Beam Training and Tracking with Limited Sampling Sets: Exploiting Environment Priors

Jianjun Zhang, *Member, IEEE*, Christos Masouros, *Senior Member, IEEE*
and Yongming Huang, *Senior Member, IEEE*

***Abstract*—Beam training and tracking (BTT) are key technologies for millimeter wave communications. However, since the effectiveness of BTT methods heavily depends on wireless environments, complexity and randomness of practical environments severely limit the application scope of many BTT algorithms and even invalidate them. To tackle this issue, from the perspective of stochastic process (SP), in this paper we propose to model beam directions as a SP and address the problem of BTT via process inference. The benefit of the SP design methodology is that environment priors and uncertainties can be naturally taken into account (e.g., to encode them into SP distribution) to improve prediction efficiencies (e.g., accuracy and robustness). We take the Gaussian process (GP) as an example to elaborate on the design methodology and propose novel learning methods to optimize the prediction models. In particular, beam training subset is optimized based on derived posterior distribution. The GP-based SP methodology enjoys two advantages. First, good performance can be achieved even for small data, which is very appealing in dynamic communication scenarios. Second, in contrast to most BTT algorithms that only predict a single beam, our algorithms output an optimizable beam subset, which enables a flexible tradeoff between training overhead and desired performance. Simulation results show the superiority of our approach.**

***Index Terms*—Beam training and tracking, Bayesian learning, 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 frequency of mmwave signals leads 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 array gains, which is realized by packing a large number of antennas into a small size thanks to the short wave-length. Nevertheless, narrow high-gain beams make beam alignment challenging, especially in mobile applications and/or dynamic environments [2], [3].

To obtain the channel state information in mmwave communications, different design methodologies have been investigated, among which the most widely accepted methodology is BTT [4]–[9]. In BTT, candidate beams at the transmitter and/or receiver are trained via a predefined search mode by choosing the ones that can optimize some performance metric, e.g., signal-to-noise ratio (SNR) [4]–[6]. [1] The existing BTT methods roughly fall into two categories. The first one is the channel-environment-independent methodology, which boils down to codebook design. In the past ten years, a variety of algorithms have been proposed to design codebooks having desired properties [6], [7], [10], [11]. The ostensible advantage of the environment-independent BTT methodology is that it can apply to different channel environments. However, since the training overhead can be prohibitively high (especially for large-scale antenna array systems), its application scope is often limited, e.g., only slow-varying environments and small-scale antenna array systems.

To reduce the overhead of BTT, the core is to reduce beam search space for future sweeping by exploiting environment priors, which constitutes the second category. Depending on how the environment prior knowledge is extracted and exploited, the environment-dependent design methodology can be further divided into three sub-categories. The first one is model-driven BTT design, whose key is to first build a motion or kinetic model and then invoke the classical Kalman filter techniques [12]–[15]. The effectiveness of this methodology critically relies on building the kinetic model, which can be very difficult in most cases, because of the complexity and randomness of practical environments. As a result, it is often utilized in the scenarios, where there exists a fixed trajectory, e.g., vehicle network or high-speed railway.

To circumvent the difficulty of manually building models, data- and model- driven BTT design methodology (i.e., the second sub-category) has been proposed, which leads to supervised learning based BTT algorithms [16]–[20]. The cost of the supervised learning based algorithms is that the number of training samples required may be very large in order to achieve satisfying performance, which brings about large burden and latency in establishing the link. In contrast to other supervised learning applications (e.g., computer vision), where a large number of training samples are ready or available, the dynamic nature of wireless communications imposes short training

The work was supported by the Engineering and Physical Sciences Research Council, UK under project EP/S028455/1, in part by the Scientific Research Fund of the Nanjing University of Aeronautics and Astronautics, and in part by the National Natural Science Foundation of China under Grants 61720106003 and 62225107. (Corresponding author: Jianjun Zhang, Yongming Huang.)

J. Zhang is with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China (e-mail: jianjun.zhang@nuaa.edu.cn). C. Masouros is with the Department of Electronic & Electrical Engineering, University College London, London WC1E7JE, U.K. (e-mail: c.masouros@ucl.ac.uk). Y. Huang is with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China. (e-mail: huangym@seu.edu.cn).

[1] In view that the main difference between beam training and beam tracking is that the size of beams used for tracking is smaller than that for training (e.g., the size of beams used for tracking can be even one), we ignore this difference in this paper.

timescales and the necessity to train with limited numbers of samples. It is therefore difficult and expensive to collect sufficient training samples. Furthermore, the collected training samples can be easily outdated, due to fast channel fading and variations. These factors greatly limit the application scope of the second sub-category of BTT solutions.

To avoid collecting training samples in advance or offline, reinforcement learning has been proposed to design BTT solutions, which constitute the third sub-category [21]–[24]. An appealing advantage of the reinforcement learning based solutions is that they can collect the samples via interacting with environments, which enables to implement learning-based BTT algorithms online [25], [26]. However, since the fundamental of reinforcement learning is Markov decision process, a shortcoming is that the convergence rate is slow, which fails to achieve good performance on the short term. Moreover, the feedback of Markov decision process is typically a real number referred to as the reward, which is affected by many factors. It is often difficult to design this feedback efficiently and make optimal decisions. Note that most learning-based BTT algorithms (designed based on either supervised learning or reinforcement learning) only predict a point estimation (i.e., a single beam). However, because of the complexity and randomness of practical environments, it is generally difficult to predict an accurate beam, which, as a result, inevitably leads to a high probability of misalignment.

To tackle the aforementioned issues, in this paper we propose to explicitly model beam direction trajectory (BDT) via SP and address the problem of BTT from the view of SP inference. Compared to the model-driven BTT methods, there is no need to manually and explicitly build a kinetic model for prediction. The randomness of channel environments can also be naturally incorporated into SP models. To enable fast inference, we choose GP as an example and formulate the problem of BTT as GP prediction. In particular, environment priors or patterns are encoded into the GP kernel to improve prediction efficiency. To accommodate complex environments, neural network techniques are exploited to enhance classical GP kernels, which guarantees that sufficient kernels (in terms of both types and amounts) are available. Compared to the supervised learning or reinforcement learning based solutions, the advantages of the GP based BTT algorithms are two-fold. First, incorporating Bayesian learning, good performance (e.g., success probability) can be achieved for small sample setting. Second, an optimizable beam interval, rather than a single beam, is generated by our BTT algorithms, which makes the algorithms more robust. They also provide a flexible tradeoff between desired performance and training overhead. The main contributions of this paper are summarized as follows:

- To identify and exploit useful priors/patterns which are difficult to be utilized by existing methods, we propose to model beam directions via SP and address the problem of BTT via process inference. In particular, we propose to distinguish two types of properties of a SP, i.e., sample path property and statistical regularity property.
- To facilitate SP inference, we choose GP as an example and formulate the problem of BTT as GP prediction. In particular, meaningful patterns of environments and their changes are encoded into the GP kernel parameterized by neural networks to improve beam prediction efficiency (e.g., high accuracy with low overhead).
- To effectively extract the priors/patterns of interest from environments and improve beam training efficiency, we propose a novel network structure and novel learning or optimization methods, whose key is to distinguish the sample path property and statistical regularity property.
- Based on Bayesian deep learning, we propose efficient BTT algorithms. We further improve the proposed algorithms via Bayesian posterior inference and optimization, which can provide a flexible tradeoff between the beam training overhead and desired performance.
- Comprehensive simulation results confirm the effectiveness of our approach. In particular, they can dynamically and adaptively amend prediction uncertainty level, which thus predicts the most economical beam range while still ensuring link reliability. Moreover, as more samples are collected, the predicted beam range decreases.
- The simulation results also reveal many useful insights in terms of modeling and multiple tradeoffs. In terms of modeling, our GP methodology can construct a desired probabilistic prediction model from given priors and data automatically. It can also refine the constructed prediction model based on collected online data.
- In terms of the sample complexity, the GP approach notably shortens the data samples required, even compared to the state-of-the-art beam index difference technique [21]. Meanwhile, it inherently realizes multiple scalable tradeoffs, e.g., effective achievable rate and beam training overhead, sample complexity and scalable robustness, the use between online data and historical data.

The remainder of this paper is organized as follows. System model of BTT is described in Section II. In Section III, the principle of BTT via SP modeling and inference is presented. To efficiently extract and exploit the priors/patterns, novel network structure and learning methods are proposed in Section IV. To provide flexible tradeoffs, Bayesian posterior inference and optimization are presented in Section V. Simulation results and conclusion are provided in Section VI and Section VII, respectively. Proofs are deferred to the appendix to improve readability. The abbreviations used in this paper and their full names are provided in the following table.

Notations: Bold uppercase $\mathbf{A}$ and bold lowercase $\mathbf{a}$ denote matrices and column vectors, 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 $\mathrm{card}(\mathcal{A})$ represent the indicator function and the cardinality of $\mathcal{A}$, respectively. $(\cdot)^{\star}$ represents an optimal quantity, e.g., an optimal solution of an optimization problem. $\mathcal{CN}(\mathbf{m}, \mathbf{R})$ stands for a complex Gaussian random vector with mean $\mathbf{m}$ and covariance matrix $\mathbf{R}$. $\mathbf{I}$ denotes an identity matrix.

II. System Model

Consider the mmwave point-to-point communication system, which consists of one base station (BS) equipped with N

Abbr.	Full Name	Abbr.	Full Name	Abbr.	Full Name
BTT	Beam Training and Tracking	HST	High-Speed Train	GP	Stochastic Process
EEAR	Expected Effective Achievable Rate	SR	Statistical Regularity	SP	Gaussian Process
AEAR	Average Effective Achievable Rate	BCI	Beam Confidence Interval	SBL	Stochastic Bandit Learning
BDT	Beam Direction Trajectory	LML	Log Marginal Likelihood	GPL	Gaussian Process Learning
KEN	Kernel Encoding Network	PTN	Process Transform Network	ML	Machine Learning
PSA	Probability of Successful Alignment	EAR	Effective Achievable Rate	DS	Direct Search

transmit antennas and a single-antenna user. To facilitate system implementation, the codebook based analog beamforming is considered in this paper, where each beam is chosen from a codebook $\mathcal{C}$ of size M [27]. Without loss of generality, $\mathcal{C}$ is constructed by uniformly sampling the beam space, i.e.,

$$\mathcal{C} = \big\{\mathbf{f}_i = \mathbf{a}(-1 + 2i/M) \,|\, i = 0, 1, 2, \cdots, M-1\big\},$$

where $\mathbf{a}(\cdot)$ represents the array response vector. For a uniform linear array, $\mathbf{a}(\cdot)$ takes the form

$$\mathbf{a}(x) = \frac{1}{\sqrt{N}}\left[1, e^{j\frac{2\pi}{\lambda}xd}, e^{j\frac{2\pi}{\lambda}2xd}, \cdots, e^{j\frac{2\pi}{\lambda}(N-1)xd}\right], \quad (1)$$

where λ denotes the signal wave-length and d represents the distance between any two adjacent antennas.

Due to the sparsity of mmwave channels, an extended Saleh-Valenzuela geometric model is considered here [4]–[10]. The channel vector between the BS and user is given by

$$\mathbf{h} = \sqrt{N/\beta}\sum_{l=1}^{L}\alpha_l \mathbf{a}(\phi_l), \quad (2)$$

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 (2), $\phi_l = \cos(\theta_l)$, where θ_l is the physical angle of departure of the l-th path. With the assumption that beam i (i.e., $\mathbf{f}_i$) is chosen by the BS, the signal received at the user is given by

$$y_i = \sqrt{P}\mathbf{h}^{\mathrm{H}}\mathbf{f}_i s + n_i, \quad (3)$$

where P is the transmit power, s represents the pilot symbol, and $n_i \sim \mathcal{CN}(0, 1)$ denotes the random noise.

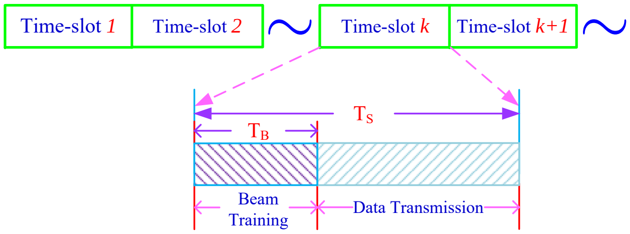

Fig. 1. Typically, each time-slot consists of two phases, i.e., beam training and data transmission.

As shown in Fig. 1, each time-slot consists of two phases, i.e., beam training (including beam alignment and tracking) and data transmission. The main task of beam training is to find out the optimal beam for the subsequent phase of data transmission. The effective achievable rate is often chosen as the performance metric, which is defined as [9]

$$R_{\text{eff}} = (1 - T_{\mathrm{B}}/T_{\mathrm{S}})\log\left(1 + P|\mathbf{h}^{\mathrm{H}}\mathbf{f}_i|^2\right), \quad (4)$$

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

The key tradeoff can be observed from (4). On one hand, to achieve a high throughput, the time allocated for T_{B} should be as little as possible, so as to reserve more time for data transmission. On the other hand, a well-trained beam yields high-gain link $|\mathbf{h}^{\mathrm{H}}\mathbf{f}_i|$, therefore improving the effective rate. The conflicting conditions impose a tradeoff in the resources spent for BTT. This issue becomes more severe in mobile scenarios, where the channels vary quickly and thus more frequent beam training is needed. Next, we tackle this issue by identifying and exploiting environment priors/patterns from the perspective of SP modeling and inference.

III. Principle of Stochastic Process Methodology

In this section, we first analyze and illuminate that there exist many useful priors and patterns, which, however, cannot be exploited effectively by conventional BTT methods. Then, we demonstrate how to tackle this issue from the perspective of SP modeling and inference.

A. *Useful Priors and Patterns for Beam Training and Tracking*

In practice, there exist environment priors and patterns that can assist BTT. For example, low-mobility may introduce strong correlation between channel instantiations, vehicular or railway scenarios may involve constrained or even fixed trajectories, indoor or open-air scenarios may provide specific geometry priors. If these priors/patterns can be well identified and exploited, they can greatly improve BTT efficiency (e.g., lower training overhead). Nevertheless, the influence of these nonparametric priors/patterns onto the communication link is often difficult to model and exploit via conventional methods from parametric statistics, which necessitates a learning-based approach from nonparametric statistics. [2]

Before proceeding, we take several typical communication scenarios to reveal that these priors/patterns, in fact, exist extensively in practice. Meanwhile, we analyze the characteristics of these priors/patterns. It is assumed that the BS can monitor multiple mobile users within its coverage and record their optimal beams within a finite time-interval, which yields multiple BDTs. Typically, these different BDTs are obtained by letting key system parameters take different values. For example, each key parameter is obtained by sampling a reasonable probability distribution, which simulates and reflects

[2] The nonparametric priors and patterns describe a SP from the perspective of sample functions, i.e., to exploit the probabilistic properties of sample functions. Typical nonparametric priors and patterns, exploited in our paper, include stochastic continuity, n-th order mean-square differentiability, mean-square smoothness of the sample functions, and so on [28].

This article has been accepted for publication in IEEE Transactions on Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2023.3239927

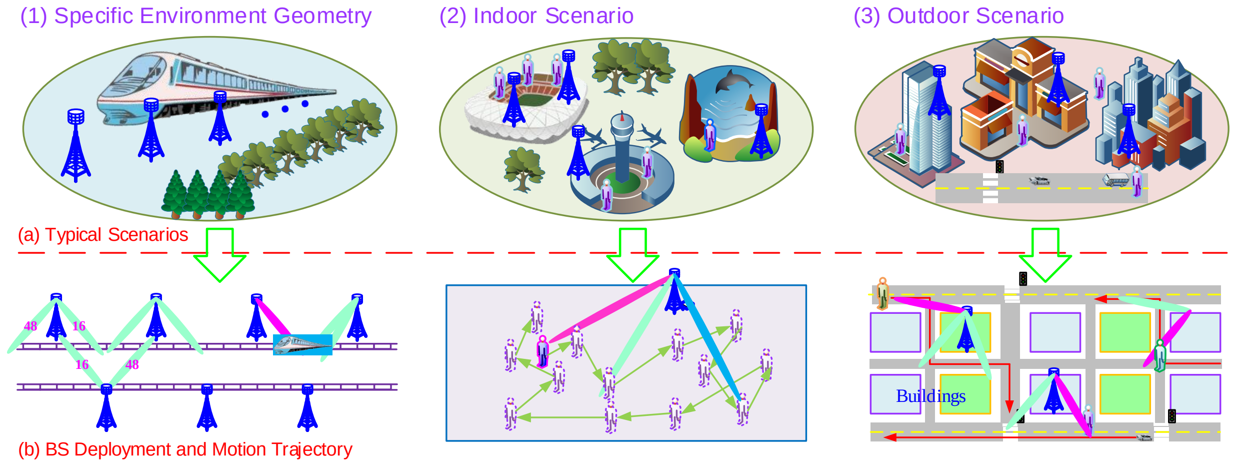

Fig. 2. An illustration of typical communication scenarios - red lines with arrows in part (b) represent motion trajectories.

practical environments as much as possible. The details of these parameters are intensively provided in Section VI.

1) Priors Induced by Environment Geometry: The priors or patterns are mainly induced by environment geometry, while the external environment may be static. This also applies to large outdoor cases, where the LOS path is often dominated. A typical example is high-speed train (HST) communication. As shown in Fig. 2-(1), because the BSs are deployed along the rails at regular distances, the trajectories of the trains are fixed, and the velocities of trains are within a reasonable range [22], [29], if multiple BDTs are observed across multiple BSs, these BDTs are almost always continuous and present an inexact periodicity, as shown in Fig. 3.

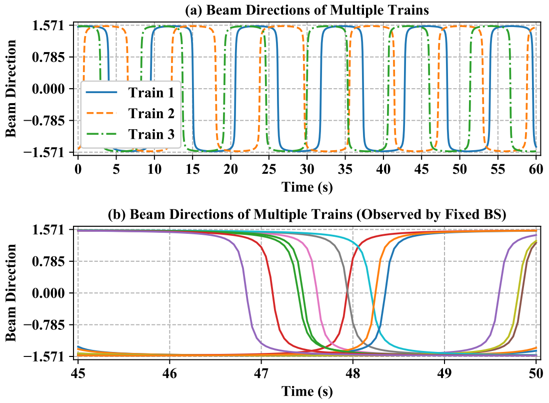

Fig. 3. Beam direction trajectories of high-speed trains.

2) Indoor Scenario: As shown in Fig. 2-(2), the user moves within a room randomly, but the geometry of the room may be static and therefore introduces correlation from one time-slot to the next. Although it is almost impossible to predict the next beam accurately due to the randomness, the difference between the beam directions at two adjacent time-slots is often small (See Fig. 4-(a)).

3) Outdoor Scenario: The outdoor physical environment can also introduce strong correlation between channel instantiations. As an example, a user moves along a street and may change to another street or occasionally stops for a moment (See Fig. 2-(3)). As shown in Fig. 4-(b), the beam directions are flat in some intervals, and may be discontinuous (typically, due to the blockage [3]).

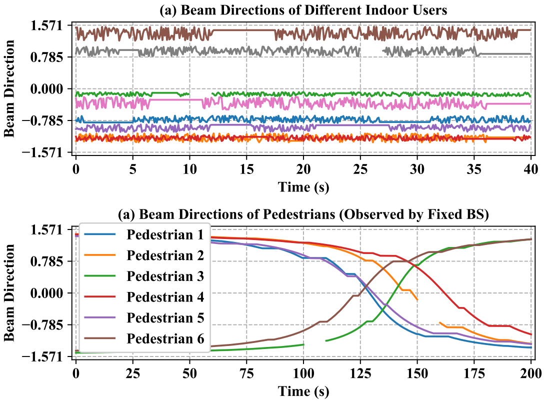

Fig. 4. Beam direction trajectories of indoor and outdoor environments.

To better analyze and characterize these priors/patterns, we propose to identify them from two perspectives, i.e., sample size and time-scale. From the perspective of sample size, we can distinguish and investigate the sample path property and statistical regularity property.

- Sample Path Property: It can be identified and reflected from a single BDT. Two typical sample path properties are beam-change continuity and bounded jump or variation. Since the environment changes continuously or the movement speed is limited or varies slowly, the BDT is a continuous function, although it may be fairly random. This sample path property is referred to as beam-change continuity. Similarly, due to a limited movement speed, the jump or variation of the BDT is bounded, although

[3] In this paper, we model and tackle blockage in an indirect way. Specifically, from the perspective of continuous-time function, a blockage event leads to a discontinuous (beam direction) function. An effective prediction model or algorithm should be able to adapt to the discontinuity and still provides an effective prediction (e.g., from the discontinuous location). Fortunately, when deep neural networks are incorporated into the GP kernel, the resultant GP model possesses the desired capability [30].

it can be very complex (e.g., the BDT is discontinuous). This property is referred to as bounded jump or variation.

- Statistical Regularity (SR) Property: The SR property is referred to as an ensemble property of a SP, which can be typically characterized via the mean function (i.e., the ensemble average), correlation function, power spectral density function, and so on. Since a single BDT is not representative in most cases (e.g., without the ergodicity assumption), we shall collect multiple realizations (i.e., multiple BDTs) of the SP, so as to investigate the SR properties. The existence of the SR property comes from the fact that a practical environment is relatively stable.

It is not difficult to understand that the BDTs of the above scenarios have the aforementioned properties. For example, a BDT of the HST communication has the property of beam-change continuity, and a BDT of the indoor environment has the property of bounded jump or variation. Moreover, if many BDTs are available, we can investigate the SR properties. The BS can obtain the BDTs required by randomly selecting a number of users and continuously recording their optimal beams for random durations. Our algorithm allows to collect the BDTs online. They can also be collected via simulations, if a simulation environment is available.

The second perspective is time-scale, namely, the two types of properties can be observed from two different time-scales. Specifically, since a SR property can only be identified from multiple BDTs or long-term observations, it, in fact, belongs to long-term information. In contrast, a sample path property can be reflected via a single BDT or short-term observations. As a result, it belongs to short-term information.

Unfortunately, although these priors/patterns are useful and should be exploited to enhance BTT, it is often difficult to incorporate them into existing BTT frameworks or algorithms. The reason for this is two-fold. First, these priors/patterns involve a considerable degree of randomness. Second, the (function) space underlying the priors/patterns is of infinite dimension. The root cause is that these priors/patterns are nonparametric, which cannot be characterized by parametric models. The two factors invalidate many deterministic and/or parametric prediction models/methods (e.g., the Kalman filter and even learning-based methods), and efficient solutions to extract and exploit these priors/patterns are still unavailable.

B. Beam Training and Tracking via Stochastic Process Modeling and Inference

To efficiently identify and exploit these priors/patterns, the key is to appropriately model and represent the BDTs. To this end, we choose the SP to model the BDTs. Let $(\Omega, \mathcal{E}, \mathbb{P})$ be a probability space, where Ω, $\mathcal{E}$ and $\mathbb{P}$ are sample space, σ-algebra on Ω and probability measure on $\mathcal{E}$, respectively. By a SP, one traditionally means a family of real random variables $\{\xi_t \,|\, t \in T\}$ ($\xi_t = \xi_t(\cdot)$) on $(\Omega, \mathcal{E}, \mathbb{P})$, with T being a set indexing ξ_t. It is assumed that each random variable $\xi_t(\omega)$ is defined for all ω. Then, for a fixed ω the values $\xi_t(\omega)$ define a function $\xi((\xi\omega)(t) = \xi_t(\omega), t \in T)$ in $\mathbb{R}^T$ and the $\mathcal{E}/\mathcal{B}$-measurability of each $\xi_t(\cdot)$ [4] implies the $\mathcal{E}/\mathcal{B}^T$-measurability of ξ, as will be shown in Theorem 1. Hence, the mapping ξ is a random element of $(\mathbb{R}^T, \mathcal{B}^T)$ and is termed a random function. As will be seen in Theorem 1, the converse also holds - if ξ is a measurable mapping from $(\Omega, \mathcal{E}, \mathbb{P})$ to $(\mathbb{R}^T, \mathcal{B}^T)$, the ω-functions $\xi_t(\omega) = (\xi\omega)(t)$ are $\mathcal{E}/\mathcal{B}$-measurable for each t, i.e., ξ_t are random variables. For a fixed ω, the function $(\xi\omega)(t)$, $(t \in T)$, is referred to as a sample function (or the sample path or realization) of the process.

Theorem 1. *For each $t \in T$, let $\xi_t = \xi_t(\omega)$ be a real function of $\omega \in \Omega$ and let ξ be the mapping from Ω to $\mathbb{R}^T$ defined as $\xi\omega = \{\xi_t(\omega) \,|\, t \in T\}$. Then, ξ_t is $\mathcal{E}/\mathcal{B}$-measurable for each $t \in T$ if and only if ξ is $\mathcal{E}/\mathcal{B}^T$-measurable.*

Proof: See Appendix A. ■

Remark 3.1 In fact, Theorem 1 indicates that the notions of SP (family of random variables) and random function are entirely equivalent. In practice, random variables are often used to describe a random phenomenon. Theorem 1 enables us to model and characterize a random phenomenon of infinite dimension from the view of random function. In particular, since the underlying function space is of infinite dimension, both the representation ability and model complexity of a SP model scale with sample size, which thus can overcome the shortcomings of parametric models.

For the considered BTT problem, beam directions within an interval T are described by a SP $B = \{b_t(\omega) \,|\, t \in T, \omega \in \Omega\}$ (referred to as a beam process), and, in particular, each BDT is denoted by $b(\omega)$. The properties of a BDT and the beam process itself (e.g., its probability distribution) correspond to the sample path property and SR property, respectively.

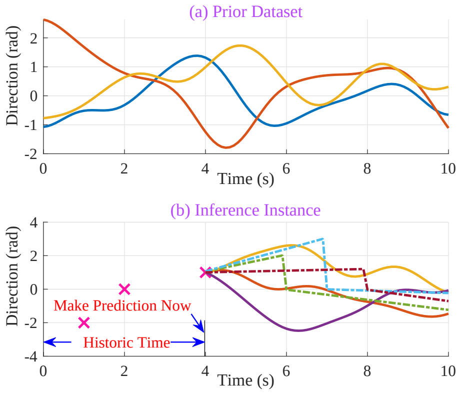

Fig. 5. A visualization of process inference that incorporates nonparametric priors and patterns (e.g., smoothness).

Having addressed the modeling issue via SP, we further tackle the inference issue. To obtain an intuitive understanding, we first use a simple inference example to qualitatively demonstrate the essence of incorporating the (nonparametric) priors/patterns into SP inference. As shown in Fig. 5, given a prior dataset (containing three sample functions) in Fig. 5-(a) and three data points (labeled by "+") in Fig. 5-(b), the nonparametric SP inference paradigm is based on the fact that

[4] The $\mathcal{E}/\mathcal{B}$-measurability of f means that f is a measurable mapping from $(\Omega, \mathcal{E})$ to $(\mathbb{R}, \mathcal{B})$, where $\mathcal{B}$ is the σ-algebra of $\mathbb{R}$.

This article has been accepted for publication in IEEE Transactions on Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2023.3239927

the prior sample functions are smooth and their variations are also small while the three dotted curves are non-smooth, and therefore infers that the three solid curves in Fig. 5-(b) have a higher probability of being the real sample function than the three dotted curves. Next, we elaborate on the quantitative inference by incorporating these nonparametric priors.

C. Beam Training and Tracking via GP - Quantitative Inference

To avoid intractability of quantitative inference caused by the SP modeling paradigm, GP is chosen in this paper. For completeness, a brief introduction of GP and GP kernel is provided in Appendix B. In general, temporal correlation or variation is very important to SP, especially for BTT. Because a GP is defined via the kernel (a second-order correlation function), mathematically the GP-based method can effectively characterize the temporal correlation or variation. A GP that describes beam directions is still denoted by $B = \{b_t(\omega)\}$.

The key of the GP modeling is that the priors/patterns are encoded into the GP kernel. For ease of understanding, the HST communication is taken as an example to demonstrate this procedure. In view that: 1) the HST communication has the property of beam-change continuity; and 2) the squared exponential kernel k_{SE} and spectral mixture kernel k_{SM} can respectively encode continuity [28] and latent patterns [31], the GP kernel for the HST communication scenario is chosen as a linear combination of the two kernels, i.e.,

$$k_{\text{B}}(t,t') = c_1 k_{\text{SM}}(t,t') + c_2 k_{\text{SE}}(t,t') + \sigma_w^2 \delta_{t,t'}, \tag{5}$$

where $c_1 > 0$ and $c_2 > 0$ are combination coefficients, $\delta_{t,t'}$ is the Kronecker delta (i.e., $\delta_{t,t'} = 1$ if and only if $t = t'$ and is zero otherwise), and σ_w^2 characterizes model uncertainty (e.g., due to quantization and/or randomness).

Let the beam direction at time $t_i\,(i = 1, \cdots, n)$ be b_{t_i}, which is denoted by b_i for convenience. The task is to predict the beam at t_{n+1} based on observations $\{b_1, b_2, \cdots, b_n\}$. Let $\mathcal{D} = \{(t_1, b_1), \cdots, (t_n, b_n)\}$ and $\mathbf{b} = [b_1, \cdots, b_n]^{\text{T}}$ denote the training dataset and discrete beam directions (i.e., beam indexes), respectively. Given the prior (i.e., kernel in (5)) and the training dataset $\mathcal{D}$, the Bayesian posterior distribution can be obtained immediately (from (29) - (31))

$$p(b_{t_{n+1}} \,|\, \mathcal{D}, t_{n+1}) \sim \mathcal{N}\big(\mu_{t_{n+1}}, \sigma_{t_{n+1}}^2\big) \tag{6}$$

$$\mu_{t_{n+1}} = \mathbf{k}_*^{\text{T}}(\mathbf{K} + \sigma_w^2\mathbf{I})^{-1}\mathbf{b} \tag{7}$$

$$\sigma_{t_{n+1}}^2 = k_{**} - \mathbf{k}_*^{\text{T}}(\mathbf{K} + \sigma_w^2\mathbf{I})^{-1}\mathbf{k}_*, \tag{8}$$

where the i-th element of $\mathbf{k}_*$ and the (i,j)-th element of $\mathbf{K}$ are calculated as $k_{\text{B}}(t_i, t_{n+1})$ and $k_{\text{B}}(t_i, t_j)$, respectively. k_{**} is calculated as $k_{\text{B}}(t_{n+1}, t_{n+1})$. Let $c > 0$ be a real number, which will be further optimized later (in Section V). For the moment, it is assumed to be a constant, e.g., $c = 2$ or 3 by following the convention of machine learning community. Given (6) - (8) and c, we sweep the beams within the beam confidence interval (BCI), i.e.,

$$\mathcal{S}_{c\sigma} = \lfloor \mu_{t_{n+1}} - c\sigma_{t_{n+1}}, \mu_{t_{n+1}} + c\sigma_{t_{n+1}} \rceil. \tag{9}$$

The method to determine BCI is referred to as $c\sigma$ criterion.

A remaining problem is to determine the hyper-parameters in the GP kernel (e.g., c_1 and c_2 in (5)), which is related to Bayesian model selection [28] and can be solved via the marginal likelihood $p(\mathbf{b} \,|\, \mathbf{t} = [t_1, \cdots, t_n])$ maximization. In practice, we often equivalently maximize the log marginal likelihood (LML) $\log p(\mathbf{b} \,|\, \mathbf{t})$, which is calculated as

$$\begin{aligned}\log p(\mathbf{b} \,|\, \mathbf{t}) = &-\frac{1}{2}\mathbf{b}^{\text{T}}(\mathbf{K} + \sigma_w^2\mathbf{I})^{-1}\mathbf{b} \\ &- \frac{1}{2}\log\det(\mathbf{K} + \sigma_w^2\mathbf{I}) - \frac{n}{2}\log 2\pi.\end{aligned} \tag{10}$$

Based on the above discussion, we propose the first BTT algorithm, which is summarized in Algorithm 1. The input of the algorithm is the dataset up to time-slot t_{n_0}. In step 3-(1), we first maximize the LML in (10) to determine the hyper-parameters. Then, we determine the BCI in steps 3-(2) and 3-(3). Next, we find out the optimal beam by sweeping all beams within the BCI. With the optimal beam available, we perform data transmission (in step 3-(5)). In step 3-(6), we enlarge the dataset. The method to find out the optimal beam for t_{n+2} is similar. As time t increases, more data points are accumulated, and better performance can be achieved.

Algorithm 1: Beam Training and Tracking via GP

```
1 input: initial dataset $\mathcal{D} = \{(t_1, b_1), \cdots, (t_{n_0}, b_{n_0})\}$
    and $c\sigma$ criterion
2 initialize: let counter of time-slot $n = n_0$
3 loop (for each time-slot $n$)
      (1) maximize LML in (10) to optimize hyper
              parameters in GP kernel
      (2) compute matrix/vector/scalar $\mathbf{K}$, $\mathbf{k}_*$ and $k_{**}$
      (3) determine BCI according to (6) - (9)
      (4) sweep all beams within BCI to find optimal
              beam $b_{n+1}^\star$
      (5) perform data transmission with beam $b_{n+1}^\star$
      (6) update $n$ and $\mathcal{D}$ as follows:
              $n \leftarrow n + 1$ and $\mathcal{D} \leftarrow \mathcal{D} \cup \{(t_n, b_n)\}$
  end-loop
```

Remark 3.2 An appealing advantage of the SP-based algorithm is that good performance can be achieved (e.g., low beam training overhead and large effective achievable rate) even with a small amount of training samples. Moreover, in contrast to most BTT algorithms that predict a single beam, the SP-based method can take environment priors/patterns and uncertainties into account and outputs a confidence interval, which thus greatly improves robustness to randomness.

In Algorithm 1, the GP kernel is chosen heuristically, which may cause some performance loss. Moreover, as t increases, $\mathcal{D}$ becomes larger and larger, which imposes a computational burden. Next, we address the two issues by proposing novel network structure and training or optimization methods.

IV. Efficient Network Architecture and Training Approach for GP Learning

In this section, we propose a novel network architecture and efficient training methods, which enable to efficiently exploit complex patterns and facilitate practical implementation.

A. *Network Architecture for Efficient GP Learning*

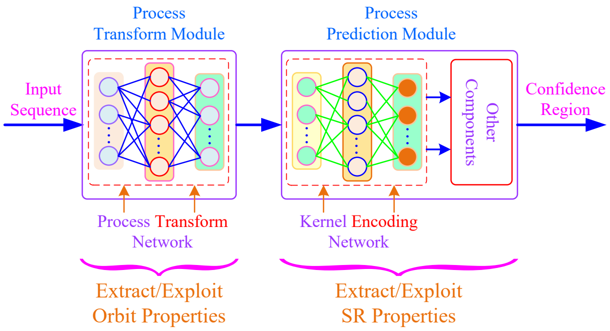

Fig. 6. Network architecture tailored for GP learning. It consists of two modules implemented via (deep) neural networks. KEN extracts and exploits SR properties or long-term information, while PTN extracts and exploits sample path properties or short-term information.

To improve the previous design, we integrate (deep) neural networks into GP. [5] Specifically, we introduce two modules (i.e., PTN and KEN) which are mainly implemented by two (deep) neural networks, as shown in Fig. 6. The individual roles of the two modules are as follows:

- Kernel encoding network (KEN) is introduced to provide required or desired kernel for the GP model, which can be obtained by enhancing an existing simple kernel via neural network. The key role of KEN is to encode the underlying SR properties of a beam process.
- Process transform network (PTN) is designed to exploit the sample path properties. Via the nonlinear transform of PTN, the transformed feature sufficiently matches the KEN-based prediction model. Moreover, PTN helps to avoid dramatic performance degradation in unexpected cases (e.g., without an accurate prior).

To bring the advantages of the network architecture into full play, it is vital to tailor an efficient training method as per the situation in which it will be utilized.

B. *Efficient Training Methods*

According to whether multiple sample functions are available and to what extent different BS environments share the similarity, we consider three different training or optimization methods, so as to efficiently extract and exploit the priors or patterns and facilitate system implementation.

1) Semi-Offline Training: If an empirical dataset (denoted by $\{b(\omega_1), b(\omega_2), \cdots, b(\omega_n)\}$ [6]) is available and the different environments share a large similarity (e.g., a BS records historical data and optimizes the learning model to improve future performance within its coverage area), an appropriate method to optimize the model is semi-offline training.

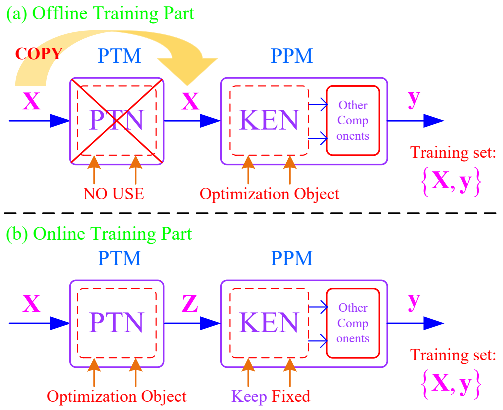

Fig. 7. An illustration of the semi-offline training method.

As shown in Fig. 7, the semi-offline training method consists of two phases (i.e., offline phase and online phase), which are in charge of optimizing KEN and PTN, respectively. The key of semi-offline method is as follows. First, KEN is optimized offline, while PTN is optimized online or when the model is put into use (e.g., PTN is optimized every K time-slots). Second, when optimizing KEN, PTN is dropped out (i.e., the training dataset used to optimize KEN is the input of PTN, not its output), as shown in Fig. 7-(a). Finally, the parameters of KEN keep fixed while optimizing PTN (See Fig. 7-(b)).

The motivation and rationality for the structure and training method may be explained as follows. On the one hand, the GP kernel, defined as $k(\mathbf{x}, \mathbf{x}')$, characterizes the statistical property of the underlying process. Since the input of the GP kernel is feature $\mathbf{x}$, it is reasonable to drop or ignore PTN when training KEN, so as to avoid possible interfering effect from PTN. Beside, the method to train the KEN-based GP model (without PTN) via maximizing LML can learn desired properties from the training dataset which well represents the beam process. On the other hand, each BDT is a random function, which may contain unexpected randomness, especially in a non-stationary environment. Therefore, when the trained KEN is put into use, the PTN (added before KEN) can tackle unpredictable randomness and enhance model robustness.

2) Fully-Online Training: When the learning model is put into an entirely new environment or the ambient environment changes frequently (and, as a result, different channel conditions share few similarity), the beam direction function at different time-blocks varies dramatically. In this case, the fully-online training method is more appropriate. Specifically, PTN and KEN are optimized simultaneously or synchronously, e.g., they are optimized every K time-slots.

3) Semi-Online Training: If the learning model is still put into an entirely new environment, but the ambient environments share relatively large similarity [7], and thus the beam direction function at different time-blocks has significant

[5] The reason why the neural network is chosen here is as follows. First, thanks to the powerful ability in terms of generalization and fitting, it can approximate an arbitrary meaningful function. Second, the bivariate function constructed here is a valid GP kernel [28], [31]. Third, many efficient optimizers (e.g., Adam) have been developed to train the learning model. In particular, sophisticated deep learning libraries (e.g., Tensorflow and Tytorch) are available, which facilitate fast validation and implementation. If other models satisfy these conditions, they may also constitute an effective solution.

[6] In practice (e.g., in Section VI), we can only obtain and use a finite number of discrete points of each sample function $b(\omega_i)$.

[7] For example, in the HST communication scenario, beam direction functions at adjacent BSs often share large similarity (See Fig. 3).

SR property. In this case, an efficient optimization method should adequately account for both long-term and short-term characteristics, which is, however, non-trivial. To tackle this issue, we propose the semi-online learning method.

The key of semi-online training includes two points, i.e., to differentiate training time-scales (i.e., the two networks are not trained simultaneously) and method to use training samples.

- Training time-scales: For clarity, the time-scales of optimizing the two modules are illustrated in Fig. 8. 1) KEN training: The time-scale of optimizing KEN is relatively large, so as to accumulate enough data samples to extract the SR property. 2) PTN training: PTN is in charge of processing short-term information (e.g., to exploit the sample path property). Hence, less samples are required, and the time-scale of optimizing PTN is small.
- Method of using training samples: It is assumed that each training round consists of K time-slots. The training scheme is as follows. 1) PTN training: At the end of each training round, PTN is trained based on the most recent L data samples. Note that similar to Fig. 7-(b), the parameters of KEN remain fixed while optimizing PTN. 2) KEN training: At the end of M training rounds, KEN is optimized with MK samples collected within the past M rounds. Similar to the semi-offline method, PTN is dropped out when optimizing KEN.

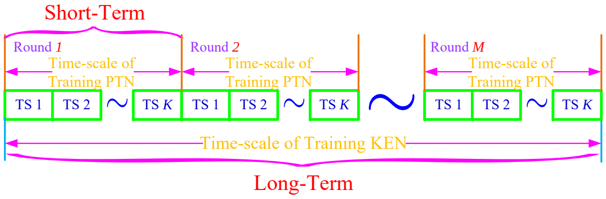
Short-Term
Round 1
Time-scale of Training PTN
Round 2
Time-scale of Training PTN
Round M
Time-scale of Training PTN
TS 1
TS 2
TS K
TS 1
TS 2
TS K
TS 1
TS 2
TS K
Time-scale of Training KEN
Long-Term

Fig. 8. Time-scales of optimizing the two networks. TS is short for time-slot.

Remark 4.1 It is important to separately train PTN and KEN. First, the decoupled training scheme greatly reduces computational complexity of both learning and prediction. Second, it ensures that each network/module can effectively extract its required information. Particularly, both long-term and short-term information can be separated out and processed more efficiently. Third, the prior can be encoded into KEN effectively. Finally, for all above training methods, both model optimization and beam prediction are done in the phase of data transmission, which do not occupy the precious communication or data transmission time.

C. Beam Training and Tracking Design - An Example for the General Case

Without loss of generality, we choose the fully-connected network and semi-online training method as an example to present a detailed algorithm. The input and output of PTN are (relative) time and transformed feature, respectively. Let $\boldsymbol{\theta}_{\mathrm{P}}$ denote the parameters (e.g., weights and biases) of PTN. The input-output relationship of PTN can be written as

$$z = h(t, \boldsymbol{\theta}_{\mathrm{P}}). \tag{11}$$

Similarly, the input-output relationship of KEN is denoted by $g(\cdot, \boldsymbol{\theta}_{\mathrm{K}})$, with parameters $\boldsymbol{\theta}_{\mathrm{K}}$. The input and output of the neural network within KEN are the transformed feature of PTN and transformed feature fed to the base kernel of the GP, respectively. Let $k_{\mathrm{B}}(\cdot,\cdot\,|\,\boldsymbol{\theta}_{\mathrm{B}})$ with parameters $\boldsymbol{\theta}_{\mathrm{B}}$ denote the base kernel of the GP (e.g., squared exponential kernel). The overall kernel is given by

$$k_{\mathrm{E}}(z_i, z_j) = k_{\mathrm{B}}(g(z_i, \boldsymbol{\theta}_{\mathrm{K}}), g(z_j, \boldsymbol{\theta}_{\mathrm{K}})\,|\,\boldsymbol{\theta}_{\mathrm{B}}). \tag{12}$$

Next, we detail the training of PTN and KEN. The loss function for GP learning is negative LML, i.e., to minimize $-\log p(\mathbf{b}\,|\,\mathbf{t})$ (or equivalently maximize $\log p(\mathbf{b}\,|\,\mathbf{t})$). Specifically, given a training dataset $\mathcal{D} = \{(t_1, b_1), \cdots, (t_n, b_n)\}$, KEN is optimized by maximizing

$$\begin{aligned} &L(\boldsymbol{\theta}_{\mathrm{K}}, \boldsymbol{\theta}_{\mathrm{B}}) \\ &= -\frac{1}{2}\mathbf{b}^{\mathrm{T}}(\mathbf{K}_{\mathrm{E}} + \sigma_w^2\mathbf{I})^{-1}\mathbf{b} - \frac{1}{2}\log\det(\mathbf{K}_{\mathrm{E}} + \sigma_w^2\mathbf{I}). \end{aligned} \tag{13}$$

Given $\mathcal{D}$, the (i,j)-th element of $\mathbf{K}_{\mathrm{E}}$ is calculated as

$$k_{ij} = k_{\mathrm{B}}\big(g(t_i, \boldsymbol{\theta}_{\mathrm{K}}), g(t_j, \boldsymbol{\theta}_{\mathrm{K}})\,|\,\boldsymbol{\theta}_{\mathrm{B}}\big).$$

To optimize $L(\boldsymbol{\theta}_{\mathrm{K}}, \boldsymbol{\theta}_{\mathrm{B}})$, the back-propagation method can be invoked. In fact, the derivatives of $L(\boldsymbol{\theta}_{\mathrm{K}}, \boldsymbol{\theta}_{\mathrm{B}})$ with respect to $\boldsymbol{\theta}_{\mathrm{K}}$ and $\boldsymbol{\theta}_{\mathrm{B}}$ can be computed by using the chain rules.

Given a training dataset $\mathcal{D}' = \{(t'_1, b'_1), \cdots, (t'_m, b'_m)\}$, PTN can be trained similarly, i.e., to maximize the following LML

$$L(\boldsymbol{\theta}_{\mathrm{P}}) = -\frac{1}{2}\mathbf{b}^{\mathrm{T}}(\mathbf{K}_{\mathrm{P}} + \sigma_w^2\mathbf{I})^{-1}\mathbf{b} - \frac{1}{2}\log\det(\mathbf{K}_{\mathrm{P}} + \sigma_w^2\mathbf{I}),$$

where matrix $\mathbf{K}_{\mathrm{P}}$ is calculated as

$$k_{ij} = k_{\mathrm{B}}\Big(g\big(h(t_i, \boldsymbol{\theta}_{\mathrm{P}}), \boldsymbol{\theta}_{\mathrm{K}}\big), g\big(h(t_j, \boldsymbol{\theta}_{\mathrm{P}}), \boldsymbol{\theta}_{\mathrm{K}}\big)\,\Big|\,\boldsymbol{\theta}_{\mathrm{B}}\Big).$$

Note that the parameters of KEN (i.e., $\boldsymbol{\theta}_{\mathrm{B}}$ and $\boldsymbol{\theta}_{\mathrm{K}}$) should be fixed when computing $\mathbf{K}_{\mathrm{P}}$ and optimizing $\boldsymbol{\theta}_{\mathrm{P}}$.

The designed BTT algorithm is summarized in Algorithm 2 for clarity. [8] K and M are provided in advance, which depend on a specific scenario. Before running the algorithm, the parameters of the two neural networks are initialized, e.g., randomly. In each time-slot, we first compute the kernel (matrix) and determine the hyper-parameters by maximizing LML (in step 3-(1)). In steps 3-(2) - 3-(3), we perform GP prediction and obtain a BCI. In step 3-(4), we find out the optimal beam by sweeping all beams within the BCI. With the optimal beam available, data can be transmitted within the remaining time of the time-slot in step 3-(5). In step 3-(6), we update the dataset. We check the condition and update the parameters of the networks if the condition is satisfied.

[8] Although the concept of continuous beam direction is utilized previously, when it comes to numerical computation in practice, we only need quantized beam directions at discrete-time points. Since our algorithm can explicitly model various uncertainties via GP kernel design (e.g., (5)) and can predict an interval, it is robust to various uncertainties (e.g., quantization). Simulation results show that the performance loss due to quantization and time discretization can be safely ignored (e.g., when $N \geq 64$).

```
Algorithm 2: Beam Training and Tracking via GP
1 input: time-scales $K$ (short-term) and $M$ (long-term);
      initial dataset $\mathcal{D} = \{(t_1, b_1), \cdots, (t_{n_0}, b_{n_0})\}$; and
      $c\sigma$ criterion
2 initialize: hyper-parameters and weights of KEN and
      PTN; let time-slot counter $n = n_0$
3 loop (for each time-slot $n$)
      (1) maximize LML in (10) to optimize hyper
            parameters in GP kernel
      (2) compute matrix/vector/scalar $\mathbf{K}$, $\mathbf{k}_*$ and $k_{**}$
      (3) determine BCI according to (6) - (9)
      (4) sweep all beams within BCI to find optimal
            beam $b_{n+1}^\star$
      (5) perform data transmission with beam $b_{n+1}^\star$
      (6) update $n$ and $\mathcal{D}$ as follows:
            $n \leftarrow n+1$ and $\mathcal{D} \leftarrow \mathcal{D} \cup \{(t_n, b_n)\}$
      (7) update parameters of two neural networks
            if $n \bmod K = 0 \Longrightarrow$ update PTN
            if $n \bmod (KM) = 0 \Longrightarrow$ update KEN
  end-loop
```

V. Performance Tradeoff via Bayesian Posterior Inference

In the previous algorithms, we choose parameter c in the $c\sigma$ criterion heuristically. In fact, c balances probability of successful alignment (PSA) and other performance metric of interest. We take the performance metric "effective achievable rate" (EAR) as an example to illustrate this point. In general, as parameter c becomes large, PSA increases monotonously. On the one hand, if $c > 0$ is too small, the BCI constructed often fails to contain the real beam direction, i.e., PSA is very small. In this case, $P|\mathbf{h}^{\mathrm{H}}\mathbf{f}_i|^2$ in (4) is also very small, which thus leads to small EAR. On the other hand, if c is too large, the resultant large beam training overhead will also reduce EAR, although PSA is almost 1 in this case. In summary, EAR first increases and then decreases, as c increases. Next, we investigate how to achieve a desired tradeoff between the two performance metrics, i.e., EAR and PSA.

Given a training dataset $\mathcal{D}_n = \{(t_1, b_1), \cdots, (t_n, b_n)\}$, we can obtain the posterior distribution of $b_{t_{n+1}}$, i.e.,

$$p(b_{t_{n+1}} \,|\, \mathcal{D}_n, t_{n+1}) \sim \mathcal{N}(\mu_{n+1}, \sigma_{n+1}^2), \tag{14}$$

where μ_{n+1} and σ_{n+1}^2 are given in (7) - (8). From Bayesian perspective, with the Bayesian posterior distribution available, meaningful statistical inference can be made. In particular, the (posterior) expected effective achievable rate (EEAR) can be characterized explicitly. To simplify performance analysis and reveal useful insights, we first consider the single-path case, i.e., $L = 1$ in (2), and later discuss the more general multi-path case. Note that the specific single-path case itself is also very important, e.g., when the LOS link is dominated [32]. The (posterior) PSA performance is concluded below.

Lemma 1. *Let $\mathcal{I}(c) = \lfloor \mu - c\sigma, \mu + c\sigma \rceil$ be the interval consisting of swept beams. For the $c\sigma$ criterion, PSA, denoted by $p_{succ}(c)$, can be respectively upper bounded and lower bounded by*

$$p_{succ}(c, \alpha) \le p_0(c) \tag{15}$$

$$p_{succ}(c, \alpha) \ge p_0(c) \max\left\{0, \left(1 - 0.5|\mathcal{I}(c)| e^{-\frac{1}{4}\lambda^\star}\right)\right\}, \tag{16}$$

where $p_0(c)$ and $\lambda^\star$ are respectively given by

$$\begin{aligned} p_0(c) &= \int_{-c}^{c} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx, \\ \lambda^\star &= 2L_p P N |\alpha_0|^2 \beta^{-1}. \end{aligned} \tag{17}$$

Proof: See Appendix C. ■

As $\lambda^\star$ increases, $\exp(-\lambda^\star/4)$ tends to zero dramatically. Hence, for a sufficiently large $\lambda^\star$ (e.g., the transmit power P is large), $p_{\mathrm{succ}}(c)$ can be safely approximated by $p_0(c)$, i.e.,

$$p_{\mathrm{succ}}(c) \approx p_0(c).$$

We further proceed to characterize the (posterior) EEAR performance. To accommodate various channel fadings (e.g., Rayleigh or Rician), the path gain α is assumed to follow the Nakagami fading. Then, $|\alpha|^2$ follows a Gamma distribution having probability density function

$$f_{|\alpha|^2}(x) = \frac{m^m x^{m-1}}{\Gamma(m)} \exp(-mx), \quad x \ge 0, \tag{18}$$

with shape m and scale $1/m$, where m denotes the Nakagami fading parameter. The following theorem characterizes the (posterior) EEAR performance.

Theorem 2. *With the assumption that α follows the Nakagami fading, the upper bound and lower bound of (posterior) EEAR $E(m, c)$ are provided in* (19) *and* (20)*, respectively.*

Proof: It can be verified that EEAR conditioned on the path gain α can be calculated as

$$r(\alpha) = \left(1 - (|\mathcal{I}(c)| + 1) T_0 T_{\mathrm{S}}^{-1}\right) \log\left(1 + PN|\alpha|^2 \beta^{-1}\right).$$

Then, the upper bound of EEAR is given by

$$E(m, c) \le p_0(c) \int_0^\infty r(x) f_{|\alpha|^2}(x) dx.$$

The lower bound can be obtained similarly. ■

For a sufficiently large $\lambda^\star$, $p_{\mathrm{succ}}(c) \approx p_0(c)$ holds true. In this case, the upper bound and lower bound of the (posterior) EEAR approximately coincide, i.e.,

$$E(m, c) \approx p_0(c)\left(1 - (|\mathcal{I}(c)| + 1) T_0 / T_{\mathrm{S}}\right) C(P, \beta). \tag{21}$$

where $C(P, \beta)$ is independent of P and β and is given by

$$C(P, \beta) = \int_0^\infty \log\left(1 + \frac{PNx}{\beta}\right) \frac{m^m x^{m-1}}{\Gamma(m) \exp(mx)} dx.$$

Moreover, for a sufficiently large m (e.g., the LOS channel), $E(m, c)$ asymptotically satisfies

$$\begin{aligned} E(c) &= \lim_{m \to \infty} E(m, c) \\ &\approx p_0(c)\left(1 - (|\mathcal{I}(c)| + 1) T_0 / T_{\mathrm{S}}\right) \cdot \log\left(1 + PN/\beta\right). \end{aligned}$$

In general, the widely used 3σ criterion is too conservative, which may cause large beam training overhead and thus lower

$$E(m,c) \leq p_0(c)\big(1-(|\mathcal{I}(c)|+1)T_0/T_{\rm S}\big)\cdot\int_0^{\infty}\log\left(1+\frac{PNx}{\beta}\right)\frac{m^m x^{m-1}}{\Gamma(m)\exp(mx)}dx. \tag{19}$$

$$E(m,c) \geq p_0(c)\big(1-(|\mathcal{I}(c)|+1)T_0/T_{\rm S}\big)\cdot\int_0^{\infty}\log\left(1+\frac{PNx}{\beta}\right)\frac{m^m x^{m-1}}{\Gamma(m)\exp(mx)}\cdot\max\left\{0,\big(1-0.5|\mathcal{I}(c)|e^{-\frac{1}{4}\lambda^\star}\big)\right\}dx. \tag{20}$$

EEAR. Based on the above discussion, we can derive an "optimal" $c\sigma$ criterion to maximize EEAR. To this end, we shall solve the following problem

$$\begin{aligned}&\max_{c}\ E(m,c)\\ &\text{s.t.}\quad c\geq 0.\end{aligned} \tag{22}$$

It is observed that problem (22) is equivalent to the following optimization problem

$$\begin{aligned}&\max_{c}\ \left(1-\frac{(|\mathcal{I}(c)|+1)T_0}{T_{\rm S}}\right)\int_0^c \exp\left(-\frac{x^2}{2}\right)dx\\ &\text{s.t.}\quad c\geq 0.\end{aligned} \tag{23}$$

An analytical solution of problem (23) is unavailable. Fortunately, the objective function in (23) is unimodal and has good properties, as concluded in the following theorem.

Theorem 3. *Let*

$$f(c)=\left(1-\frac{(2c\sigma+1)T_0}{T_S}\right)\int_0^c\exp\left(-\frac{x^2}{2}\right)dx. \tag{24}$$

Then $f(c)$ is a unimodal function and has a unique maximal point within $(0, 0.5T_S T_0^{-1}\sigma^{-1}-0.5\sigma^{-1})$.

Proof: See Appendix D. ■

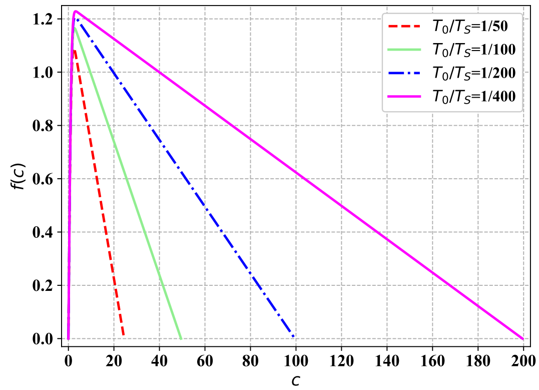

Fig. 9. The graph of function $f(c)$ for different settings of T_0/T_S ($\sigma=1$).

The graph of $f(c)$ is demonstrated in Fig. 9, which clearly illustrates the variation tendency of $f(c)$. As shown in Fig. 9 and proved in Theorem 3, $f(c)$ first increases monotonously and then decreases monotonously, and thus contains only one maximum. Based on Theorem 3, classical derivative-free search methods (e.g., binary search or golden search [33]) can be used to find the optimal value $c^\star$. Here, the golden search is used to find $c^\star$, which is summarized in Algorithm 3 for clarity. The golden ratio, i.e., $r=(\sqrt{5}-1)/2$, is chosen to shorten the search interval in a constant ratio [33]. With $c^\star$ available, the beam search subspace is refined as

$$\mathcal{I}(c)=\lfloor\mu-c^\star\sigma,\mu+c^\star\sigma\rceil. \tag{25}$$

Algorithm 3: Optimal Beam Subspace Search Algorithm

```
input: variance of posterior prediction distribution $\sigma^2$;
    duration of one time-slot $T_{\rm S}$; duration of transmitting
    one beam $T_{\rm B}$; tolerance $\delta>0$; let $r=(\sqrt{5}-1)/2$
initialize: $a=0$, $b=0.5\sigma^{-1}(T_{\rm S}T_0^{-1}-1)$, $y_{\rm p}=f(p)$
    $p=a+(1-r)(b-a)$, $q=a+r(b-a)$, $y_{\rm q}=f(q)$
loop
    if $y_{\rm p}<y_{\rm q}$
        $b\leftarrow q$, $q\leftarrow p$, $y_{\rm q}\leftarrow y_{\rm p}$, $p\leftarrow a+(1-r)(b-a)$
        $y_{\rm p}=f(p)$
    else
        $a\leftarrow p$, $p\leftarrow q$, $y_{\rm p}\leftarrow f(q)$, $q\leftarrow a+r(b-a)$
        $y\leftarrow f(q)$
    end
  until $|b-a|<\delta$ and $|y_{\rm p}-y_{\rm q}|<\delta$
output: optimal criterion $c^\star=(a+b)/2$
    beam search subspace $\lfloor\mu-c^\star\sigma,\mu+c^\star\sigma\rceil$
```

Finally, we extend the analysis for the single-path case to the multi-path case. The (posterior) probability $p_{\rm succ}(c)$ can be calculated as (See the proof of Lemma 1 for details)

$$\begin{aligned}p_{\rm succ}(c)&=\mathbb{P}\big(\mathbf{f}_{i^\star}\text{ is found}\,|\,i^\star\in\mathcal{I}(c)\big)\mathbb{P}\big(i^\star\in\mathcal{I}(c)\big)\\&=\mathbb{P}\big(\mathbf{f}_{i^\star}\text{ is found}\,|\,i^\star\in\mathcal{I}(c)\big)\cdot p_0(c),\end{aligned}$$

where $i^\star$ denotes the index of optimal beam. The key operation of the $c\sigma$ criterion is to sweep all beams within $\mathcal{I}(c)$. Under normal conditions, $\mathbb{P}\big(\mathbf{f}_{i^\star}\text{ is found}\,|\,i^\star\in\mathcal{I}(c)\big)\approx 1$ also holds for the multi-path case. The practical meaning is that if the optimal beam is known to be within an interval, it can be found out via sweeping the interval, which is, in fact, the foundation of sweeping-based beam alignment techniques.

Thanks to the approximation, the posterior EEAR, denoted by $\bar{R}_{\rm eff}$, can be approximated as

$$\bar{R}_{\rm eff}\approx p_0(c)\cdot\big(1-(|\mathcal{I}(c)|+1)T_0/T_{\rm S}\big)\cdot\mathbb{E}(R(\{\alpha_l,\phi_l\})), \tag{26}$$

where $\mathbb{E}(R(\{\alpha_l,\phi_l\}))$ denotes the achievable rate (conditioned on multiple factors). It is observed from (26) that although the analytic expression of $\mathbb{E}(R(\{\alpha_l,\phi_l\}))$ is unavailable, it does not affect the optimization of $\bar{R}_{\rm eff}$, as the optimization variable is c. To maximize $\bar{R}_{\rm eff}$, it is sufficient to maximize $p_0(c)\cdot\big(1-(|\mathcal{I}(c)|+1)T_0/T_{\rm S}\big)$, which coincides with the single-path case. Therefore, Algorithm 3 (derived for the single-path case) is also applicable to the multi-path case.

VI. Numerical Results

In this section, simulation results are provided to demonstrate the performance of the proposed algorithms. We compare our approach, i.e., GP learning (GPL) based algorithms, to the classical and state-of-the-art benchmarks, including the conventional direct search (DS) algorithm, two machine learning (ML) based algorithms, i.e., the direct upper confidence bound (DUCB) based algorithm [34] and the stochastic bandit learning (SBL) based algorithm [21]) and the oracle aided algorithm. [9] For convenience, a uniform distribution taking values in $[a, b]$ is denoted by $\mathrm{U}(a, b)$.

A. Simulation Environment

The datasets used to train prediction models are generated via software simulation or hardware device. For the software simulation part, the uniform linear array is chosen here. The size of the codebook $\mathcal{C}$ satisfies $M = N$, and two cases (i.e., $N = 64$ and $N = 128$) are considered. For all simulation experiments, the channel model in (2) includes one LOS path and three NLOS paths. The angles of departure of the NLOS paths are distributed as $U[0, 2\pi]$. 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 $\mathcal{CN}(0, \sigma_{\mathrm{N}}^2)$. The other system parameters are described below.

1) High-speed Train Communication: We refer to [22] to set relevant system parameters, while [22] is further referred to the 3GPP mmwave high-speed train scenario. In contrast to [22], where the relevant parameters take fixed values, a more practical setting is considered, so as to accommodate possible errors. Specifically, the distance between two adjacent BSs is distributed as $\mathrm{U}(480, 520)$ (m) and the distance between each BS and the side track is distributed as $\mathrm{U}(4.5, 5.5)$ (m). The velocity of the train is distributed as $\mathrm{U}(280, 380)$ (km/h). The length of each time-slot is set to 10 ms.

2) Outdoor Environment: The coverage radius of the BS is fixed to 400 (m), and the height of the BS is set to 20 (m). The distance between the road and cell center is set to 10 (m). A pedestrian walks along the road, with speed distributed as $\mathrm{U}(3.6, 5.4)$ (km/h). The pedestrian may randomly stop for a moment, and the stopping time is distributed as $\mathrm{U}(1.0, 11.0)$ (s). The blockage phenomena randomly occur with probability 0.02, i.e., the probability that jumps or discontinuous points occur is 0.02. The length of each time-slot is set to 50 ms.

3) Indoor Environment: Instead of software simulation, the dataset used to optimize the prediction model is generated via the hardware. Specifically, an indoor test room is considered, whose size is about 6×10 (m^2). The transmit antenna array (a planar array of size 8×8) is fixed on a shelf, whose height is about 2.2 (m). The users in the room move randomly, with speeds varying within about $(0.2, 2.0)$ (m/s). The duration of each time-slot is set to 100 ms. Two GP models are constructed to predict the elevation angle and azimuth angle.

For the GPL-based BTT algorithms, the neural network of PTN consists of 3 layers (with 16, 4 and 1 neurons), while the neural network of KEN consists of 4 layers (with 16, 8, 4 and 2 neurons). For both PTN and KEN, the ReLU function is chosen as the activation function. All BDTs (including the test BDTs) are obtained by first sampling the probability distributions of key system parameters and then simulating the beam trajectories based on the sampled parameters. The Adam optimizer is chosen to train the model, with learning rate 0.002. The number of training epochs is 32. The training procedure is stopped if: 1) average training loss (calculated over 64 samples) is less than -2.0; or 2) the difference between average losses in two adjacent iterations is less than 0.02.

B. Performance of Beam Training and Tracking

Next, we evaluate the performance of the proposed BTT algorithms. PSA and average effective achievable rate (AEAR) are chosen as performance metrics to evaluate different algorithms. Two settings of LOS paths are considered: (1) the LOS path with a constant path gain, i.e., $|\alpha_{\mathrm{LOS}}| = 1$; and (2) the LOS path with a fading path gain, i.e., $\alpha_{\mathrm{LOS}} \sim \mathcal{CN}(\bar{m}, 1)$ with $|\bar{m}| = 1$. If not explicitly stated, GPL is referred to as Algorithm 2. It is known that the size of the training dataset affects the performance of most ML algorithms. Hence, we first evaluate the small sample performance of the three ML based algorithms, i.e., DUCB, SBL and GPL. Fig. 10 and Fig. 11 demonstrate the PSA and AEAR performance vs. the size of available training set T, respectively.

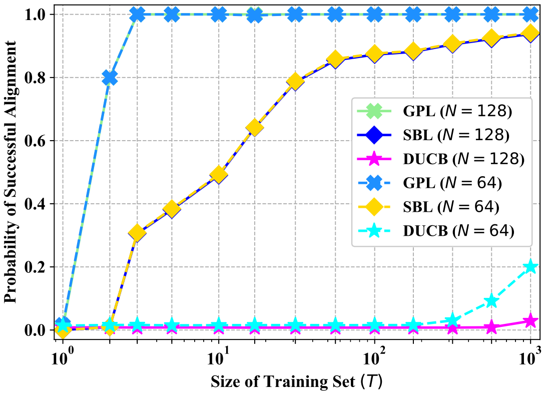

Fig. 10. The PSA performance of different algorithms - semi-offline training, $\mathrm{SNR} = 5\mathrm{dB}$ and $|\alpha_{\mathrm{L}}| = 1$ (Scenario 2).

It is observed from the two figures that DUCB achieves the worst performance (in terms of both PSA and AEAR) among the three BTT algorithms. The reason for this is that DUCB is mainly applicable to quasi-static mmwave channels, while the considered channels vary rapidly. We can also observe that GPL achieves the best PSA and AEAR performance and its AEAR performance approaches that of the ideal oracle aided algorithm which can access the optimal beams without any training overhead. More importantly, it can be observed that good performance can be achieved even when the number of available training samples is as low as 5. The small sample performance of GPL is appealing in wireless communications,

[9] The direct search (DS) algorithm searches the optimal beam within the codebook exhaustively. Note that the oracle aided algorithm is only served as a benchmark, which can always find out the optimal beam without causing any beam training overhead, i.e., $T_{\mathrm{B}} = 0$.

since they typically vary rapidly. Although SBL performs well when T is large, its performance degenerates quickly as the size of available training dataset decreases.

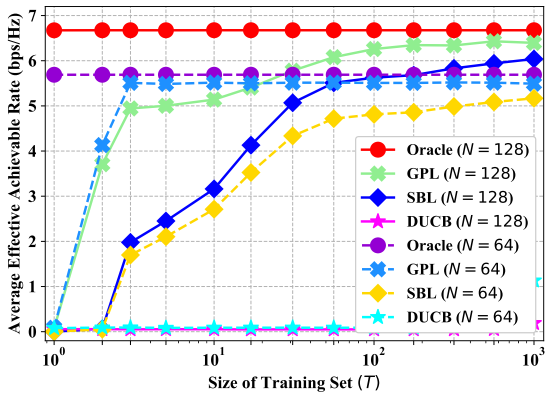

Fig. 11. The AEAR performance of different algorithms - semi-offline training, SNR $= 5$dB and $|\alpha_{\mathrm{L}}| = 1$ (Scenario 2).

The reason why the proposed BTT algorithm achieves good small sample performance is three-fold. First, in contrast to most prediction models (e.g., the neural network prediction model), the GP learning based model is a probabilistic model, which equivalently fits an infinite number of regression models and reasonably allocates the probabilities among these models according to the observed samples. Second, the environment priors and uncertainties are implicitly considered and encoded into the GP kernel, which further improves the performance. Finally, most of the existing prediction algorithms only output a point estimation, while our algorithm generates an interval estimation, which thus enhances the robustness.

It is observed that for the two cases $N = 64$ and $N = 128$, although GPL achieves almost the same PSA performance, as T increases, the AEAR metric corresponding to $N = 64$ is first better than that of $N = 128$ and then worse than it. The reason is as follows. For a BCI of same size, more beams are needed to sweep the BCI for a codebook of narrower beams. When T is small, the BCI is relatively large, which leads to large beam training overhead and thus degenerates AEAR. When T is sufficiently large, the BCI is narrow and thus the difference in terms of sweeping overhead is small. But since the array gain corresponding to $N = 128$ is larger than that of $N = 64$, the former achieves better AEAR performance.

To better evaluate the GPL approach, the PSA and AEAR performance of different algorithms for fading LOS paths is provided in Fig. 12 and 13, respectively. It is not surprising that DUCB achieves the worst performance in terms of both PSA and AEAR. Thanks to the good small sample performance and robustness, it is expected that GPL outperforms SBL, in terms of both PSA and AEAR. Not surprisingly, all algorithms perform better and better, as the SNR increases.

It is interesting to see that both GPL and SBL outperform DS in terms of PSA. The reason for this is that DS is more vulnerable to noise, especially for low SNR or low path gain. We explain this phenomenon mathematically. Let the beams used for sweeping be $\mathcal{T}$, which is a subset of $\mathcal{C}$. Let $\mathbf{f}_{i^\star}$ be the optimal beam. Then, $\mathbf{f}_{i^\star}$ can be identified if and only if

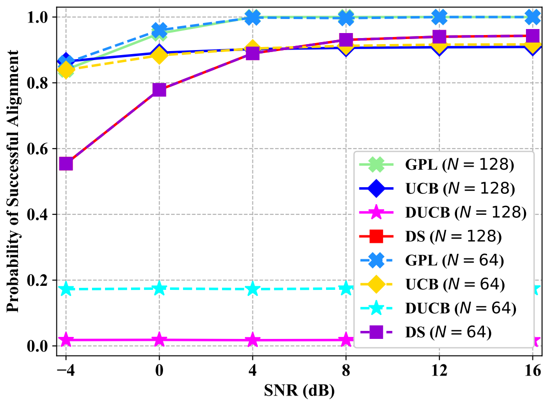

Fig. 12. The PSA performance of different algorithms - HST scenario, semi-online training and $\alpha_{\mathrm{LOS}} \sim \mathcal{CN}(\bar{m}, 1)$ with $|\bar{m}| = 1$.

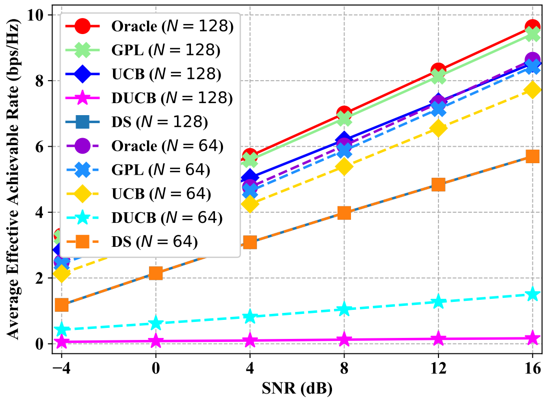

Fig. 13. The AEAR performance of different algorithms - HST scenario, semi-online training and $\alpha_{\mathrm{LOS}} \sim \mathcal{CN}(\bar{m}, 1)$ with $|\bar{m}| = 1$.

$$Y_{i^\star} = |\mathbf{1}^{\mathrm{T}}\mathbf{y}_{i^\star}|^2/L_{\mathrm{p}} > Y_i = |\mathbf{1}^{\mathrm{T}}\mathbf{y}_i|^2/L_{\mathrm{p}}, (\forall\, i \neq i^\star), \quad (27)$$

where $\mathbf{y}_i$ is the received signals of $\mathbf{f}_i$ ($\mathbf{f}_i \in \mathcal{T}$). Intuitively, if more variables $\{Y_i\}$ are involved in (27), the probability that all events in (27) occur simultaneously becomes smaller. Note that DS uses the entire codebook $\mathcal{C}$, while GPL and SBL use a small subset of $\mathcal{C}$. Therefore, the impact of noise on DS is much larger than that on the other two algorithms.

In contrast to most BTT algorithms, an important advantage of GPL is that an optimizable interval estimation is provided, which enables a flexible tradeoff between the PSA and AEAR performance by adjusting the BCI width. The PSA and AEAR performance achieved by GPL (for the indoor scenario) with different BCI choices is demonstrated in Fig. 14 and Fig. 15, respectively. The abbreviation “POST-OPT” denotes Bayesian posterior optimization (i.e., Algorithm 3).

It is observed from Fig. 14 that the 3σ criterion achieves the best PSA performance, which is as expected. However, the 3σ criterion is often too conservative. Although it avoids beam misalignments, it consumes too much precious communication time-resource, which, in turn, reduces the AEAR performance, as shown in Fig. 15. Compared to the 3σ criterion, the 1σ criterion is, however, too confident, which leads to frequent

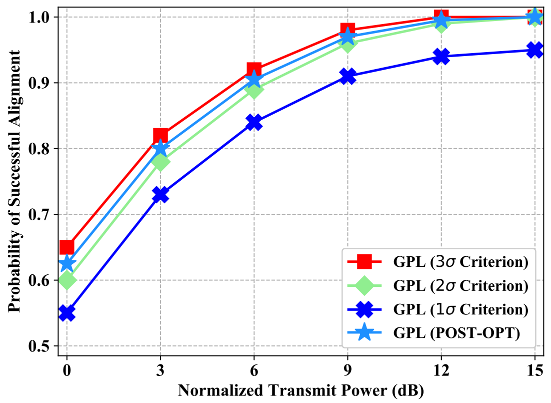

Fig. 14. The PSA performance of GPL with different BCI choices - fully-online training.

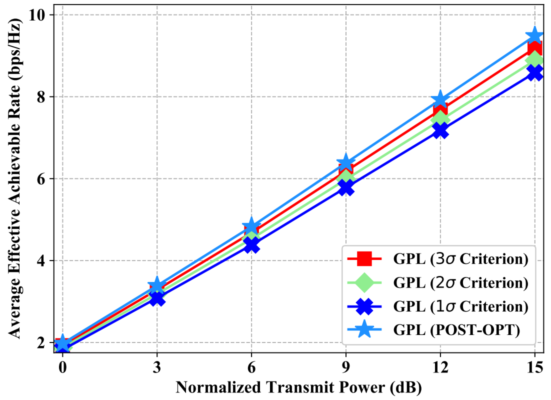

Fig. 15. The AEAR performance of GPL with different BCI choices - fully-online training.

beam misalignments. As a result, its AEAR performance is also far from satisfactory. In contrast to the previous two BCI choices, the optimized BCI can automatically achieve an ideal tradeoff between beam training and data transmission, which helps to obtain the best AEAR performance.

VII. Conclusion

From the perspective of identifying and exploiting challenging priors and patterns, in this paper we proposed to model BDTs via SP and addressed the problem of BTT via process inference. We formulated the problem of BTT as GP prediction by encoding environment priors/patterns and uncertainties into the GP kernel, so as to achieve the small sample performance. To improve prediction efficiency, novel network structure and efficient learning methods were proposed. Bayesian posterior inference and optimization were further proposed to enhance the designed algorithms, which can achieve a flexible tradeoff between desired performance and beam training overhead.

Appendix A
Proof of Theorem 1

For any index $u = (t_1, t_2, \cdots, t_k)$, the projection $\pi_u = \pi_{t_1,\cdots,t_k}$ from $\mathbb{R}^T$ to $\mathbb{R}^k$ is clearly $\mathcal{B}^T/\mathcal{B}^k$-measurable, since if $C \in \mathcal{B}^k$, $\pi_u^{-1}C$ is a cylinder and hence is in $\mathcal{B}^T$. Therefore, if ξ is $\mathcal{E}/\mathcal{B}^T$-measurable, $\xi_t = \pi_t \xi$ is $\mathcal{E}/\mathcal{B}$-measurable for each t.

Conversely, if each ξ_t is $\mathcal{E}/\mathcal{B}$-measurable, $(\xi_{t_1}, \xi_{t_2}, \cdots, \xi_{t_k})$ is clearly $\mathcal{E}/\mathcal{B}^k$-measurable, i.e., $\pi_u\xi$ is $\mathcal{E}/\mathcal{B}^k$-measurable for $u = (t_1, t_2, \cdots, t_k)$. Hence if $C \in \mathcal{B}^k$, $\xi^{-1}\pi_u^{-1}C = (\pi_u\xi)^{-1}C \in \mathcal{E}$ or $\xi^{-1}E \in \mathcal{E}$ for each cylinder E. Since these cylinders generate $\mathcal{B}^T$, if follows that ξ is $\mathcal{E}/\mathcal{B}^T$-measurable as required.

Appendix B
Gaussian Process and Typical Kernels

A. Gaussian Process Regression

A stochastic process $f(\mathbf{x})$ is referred to as a GP if and only if for any finite number of points $\mathbf{x}_1, \cdots, \mathbf{x}_n$ $(\forall\, i, \mathbf{x}_i \in \mathbb{R}^D)$, the joint probability density function $p(f(\mathbf{x}_1), \cdots, f(\mathbf{x}_n))$ is Gaussian [28]. A GP is completely characterized by its mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$, which are, in a sense, similar to the mean and variance of the multi-variate Gaussian distribution. The mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$ are respectively defined as

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})] \\ k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]. \end{aligned} \tag{28}$$

For simplicity, the mean function is usually taken to be zero in practice, i.e., $m(\mathbf{x}) = 0$.

GP can be used for regression [28], i.e., to predict or infer $f(\mathbf{x}_\star)$ for an unseen $\mathbf{x}_*$ based on a set of observations $\mathcal{D} = \{(\mathbf{x}_i, y_i) \,|\, y_i = f(\mathbf{x}_i) + w_i, w_i \sim \mathcal{N}(0, \sigma_w^2), i = 1, \cdots, n\}$, where $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^D$ and $y \in \mathbb{R}$ represent input vector and output scalar, respectively. In contrast to other regression methods, GP regression is based on Bayesian inference, which outputs a probability distribution, rather than a point estimate for the quantity of interest. Given dataset $\mathcal{D}$, the conditional (or predictive) distribution of $f_* = f(\mathbf{x}_*)$ at $\mathbf{x}_*$ is given by

$$p(f_*|\mathcal{D}, \mathbf{x}_*) \sim \mathcal{N}\big(\mu(\mathbf{x}_*), \sigma^2(\mathbf{x}_*)\big) \tag{29}$$

$$\mu(\mathbf{x}_*) = \mathbf{k}_*^{\mathrm{T}}(\mathbf{K} + \sigma_w^2\mathbf{I})^{-1}\mathbf{y} \tag{30}$$

$$\sigma^2(\mathbf{x}_*) = k_{**} - \mathbf{k}_*^{\mathrm{T}}(\mathbf{K} + \sigma_w^2\mathbf{I})^{-1}\mathbf{k}_*, \tag{31}$$

where the (i, j)-th element of $\mathbf{K}$ and the i-th element of $\mathbf{k}_*$ are calculated as $k(\mathbf{x}_i, \mathbf{x}_j)$ and $k(\mathbf{x}_i, \mathbf{x}_*)$, respectively. The scalar k_{**} is calculated as $k(\mathbf{x}_*, \mathbf{x}_*)$.

B. Typical Covariance Functions

A covariance function (also referred to as kernel in literatures) is crucial for GP predictor, because it encodes the prior about the function to be learned, i.e., the specification of the kernel implies a distribution over functions. Next, we briefly introduce two classical kernels [28], [31].

1) Squared Exponential Kernel: The squared exponential kernel takes the form

$$k_{\mathrm{SE}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2l^2}\|\mathbf{x} - \mathbf{x}'\|^2\right), \tag{32}$$

where σ_f^2 and l denote the signal variance and length-scale, respectively. The physical meaning of parameter l is that if we think that the GP varies rapidly, the length-scale l should be shorter [28]. Hence, the degree of variation of a GP is achieved by simply adjusting the parameters of the kernel. Since k_{SE} is infinitely differentiable, a GP with this kernel is smooth.

2) Spectral Mixture Kernel: To provide more flexibility, the spectral mixture kernel takes the form

$$k_{\text{SM}}(\mathbf{x},\mathbf{x}') = \sum_{q=1}^{Q} a_q \frac{|\mathbf{\Sigma}_q|^{0.5}}{(2\pi)^{D/2}} \exp\left(-\frac{1}{2}\left\|\mathbf{\Sigma}_q^{0.5}(\mathbf{x}-\mathbf{x}')\right\|^2\right) \cdot \cos\langle \mathbf{x}-\mathbf{x}', 2\pi\boldsymbol{\mu}_q\rangle, \tag{33}$$

where $\{\alpha_q\}, \{\mathbf{\Sigma}_q\}$ and $\{\boldsymbol{\mu}_q\}$ are mixture weights, bandwidths (inverse length-scales) and frequencies, respectively. It is referred to [31] for more details about this kernel. Compared to the squared exponential kernel, the spectral mixture kernel is more expressive. To further enhance the flexibility, the neural network is incorporated into the GP in this paper.

Appendix C
Proof of Lemma 1

The key operation corresponding to the $c\sigma$ criterion is to sweep the interval $\mathcal{I}(c)$. The (posterior) probability $p_{\text{succ}}(c)$, i.e., the probability that the true beam $\mathbf{f}^\star$ (with beam index $i^\star$) can be found via beam sweeping, can be calculated as

$$p_{\text{succ}}(c) = \mathbb{P}\big(\mathbf{f}_{i^\star} \text{ is found} \,|\, i^\star \in \mathcal{I}(c)\big)\mathbb{P}\big(i^\star \in \mathcal{I}(c)\big).$$

According to Algorithm 2 and (29) - (31), the probability $\mathbb{P}\big(i^\star \in \mathcal{I}(c)\big)$ can be calculated as

$$\mathbb{P}\big(i^\star \in \mathcal{I}(c)\big) = \int_{\mu - c\sigma}^{\mu + c\sigma} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx. \tag{34}$$

We consider the conditional probability $\mathbb{P}\big(\mathbf{f}_{i^\star} \text{ is found} \,|\, i^\star \in \mathcal{I}(c)\big)$. For convenience, let $p_0 = \mathbb{P}\big(\mathbf{f}_{i^\star} \text{ is found} \,|\, i^\star \in \mathcal{I}(c)\big)$. The event that $\mathbf{f}_{i^\star}$ is found under the condition $i^\star \in \mathcal{I}(c)$ occurs if and only if $Y^\star = |\mathbf{1}^{\mathrm{T}}\mathbf{y}^\star|^2/L_{\mathrm{p}} > Y_i = |\mathbf{1}^{\mathrm{T}}\mathbf{y}_i|^2/L_{\mathrm{p}}$ for all $i \neq i^\star, i \in \mathcal{I}(c)$. According to the union bound formula, p_0 can be lower bounded by

$$\begin{aligned}
&\mathbb{P}\big(\mathbf{f}_{i^\star} \text{ is found} \,|\, i^\star \in \mathcal{I}(c)\big) \\
=&\mathbb{P}(Y_{i^\star} > Y_i \,|\, \forall\, i \neq i^\star, i \in \mathcal{I}(c)) \\
\geq& \max\left\{0, 1 - \sum_{i \neq i^\star, i \in \mathcal{I}(c)} \mathbb{P}(Y_{i^\star} \leq Y_i)\right\}.
\end{aligned}$$

It can be verified that $2Y_i = 2|\mathbf{1}^{\mathrm{T}}\mathbf{y}_i|^2/L_{\mathrm{p}}$ obeys a noncentral chi-squared distribution with degrees of freedom 2, whose probability density function is given by

$$f_i(x) = \frac{1}{2} e^{-\frac{1}{2}(x+\lambda_i)} \sum_{n=0}^{\infty} \frac{\lambda_i^n x^n}{4^n (n!)^2}, \; (x \geq 0),$$

where $\lambda_i = 2PNL_{\mathrm{p}}\beta^{-1}\left|\alpha_0 \mathbf{a}^{\mathrm{H}}(\phi)\mathbf{f}_i\right|^2$ is the noncentrality parameter. Since $Y_{i^\star}$ and Y_i are independent, the probability $\mathbb{P}(Y_{i^\star} \leq Y_i)$ can be calculated as

$$\begin{aligned}
\mathbb{P}(Y_{i^\star} \leq Y_i) &= \iint_{x \geq 0, y \geq 0, x \leq y} f_{i^\star}(x) f_i(y)\, dx dy \\
&= \int_0^{\infty} f_{i^\star}(x) dx \int_x^{\infty} f_i(y) dy.
\end{aligned}$$

For $i \neq i^\star$, Y_i is degenerated into an exponential distribution since $\lambda_i = 0$. Similar to the derivation in [7], $\mathbb{P}(Y_{i^\star} \leq Y_i)$ is calculated by $\frac{1}{2}\exp\left(-\frac{1}{4}\lambda_{i^\star}\right)$. Thus, p_0 is lower bounded by

$$\mathbb{P}(\mathbf{f}_{i^\star} = \mathbf{f}^\star) \geq \max\left\{0, 1 - \frac{|\mathcal{I}(c)|}{2}\exp\left(-\frac{1}{4}\lambda_{i^\star}\right)\right\},$$

with $\lambda_{i^\star} = \lambda^\star = 2PNL_{\mathrm{p}}|\alpha_{\mathrm{C}}|^2\beta^{-1}$. The remaining derivation is straightforward, which is omitted.

Appendix D
Proof of Theorem 3

To prove the unimodality of $f(c)$, we shall investigate the properties of the first-order and second-order derivative functions of f in $(0, \infty)$, which are respectively calculated as

$$\begin{aligned}
f'(c) =& -2\sigma T_0/T_{\mathrm{S}} \int_0^c \exp\left(-\frac{x^2}{2}\right) dx + \\
&\big(1 - (2c\sigma + 1)T_0/T_{\mathrm{S}}\big)\exp(-c^2/2) \\
f''(c) =& \exp\left(-\frac{c^2}{2}\right)\left(\frac{2c^2\sigma T_0}{T_{\mathrm{S}}} - c\left(1 - \frac{T_0}{T_{\mathrm{S}}}\right) - \frac{4\sigma T_0}{T_{\mathrm{S}}}\right).
\end{aligned}$$

We first investigate the properties of $f''(c)$, which is the product of positive function $\exp(-c^2/2)$ and quadratic function $g(c) = \big(2c^2\sigma T_0/T_{\mathrm{S}} - c(1 - T_0/T_{\mathrm{S}}) - 4\sigma T_0/T_{\mathrm{S}}\big)$. Because: 1) the coefficient of the quadratic term of $g(c)$ is $2\sigma T_0/T_{\mathrm{S}} > 0$; 2) the minimum point (or axis of symmetry) of $g(c)$ satisfies $c^\star = (T_{\mathrm{S}} - T_0)/(4\sigma T_0) > 0$; and 3) $g(0) = -4\sigma T_0/T_{\mathrm{S}}$, there must exist a point $c_0 > c^\star$ such that $g(c) < 0$ for $c \in (0, c_0)$ and $g(c) > 0$ for $c \in (c_0, \infty)$. Since $\exp(-c^2/2) > 0$, $f''(c)$ and $g(c)$ keep the same positivity or negativity, i.e., $f''(c) < 0$ for $c \in (0, c_0)$ and $f''(c) > 0$ for $c \in (c_0, \infty)$. Hence, $f'(c)$ monotonously decreases in $(0, c_0)$ and increases in (c_0, ∞).

Because $f'(0) = 1 - T_0/T_{\mathrm{S}} > 0$ and $\lim_{c \to \infty} f'(c) = -\sqrt{2\pi}\sigma T_0/T_{\mathrm{S}} < 0$, the zero point theorem asserts that there exists at least one point c_{z} such that $f'(c_{\mathrm{z}}) = 0$. Note also that because: 1) $f'(0) = 1 - T_0/T_{\mathrm{S}} > 0$; and 2) $f'(c)$ first monotonously decreases and then increases, f' has at most two zero points within $(0, \infty)$. We next prove that f' has only one zero point. Suppose that there are two zero points, denoted by c_1 and c_2. The continuity of f' and positivity of f'' in (c_0, ∞) asserts that $f'(c) > 0$ for $c \in (c_2, \infty)$, which, however, contradicts with the fact that $f'(c) < 0$ when c is sufficiently large (as $\lim_{c \to \infty} f'(c) = -\sqrt{2\pi}\sigma T_0/T_{\mathrm{S}} < 0$). Hence, there exists one and only one zero point of f', which is also the maximum point of f.

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] D. Zhu, J. Choi, Q. Cheng, W. Xiao, and R. W. Heath, "High-resolution angle tracking for mobile wideband millimeter-wave systems with antenna array calibration," *IEEE Trans. Wireless Commun.*, vol. 17, no. 11, pp. 7173–7189, 2018.

[3] 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.

[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.

[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] Z. Xiao, T. He, P. Xia, and X. G. Xia, "Hierarchical codebook design for beamforming training in millimeter-wave communication," *IEEE Trans. Wireless Commun.*, vol. 15, no. 5, pp. 3380–3392, May 2016.

[11] 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.

[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] J. C. Aviles and A. Kouki, "Position-aided mm-wave beam training under nlos conditions," *IEEE Access*, vol. 4, pp. 8703–8714, 2016.

[18] Z. Wei, Y. Zhao, X. Liu, and Z. Feng, "Doa-lf: A location fingerprint positioning algorithm with millimeter-wave," *IEEE Access*, vol. 5, pp. 22 678–22 688, 2017.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

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

[29] C.-X. Wang, A. Ghazal, B. Ai, Y. Liu, and P. Fan, "Channel measurements and models for high-speed train communication systems: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 974–987, 2016.

[30] R. Calandra, J. Peters, C. Rasmussen, and M. Deisenroth, "Manifold gaussian processes for regression," in *International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 1–10.

[31] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing, "Deep kernel learning," in *Proceedings of Machine Learning Research*, vol. 51. Cadiz, Spain: PMLR, 09–11 May 2016, pp. 370–378.

[32] T. S. Rappaport, E. Ben-Dor, J. N. Murdock, and Y. Qiao, "38 GHz and 60 GHz angle-dependent propagation for cellular peer-to-peer wireless communications," in *2012 IEEE ICC*, June 2012, pp. 4568–4573.

[33] J. H. Mathews and K. D. Fink, *Numerical methods using MATLAB*, 4th ed., ser. Mathematics - Computational Mathematics. Prentice Hall, 2003.

[34] M. Hashemi, A. Sabharwal, C. E. Koksal, and N. B. Shroff, "Efficient beam alignment in millimeter wave systems using contextual bandits," in *IEEE INFOCOM 2018*, April 2018, pp. 2393–2401.

Jianjun Zhang (Member, IEEE) received the M.S. degree from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2014, and the Ph.D. degree from Southeast University, Nanjing, China, in 2018.

Since July 2022, he has been a Professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. From December 2019 to April 2022, he was a Research Fellow of the electrical and electronics engineering with University College London (UCL), London, UK. From March 2019 to November 2019, he was a Post-Doctoral Researcher with the Purple Mountain Laboratories, Nanjing, China. He was the recipient of the Best Paper Award in the IEEE Globecom 2019. His current research interests include machine learning, theory and algorithms of optimization, and wireless communications and networking.

Christos Masouros (Senior Member, IEEE) received the Diploma degree in Electrical and Computer Engineering from the University of Patras, Greece, in 2004, and MSc by research and PhD in Electrical and Electronic Engineering from the University of Manchester, UK in 2006 and 2009 respectively. In 2008 he was a research intern at Philips Research Labs, UK. Between 2009-2010 he was a Research Associate in the University of Manchester and between 2010-2012 a Research Fellow at Queen's University Belfast. In 2012 he joined University College London as a Lecturer. He has held a Royal Academy of Engineering Research Fellowship between 2011-2016.

He is currently a Full Professor in the Information and Communications Engineering research group, Dept. Electrical and Electronic Engineering, University College London. His research interests lie in the field of wireless communications and signal processing with particular focus on Green Communications, Large Scale Antenna Systems, Cognitive Radio, interference mitigation techniques for MIMO and multicarrier communications. He was the recipient of the Best Paper Awards in the IEEE GlobeCom 2015 and IEEE WCNC 2019 conferences, and has been recognised as an Exemplary Editor for the IEEE Communications Letters, and as an Exemplary Reviewer for the IEEE Transactions on Communications. He is an Editor for IEEE Transactions on Communications, and IEEE Transactions on Wireless Communications. He has been an Associate Editor for IEEE Communications Letters, and a Guest Editor for IEEE Journal on Selected Topics in Signal Processing issues "Exploiting Interference towards Energy Efficient and Secure Wireless Communications" and "Hybrid Analog/Digital Signal Processing for Hardware-Efficient Large Scale Antenna Arrays". He is currently an elected member of the EURASIP SAT Committee on Signal Processing for Communications and Networking.

Yongming Huang (Senior Member, IEEE) received the B.S. and M.S. degrees from Nanjing University, Nanjing, China, in 2000 and 2003, respectively, and the Ph.D. degree in electrical engineering from Southeast University, Nanjing, in 2007.

Since March 2007 he has been a faculty in the School of Information Science and Engineering, Southeast University, China, where he is currently a full professor. He has also been the Director of the Pervasive Communication Research Center, Purple Mountain Laboratories, since 2019. From 2008 to 2009, he was visiting the Signal Processing Lab, Royal Institute of Technology (KTH), Stockholm, Sweden. He has published over 200 peer-reviewed papers, hold over 80 invention patents. His current research interests include intelligent 5G/6G mobile communications and millimeter wave wireless communications. He submitted around 20 technical contributions to IEEE standards, and was awarded a certificate of appreciation for outstanding contribution to the development of IEEE standard 802.11aj. He served as an Associate Editor for the IEEE Transactions on Signal Processing and a Guest Editor for the IEEE Journal on Selected Areas in Communications. He is currently an Editor-at-Large for the IEEE Open Journal of the Communications Society.