mathcal{D}^n$ represent the i.i.d. (independently and identically distributed) sampling of n samples. In supervised learning, each $\boldsymbol{z}_i$ takes the form $\boldsymbol{z}_i=(\boldsymbol{x}_i, y_i)$, where $\boldsymbol{x}_i\in\mathcal{X}$ and $y_i\in\mathcal{Y}$ denote input feature and target label, respectively. Given $\boldsymbol{S}$, the design goal is to find a function $h:\mathcal{X}\to\mathcal{Y}$ from a function space $\mathcal{H}$ which can make a good prediction on an unseen input feature $\boldsymbol{x}^*$. The quality of the prediction is measured by a loss function $L:\mathcal{H}\times\mathcal{Z}\to\mathbb{R}$. The design goal is to minimize the expected error, i.e.,

$$\mathcal{L}(h,\mathcal{D})=\mathbb{E}_{\boldsymbol{z}^*\sim\mathcal{D}}L(h,\boldsymbol{z}^*). \tag{39}$$

The empirical counterpart of $\mathcal{L}(h,\mathcal{D})$ is defined by

$$\hat{\mathcal{L}}(h,\boldsymbol{S})=\frac{1}{m}\sum_{i=1}^{m}L(h,\boldsymbol{z}_i). \tag{40}$$

Instead of deterministic predictors, randomized predictors, i.e., the probability measures on the hypothesis space $\mathcal{H}$, are considered. Let $\mathcal{M}(\mathcal{H})$ represent a set of probability measures over $\mathcal{H}$. Two probability measures, i.e., the prior $P\in\mathcal{M}(\mathcal{H})$ and the posterior $V\in\mathcal{M}(\mathcal{H})$, are particularly important. P and V also denote their probability densities. The Gibbs error characterizes the performance of a randomized predictor

$$\mathcal{L}(V,\mathcal{D})=\mathbb{E}_{h\sim V}\mathcal{L}(h,\mathcal{D}). \tag{41}$$

The empirical counterpart of $\mathcal{L}(V,\mathcal{D})$ is defined by

$$\hat{\mathcal{L}}(V,\boldsymbol{S})=\mathbb{E}_{h\sim V}\hat{\mathcal{L}}(h,\boldsymbol{S}). \tag{42}$$

The PAC-Bayesian learning theory bounds the unknown generalization error (GE) $\mathcal{L}(V,\mathcal{D})$ based on the empirical error (EE) $\hat{\mathcal{L}}(V,\boldsymbol{S})$. Specifically, given a prior distribution $P\in\mathcal{M}(\mathcal{H})$, a confidence level $\delta\in(0,1)$, and a positive real number $\beta>0$, with probability at least $1-\delta$ over data samples $\boldsymbol{S}\sim\mathcal{D}^n$, $\forall\, V\in\mathcal{M}(\mathcal{H})$ the following inequality holds true [34]:

$$\begin{aligned}\mathcal{L}(V,\mathcal{D})\le\hat{\mathcal{L}}(V,\boldsymbol{S})+&\beta^{-1}\big(D_{\mathrm{KL}}(V||P)\\ &-\ln\delta+\Psi(\beta,n)\big),\end{aligned} \tag{43}$$

where $D_{\mathrm{KL}}(\cdot||\cdot)$ denotes the Kullback-Leibler (KL) divergence, and $\Psi(\beta,n)$ is constant and depends on P and $\mathcal{D}$.

The posterior $V^\star$ that yields the lowest GE is of particular importance. To obtain the optimal posterior $V^\star$, it is natural to minimize the bound $\hat{\mathcal{L}}(V,\boldsymbol{S})+\beta^{-1}D_{\mathrm{KL}}(V||P)$, i.e.,

$$V^\star(h)=\underset{V\in\mathcal{M}(\mathcal{H})}{\arg\min}\ \hat{\mathcal{L}}(V,\boldsymbol{S})+\beta^{-1}D_{\mathrm{KL}}(V||P).$$

The optimal distribution $V^\star$, also known as the optimal Gibbs posterior [35], has a closed-form expression and is given by

$$V^\star(h)=\frac{P(h)\exp\big(-\beta\hat{\mathcal{L}}(h,\boldsymbol{S})\big)}{\mathbb{E}_{h\sim P}\big[\exp\big(-\beta\hat{\mathcal{L}}(h,\boldsymbol{S})\big)\big]}. \tag{44}$$

Let $\beta=m$ and the loss function L is chosen as the negative log-likelihood, i.e., $L(h,\boldsymbol{z}_i)=-\log p(\boldsymbol{z}_i\,|\,h)$. The optimal Gibbs posterior coincides with the Bayesian posterior

$$V^\star_{P,\boldsymbol{S}}(h)=\frac{P(h)\prod_{i=1}^{m}p(\boldsymbol{z}_i\,|\,h)}{\int_{\mathcal{H}}P(h)\prod_{i=1}^{m}p(\boldsymbol{z}_i\,|\,h)dh}. \tag{45}$$

Let $V:\mathcal{Z}^n\times\mathcal{M}(\mathcal{H})\to\mathcal{M}(\mathcal{H})$ denote an arbitrary learner, which takes in a dataset of size n and a prior and outputs a posterior. For the dataset $\boldsymbol{S}$ and prior P, the resultant posterior is denoted by $V(\boldsymbol{S},P)$. Note that so far, we have considered only a single learning task with distribution $\mathcal{D}$. However, our goal is to extract desired information or knowledge shared by a batch of correlated tasks. All possible tasks are collected into set $\mathcal{T}=\{\nu_i=(\mathcal{D}_i,n_i,\boldsymbol{S}_i)\}$, where $\mathcal{D}_i$, n_i and $\boldsymbol{S}_i$ represent the data distribution, the number of samples, and the sampled dataset of the underlying task, respectively. For simplicity, m tasks can also be denoted by $\mathfrak{D}=\{\boldsymbol{S}_1,\boldsymbol{S}_2,\cdots,\boldsymbol{S}_m\}$.

B. Proof of Theorem 2

Given n tasks, i.e., $\mathfrak{D}=\{\boldsymbol{S}_1,\boldsymbol{S}_2,\cdots,\boldsymbol{S}_m\}$, the problem to be addressed boils down to extracting the domain knowledge shared by the n tasks. To this end, we can impose a probability measure on $\mathcal{M}(\mathcal{H})$, denoted by $\mathcal{M}(\mathcal{M}(\mathcal{H}))$. Since each task is associated to a probability measure, i.e., an element of $\mathcal{M}(\mathcal{H})$, the domain knowledge shared by different tasks should be an element of $\mathcal{M}(\mathcal{M}(\mathcal{H}))$. Similar to ordinary Bayesian learning, the prior of the domain knowledge, referred to as hyper-prior, is represented by $\mathcal{P}\in\mathcal{M}(\mathcal{M}(\mathcal{H}))$, i.e., a distribution over priors P. Given m datasets $\{\boldsymbol{S}_1,\boldsymbol{S}_2,\cdots,\boldsymbol{S}_m\}$ from m tasks, the design or optimization goal is to update the hyper-prior $\mathcal{P}$ into hyper-posterior $\mathcal{Q}$, another element of $\mathcal{M}(\mathcal{M}(\mathcal{H}))$.

The update or optimization criterion is to minimize the GE from the available datasets $\{\boldsymbol{S}_1,\boldsymbol{S}_2,\cdots,\boldsymbol{S}_m\}$ of m tasks to future unseen tasks. The GE here, which is also referred to as transfer-error, is measured by the expected Gibbs error:

$$\mathcal{L}(\mathcal{Q},\mathcal{T})=\underset{P\sim\mathcal{Q}}{\mathbb{E}}\ \underset{(\mathcal{D},n)\sim\mathcal{T}}{\mathbb{E}}\ \underset{\boldsymbol{S}\sim\mathcal{D}^n}{\mathbb{E}}[\mathcal{L}(V(\boldsymbol{S},P),\mathcal{D})]. \tag{46}$$

Because $\mathcal{L}(\mathcal{Q},\mathcal{T})$ is unknown in practice, the empirical multi-task error is utilized instead, which is defined as

$$\mathcal{L}(\mathcal{Q},\boldsymbol{S}_1,\cdots,\boldsymbol{S}_m)=\underset{P\sim\mathcal{Q}}{\mathbb{E}}\left[\frac{1}{m}\sum_{i=1}^{m}\hat{\mathcal{L}}(V(\boldsymbol{S}_i,P),\boldsymbol{S}_i)\right]. \tag{47}$$

To obtain an objective function, we shall derive a tractable upper bound of $\mathcal{L}(\mathcal{Q},\mathcal{T})$, which consists of four steps.

1) Step 1: We bound the GE of each task $\nu_i=(\mathcal{D}_i,n_i,\boldsymbol{S}_i)$ with $\boldsymbol{S}_i\sim\mathcal{D}_i^{n_i}$. Note that given a prior distribution P and the dataset $\boldsymbol{S}_i$, the learner V generates a posterior distribution $V=V(\boldsymbol{S}_i,P)$. To derive the GE bound via the important inequality (43) [36], the hypothesis space, prior and loss function should be appropriately redefined. A "tuple hypothesis" is defined as $g=(P,h)$, where $P\in\mathcal{M}(\mathcal{H})$ and $h\in\mathcal{H}$. The "prior over hypothesis" $\pi = (\mathcal{P}, P)$ is defined as a distribution over $\mathcal{M}(\mathcal{H}) \times \mathcal{H}$, in which P is first sampled from $\mathcal{P}$ and h is then sampled from P. The "posterior over hypothesis" is defined similarly. In particular, for $\rho = (\mathcal{Q}, V(\boldsymbol{S}_i, P))$, P is sampled first from $\mathcal{Q}$ and h is then sampled from $V = V(\boldsymbol{S}, P)$.

The KL divergence between ρ and π is calculated as

$$\begin{aligned}
&D_{\text{KL}}(\rho||\pi) = \mathop{\mathbb{E}}_{g\sim\rho}\left(\log\frac{\rho(g)}{\pi(g)}\right)\\
&=\mathop{\mathbb{E}}_{P\sim\mathcal{Q}}\mathop{\mathbb{E}}_{h\sim V(\boldsymbol{S}_i,P)}\left(\log\frac{\mathcal{Q}(P)V(\boldsymbol{S}_i,P)(h)}{\mathcal{P}(P)P(h)}\right)\\
&=\mathop{\mathbb{E}}_{P\sim\mathcal{Q}}\left(\log\frac{\mathcal{Q}(P)}{\mathcal{P}(P)}\right)+\mathop{\mathbb{E}}_{P\sim\mathcal{Q}}\mathop{\mathbb{E}}_{h\sim V(\boldsymbol{S}_i,P)}\left(\log\frac{V(\boldsymbol{S}_i,P)(h)}{P(h)}\right)\\
&=D_{\text{KL}}(\mathcal{Q}||\mathcal{P})+\mathop{\mathbb{E}}_{P\sim\mathcal{Q}}D_{\text{KL}}(V(\boldsymbol{S}_i,P)||P).
\end{aligned}$$

Similar to the derivation of (43) and letting $\beta = n_i$, we can obtain the inequality in (48), for confidence level $\delta_{1,i} \in (0, 1]$.

2) Step 2: In this step, we further bound the GE of the task-environment level. In particular, $\{\nu_i = (\mathcal{D}_i, n_i, \boldsymbol{S}_i)\}$ are tasks drawn i.i.d. from the task-environment distribution $\mathcal{T}$. In this case, P, $\mathcal{P}$ and $\mathcal{Q}$ respectively play the roles of g, π and ρ, i.e., $g = P$, $\pi = \mathcal{P}$ and $\rho = \mathcal{Q}$. Given the hyper-prior $\mathcal{P}$ and hyper-posterior $\mathcal{Q}$, for confidence level δ_2, we can obtain similarly a probability inequality, which is given by

$$\begin{aligned}
\mathbb{P}\Bigg\{\mathcal{L}(\mathcal{Q},\mathcal{T}) \leq &\frac{1}{m}\sum_{i=1}^{m}\mathcal{L}(\mathcal{Q},\mathcal{D}_i)+\frac{1}{m}\bigg(D_{\text{KL}}(\mathcal{Q}||\mathcal{P})\\
&-\ln\delta_2+\Psi(m,m)\bigg)\Bigg\} \geq 1-\delta_2. \qquad (49)
\end{aligned}$$

3) Step 3: We define $m+2$ events $A_1, A_2, \cdots, A_m$, B and C, which are given in (50) - (52). Note that $\delta \in (0, 1)$ in (50) - (52) is a small positive real number. For these events, from Step 1 and Step 2, we have obtained $\mathbb{P}(A_i) \geq 1-\delta/(2m)$ and $\mathbb{P}(B) \geq 1 - \delta/2$. Note that C holds if $\{A_i\}$ and B all hold, which implies that $\big(\cap_{i=1}^m A_i\big) \cap B \subset C$ holds. According to the union bound formula (of probability), we can obtain

$$\begin{aligned}
\mathbb{P}(C) \geq&\mathbb{P}(A_1\cap A_2\cap\cdots\cap A_m\cap B)\\
=&1-\mathbb{P}(A_1^c\cup A_2^c\cup\cdots\cup A_m^c\cup B^c)\\
\geq&1-\mathbb{P}(A_1^c)-\mathbb{P}(A_2^c)-\cdots\mathbb{P}(A_m^c)-\mathbb{P}(B^c)\geq 1-\delta.
\end{aligned}$$

The above inequality implies that with probability at least $1-\delta$, the transfer-error is upper bounded by $U(\mathcal{Q}, \boldsymbol{S}_1, \cdots, \boldsymbol{S}_m) + C(m, \{n_i\}, \delta)$, with $U(\mathcal{Q}, \boldsymbol{S}_1, \cdots, \boldsymbol{S}_m)$ given in (53). In view that $C(m, \{n_i\}, \delta)$ is a constant, to minimize the transfer-error we should minimize $U(\mathcal{Q}, \boldsymbol{S}_1, \cdots, \boldsymbol{S}_m)$.

4) Step 4: Note that since the Gibbs posterior minimizes PAC-Bayesian error bound, it is naturally chosen here. In this case, $V(\boldsymbol{S}_i, P)$ (for $\boldsymbol{S}_i$) takes the form

$$V(\boldsymbol{S}_i,P)(h)=\frac{P(h)\exp\big(-n_i\hat{\mathcal{L}}(h,\boldsymbol{S}_i)\big)}{\mathbb{E}_{h\sim P}\big[\exp\big(-n_i\hat{\mathcal{L}}(h,\boldsymbol{S}_i)\big)\big]}. \qquad (54)$$

Given $V(\boldsymbol{S}_i, P)$, the summation of the first and third terms of $U(\mathcal{Q}, \boldsymbol{S}_1, \cdots, \boldsymbol{S}_m)$ can be simplified as

$$\begin{aligned}
&\mathcal{L}(\mathcal{Q},\boldsymbol{S}_1,\cdots,\boldsymbol{S}_m)+\frac{1}{m}\sum_{i=1}^{m}\frac{1}{n_i}\mathop{\mathbb{E}}_{P\sim\mathcal{Q}}\big[D_{\text{KL}}(V(\boldsymbol{S}_i,P)||P)\big]\\
=&\frac{1}{m}\sum_{i=1}^{m}\mathop{\mathbb{E}}_{P\sim\mathcal{Q}}\left(\hat{\mathcal{L}}(V(\boldsymbol{S}_i,P),\boldsymbol{S}_i)+\frac{1}{n_i}D_{\text{KL}}(V(\boldsymbol{S}_i,P)||P)\right)\\
=&\frac{1}{m}\sum_{i=1}^{m}\mathop{\mathbb{E}}_{P\sim\mathcal{Q}}\mathop{\mathbb{E}}_{h\sim V}\left(\hat{\mathcal{L}}(h,\boldsymbol{S}_i)+\frac{1}{n_i}\ln\frac{V(\boldsymbol{S}_i,P)(h)}{P(h)}\right)\\
\overset{(*)}{=}&\frac{1}{m}\sum_{i=1}^{m}-\frac{1}{n_i}\mathop{\mathbb{E}}_{P\sim\mathcal{Q}}\ln\left(\mathop{\mathbb{E}}_{h\sim P}\big[\exp\big(-n_i\hat{\mathcal{L}}(h,\boldsymbol{S}_i)\big)\big]\right), \qquad (55)
\end{aligned}$$

where $(*)$ is due to (54). As a result, $mU(\mathcal{Q}, \boldsymbol{S}_1, \cdots, \boldsymbol{S}_m)$ can be simplified as

$$\begin{aligned}
mU(\mathcal{Q},\boldsymbol{S}_1,\cdots,\boldsymbol{S}_m)=&\left(1+\sum_{i=1}^{m}\frac{1}{n_i}\right)D_{\text{KL}}(\mathcal{Q}||\mathcal{P})\\
&-\sum_{i=1}^{m}\frac{1}{n_i}\mathop{\mathbb{E}}_{P\sim\mathcal{Q}}\ln\left(\mathop{\mathbb{E}}_{h\sim P}\big[\exp\big(-n_i\hat{\mathcal{L}}(h,\boldsymbol{S}_i)\big)\big]\right).
\end{aligned}$$

Note that the use of $mU(\mathcal{Q}, \boldsymbol{S}_1, \cdots, \boldsymbol{S}_m)$ as the optimization objective does not affect the optimality. By minimizing $mU(\mathcal{Q}, \boldsymbol{S}_1, \cdots, \boldsymbol{S}_m)$ with respect to $\mathcal{Q}$ and similar to (44), we can obtain the optimal posterior $\mathcal{Q}^\star$ given by

$$\begin{aligned}
&\mathcal{Q}^\star(P)\\
=&\frac{\mathcal{P}(P)\exp\Big(\big(1+\sum_{i=1}^{m}n_i^{-1}\big)^{-1}\sum_{i=1}^{m}\frac{1}{n_i}Z(\boldsymbol{S}_i,P)\Big)}{\mathbb{E}_{P\sim\mathcal{P}}\Big[\exp\Big(\big(1+\sum_{i=1}^{m}n_i^{-1}\big)^{-1}\sum_{i=1}^{m}\frac{1}{n_i}Z(\boldsymbol{S}_i,P)\Big)\Big]},
\end{aligned}$$

where $Z(\boldsymbol{S}_i, P) = \ln\Big(\mathop{\mathbb{E}}_{h\sim P}\big[\exp\big(-n_i\hat{\mathcal{L}}(h, \boldsymbol{S}_i)\big)\big]\Big)$ is introduced for convenience. When the GP regression is chosen as the learner V, $Z(\boldsymbol{S}_i, P)$ is, in fact, the marginal log-likelihood given in (11). We have completed the proof.

References

[1] M. Xiao, S. Mumtaz, Y. Huang, L. Dai, Y. Li, M. Matthaiou, G. K. Karagiannidis, E. Björnson, K. Yang, C. L. I, and A. Ghosh, "Millimeter wave communications for future mobile networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 1909–1935, Sept 2017.

[2] R. W. Heath, N. Gonzlez-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, "An overview of signal processing techniques for millimeter wave MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 436–453, April 2016.

[3] S. He, J. Wang, Y. Huang, B. Ottersten, and W. Hong, "Codebook-based hybrid precoding for millimeter wave multiuser systems," *IEEE Trans. Signal Process.*, vol. 65, no. 20, pp. 5289–5304, Oct 2017.

[4] J. Singh and S. Ramakrishna, "On the feasibility of codebook-based beamforming in millimeter wave systems with multiple antenna arrays," *IEEE Trans. Wireless Commun.*, vol. 14, no. 5, pp. 2670–2683, May 2015.

[5] M. Gao, B. Ai, Y. Niu, Z. Zhong, Y. Liu, G. Ma, Z. Zhang, and D. Li, "Dynamic mmwave beam tracking for high speed railway communications," in *2018 IEEE WCNCW*, April 2018, pp. 278–283.

[6] S. Hur, T. Kim, D. Love, J. Krogmeier, T. Thomas, and A. Ghosh, "Millimeter wave beamforming for wireless backhaul and access in small cell networks," *IEEE Trans. Commun.*, vol. 61, no. 10, pp. 4391–4403, October 2013.

[7] J. Zhang, Y. Huang, Q. Shi, J. Wang, and L. Yang, "Codebook design for beam alignment in millimeter wave communication systems," *IEEE Trans. Commun.*, vol. 65, no. 11, pp. 4980–4995, Nov 2017.

[8] V. Va, H. Vikalo, and R. W. Heath, "Beam tracking for mobile millimeter wave communication systems," in *2016 IEEE GlobalSIP*, Dec 2016, pp. 743–747.

$$\mathbb{P}\bigg\{\mathcal{L}(\mathcal{Q},\mathcal{D}_i) \leq \mathcal{L}(\mathcal{Q},\boldsymbol{S}_i) + \frac{1}{n_i}\Big(D_{\mathrm{KL}}(\mathcal{Q}||\mathcal{P}) + \mathop{\mathbb{E}}_{P\sim\mathcal{Q}} D_{\mathrm{KL}}(Q(\boldsymbol{S}_i,P)||P) + \ln\frac{1}{\delta_{1,i}} + \Psi(n_i,n_i)\Big)\bigg\} \geq 1-\delta_{1,i}. \tag{48}$$

$$A_i = \bigg\{\mathcal{L}(\mathcal{Q},\mathcal{D}_i) \leq \mathcal{L}(\mathcal{Q},\boldsymbol{S}_i) + \frac{1}{n_i}\Big(D_{\mathrm{KL}}(\mathcal{Q}||\mathcal{P}) + \mathop{\mathbb{E}}_{P\sim\mathcal{Q}} D_{\mathrm{KL}}(Q(\boldsymbol{S}_i,P)||P) + \ln\frac{2m}{\delta} + \Psi(n_i,n_i)\Big)\bigg\}. \tag{50}$$

$$B = \bigg\{\mathcal{L}(\mathcal{Q},\mathcal{T}) \leq \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}(\mathcal{Q},\mathcal{D}_i) + \frac{1}{m}\bigg(D_{\mathrm{KL}}(\mathcal{Q}||\mathcal{P}) - \ln\frac{\delta}{2} + \Psi(m,m)\bigg)\bigg\}. \tag{51}$$

$$C = \bigg\{\mathcal{L}(\mathcal{Q},\mathcal{T}) \leq \mathcal{L}(\mathcal{Q},\boldsymbol{S}_1,\cdots,\boldsymbol{S}_m) + \frac{1}{m}\bigg(1+\sum_{i=1}^{m}\frac{1}{n_i}\bigg)D_{\mathrm{KL}}(\mathcal{Q}||\mathcal{P}) + \frac{1}{m}\sum_{i=1}^{m}\frac{1}{n_i}\mathop{\mathbb{E}}_{P\sim\mathcal{Q}}\big[D_{\mathrm{KL}}(V(\boldsymbol{S}_i,P)||P)\big] + C(m,\{n_i\},\delta)\bigg\} \quad \text{with} \quad C(m,\{n_i\},\delta) = \frac{1}{m}\ln\frac{2}{\delta} + \frac{1}{m}\Psi(m,m) + \frac{1}{m}\sum_{i=1}^{m}\frac{1}{n_i}\bigg(\ln\frac{2m}{\delta} + \Psi(n_i,n_i)\bigg). \tag{52}$$

$$U(\mathcal{Q},\boldsymbol{S}_1,\cdots,\boldsymbol{S}_m) = \mathcal{L}(\mathcal{Q},\boldsymbol{S}_1,\cdots,\boldsymbol{S}_m) + \frac{1}{m}\bigg(1+\sum_{i=1}^{m}\frac{1}{n_i}\bigg)D_{\mathrm{KL}}(\mathcal{Q}||\mathcal{P}) + \frac{1}{m}\sum_{i=1}^{m}\frac{1}{n_i}\mathop{\mathbb{E}}_{P\sim\mathcal{Q}}\big[D_{\mathrm{KL}}(V(\boldsymbol{S}_i,P)||P)\big]. \tag{53}$$

[9] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic, "Deep learning coordinated beamforming for highly-mobile millimeter wave systems," *IEEE Access*, vol. 6, pp. 37 328–37 348, 2018.

[10] D. Zhang, A. Li, M. Shirvanimoghaddam, P. Cheng, Y. Li, and B. Vucetic, "Codebook-based training beam sequence design for millimeter wave tracking systems," *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5333–5349, Nov 2019.

[11] Z. Xiao, P. Xia, and X. G. Xia, "Codebook design for millimeter-wave channel estimation with hybrid precoding structure," *IEEE Trans. Wireless Commun.*, vol. 16, no. 1, pp. 141–153, Jan 2017.

[12] F. Liu, W. Yuan, C. Masouros, and J. Yuan, "Radar-assisted predictive beamforming for vehicular links: Communication served by sensing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7704–7719, 2020.

[13] S. G. Larew and D. J. Love, "Adaptive beam tracking with the unscented kalman filter for millimeter wave communication," *IEEE Signal Process. Lett.*, vol. 26, no. 11, pp. 1658–1662, 2019.

[14] F. Liu, P. Zhao, and Z. Wang, "EKF-based beam tracking for mmwave MIMO systems," *IEEE Commun. Lett.*, vol. 23, no. 12, pp. 2390–2393, 2019.

[15] W. Yuan, F. Liu, C. Masouros, J. Yuan, D. W. K. Ng, and N. Gonzlez-Prelcic, "Bayesian predictive beamforming for vehicular networks: A low-overhead joint radar-communication approach," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1442–1456, 2021.

[16] V. Va, J. Choi, T. Shimizu, G. Bansal, and R. W. Heath, "Inverse multipath fingerprinting for millimeter wave V2I beam alignment," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4042–4058, May 2018.

[17] K. Satyanarayana, M. El-Hajjar, A. A. M. Mourad, and L. Hanzo, "Deep learning aided fingerprint-based beam alignment for mmwave vehicular communication," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 10 858–10 871, Nov 2019.

[18] V. Va, T. Shimizu, G. Bansal, and R. W. Heath, "Online learning for position-aided millimeter wave beam training," *IEEE Access*, vol. 7, pp. 30 507–30 526, 2019.

[19] J. Zhang, Y. Huang, Y. Zhou, and X. You, "Beam alignment and tracking for millimeter wave communications via bandit learning," *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5519–5533, 2020.

[20] M. Cheng, J. Wang, J. Wang, M. Lin, Y. Wu, and H. Zhu, "A fast beam searching scheme in mmwave communications for high-speed trains," in *2019 IEEE ICC*, May 2019, pp. 1–6.

[21] W. Wu, N. Cheng, N. Zhang, P. Yang, W. Zhuang, and X. Shen, "Fast mmwave beam alignment via correlated bandit learning," *IEEE Trans. Wireless Commun.*, vol. 18, no. 12, pp. 5894–5908, 2019.

[22] M. B. Booth, V. Suresh, N. Michelusi, and D. J. Love, "Multi-armed bandit beam alignment and tracking for mobile millimeter wave communications," *IEEE Commun. Lett.*, vol. 23, no. 7, pp. 1244–1248, 2019.

[23] J. Zhang, Y. Huang, J. Wang, and X. You, "Intelligent beam training for millimeter-wave communications via deep reinforcement learning," in *2019 IEEE GLOBECOM*, Dec 2019, pp. 1–7.

[24] J. Zhang, Y. Huang, J. Wang, X. You, and C. Masouros, "Intelligent interactive beam training for millimeter wave communications," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 2034–2048, 2021.

[25] W. Xu, Y. Ke, C.-H. Lee, H. Gao, Z. Feng, and P. Zhang, "Data-driven beam management with angular domain information for mmwave uav networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 11, pp. 7040–7056, 2021.

[26] J. Zhang and C. Masouros, "Learning-based predictive transmitter-receiver beam alignment in millimeter wave fixed wireless access links," *IEEE Trans. Signal Process.*, vol. 69, pp. 3268–3282, 2021.

[27] J. Zhang, W. Xu, H. Gao, M. Pan, Z. Feng, and Z. Han, "Position-attitude prediction based beam tracking for uav mmwave communications," in *2019 IEEE ICC*, 2019, pp. 1–7.

[28] J. Zhang, W. Xu, H. Gao, M. Pan, Z. Han, and P. Zhang, "Codebook-based beam tracking for conformal array-enabled uav mmwave networks," *IEEE Internet Things J.*, vol. 8, no. 1, pp. 244–261, 2021.

[29] H.-L. Song and Y.-C. Ko, "Beam alignment for high-speed uav via angle prediction and adaptive beam coverage," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 10 185–10 192, 2021.

[30] S. Sarkka, A. Solin, and J. Hartikainen, "Spatiotemporal learning via infinite-dimensional bayesian filtering and smoothing: A look at gaussian process regression through kalman filtering," *IEEE Signal Process. Mag.*, vol. 30, no. 4, pp. 51–61, 2013.

[31] J. Zhang, C. Masouros, and Y. Huang, "Beam training and tracking with limited sampling sets: Exploiting environment priors," *IEEE Trans. Commun.*, vol. 71, no. 5, pp. 3008–3023, 2023.

[32] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[33] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[34] P. Alquier, J. Ridgway, and N. Chopin, "On the properties of variational approximations of Gibbs posteriors," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 8374–8414, 2015.

[35] O. Catoni, "Pac-Bayesian supervised classification: The thermodynamics of statistical learning," *Lecture Notes-Monograph Series*, vol. 56, pp. 1–163, 2007.

[36] R. Amit and R. Meir, "Meta-learning by adjusting priors based on extended PAC-Bayes theory," *2018 ICML*, 2018.