

Learning-Based Predictive Transmitter-Receiver Beam Alignment in Millimeter Wave Fixed Wireless Access Links

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

Abstract—Millimeter wave (mmwave) fixed wireless access is a key enabler of 5G and beyond small cell network deployment, exploiting the abundant mmwave spectrum to provide Gbps backhaul and access links. Large antenna arrays and extremely directional beamforming are necessary to combat the mmwave path loss. However, narrow beams increase sensitivity to physical perturbations caused by environmental factors. To address this issue, in this paper we propose a predictive transmit-receive beam alignment process. We construct an explicit mapping between transmit (or receive) beams and physical coordinates via a Gaussian process, which can incorporate environmental uncertainties. To make full use of underlying correlations between transmitter and receiver and accumulated experiences, we further construct a hierarchical Bayesian learning model and design an efficient beam predictive algorithm. To reduce dependency on physical position measurements, a reverse mapping that predicts physical coordinates from beam experiences is further constructed. The designed algorithms enjoy two folds of advantages. Firstly, thanks to Bayesian learning, a good performance can be achieved even for a small sample setting as low as 10 samples in our scenarios, which drastically reduces training time and is therefore very appealing for wireless communications. Secondly, in contrast to most existing algorithms that only predict one beam in each time-slot, the designed algorithms generate the most promising beam subset, which improves robustness to environment uncertainties. Simulation results demonstrate the effectiveness and superiority of the designed algorithms against the state of the art.

Index Terms—Beam alignment, Bayesian learning, beam prediction, beam training, fixed wireless access, Gaussian processes, millimeter wave communications.

I. INTRODUCTION

MILLIMETER wave (mmwave) fixed wireless access (FWA), which is also referred to as mmwave distribution network and supported by the IEEE 802.11ay, enables different deployment scenarios, including broadband residential access, WiFi access point (AP), small cell backhaul, and home media sharing [1]. By operating in the mmwave bands, mmwave FWA provides much increased capacity compared to other WiFi systems that operate in the microwave bands. Moreover, since mmwave links are highly directional, it also presents significant opportunities for spatial reuse,

offering more flexibility in the deployment of FWA systems. As a promising cost-efficient high-performance alternative and complement to the conventional fixed access networks, mmwave FWA offers many advantages to service providers, including high coverage, low upfront cost, and less coordination with various building owners.

The advantages of mmwave FWA systems particularly rely on the features of mmwave bands, including the availability of large available bandwidth and the line-of-sight (LOS) nature of mmwave communications which helps to control interference between systems [2]. However, it is far from easy to reap these benefits due to the propagation features of mmwave signals. In fact, millimeter wave systems require a large directional gain in order to combat their relatively high path loss compared to systems with lower frequencies and the additional losses due to rain and oxygen absorption. Fortunately, the small wavelength of mmwave signals makes it convenient to compensate the above by packing a large number of antennas into a small space, which also enables adaptive alignment of transmit and receive beams.

Although beamforming techniques at mmwave have been widely studied (e.g., in the standards IEEE 802.15.3c and IEEE 802.11ad), beamforming for mmwave FWA systems is more challenging. The reason is two-fold. Firstly, the distance between the transceivers in mmwave FWA systems is often longer in outdoor environments, which requires a much larger beamforming gain (or equivalently, narrow beams). Secondly, other environmental factors such as wind and precipitation can cause a large receive SNR (signal-to-noise ratio) degradation and thus require a more subtle beam alignment (BA) [3]. In particular, since the transceiver units of mmwave FWA systems are mounted to outdoor facility such as poles, pillars or street lamps, vibration and movement induced by wind flow and gusts, traffic vibrations, or other disruptions may cause unacceptable outage probability if BA is not frequently performed [4]. This is particularly challenging for the pencil-beams employed in these links, where fractions of a degree in azimuth angle can cause SNR loss of a few dB.

To address these issues, an adaptive BA algorithm using a hierarchical multi-resolution codebook was proposed in [4], which, to some extent, can avoid the costly exhaustive search of all pairs of transmit and receive beams. Note that the performance of the hierarchical search scheme heavily relies on the adopted hierarchical codebook. Because of the importance of the hierarchical codebook, it has attracted much attention and has been widely researched [5]–[9]. For example, by exploiting

Copyright (c) 2021 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org. (Corresponding author: Jianjun Zhang.)

The work was supported by the Engineering and Physical Sciences Research Council, UK under project EP/S028455/1.

J. Zhang and C. Masouros are with the Department of Electronic & Electrical Engineering, University College London, London WC1E7JE, U.K. (E-mail: jianjun.zhang, c.masouros@ucl.ac.uk).

sub-array and deactivation techniques, a hierarchical codebook design was devised in [8], where closed-form expressions were provided to generate the codebook. Nevertheless, the training overhead of the adaptive search based algorithms is still high for large-scale antenna array systems. More importantly, in practical dynamic environments, mmwave channels fade fast and the period of each coherence block is too short to allocate enough time for frequent and accurate BA [10].

To reduce the overhead of beam training, one promising way is to extract and exploit information from the training history, so as to reduce the beam search space for future training. Such an idea is facilitated by the fast development of machine learning (ML), which leads to ML based beam training algorithms [10]–[17]. The existing ML based solutions roughly fall into two categories, i.e., supervised learning (SL) based category and reinforcement learning (RL) based category. The SL based algorithms occupy most of the existing ML based beam training solutions [10]–[15]. To achieve a satisfying performance, SL based algorithms require a large number of training samples in advance, which limits its application scope. The second category is based on RL or more general sequential decision and optimization [16]–[21]. Based on the multi-armed bandit (MAB), a lightweight RL approach, a beam training algorithm was proposed in [16], which could utilize contextual information effectively. However, due to the limited structure of the MAB, it is difficult to discover useful patterns and make complex decisions via MAB. The temporal correlation information is well exploited in [20], [21] to design beam training and/or data transmission algorithms, e.g., spatial information through random mobilities of users is exploited in [21].

To achieve a good performance for small sample setting, in this paper we leverage the Gaussian process (GP) to construct an explicit mapping between transmit (or receive) beams and physical coordinates. To enhance the basic GP based beam prediction algorithm, we further design two algorithms that can respectively make full use of underlying correlations between the transmitter and receiver and reduce the dependency on physical position measurements. Thanks to the Bayesian learning based design paradigm, a good performance can be achieved even for the small sample setting. Instead of a single beam, a promising beam subset is generated by the designed algorithms, which makes the algorithms more robust. The main contributions are summarized as follows:

- Aiming at achieving a good performance for small sample sets, we propose a BA algorithm by exploiting the GP to construct a probabilistic mapping between transmit (or receive) beams and physical coordinates. An important advantage of the GP-based design is that the environmental uncertainty is incorporated into the BA algorithm.
- To take full advantage of accumulated experiences and (in particular) the correlation between the transmitter and receiver, we further propose an enhanced BA algorithm based on the PAC-Bayesian learning theory. Compared to the original BA algorithm, the enhanced design dramatically reduces the computational complexity.
- We construct a reverse mapping which can predict physical coordinates from the record of past optimal beams,

based on which we propose a novel BA algorithm. The novel algorithm consists of both coordinate prediction and beam prediction, which helps to reduce (or even avoid) coordinate measurements and thus reduce training time.

- Comprehensive simulation results are provided to demonstrate the effectiveness and superiority of the proposed BA algorithms. It is shown that the proposed algorithms are robust to environmental changes and uncertainties and can achieve a good performance even for the small sample setting.

The remainder of this paper is organized as follows. The system model of mmwave FWA backhaul is described in Section II. In Section III, a predictive BA algorithm is designed based on Bayesian deep learning. To exploit the correlation between the transmitter and receiver, an enhanced BA design is proposed in Section IV. To reduce the training time and samples, a physical coordinate (position) prediction algorithm is designed based on the record of past optimal beams in Section V. Simulation results and conclusions are given in Section VI and Section VII, respectively. GP and PAC-Bayesian learning theory are provided in Appendix A to facilitate the development of the GP based BA algorithms. Tedious proof is deferred to Appendix B to improve readability.

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. Particularly, bold italic lowercase letters \mathbf{x} represent physical coordinates. Calligraphic letters \mathcal{A} stand for sets or distributions. $\mathbb{P}(\cdot)$, $\mathbb{E}(\cdot)$ and $(\cdot)^H$ stand for probability, mathematical expectation and Hermitian operators, respectively. $\mathbb{I}\{\cdot\}$ and $\text{card}(\mathcal{A})$ denote an indicator function and the cardinality of the set \mathcal{A} , respectively. $(\cdot)^*$ 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} . $A \sim U(a, b)$ indicates that A is a random variable that is uniformly distributed in interval $[a, b]$.

II. SYSTEM MODEL

Consider a millimeter wave (mmwave) wireless backhaul system, which consists of one transmitter (Tx) and one receiver (Rx), as shown in Fig. 1. The Tx and Rx are equipped with N_T transmit antennas and N_R receive antennas, respectively. To facilitate practical system implementation, analog beamforming based on codebooks is considered in this paper, where each transmit (or receive) beam is chosen from a codebook \mathcal{C}_T (or \mathcal{C}_R) of size M_T (or M_R) [2].

Without loss of generality, \mathcal{C}_T and \mathcal{C}_R are constructed by sampling beam spaces. Specifically, for uniform linear arrays (ULAs), \mathcal{C}_T is constructed as

$$\mathcal{C}_T = \{\mathbf{f}_i = \mathbf{a}_T(-1 + 2i/M_T) \mid i = 0, 1, \dots, M_T - 1\}. \quad (1)$$

$\mathbf{a}_T(\cdot)$ in (1) takes the form

$$\mathbf{a}_T(x) = \frac{1}{\sqrt{N_T}} \left[1, e^{j \frac{2\pi}{\lambda_T} d_T x}, e^{j \frac{2\pi}{\lambda_T} 2d_T x}, \dots, e^{j \frac{2\pi}{\lambda_T} (N_T-1)d_T x} \right],$$

where λ_T and d_T are the signal wave-length and the distance between any two adjacent antennas, respectively. \mathcal{C}_R can be

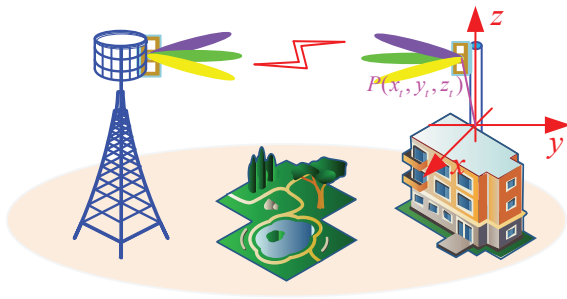


Fig. 1. An illustration of a millimeter wave MIMO FWA backhaul system.

constructed similarly and denoted by

$$\mathcal{C}_R = \{\mathbf{w}_j = \mathbf{a}_R(-1 + 2j/M_R) \mid j = 0, 1, \dots, M_R - 1\}.$$

Due to the sparsity of mmwave channels, an extended Saleh-Valenzuela geometric model is considered in this paper. The wideband mmwave transmission is suffered from beam squint effect [22], [23]. For simplicity, the narrow-band channel model is considered in this paper. The extension to the wideband case is deferred to future works. The channel matrix between the Tx and the Rx is given by

$$\mathbf{H} = \sqrt{N_T N_R} d^{-\tau} \sum_{l=1}^L \alpha_l \mathbf{a}_R(\phi_l) \mathbf{a}_T^H(\psi_l), \quad (2)$$

where d is the distance between the Tx and the Rx, τ is the path-loss exponent, L is the number of propagation paths, and α_l is the complex small-scale fading path gain of the l -th path. In (2), ϕ_l and ψ_l take the form $\phi_l = \cos(\theta_l)$ and $\psi_l = \cos(\vartheta_l)$, where θ_l and ϑ_l represent the physical angle of arrival (AoA) and physical angle of departure (AoD) of the l -th path, respectively. \mathbf{a}_R in (2) represents Rx array response vector.

Let the transmit beam be $\mathbf{f} \in \mathcal{C}_T$. The signal received at the Rx with receive beam $\mathbf{w} \in \mathcal{C}_R$ is given by

$$\mathbf{y} = \sqrt{P} \mathbf{w}^H \mathbf{H} \mathbf{f} \mathbf{s} + \mathbf{n}, \quad (3)$$

where P is the transmit power, \mathbf{s} of size $L_p \times 1$ is the pilot sequence, and $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \sigma_n^2 \mathbf{I})$ is the noise vector. For the codebook-based beamforming system, each time-slot consists of two phases, i.e., beam training and data transmission. The main task of beam training is to search the optimal transmit and receive beams (i.e., find out the beams with the highest measured signal power) for the subsequent phase of data transmission. The effective achievable rate (EAR) is adopted as a metric to measure the throughput performance of a beam training algorithm, which is defined as [10]

$$R_{\text{eff}} = (1 - T_B/T_S) \log(1 + P |\mathbf{w}^H \mathbf{H} \mathbf{f}|^2), \quad (4)$$

where T_B and T_S denote the duration of beam training within a time-slot and the duration of a time-slot, respectively.

To achieve a high throughput, the time allocated for beam training T_B should be as little as possible, so as to reserve more time for data transmission. Meanwhile, the array gain $|\mathbf{w}^H \mathbf{H} \mathbf{f}|$ should be as large as possible. However, the two goals are in conflict with each other. On the one hand, a large array gain can be achieved only when the optimal beams

are found. On the other hand, finding the optimal beams is time-consuming for a large antenna array and often requires a large training overhead. This issue becomes more severe in the FWA system, where more frequent BA is required due to environmental factors, typically, the vibrations caused by wind or other external stimulations.

To mitigate this issue, we will propose an efficient approach based on Bayesian learning. Similar to [11]–[15], where the information of location is exploited to construct a mapping based on neural networks, the key is to construct an explicit mapping between beam directions and coordinates. Essentially, BA is to match the beam directions with the AoDs/AoAs of key channel paths. The AoDs/AoAs are determined by the locations (of Tx and Rx) and their surroundings. Since the LOS path is often present and dominates and the influence of the surroundings can be ignored for the considered FWA system, the AoDs/AoAs are mainly determined by the locations. Therefore, it is reasonable and efficient to construct an explicit mapping between the beam directions and coordinates.

III. TRANSMIT-RECEIVE BEAM ALIGNMENT DESIGN VIA BAYESIAN DEEP LEARNING

In this section, we will propose an efficient BA algorithm based on Gaussian Processes (GPs). In particular, we assume that the Tx and Rx predict their beams independently.

A. Predictive Beam Alignment via GPs - A Naive Algorithm

In a mmwave backhaul communication system, once the Tx and Rx coordinate positions P and Q are determined, the AoD and AoA (and thus the optimal transmit and receive beams) can also be found, i.e., the AoD and AoA are functions of the (physical) coordinates P and Q . For simplicity, we focus on a displacement at the Tx and assume that the Rx is static with coordinates $(0, d_0, H_R)$ where d_0 is the horizontal distance between the Tx and Rx and H_R is the height of the Rx, as shown in Fig. 2. If no vibration occurs (i.e., the Tx coordinates are $(0, 0, H_T)$ with H_T denoting the height of the Tx pillar), the beam direction vector is \vec{v}_0 . If a displacement occurs (e.g., due to wind flow) and the coordinates of the Tx are $P(x_t, y_t, z_t)$, then the beam direction vector is given by

$$\vec{v}_t = \frac{(-x_t, d_0 - y_t, H_R - z_t)}{\sqrt{x_t^2 + (d_0 - y_t)^2 + (H_R - z_t)^2}}. \quad (5)$$

The relative AoA θ_t can be calculated using the geometry in Fig. 2 and the law of cosines as

$$\theta_t = \arccos\left(\frac{|P_0 Q|^2 + |P_t Q|^2 - |P_0 P_t|^2}{2|P_0 Q| \cdot |P_t Q|}\right). \quad (6)$$

Note that the expressions derived in (5) and (6) are obtained under some idealistic assumptions. However, things are often not that simple. Typically, the pillar shape may be deformed, as shown in Fig. 3. In fact, affected by the material and usage time of the pillar, local meteorological or weather conditions and so on, it is challenging (and even impossible) to derive an accurate and deterministic expression to characterize the relationship between the physical coordinates and the AoD/AoA (or the beam direction vectors). It is also difficult

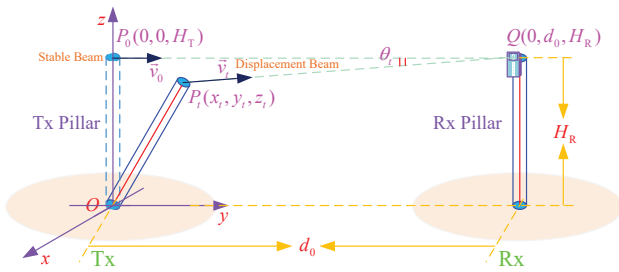


Fig. 2. The relationship between coordinate positions and beam directions.

to incorporate environmental uncertainties into a deterministic model. Accordingly, henceforth we instead adopt a stochastic process to model the displacement of the Tx and Rx beams as a result of complex environmental perturbations, which we use for our learning based beam alignment approach.

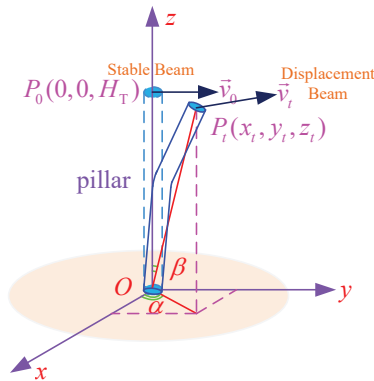


Fig. 3. An accurate and deterministic expression may be unavailable due to all kinds of non-ideal factors, e.g., the pillar shape is deformed.

Instead of the deterministic modeling via NNs, we consider a non-parametric stochastic modeling via stochastic processes, so as to incorporate uncertainties and improve robustness¹. In particular, GPs are chosen in this paper, thanks to the inference tractability (without complex and difficult sampling), sufficient flexibility (to approximate an arbitrary function) and the good small sample performance. Due to the tractability, closed-form expressions of GP posterior probabilities are available, which enables to compute quantities of interest (e.g., posterior prediction performance). For completeness, a brief introduction of GPs is provided in Appendix A. Let $\mathbf{x}(t) = (x_t, y_t, z_t)$ be the coordinates of point P at time t . The prediction model, i.e., the mapping between the beam index $b_T(\cdot)$ of transmit beam and coordinates $\mathbf{x}(t)$, denoted by f , can be written as

$$b_T(\mathbf{x}(t)) = f(\mathbf{x}(t)) + p(\mathbf{x}(t)), \quad (7)$$

where $p(\mathbf{x})$ incorporates possible uncertainties (e.g., due to system or modeling error)². The GP that describes the transmit

¹Since the parameters of the considered stochastic process take value in a Euclidean space of dimension greater than 1, the stochastic process is, in fact, a random field. However, we still adopt the terminology - stochastic process.

²Note that in practice we, in fact, predict the beam direction $b_T(\mathbf{x}(t))$ by constructing the mapping $b_T(\mathbf{x}(t)) = f(\mathbf{x}(t))$, where $b_T(\cdot)$ and $f(\cdot)$ may be vector (functions). Then, b_T is mapped into a beam index via quantization according to the codebook. To avoid complex notations and descriptions, it is assumed to predict the beam index in this paper.

beams is denoted by $B_T(\mathbf{x})$ and referred to as transmit beam process (TBP). The beam process for Rx, i.e., receive beam process (RBP) $B_R(\mathbf{x})$, is defined similarly.

For a GP, the pattern of beam change (e.g., the periodicity of $\mathbf{x}(t)$) is encoded into its kernel, denoted by $k_T(\mathbf{x}(t), \mathbf{x}(t'))$. From the view of Bayesian inference, the kernel encodes the prior. Typically, $k_T(\cdot, \cdot)$ can be a linear combination of the SP kernel $k_{SP}(\cdot, \cdot)$ and the SM kernel $k_{SM}(\cdot, \cdot)$, i.e.,

$$k_T(\mathbf{x}, \mathbf{x}') = c_1 k_{SP}(\mathbf{x}, \mathbf{x}') + c_2 k_{SM}(\mathbf{x}, \mathbf{x}') + \sigma_p^2 \delta_{\mathbf{x}, \mathbf{x}'}, \quad (8)$$

where $c_1 > 0$ and $c_2 > 0$ are the combination coefficients, and $\delta_{\mathbf{x}, \mathbf{x}'}$ is a Kronecker delta (i.e., $\delta_{\mathbf{x}, \mathbf{x}'} = 1$ if and only if $\mathbf{x} = \mathbf{x}'$). However, the practical environment is often very complex. The kernel provided in (8) may be not rich enough to characterize the practical beam processes.

Thanks to the powerful representation and fitting ability of neural networks (NNs), we tackle this problem by parameterizing TBP via NNs. To parameterize a GP, it is sufficient to parameterize its mean function and kernel.³ In contrast to the mean function, it is a bit difficult to parameterize the kernel, since its positive-definiteness must be guaranteed. In this paper, we adopt the method in [24]. Let $k_B(\cdot, \cdot)$ be a base kernel, which can be a SE kernel, a PE kernel, a SM kernel, and so on. The NN is used as a feature extractor on top of which the base kernel is applied. We take the SE kernel as an example and let $\Phi(\mathbf{x}; \phi)$ be the NN with parameters ϕ (e.g., the weights and biases). Then, the parametric kernel $k_p(\cdot, \cdot)$ reads as

$$\begin{aligned} k_p(\mathbf{x}, \mathbf{x}') &= k_B(\Phi(\mathbf{x}; \phi), \Phi(\mathbf{x}'; \phi)) \\ &= \sigma_f^2 \exp\left(-\frac{1}{2l^2} \|\Phi(\mathbf{x}; \phi) - \Phi(\mathbf{x}'; \phi)\|^2\right). \end{aligned}$$

Let $k_T(\cdot, \cdot)$ represent the parameterized kernel of TBP. The NN with parameters ϕ_T for $k_T(\cdot, \cdot)$ is referred to as transmit kernel encoding network (TKEN). The parametric TBP is denoted by $B_T(\mathbf{x}) = \mathcal{GP}(\mathbf{0}, k_T(\mathbf{x}, \mathbf{x}'; \phi_T))$.

Let the beam index at point \mathbf{x}_i ($i = 1, \dots, n$) be $b(\mathbf{x}_i)$.⁴ For convenience, $b(\mathbf{x}_i)$ is abbreviated as b_i , and other variables are denoted similarly. The task is to predict the beam at \mathbf{x}_{n+1} . Let $\mathcal{S}_n = \{(\mathbf{x}_1, b_1), \dots, (\mathbf{x}_n, b_n)\}$ and $\mathbf{b}_n = [b_1, \dots, b_n]^T$ be the training set and the label set, respectively. By using (26) - (28), the Bayesian posterior distribution is given by

$$p(B(\mathbf{x}_{n+1}) | \mathcal{S}_n, \mathbf{x}_{n+1}) \sim \mathcal{N}(\mu(\mathbf{x}_{n+1}), \sigma^2(\mathbf{x}_{n+1})) \quad (9)$$

$$\mu(\mathbf{x}_{n+1}) = \mathbf{k}_*^T (\mathbf{K} + \sigma_p^2 \mathbf{I})^{-1} \mathbf{b}_n \quad (10)$$

$$\sigma^2(\mathbf{x}_{n+1}) = k_{**} - \mathbf{k}_*^T (\mathbf{K} + \sigma_p^2 \mathbf{I})^{-1} \mathbf{k}_*. \quad (11)$$

Please refer to Appendix A (in particular (26) - (28)) for the definition and calculation of k_{**} , \mathbf{k}_* and \mathbf{K} .

For simplicity, we take uniform linear array as an example. To improve robustness, instead of predicting and sweeping one single beam, we sweep the beams within an interval, which is referred to as beam confidence interval (BCI), i.e.,

$$\mathcal{I}_{c\sigma} = [\mu(\mathbf{x}_{n+1}) - c\sigma(\mathbf{x}_{n+1}), \mu(\mathbf{x}_{n+1}) + c\sigma(\mathbf{x}_{n+1})], \quad (12)$$

³The mean function can be directly parameterized via NNs. Similar to most literatures, we set the mean function to zero and concentrate on the kernel.

⁴Note that $\{b(\mathbf{x})\}$ is a sample function (or path) of the stochastic process $\{B(\mathbf{x})\}$, while $\{B(\mathbf{x})\}$ has infinite sample functions.

where c usually takes 3, 2 and 2.5 in practice. In fact, $c = 3$ and $c = 2$ respectively correspond to the 3σ criterion in mathematical statistics and the 2σ criterion in machine learning community, while $c = 2.5$ is introduced to balance the two most common criteria. As an example, if $c = 3$ is chosen, the posterior prediction performance (or probability of successful probability) is $\int_{-c}^c \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx \approx 0.9975 \approx 1$. Note that the choice of the value of c implies a tradeoff between different performance metrics. A smaller value of c implies a lower training overhead while a worse reliability.

A remaining problem is to determine the hyper-parameters, i.e., the parameters of the base kernel and the weights of the NN. This problem is related to Bayesian model selection [25] and can be addressed via marginal likelihood (or evidence) $p(\mathbf{b}_n | \mathbf{X}_n = [\mathbf{x}_1, \dots, \mathbf{x}_n])$ maximization. The marginal likelihood is the integral of the likelihood times the prior, i.e.,

$$p(\mathbf{b}_n | \mathbf{X}_n) = \int p(\mathbf{b}_n | f, \mathbf{X}_n) p(f | \mathbf{X}_n) df, \quad (13)$$

where f is the mapping in (7) that needs to be inferred from data, $p(f | \mathbf{X}_n) = \mathcal{N}(\mathbf{0}, \mathbf{K})$ is the prior, and $p(\mathbf{b}_n | f, \mathbf{X}_n) = p(\mathbf{b}_n | f(\mathbf{X}_n), \mathbf{X}_n) = \mathcal{N}(f(\mathbf{X}_n), \sigma_p^2 \mathbf{I})$ is the likelihood. Via appropriate algebraic operations, $p(\mathbf{b}_n | \mathbf{X}_n)$ is calculated to be $\mathcal{N}(\mathbf{0}, \mathbf{K} + \sigma_p^2 \mathbf{I})$. In practice, we maximize the log marginal likelihood (LML), i.e., $\log p(\mathbf{b}_n | \mathbf{X}_n)$, which is calculated as

$$\log p(\mathbf{b}_n | \mathbf{X}_n) = -\frac{1}{2} \mathbf{b}_n^T (\mathbf{K} + \sigma_p^2 \mathbf{I})^{-1} \mathbf{b}_n - \frac{1}{2} \log \det(\mathbf{K} + \sigma_p^2 \mathbf{I}) - \frac{n}{2} \log 2\pi. \quad (14)$$

Given the training set (or samples) $\mathcal{S}_n = \{(\mathbf{x}_1, b_1), (\mathbf{x}_2, b_2), \dots, (\mathbf{x}_n, b_n)\}$, the parameters can be determined by maximizing $\log p(\mathbf{b}_n | \mathbf{X}_n)$. This is a nonlinear optimization problem, which can be solved via nonlinear optimization methods.

Algorithm 1: Bayesian Beam Alignment/Training Algorithm

1 **input:** initial experiences $\mathcal{S}_T = \{(\mathbf{x}_1, b_1), \dots, (\mathbf{x}_{n_0}, b_{n_0})\}$ and $\mathcal{S}_R = \{(\mathbf{x}'_1, b'_1), \dots, (\mathbf{x}'_{n_0}, b'_{n_0})\}$; $c\sigma$ criterion

2 **initial:** let time-slot counter $n = n_0$

3 **loop** (in each time-slot n)

(1) **maximize** LML in (14) to optimize GP predictive model

(2) **compute** covariance matrix/vector/scalar \mathbf{K} , \mathbf{k}_* and k_{**}

(3) **determine** transmit BCI according to (9) - (12)

(4) **determine** receive BCI similarly

(5) **sweep** all beam pairs constructed from BCIs to find out optimal transmit beam b_{n+1}^* and receive beam $b_{n+1}'^*$

(6) **perform** data transmission with beams b_{n+1}^* and $b_{n+1}'^*$

(7) **update** n , \mathcal{S}_T and \mathcal{S}_R : $n \leftarrow n + 1$

$\mathcal{S}_T \leftarrow \mathcal{S}_T \cup \{(\mathbf{x}_n, b_n)\}$ and $\mathcal{S}_R \leftarrow \mathcal{S}_R \cup \{(\mathbf{x}'_n, b'_n)\}$

end-loop

For clarity, the designed BA algorithm is summarized in Algorithm 1. The input of Algorithm 1 is the training datasets \mathcal{S}_T (for Tx) and \mathcal{S}_R (for Rx) up to time-slot t_{n_0} . Note that \mathcal{S}_T and \mathcal{S}_R can be empty and Algorithm 1 is implemented online in this case. In step 3-(1), we first maximize the LML to determine the hyper-parameters. Then, we determine the transmit BCI in steps 3-(2) and 3-(3). The receive BCI is determined

similarly. In step 3-(5), we find out the optimal transmit and receive beams by sweeping all beam pairs constructed from the BCIs. With the optimal beams available, we can perform data transmission, in step 3-(6). In step 3-(7), we enlarge the training sets by adding the current experiences into \mathcal{S}_T and \mathcal{S}_R . The step to find out the optimal beams for time-slot t_{n+2} is similar. As t increases, a better performance can be achieved with more data accumulated. If the BCIs are large or do not contain the real beams, it implies that the current model (e.g., GP kernel) is inappropriate and Algorithm 1 should be restarted.

If the assumption deviates from practice due to some accidental factors, the predicted beam confidence interval does not contain the real beam direction or the interval is relatively large, which thus affects the prediction performance or other performance metrics (e.g., effective achievable rate). In this case, it is sufficient to recollect a training dataset and optimize the model. Fortunately, since our algorithms can run online and require only a small number of samples (due to a good small sample performance), they will adapt to the new environment quickly again.

Remark 3.1 An important advantage of Algorithm 1 is that a good performance can be achieved even for a small sample setting, which is desirable for ever-changing communication environments. Moreover, in contrast to most of the existing algorithms that predict a single beam each time, Algorithm 1 takes environmental uncertainties into account and outputs a beam subset, which greatly improves its robustness.

Note that although Algorithm 1 has appealing advantages, we hope that we can perfect it by overcoming the following drawbacks. Firstly, to obtain the BCI (or the posterior distribution), a complex optimization problem has to be solved in each time-slot, i.e., maximize the LML in (14). As more and more experiences are accumulated, the scale of the problem also increases, which may be prohibitive. On the other hand, since the Tx and Rx are in a similar/same environment, their experiences are correlated. It is hoped that we can exploit the correlated experiences to improve the performance. Secondly, it is troublesome to measure the coordinates in each time-slot. We hope that the measurements can be reduced and even avoided. Next, we will address the two issues.

IV. BEAM ALIGNMENT WITH CORRELATION EXTRACTION AND EXPLOITATION

In this section, we perfect the designed BA algorithm by extracting and exploiting underlying correlation between the changes of the Tx coordinates and Rx coordinates.

A. Correlation Capture - Hierarchical Prior Modeling

The changes of the Tx coordinates and Rx coordinates are often correlated in practice. In particular, the antenna arrays of the Tx and Rx may be placed at the top of the pillars (e.g., installed at the top of the street lamps). Because of the same or similar external affections (e.g., the wind), the changes of the coordinates share many features. For example, the changes of the coordinates share a same or similar periodicity, although their magnitudes may be different. It is expected to reduce the

scale of the optimization problem in (14) while improve the system performance by leveraging the correlation.

First, we consider how to characterize the correlation. Let $k_T(\cdot, \cdot)$ and $k_R(\cdot, \cdot)$ denote the GP kernels of the TBP and RBP which are parameterized by NNs with weights ϕ_T and ϕ_R , respectively. The NNs of $k_T(\cdot, \cdot)$ and $k_R(\cdot, \cdot)$ are respectively referred to as transmit and receive kernel encoding network (TKEN and RKEN). The parameterized beam processes are denoted by

$$\begin{aligned} B_T(\mathbf{x}) &= \mathcal{GP}(0, k_T(\mathbf{x}, \mathbf{x}'; \phi_T)) \\ B_R(\mathbf{x}) &= \mathcal{GP}(0, k_R(\mathbf{x}, \mathbf{x}'; \phi_R)). \end{aligned} \quad (15)$$

From the perspective of Bayesian inference, $k_T(\cdot)$ is, in fact, the prior of TBP. With data $\mathcal{S}_T = \{(\mathbf{x}_1, b_1), \dots, (\mathbf{x}_n, b_n)\}$ available, $k_T(\cdot)$ is updated into the posterior $k'_T(\cdot)$. The perspective also applies to RBP.

The key is to capture, represent and exploit the correlation. In view that TBP and RBP are affected by the same or similar external effects, it is reasonable to assume that the GP kernels $k_T(\mathbf{x}, \mathbf{x}'; \phi_T)$ and $k_R(\mathbf{x}, \mathbf{x}'; \phi_R)$ are sampled from a prior distribution. Note that since $k_T(\mathbf{x}, \mathbf{x}'; \phi_T)$ and $k_R(\mathbf{x}, \mathbf{x}'; \phi_R)$ themselves are priors, the introduced prior distribution is, in fact, a prior for the priors, i.e., a hyper-prior (See Fig. 4).

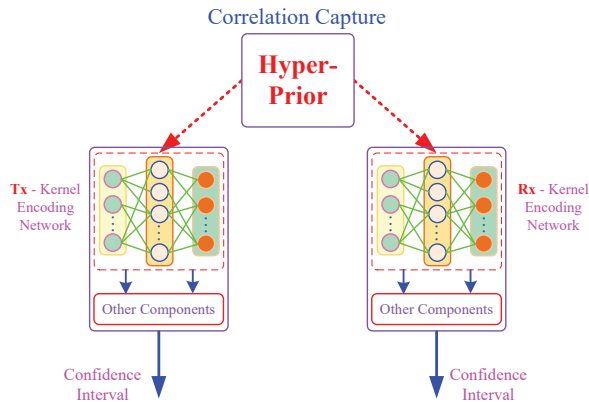


Fig. 4. An illustration of hierarchical prior modeling - the kernels of TBP and RBP are parameterized via NNs, while the correlation between TBP and RBP is captured and represented via a hyper-prior.

Although there are different methods to introduce a hyper-prior for the GP kernels, we adopt a simple but very effective method. Firstly, the NNs of $k_T(\mathbf{x}, \mathbf{x}'; \phi_T)$ and $k_R(\mathbf{x}, \mathbf{x}'; \phi_R)$ shall the same network structure, i.e., the same number of layers, the same number of neurons in each layer, and so on. Secondly, the weights and biases in the same location of the NNs share a common probability distribution, which has dual attributes, i.e., prior (used to capture prior correlations and initialize the weights and biases) and posterior (used to parameterize the GP kernels via sampling). For simplicity, the hyper-prior is chosen as zero-mean isotropic Gaussian, i.e., $\mathcal{P}(\theta) = \mathcal{N}(\mathbf{0}, \sigma_p^2 \mathbf{I})$, where θ represents ϕ_T or ϕ_R .

B. Correlation Extraction and Exploitation - Hyper-Posterior and Posterior Inference

Given the hyper-prior \mathcal{P} and experiences $\mathcal{S}_T = \{(\mathbf{x}_i, b_i) \mid i = 1, \dots, n\}$ and $\mathcal{S}_R = \{(\mathbf{x}'_i, b'_i) \mid i = 1, \dots, n\}$, the key is to

extract correlation and use it to improve system performance. Since the correlation is contained in the hyper-prior and experiences \mathcal{S}_T and \mathcal{S}_R . From the Bayesian perspective, we shall transform the hyper-prior into a hyper-posterior. We hope that the extracted correlation can compensate for the performance loss due to a limited number of samples. More specifically, it is hoped that the generalization error (GE) could be as small as possible. Let $Z(\mathcal{S}, P) = \int_{\mathcal{H}} P(h) \prod_{j=1}^m p(z_j | h) dh$ denote the marginal likelihood with a sample set $\mathcal{S} = \{z_1, z_2, \dots, z_m\}$. The hyper-posterior that yields the smallest GE is characterized in the following theorem.

Theorem 1. Given the hyper-prior \mathcal{P} and experiences \mathcal{S}_T and \mathcal{S}_R , the hyper-posterior that achieves the optimal PAC-Bayes GE bound is given by ⁵

$$\mathcal{Q}^*(P) = \frac{\mathcal{P}(P) \exp([\ln Z(\mathcal{S}_T, P) + \ln Z(\mathcal{S}_R, P)]/2)}{R(\mathcal{P}, \mathcal{S}_T, \mathcal{S}_R)}, \quad (16)$$

where $R(\mathcal{P}, \mathcal{S}_T, \mathcal{S}_R)$ is given by

$$\begin{aligned} R(\mathcal{P}, \mathcal{S}_T, \mathcal{S}_R) &= \int \mathcal{P}(P) \exp\left(\frac{\ln Z(\mathcal{S}_T, P) + \ln Z(\mathcal{S}_R, P)}{2}\right) dP. \end{aligned}$$

Proof: See Appendix B. ■

The optimal hyper-posterior $\mathcal{Q}^*(P)$ in (16) is derived via upper bounding the GE performance, which are provided in (40) or (42). Theorem 1, in fact, provides an efficient method to extract the correlation, which is realized by transforming the hyper-prior along with the experiences of Tx and Rx into the hyper-posterior. In particular, the closed-form expression in (16), which is up to the constant term $R(\mathcal{P}, \mathcal{S}_T, \mathcal{S}_R)$, is provided to compute the optimal hyper-posterior $\mathcal{Q}^*(P)$.

In practice, it may be difficult (and even unnecessary) to accurately compute $\mathcal{Q}^*(P)$. Hence, tractable approximations of \mathcal{Q}^* may be more appealing. For simplicity, we next consider a simple but effective approximation - maximum a posterior (MAP). Another commonly used approximation is variational inference (VI) [26], which is omitted due to space limitation. The MAP approximates $\mathcal{Q}^*(P)$ by a single-point (or Dirac) measure, i.e.,

$$\hat{P} = \arg \max_{P \in \mathcal{M}(\mathcal{H})} \mathcal{Q}^*(P) = \arg \max_{P \in \mathcal{M}(\mathcal{H})} \ln \mathcal{Q}^*(P).$$

Via simple algebraic operations, it can be verified that \hat{P} can be obtained by minimizing the following cost function

$$c(P) = -\frac{\ln Z(\mathcal{S}_T, P) + \ln Z(\mathcal{S}_R, P)}{2} - \ln \mathcal{P}(P). \quad (17)$$

⁵ $\mathcal{P}(\cdot)$ represents a probability distribution/measure on the hypothesis space \mathcal{H} (i.e., the function space that consists of all prediction models), namely, P is a GP. Each GP is determined by its kernel parameterized a NN. Hence, \mathcal{P} , a distribution on NN parameters, represents a probability measures on the space of all GPs, which is denoted by $\mathcal{P}(P) = \mathcal{P}(\theta)$. The two spaces (i.e., the hypothesis space that contains the prediction models of interest and the space of NN parameters that correspond to GP kernels) are two different spaces. For the GP regression setting, $\ln Z(\mathcal{S}, P)$ is, in fact, the LML in (14).

C. Beam Alignment with Correlation Exploitation

Based on the aforementioned discussion, we can propose a more efficient BA algorithm which can extract and exploit the correlation contained in the experiences of Tx and Rx to further improve the system performance. The principle of the designed BA algorithm is outlined in Fig. 5.

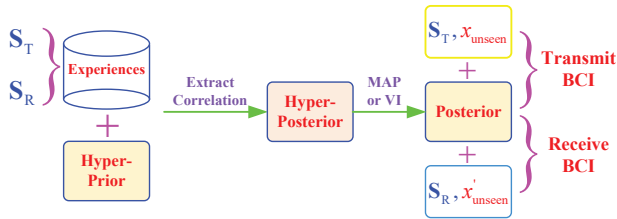


Fig. 5. The principle of beam prediction with correlation exploitation.

Via the hierarchical Bayesian modeling, a reasonable hyper-prior is introduced to capture the correlation. With sufficient experiences available, by extracting the correlation, the hyper-prior is transformed into a hyper-posterior. A good posterior can be further obtained via the MAP approximation. Now, a sufficiently good predictive model has been obtained, since the weights and biases of the NNs in TKEN and RKEN have been determined. As a result, there is no need to maximize the LML in (14). Note that the predictive model is shared by the Tx and Rx. For both Tx and Rx, once unseen coordinates are arrived, a BCI can be obtained according to Eq.(9) - Eq.(11). The BA algorithm is summarized in Algorithm 2.

Algorithm 2 mainly consists of two parts. The first part is to extract the correlation from the experiences, while the second part is to exploit the correlation to improve the system performance. To determine a BCI, or equivalently, the posterior distribution $p(B(\mathbf{x}_{n+1}) | \mathcal{S}_n, \mathbf{x}_{n+1}) \sim \mathcal{N}(\mu(\mathbf{x}_{n+1}), \sigma^2(\mathbf{x}_{n+1}))$ in (9), the experiences \mathcal{S}_n are still required. Alternatively, we can choose a subset of \mathcal{S}_n to further reduce the computational complexity. Finally, to extract the correlation, the experiences \mathcal{S}_T and \mathcal{S}_R have to be gathered together. With the assumption that Tx has a larger computational capacity, a preferable option is that Tx undertakes the task and Rx sends its experiences to Tx. Accordingly, the EAR in (4) is modified as

$$R_{\text{eff}} = (1 - (T_B + T_E)/T_S) \log(1 + P|\mathbf{w}^H \mathbf{H} \mathbf{f}|^2), \quad (18)$$

where T_E is the overhead of collecting the experiences.

Remark 4.1 Compared to Algorithm 1, an important advantage of Algorithm 2 is that there is no need to optimize the (predictive) model by solving a complex optimization problem in each time-slot. Therefore, the computational complexity of Algorithm 2 is much less than that of Algorithm 1.

V. COORDINATE PREDICTION VIA BEAMS

To avoid measuring coordinates frequently, in this section we propose an efficient algorithm to predict the coordinates based on the record of past optimal beams. Then, we incorporate the coordinate prediction algorithm into BA.

Algorithm 2: Beam Alignment with Correlation Exploitation

1 **input:** experiences $\mathcal{S}_T = \{(\mathbf{x}_1, b_1), \dots, (\mathbf{x}_N, b_N)\}$ and $\mathcal{S}_R = \{(\mathbf{x}'_1, b'_1), \dots, (\mathbf{x}'_N, b'_N)\}$; $c\sigma$ criterion

2 **extract correlation**

- (1) **introduce** hyper-prior for NNs of TKEN and RKEN
- (2) **transform** the hyper-prior into a hyper-posterior
- (3) **determine** posterior via MAP or VI approximation

3 **loop** (in each time-slot $n \geq N + 1$)

- (1) **compute** covariance matrix/vector/scalar \mathbf{K} , \mathbf{k}_* and k_{**}
- (2) **determine** transmit BCI and receive BCI
- (3) **sweep** all beam pairs constructed from BCIs to find out optimal transmit beam b_{n+1}^* and receive beam b'_{n+1}
- (4) **perform** data transmission with beams b_{n+1}^* and b'_{n+1}
- (5) **update** n , \mathcal{S}_T and \mathcal{S}_R : $n \leftarrow n + 1$
 $\mathcal{S}_T \leftarrow \mathcal{S}_T \cup \{(\mathbf{x}_n, b_n)\}$ and $\mathcal{S}_R \leftarrow \mathcal{S}_R \cup \{(\mathbf{x}'_n, b'_n)\}$

end-loop

A. Coordinate Prediction via Beams

For convenience, the Tx is taken as an example to elaborate on the coordinate prediction. Up to now, to predict the beam direction in time-slot $n + 1$, the coordinates \mathbf{x}_{n+1} have to be provided. However, it may be costly and troublesome to accurately measure the coordinates in each time-slot. Therefore, an algorithm that requires less measurements or can automatically estimate the coordinates is more appealing. Fortunately, the coordinates can be estimated from past beams (i.e., the corresponding beam directions of the beams) which, for convenience, are referred to as beam experiences (BEs). The rationality is that the environment and also the coordinates often change continuously, which leads to continuous changing beam direction. In other words, the continuous change of the beam direction, in fact, reflects the change of the coordinates. Therefore, based on the continuity, the coordinates \mathbf{x}_{n+1} can be estimated from on the BEs.

The mapping relationship between the coordinates $\mathbf{x}(t)$ and the beam direction $b(t)$ ⁶ can be written as

$$\mathbf{x}(t) = \mathbf{g}(b(t)) + \mathbf{q}(t), \quad (19)$$

where $\mathbf{q}(t)$ incorporates possible uncertainties caused by system or modeling error. To facilitate inference, a multi-output GP (MO-GP) is used to characterize the mapping in (19). For simplicity, the MO-GP is decomposed as K single-output GPs with K denoting the dimension of each coordinate vector, although it may incur some performance loss.

Algorithm 3: Coordinate Prediction via Beam Experiences

1 **input:** beam experiences $\mathcal{D} = \{(b_1, \mathbf{x}_1), \dots, (b_n, \mathbf{x}_n)\}$, beam b_{n+1} and $c\sigma$ criterion

2 **maximize** LML to optimize GP predictive model

3 **determine** predictive distribution of each dimension

4 **determine** coordinate confidence region \mathcal{R}_{n+1}

5 **output:** coordinate confidence region \mathcal{R}_{n+1}

⁶Notice that a beam direction is a continuous quantity, while a beam index is a discrete quantity (an integer). Here, $b(t)$ represents a continuous quantity. The beam index can be obtained via quantization with a codebook.

Let $\mathcal{D} = \{(b_1, \mathbf{x}_1), \dots, (b_n, \mathbf{x}_n)\}$ denote the BEs, i.e., the training set for coordinate prediction (CP). The task of predicting the coordinates \mathbf{x}_{n+1} based on \mathcal{D} is similar to the prediction of the beam b_{n+1} based on $\mathcal{S}_n = \{(\mathbf{x}_1, b_1), \dots, (\mathbf{x}_n, b_n)\}$. Due to space limitation, the details are omitted. For clarity, the algorithm to predict coordinates is summarized in Algorithm 3, which applies to both Tx and Rx. Note that the output of the CP algorithm is often a region of \mathbb{R}^K which is referred to as coordinate confidence region (CCR), i.e., a region that contains the coordinates with a high probability.

B. Beam Alignment Incorporating Coordinate Prediction

In this subsection, we take Tx as an example to show how to incorporate beam prediction (BP) and coordinate prediction into BA. The design principle is shown in Fig. 6. It consists of two modules, i.e., BP module and CP module. The individual roles of the two modules are as follows:

- The role of the BP module is to provide a reliable BCI. The optimal beam can be found by sweeping all beams within the BCI.
- The CP module is in charge of determining an effective CCR, based on which the BP module can estimate or predict an effective BCI.

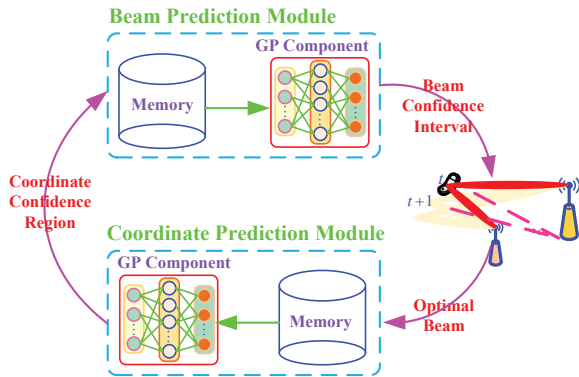


Fig. 6. A sketch of the BA design principle incorporating CP and BP.

Note that the BP module takes as input a scalar or a vector, while the output of the CP module is a Euclidean region, e.g., \mathcal{R} . To tackle this issue, we discretize \mathcal{R} by sampling L points within \mathcal{R} and obtain a discrete point set $\mathcal{D} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_L\}$. For each point $\mathbf{y}_i \in \mathcal{D}$, the BP module is invoked and yields a BCI $\mathcal{I}(\mathbf{y}_i)$ with mean value b_i . Then, by sweeping all beams within $\bigcup_{i=1}^L \mathcal{I}(\mathbf{y}_i)$, we can find the optimal beam b^* . The coordinates for b^* , denoted by \mathbf{y}^* , are chosen as the point in \mathcal{D} whose BCI contains b^* and is also narrowest.

Now, we can present a BA algorithm with both BP and CP, which is summarized in Algorithm 4 for clarity. In each time-slot, the main steps are as follows. In step 3-(1), we first determine a CCR via Algorithm 3 based on the BEs \mathcal{D} . We discretize the CCR and obtain a discrete point set \mathcal{D} in step 3-(2). In step 3-(3), we estimate a BCI for each sampled point. By sweeping all beams within the union of all BCIs in step 3-(4), we can find the optimal beam b_{n+1} . In step 3-(5), based on the optimal beam, we further determine the optimal

coordinates \mathbf{x}_{n+1} . In steps 3-(6) and 3-(7), we perform data transmission and update the experiences, respectively.

Algorithm 4: Tx (or Rx) Beam Alignment with Both BP and CP

```

1 input: initial BP dataset  $\mathcal{D} = \{(b_1, \mathbf{x}_1), \dots, (b_{n_0}, \mathbf{x}_{n_0})\}$ ;
    $c\sigma$  criterion
2 initial: construct CP dataset  $\mathcal{S} = \{(\mathbf{x}_1, b_1), \dots, (\mathbf{x}_{n_0}, b_{n_0})\}$ ;
   let time-slot counter  $n = n_0 + 1$ 
3 loop (in each time-slot  $n$ )
   (1) determine coordinate confidence region via Algorithm 3
   (2) sample CCR to obtain a point set  $\mathcal{D} = \{\mathbf{y}_1, \dots, \mathbf{y}_L\}$ 
   (3) determine BCI  $\mathcal{I}(\mathbf{y}_i)$  for each point  $\mathbf{y}_i$  via BP module
   (4) sweep beams within  $\bigcup \mathcal{I}(\mathbf{y}_i)$  to find optimal beam  $b_{n+1}$ 
   (5) determine optimal coordinates  $\mathbf{x}_{n+1}$  for beam  $b_{n+1}$ 
   (6) perform data transmission with found optimal beam
   (7) update  $n$ ,  $\mathcal{D}$  and  $\mathcal{S}$ :  $n \leftarrow n + 1$ ,
        $\mathcal{D} \leftarrow \mathcal{D} \cup \{(b_n, \mathbf{x}_n)\}$  and  $\mathcal{S} \leftarrow \mathcal{S} \cup \{(\mathbf{x}_n, b_n)\}$ 
end-loop

```

In Algorithm 4, it is assumed that the initial dataset $\mathcal{D} = \{(b_1, \mathbf{x}_1), \dots, (b_{n_0}, \mathbf{x}_{n_0})\}$ is available. Typically, the dataset can be collected in advance or online (but the frequency of measuring the coordinates is much lower than that of BA). If this dataset is unavailable or difficult to obtain, it is still able to predict beams efficiently via GPs. Note that one can always obtain a dataset like $\mathcal{D}' = \{(t_1, b_1), \dots, (t_{n_0}, b_{n_0})\}$ ⁷. Then, Algorithm 1 can be used with only a minor modification, i.e., to replace the coordinates \mathbf{x} by the time t . For completeness, the modified algorithm is summarized in Algorithm 4'.

Algorithm 4': Tx (or Rx) Beam Alignment with Beam Dataset

```

1 input: initial beam dataset  $\mathcal{D}' = \{(t_1, b_1), \dots, (t_{n_0}, b_{n_0})\}$ ;
    $c\sigma$  criterion
2 initial: let time-slot counter  $n = n_0 + 1$ 
3 loop (in each time-slot  $n$ )
   (1) maximize LML to optimize GP predictive model
   (2) compute covariance matrix/vector/scalar  $\mathbf{K}$ ,  $\mathbf{k}_*$  and  $k_{**}$ 
   (3) determine transmit BCI according to (9) - (12)
   (4) sweep transmit BCI to search optimal beam  $b_{n+1}^*$ 
   (5) perform data transmission with optimal beam  $b_{n+1}^*$ 
   (6) update  $n$ ,  $\mathcal{D}'$ :  $n \leftarrow n + 1$ ,  $\mathcal{D}' \leftarrow \mathcal{D}' \cup \{(t_n, b_n)\}$ 
end-loop

```

Remark 5.1 Compared with Algorithm 1 and Algorithm 2, the most important feature of Algorithm 4 is that there is no need to measure the coordinates in real-time. However, since two GPs are introduced in Algorithm 4, it may consume more computing and storage resources. Compared with Algorithm 1, Algorithm 2 and Algorithm 4, which are mainly model-driven, Algorithm 4' is more inclined to be data-driven.

Up to now, we have focused on single-beam (or beam pairs) training and tracking. To train and track multiple beams, we can employ multiple GPs to construct multiple mappings and in each time-slot we sweep all beams within the BCIs provided

⁷Note that b_i is the beam at time-slot i , which can be obtained via hierarchical or exhaustive search or other algorithms. Note that t_i can be the real time or simply i/T_0 with T_0 a normalized (time) constant.

by all GPs. The detailed operations are similar to the case of single-beam training and tracking, which are omitted.

VI. NUMERICAL RESULTS

In this section, simulation results are provided to demonstrate the performance of the proposed BA algorithms. The first subsection describes the simulation environment, while the second subsection provides the simulation results.

A. Simulation Environment

A communication environment shown in Fig. 2 is chosen to evaluate the algorithms. In contrast to Fig. 2, both Tx and Rx are affected by the wind. It should be pointed out that although the specific environment is chosen, our algorithms also apply to other environments. In Section III-A, we have derived the expressions of Tx beam direction vector and AoA when given the Tx coordinates $P(x_t, y_t, z_t)$. Let β_t and α_t be the Tx elevation angle and azimuth angle, respectively. Then, the Tx coordinates $P(x_t, y_t, z_t)$ can be calculated as

$$\begin{aligned} x_t &= H \sin(\beta_t) \cos(\alpha_t), & y_t &= H \sin(\beta_t) \sin(\alpha_t) \\ z_t &= H \cos(\beta_t). \end{aligned} \quad (20)$$

In practice, the wind often disrupts the (stable) beam along a fixed direction within an appropriate time-interval. Let U be the time-interval. In each $[jU, (j+1)U)$, the azimuth α is fixed to a constant. Since the expressions in (20) are obtained under an ideal case, we add some stochastic elements into them to model possible non-ideal factors (e.g., the deformation of the pillar). Specifically, H in (20) is modified as

$$H_t = H_0 \sqrt{\sin^2(k\beta_t + n_1) + \cos^2(k\beta_t + n_2)}, \quad (21)$$

where H_0 is the original length of the pillar, k is distributed as $\mathcal{N}(1, \sigma_\beta^2)$ and n_i ($i = 1, 2$) are distributed as $\sim \mathcal{N}(0, \sigma_i^2)$.

To incorporate external effects in terms of time-dimension, a damped periodical stochastic process is used to model the change of the elevation angle β . The stochastic process in the interval $[0, U)$ can be mathematically expressed as

$$\beta_0(t) = (Ad(t) \sin(2\pi f_d(t - t_0) + \phi_0) + b(t) + N(t)) \cdot \mathbb{I}_{\{0 \leq t < U\}}. \quad (22)$$

The meaning of the parameters in (22) is explained below:

- A is a random variable characterizing the angle spread. It reflects the initial strength of the external effects. $d(t)$ describes the attenuated behavior of the external effects. It is a decreasing function, e.g., an exponential function $\exp(-\eta t)$ with η a positive random variable.
- t_0 and ϕ_0 denote initial random time and initial random phase. f_d characterizes the frequency of the disturbance effects. t_0 is assumed to be uniformly distributed in an interval $[a_1, b_1]$, and it is similar for ϕ_0 and f_d .
- $b(t)$, a constant function, denotes the stable beam direction. $N(t)$ absorbs other possible stochastic disturbance factors. In each time-slot, it randomly and independently takes a value within a set, e.g., $[-0.5, 0.5]$.

A complete expression of β can be expressed as

$$\beta(t) = \sum_{n=0}^{\infty} \beta_0(t - nU). \quad (23)$$

For clarity, an instance is provided in Fig. 7. The simulation setting of Rx is similar to that of Tx, which is omitted.

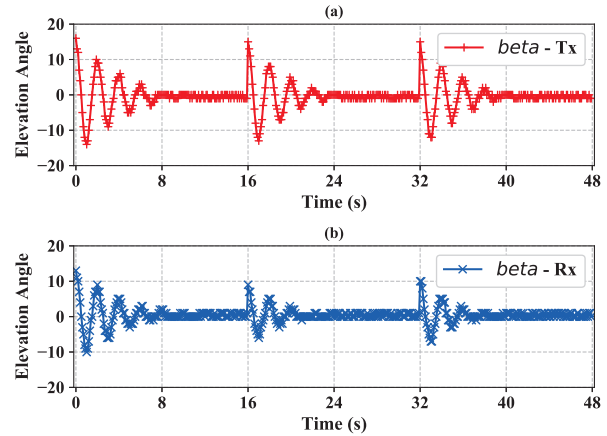


Fig. 7. An instance of two stochastic processes of the elevation angle β .

The sizes of Tx and Rx codebooks \mathcal{C}_T and \mathcal{C}_R satisfy $M_T = N_T$ and $M_R = N_R$. The channel model in (2) consists of both LOS path and NLOS paths. The NLOS part of the channel model consists of 3 clusters and each cluster consists of 4 channel paths [27]. The AoD and AoA of the LOS path are determined according to (21), while the AoDs and AoAs of all NLOS paths are uniformly distributed in an interval, e.g., $[0, 2\pi]$ or $[0, \pi]$. The LOS path gain is distributed as $\alpha_{\text{LOS}} \sim \mathcal{CN}(1/\sqrt{2} + i/\sqrt{2}, 0.1)$. The average power ratio between the LOS path gain α_{LOS} and each NLOS path gain α_{NLOS} is 10dB. The path gain of each NLOS path is distributed as $\mathcal{CN}(0, \sigma_{\text{NLOS}}^2)$, where σ_{NLOS}^2 is calculated according to the gain of the LOS path. The path loss or attenuation d^r in (2) is fixed to $d^r = N_T N_R / 16$. The SNR γ is defined as $\gamma = P / \sigma_n^2$ with P and σ_n^2 given in (3). U in (22) is set to $U = 160$.

B. Performance of Beam Alignment

In this subsection, we demonstrate the performance of the proposed BA algorithms. Since the BA algorithms proposed in this paper, i.e., Algorithm 1, Algorithm 2 and Algorithm 4, are all designed based on GP learning, they are abbreviated as GPL-1, GPL-2 and GPL-4, respectively. In addition to the conventional hierarchical search (HS) based BA algorithm [4], several ML based BA algorithms proposed recently, i.e., direct upper confidence bound (DUCB) [16], hierarchical posterior matching (HPM) [20], partially observable Markov decision process (POMDP) with some modifications [21] and stochastic bandit learning (SBL) [19], are adopted for comparison. EAR [19] and probability of successful alignment (PSA) are used as performance metrics to evaluate different algorithms.

First, we evaluate different BA algorithms by comparing the EAR and PSA performances vs. the size of available training set T . The algorithms designed in this paper are implemented

online, and the first T samples are chosen as the training set and the remaining samples are used for testing. It is well known that the size of the training set has a significant impact on the performance of most ML algorithms. T of a small value corresponds to a small sample performance of a learning algorithm. Note that the small sample case is very common in wireless communications, since the channels often vary rapidly. For GPL-1, the size of its initial training set, i.e., n_0 in GPL-1, is set to $n_0 = 10$. The average EAR performance of different BA algorithms is shown in Fig. 8.

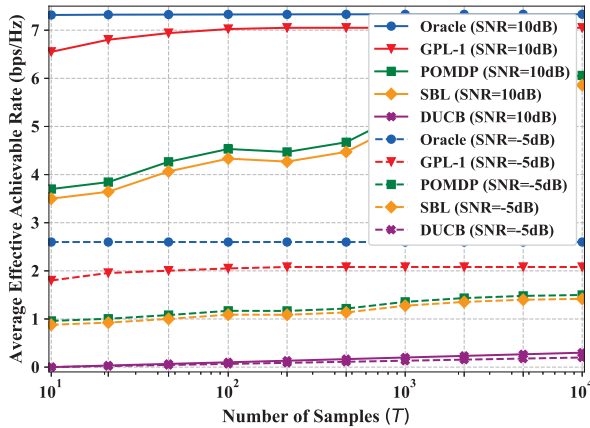


Fig. 8. The average EAR performance vs. the size of the training set: $f_d = 0.3$ and $A \sim U(3, 29)$.

It is seen that GPL-1 achieves the best EAR performance among the four BA algorithms and approaches the oracle-aided ideal algorithm which accesses the optimal beams without any training overhead. It is also seen that GPL-1 can achieve a good performance for a small sample setting (e.g., when $T \geq 46 \approx 10^{1.66}$), which is appealing in mmwave communications. The reason for this is three-fold. Firstly, in contrast to the data-driven design paradigm, the algorithms in this paper are designed based on the paradigm that is driven by (communication) model and empirical data collaboratively [28]. Secondly, instead of predicting a single beam accurately, only an interval that contains the beam of interest is required to predict, which further reduces the number of required samples. Thirdly, an appealing and important advantage of Bayesian learning is that a good performance can be achieved even for a small sample setting, which is attributed to the model averaging function of Bayesian learning. In contrast, both POMDP and SBL can achieve a good performance only under a large sample setting. As the size of the available training set decreases, their performance degenerates quickly.

Fig. 9 demonstrates the PSA performance of different BA algorithms. It is not surprising that GPL-1 achieves the best PSA performance for the two cases (i.e., low SNR and high SNR), which further shows the effectiveness and superiority of GPL-1 under the small sample setting. It is observed that for the two performance metrics (i.e., EAR and PSA), DUCB achieves the worst performance among the four algorithms. The reason for this is that the considered mmwave channels vary rapidly, and DUCB is mainly applicable to slow-varying

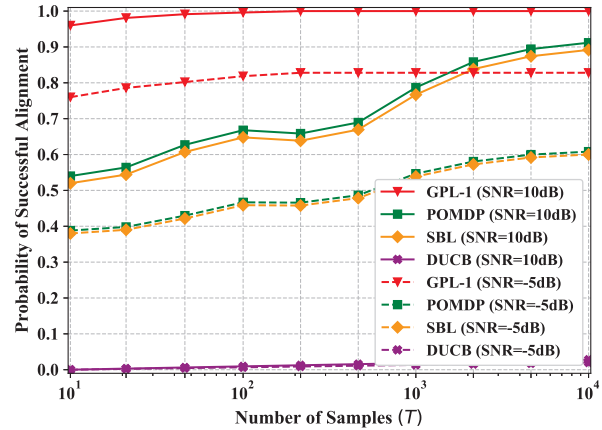


Fig. 9. The PSA performance vs. the size of the training set: $f_d = 0.3$ and $A \sim U(3, 29)$.

mmwave channels. POMDP performs a little better than SBL, since more information can be incorporated into MDP states, which helps to make better decisions.

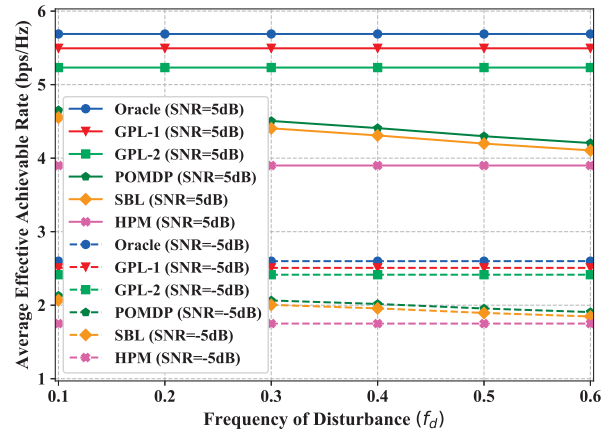


Fig. 10. The average EAR performance vs. the rate of environment change: $A \sim U(15, 30)$, $T = 160$ for GPL-x and $T = 3200$ for SBL and POMDP.

To further evaluate the proposed BA algorithms, we compare the EAR and PSA performance for changing environments which are characterized by f_d in (22). The simulation results of GPL-2 are also provided. The EAR performance of different BA algorithms is shown in Fig. 10.

It is observed that for two SNR settings, GPL-1 achieves the best EAR performance among the five BA algorithms and GPL-2 achieves the second best one. More importantly, both GPL-1 and GPL-2 can achieve a good EAR performance for the changing environments. The reason for this is that the two BA algorithms predict beams based on coordinate positions. As long as the coordinates can be obtained, the GPs based designs can provide good predictions. In contrast, the rate of environment change has an influence on SBL and POMDP. As the environment changes faster, the EAR performance of the two algorithms decreases accordingly. It is also observed that SBL performs better than HPM. The reason for this is that beam index difference technique is incorporated into SBL,

making it well-suited for varying environments.

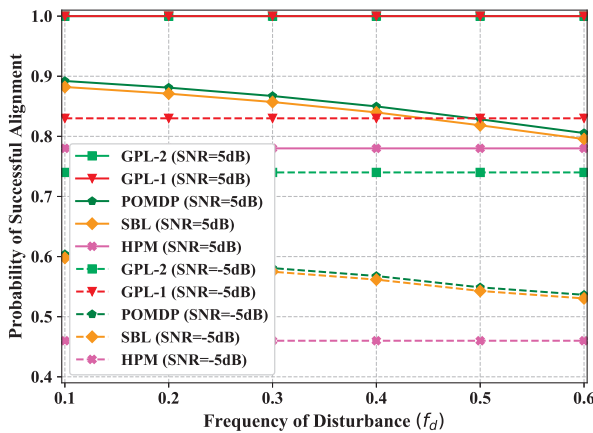


Fig. 11. The PSA performance vs. the rate of environment change: $A \sim U(15, 30)$, $T = 160$ for GPL-x and $T = 3200$ for SBL and POMDP.

Fig. 11 shows the PSA performance of the five algorithms for the changing environments. Similar to the EAR metric, GPL-1 and GPL-2 achieve the best PSA performance, while the PSA performance of SBL and POMDP decreases as the environment changes faster. Observing from Fig. 10 and Fig. 11, one can find that GPL-1 achieves a better performance than GPL-2 in terms of both EAR and PSA. The reason for this is two-fold. Firstly, gathering together the experiences of Tx and/or Rx incurs some overhead, e.g., consume a part of time resource. Secondly, GPL-1 optimizes the GP predictive model in each time-slot, while GPL-2 optimizes the predictive model once and then uses the model for a long time. Note that optimizing the predictive model involves solving a complex optimization problem, which may be prohibitive in some cases.

Finally, we demonstrate the EAR and PSA performance of different BA algorithms for varying SNR, as shown in Fig. 12 and 13. The simulation results of GPL-4 are provided as well. It is observed that GPL-1 and GPL-4 achieve a better performance (in terms of both EAR and PSA) than the existing three BA algorithms. Moreover, the two GP learning based BA algorithms are immune to the environment change, which is desirable in practice. It is seen that GPL-1 achieves a bit better performance than GPL-4. The reason for this is that the estimated coordinates may not be very accurate, which may affect the final performance. However, an important advantage of GPL-4 is that there is no need to measure the coordinates in each time-slot, which can save a lot of trouble. As a non-adaptive algorithm, it is not surprising that HS achieves the worst performance among the five BA algorithms.

VII. CONCLUSION

In this paper, aiming at achieving the goal of obtaining a good performance for the small sample setting, we designed a GP learning based BA algorithm which takes as input the coordinates. Then, we enhanced the BA algorithm from two aspects, i.e., reducing the computational complex to improve the data efficiency and avoiding measuring the coordinates frequently. By extracting and exploiting the correlation between

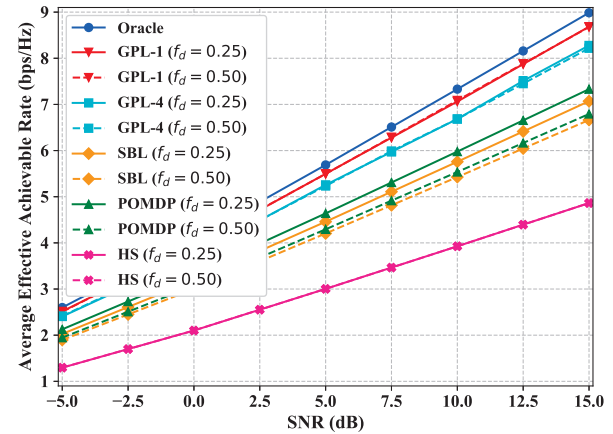


Fig. 12. The average EAR performance vs. SNR: $A \sim U(10, 30)$, $T = 160$ for GPL-x and $T = 3200$ for SBL and POMDP.

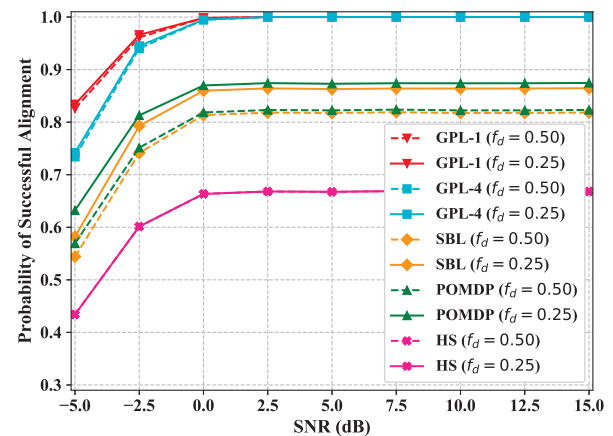


Fig. 13. The PSA performance vs. SNR: $A \sim U(10, 30)$, $T = 160$ for GPL-x and $T = 3200$ for SBL and POMDP.

Tx and Rx, we accomplished the first goal. We completed the second goal by introducing another GP that predicts the coordinates. Thanks to the Bayesian learning based design paradigm, we can obtain a good and robust performance in terms of both EAR and PSA for the small sample setting. Finally, simulation results were provided to demonstrate the effectiveness and superiority of the proposed algorithms.

APPENDIX A PAC-BAYESIAN LEARNING

In this appendix, Gaussian Process (GP) regression and PAC-Bayesian learning are briefly introduced to facilitate developing efficient BA algorithms.

A. Gaussian Processes and Regression

A stochastic process $f(\mathbf{x})$ is called a Gaussian process (GP) if (and only if) for any finite number of points $\mathbf{x}_1, \dots, \mathbf{x}_n$, the joint probability density function, $p(f(\mathbf{x}_1), \dots, f(\mathbf{x}_n))$, is Gaussian [29].⁸ A GP is completely characterized by its mean

⁸ $\{f(\mathbf{x}_n)\}$ are, in fact, random variables, while $\{\mathbf{x}_n\}$ denote indices.

function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$, which is, in a sense, similar to the multi-variate Gaussian distribution. The mean function $m(\mathbf{x})$ and the covariance function $k(\mathbf{x}, \mathbf{x}')$ of a GP $f(\mathbf{x})$ are respectively defined as

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})] \\ k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]. \end{aligned} \quad (24)$$

Accordingly, the GP is written as

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \quad (25)$$

For simplicity, $m(\mathbf{x})$ is assumed to be zero, i.e., $m(\mathbf{x}) = 0$.

The task of GP regression is to infer $f(\mathbf{x}^*)$ for an unseen \mathbf{x}^* , given noisy observations $\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid y_i = f(\mathbf{x}_i) + w_i, w_i \sim N(0, \sigma_w^2), i = 1, \dots, n\}$. According to the definition of GPs, the joint distribution of $\mathbf{y} = [y_1, \dots, y_n]^T$ and the test output $f_* = f(\mathbf{x}^*)$ is given by

$$\begin{bmatrix} \mathbf{y} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} k(\mathbf{X}, \mathbf{X}) + \sigma_w^2 \mathbf{I} & k(\mathbf{X}, \mathbf{x}^*) \\ k(\mathbf{x}^*, \mathbf{X}) & k(\mathbf{x}^*, \mathbf{x}^*) \end{bmatrix}\right),$$

where $k(\mathbf{X}, \mathbf{X})$ with $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]$ represents the $n \times n$ matrix of the covariances evaluated at all pairs of the columns of \mathbf{X} , and $k(\mathbf{X}, \mathbf{x}^*)$, $k(\mathbf{x}^*, \mathbf{X})$ and $k(\mathbf{x}^*, \mathbf{x}^*)$ are calculated similarly. The conditional distribution of $f_* = f(\mathbf{x}^*)$, i.e., GP predictive distribution, at a test input \mathbf{x}^* is given by

$$p(f_* | \mathcal{D}, \mathbf{x}^*) \sim \mathcal{N}(\mu(\mathbf{x}^*), \sigma^2(\mathbf{x}^*)) \quad (26)$$

$$\mu(\mathbf{x}^*) = \mathbf{k}_*^T (\mathbf{K} + \sigma_w^2 \mathbf{I})^{-1} \mathbf{y} \quad (27)$$

$$\sigma^2(\mathbf{x}^*) = k_{**} - \mathbf{k}_*^T (\mathbf{K} + \sigma_w^2 \mathbf{I})^{-1} \mathbf{k}_*, \quad (28)$$

where $\mathbf{K} = k(\mathbf{X}, \mathbf{X})$, $\mathbf{k}_* = k(\mathbf{X}, \mathbf{x}^*)$ and $k_{**} = k(\mathbf{x}^*, \mathbf{x}^*)$ are introduced for simplicity.

The covariance function $k(\cdot, \cdot)$, also referred to as a kernel, is crucial for GP regression, because it encodes the prior about the function to be learned, i.e., the specification of the kernel implies a distribution over functions. Briefly speaking, the kernel defines nearness or similarity under the GP perspective [25]. However, an arbitrary function of input pairs \mathbf{x} and \mathbf{x}' may not be a kernel. Next, we briefly introduce several kinds of kernels, which are widely used in practice [25].

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

$$k_{\text{SE}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2l^2} \|\mathbf{x} - \mathbf{x}'\|^2\right), \quad (29)$$

where σ_f^2 and l denote the signal variance and length-scale, respectively [25]. The physical meaning of the parameter l is that if we think that the GP varies rapidly, the length-scale l should be shorter. Hence, the degree of variation of a GP is achieved by simply adjusting the parameters of the kernel. Note also that since the SE kernel is infinitely differentiable, the GP with this kernel is smooth.

2) *Periodic Kernel*: The expression of the periodic kernel (PE) is given below

$$k_{\text{per}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{2}{l^2} \sin^2\left(p^{-1}\pi \|\mathbf{x} - \mathbf{x}'\|^2\right)\right), \quad (30)$$

where p characterizes the periodicity, and the meaning of the other parameters is similar to the SE kernel [30].

3) *Spectral Mixture Kernel*: To provide more flexibility, the spectral mixture (SM) kernel takes the form

$$k_{\text{SM}}(\mathbf{x}, \mathbf{x}') = \sum_{q=1}^Q a_q \frac{|\Sigma_q|^{0.5}}{(2\pi)^{D/2}} \exp\left(-\frac{1}{2} \|\Sigma_q^{0.5}(\mathbf{x} - \mathbf{x}')\|^2\right) \cos\langle \mathbf{x} - \mathbf{x}', 2\pi \boldsymbol{\mu}_q \rangle, \quad (31)$$

where $\{\alpha_q\}$, $\{\Sigma_q\}$ and $\{\boldsymbol{\mu}_q\}$ are mixture weights, bandwidths (inverse length-scales) and frequencies, respectively. The SM kernel is more expressive, which helps to discover interesting and important structures and patterns of a stochastic process.

B. PAC-Bayesian Learning

A learning task is characterized by an unknown data distribution \mathcal{D} over a domain \mathcal{Z} , from which a set of m points $\mathcal{S} = \{z_i \mid z_i \sim \mathcal{D}, i = 1, \dots, m\}$. Let $\mathcal{S} \sim \mathcal{D}^m$ represent the i.i.d. (independent and identically distributed) sampling of m data points. In supervised learning, each z_i takes the form $z_i = (\mathbf{x}_i, y_i)$, where $\mathbf{x}_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ are input feature and target label, respectively. Given \mathcal{S} , the goal is to find a hypothesis (usually a function $h: \mathcal{X} \rightarrow \mathcal{Y}$) from a hypothesis space \mathcal{H} that can make a good prediction on an unseen input feature $\mathbf{x}^* \sim \mathcal{D}_{\mathbf{x}}$. The quality of the prediction is measured by a loss function $L: \mathcal{H} \times \mathcal{Z} \rightarrow \mathbb{R}$. The design goal is to minimize the expected error for the data distribution, i.e.,

$$\mathcal{L}(h, \mathcal{D}) = \mathbb{E}_{z^* \sim \mathcal{D}} L(h, z^*). \quad (32)$$

Since \mathcal{D} is unknown in practice, the empirical error is used instead, which is given by

$$\hat{\mathcal{L}}(h, \mathcal{S}) = \frac{1}{m} \sum_{i=1}^m L(h, z_i). \quad (33)$$

Note that since only a finite number of data points are used for training, the uncertainty of the predictions is inevitable. For this reason, we are concerned with randomized predictors, i.e., the probability measures on the hypothesis space \mathcal{H} . Let $\mathcal{M}(\mathcal{H})$ represent a set of probability measures over \mathcal{H} . Two probability measures, i.e., the prior $P \in \mathcal{M}(\mathcal{H})$ and the posterior $Q \in \mathcal{M}(\mathcal{H})$, are considered. P and Q also denote the respective probability densities. It should be pointed out that the prior should be independent of the observed data set \mathcal{S} , while the posterior may depend on it. The performance of a randomized predictor is characterized by the Gibbs error

$$\mathcal{L}(Q, \mathcal{D}) = \mathbb{E}_{h \sim Q} \mathcal{L}(h, \mathcal{D}). \quad (34)$$

Similarly, its empirical counterpart is defined as

$$\hat{\mathcal{L}}(Q, \mathcal{S}) = \mathbb{E}_{h \sim Q} \hat{\mathcal{L}}(h, \mathcal{S}). \quad (35)$$

The generalization error (GE) $\mathcal{L}(Q, \mathcal{D})$ is unknown due to \mathcal{D} . Hence, the empirical error (EE) $\hat{\mathcal{L}}(Q, \mathcal{S})$ is considered instead, which may result in overfitting and thus poor generalization performance. It is meaningful to bound the unknown GE based on the EE. The PAC-Bayesian learning provides such a guarantee with a high probability [26].

Lemma 1. *Given a data distribution \mathcal{D} , a hypothesis space \mathcal{H} , a loss function $L: \mathcal{H} \times \mathcal{Z} \rightarrow \mathbb{R}$, a prior distribution*

$P \in \mathcal{M}(\mathcal{H})$, a confidence level $\delta \in (0, 1)$, and $\beta > 0$, with probability at least $1 - \delta$ over samples $\mathcal{S} \sim \mathcal{D}^m$, $\forall Q \in \mathcal{M}(\mathcal{H})$ the following inequality holds true:

$$\mathcal{L}(Q, \mathcal{D}) \leq \hat{\mathcal{L}}(Q, \mathcal{S}) + \beta^{-1} (D_{\text{KL}}(Q||P) - \ln \delta + \Psi(\beta, m)), \quad (36)$$

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

In practice, we are particularly interested in the posterior Q^* that yields the lowest GE. For this purpose, it is natural to minimize the bound $\hat{\mathcal{L}}(Q, \mathcal{S}) + \beta^{-1} D_{\text{KL}}(Q||P)$ in (36), i.e.,

$$Q^*(h) = \arg \min_{Q \in \mathcal{M}(\mathcal{H})} \hat{\mathcal{L}}(Q, \mathcal{S}) + \beta^{-1} D_{\text{KL}}(Q||P).$$

Fortunately, the optimal distribution Q^* , known as the optimal Gibbs posterior [31], has a closed-form and is given by

$$Q^*(h) = \frac{P(h) \exp(-\beta \hat{\mathcal{L}}(h, \mathcal{S}))}{\mathbb{E}_{h \sim P} [\exp(-\beta \hat{\mathcal{L}}(h, \mathcal{S}))]}. \quad (37)$$

In a probabilistic setting, the loss function L is defined as the negative log-likelihood of the data, i.e.,

$$L(h, \mathbf{z}_i) = -\log p(\mathbf{z}_i | h).$$

Let $\beta = m$. Then, the optimal Gibbs posterior coincides with the Bayesian posterior

$$Q_{P, \mathcal{S}}^*(h) = \frac{P(h) \prod_{i=1}^m p(\mathbf{z}_i | h)}{\int_{\mathcal{H}} P(h) \prod_{i=1}^m p(\mathbf{z}_i | h) dh}. \quad (38)$$

APPENDIX B PROOF OF THEOREM 1

Let $Q : \mathcal{Z}^m \times \mathcal{M}(\mathcal{H}) \rightarrow \mathcal{M}(\mathcal{H})$ be an arbitrary learner, which takes in a data-set of size m and a prior and outputs a posterior. For a data-set \mathcal{S} and a prior P , the resultant posterior is denoted by $Q(\mathcal{S}, P)$. Let $\mathcal{L}(Q, \mathcal{D}, Q, n)$ represent the GE of a learner Q with a hyper-posterior Q on a data distribution \mathcal{D} when n data points are sampled from \mathcal{D} , i.e.,

$$\mathcal{L}(Q, \mathcal{D}, Q, n) = \mathbb{E}_{\mathcal{S} \sim \mathcal{D}^n} \mathbb{E}_{P \sim Q} \mathbb{E}_{h \sim Q(\mathcal{S}, P)} \mathbb{E}_{\mathbf{z} \sim \mathcal{D}} L(h, \mathbf{z}). \quad (41)$$

Similar to the derivation of Theorem 2 in [32], we leverage Lemma 1 to bound the GE of TBP and RBP when learning is done by Q . To utilize Lemma 1, the hypothesis space, prior and loss function should be appropriately redefined. A ‘‘tuple hypothesis’’ is defined as $f = (P, h)$, where $P \in \mathcal{M}(\mathcal{H})$ and $h \in \mathcal{H}$. The ‘‘prior over hypothesis’’ $\pi = (\mathcal{P}, P)$ is defined as a distribution over $\mathcal{M}(\mathcal{H}) \times \mathcal{H}$ in which P is first sampled from \mathcal{P} and h is then sampled from P . According to Lemma 1, the ‘‘posterior over hypothesis’’ can be any distribution over $\mathcal{M}(\mathcal{H}) \times \mathcal{H}$, in particular, $\rho = (Q, Q(\mathcal{S}, P))$ with $\mathcal{S} \in \{\mathcal{S}_T, \mathcal{S}_R\}$, in which P is first sampled from Q and h is then sampled from $Q = Q(\mathcal{S}, P)$.

Now, we bound the GE of TBP and RBP. The KL divergence between ρ and π is calculated as

$$\begin{aligned} D_{\text{KL}}(\rho||\pi) &= \mathbb{E}_{f \sim \rho} \log \frac{\rho(f)}{\pi(f)} = \mathbb{E}_{P \sim Q} \mathbb{E}_{h \sim Q(\mathcal{S}, P)} \log \frac{Q(P)Q(\mathcal{S}, P)(h)}{\mathcal{P}(P)P(h)} \\ &= \mathbb{E}_{P \sim Q} \log \frac{Q(P)}{\mathcal{P}(P)} + \mathbb{E}_{P \sim Q} \mathbb{E}_{h \sim Q(\mathcal{S}, P)} \log \frac{Q(\mathcal{S}, P)(h)}{P(h)} \\ &= D_{\text{KL}}(Q||\mathcal{P}) + \mathbb{E}_{P \sim Q} D_{\text{KL}}(Q(\mathcal{S}, P)||P). \end{aligned}$$

By applying Lemma 1 and letting $\beta = m = n$ in (37), we can obtain that for any confidence level $\delta_0 \in (0, 1]$, the inequality in (39) holds uniformly over $(Q, Q) \in \mathcal{M}(\mathcal{M}(\mathcal{H})) \times \mathcal{M}(\mathcal{H})$. Let $\delta_0 = \delta/2$. The use of the union bound yields (40), i.e., the upper bound of the sum of the GEs of TBP and RBP.

Let $\mathcal{L}(Q, Q, n) = \mathcal{L}(Q, \mathcal{D}_T, Q, n) + \mathcal{L}(Q, \mathcal{D}_R, Q, n)$. For convenience, let $C(\delta, n) = 2/n \ln(2/\delta) + 2/n \Psi(n, n)$. When the Gibbs posterior is chosen as the learner, inequality (40) can be simplified as

$$\mathbb{P} \left\{ \mathcal{L}(Q, Q, n) \leq -\frac{1}{n} \mathbb{E}_{P \sim Q} [\ln Z(\mathcal{S}_T, P) + \ln Z(\mathcal{S}_R, P)] + \frac{2}{n} D_{\text{KL}}(Q||\mathcal{P}) + C(\delta, n) \right\} \geq 1 - \delta, \quad (42)$$

where $Z(\mathcal{S}_T, P) = \mathbb{E}_{h \sim P} [\exp(-n \hat{\mathcal{L}}(h, \mathcal{S}_T))]$ and $Z(\mathcal{S}_R, P)$ is defined similarly.

The optimal hyper-posterior distribution Q^* shall minimize the upper bound in (42), i.e.,

$$J(Q) = -\frac{1}{n} \mathbb{E}_{P \sim Q} [Z(\mathcal{S}_T, P) + Z(\mathcal{S}_R, P)] + \frac{2}{n} D_{\text{KL}}(Q||\mathcal{P}),$$

where $C(\delta, n)$ is omitted since it is independent of Q . According to (37), the optimal posterior Q^* is given by

$$Q^*(P) = \frac{\mathcal{P}(P) \exp([Z(\mathcal{S}_T, P) + Z(\mathcal{S}_R, P)]/2)}{R(\mathcal{P}, \mathcal{S}_T, \mathcal{S}_R)},$$

which completes the proof.

REFERENCES

- [1] C. Chen, O. Kedem, C. R. C. M. d. Silva, and C. Cordeiro, ‘‘Millimeter-wave fixed wireless access using ieee 802.11ay,’’ *IEEE Commun. Mag.*, vol. 57, no. 12, pp. 98–104, December 2019.
- [2] 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.
- [3] H. Zhang, S. Venkateswaran, and U. Madhow, ‘‘Channel modeling and mimo capacity for outdoor millimeter wave links,’’ in *2010 IEEE WCNC*, April 2010, pp. 1–6.
- [4] 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.
- [5] A. Alkhateeb, O. El Ayach, G. Leus, and R. Heath, ‘‘Channel estimation and hybrid precoding for millimeter wave cellular systems,’’ *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 831–846, Oct 2014.
- [6] 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.
- [7] X. Gao, L. Dai, Z. Chen, Z. Wang, and Z. Zhang, ‘‘Near-optimal beam selection for beamspace mmwave massive MIMO systems,’’ *IEEE Commun. Lett.*, vol. 20, no. 5, pp. 1054–1057, May 2016.

$$\mathbb{P}\left\{\mathcal{L}(\mathcal{Q}, \mathcal{D}, Q, n) \leq \mathbb{E}_{P \sim \mathcal{Q}} \hat{\mathcal{L}}(Q(\mathbf{S}, P), \mathbf{S}) + \frac{1}{n} \left(D_{\text{KL}}(\mathcal{Q}||P) + \mathbb{E}_{P \sim \mathcal{Q}} D_{\text{KL}}(Q(\mathbf{S}, P)||P) + \ln \frac{1}{\delta_0} + \Psi(n, n) \right)\right\} \geq 1 - \delta_0. \quad (39)$$

$$\mathbb{P}\left\{\mathcal{L}(\mathcal{Q}, \mathcal{D}_T, Q, n) + \mathcal{L}(\mathcal{Q}, \mathcal{D}_R, Q, n) \leq \mathbb{E}_{P \sim \mathcal{Q}} \left[\hat{\mathcal{L}}(Q(\mathbf{S}_T, P), \mathbf{S}_T) + \hat{\mathcal{L}}(Q(\mathbf{S}_R, P), \mathbf{S}_R) \right] + \frac{1}{n} \left(2D_{\text{KL}}(\mathcal{Q}||P) + \mathbb{E}_{P \sim \mathcal{Q}} [D_{\text{KL}}(Q(\mathbf{S}_T, P)||P) + D_{\text{KL}}(Q(\mathbf{S}_R, P)||P)] \right) + \frac{2}{n} \ln \frac{2}{\delta} + \frac{2}{n} \Psi(n, n) \right\} \geq 1 - \delta. \quad (40)$$

- [8] 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.
- [9] 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.
- [10] 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. 37328–37348, 2018.
- [11] 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.
- [12] J. C. Aviles and A. Kouki, "Position-aided mm-wave beam training under nlos conditions," *IEEE Access*, vol. 4, pp. 8703–8714, 2016.
- [13] Z. Wei, Y. Zhao, X. Liu, and Z. Feng, "Doa-lf: A location fingerprint positioning algorithm with millimeter-wave," *IEEE Access*, vol. 5, pp. 22678–22688, 2017.
- [14] 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. 10858–10871, Nov 2019.
- [15] V. Va, T. Shimizu, G. Bansal, and R. W. Heath, "Online learning for position-aided millimeter wave beam training," *IEEE Access*, vol. 7, pp. 30507–30526, 2019.
- [16] M. Hashemi, A. Sabharwal, C. E. Koksall, and N. B. Shroff, "Efficient beam alignment in millimeter wave systems using contextual bandits," in *IEEE INFOCOM 2018*, April 2018, pp. 2393–2401.
- [17] 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.
- [18] M. Scalabrin, N. Michelusi, and M. Rossi, "Beam training and data transmission optimization in millimeter-wave vehicular networks," in *2018 IEEE GLOBECOM*, Dec 2018, pp. 1–7.
- [19] J. Zhang, Y. Huang, Y. Zhou, and X. You, "Beam alignment and tracking for millimeter wave communications via bandit learning," *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5519–5533, 2020.
- [20] S. Chiu, N. Ronquillo, and T. Javidi, "Active learning and CSI acquisition for mmwave initial alignment," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2474–2489, 2019.
- [21] M. Hussain, M. Scalabrin, M. Rossi, and N. Michelusi, "Mobility and blockage-aware communications in millimeter-wave vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13072–13086, 2020.
- [22] B. Wang, M. Jian, F. Gao, G. Y. Li, and H. Lin, "Beam squint and channel estimation for wideband mmwave massive MIMO-OFDM systems," *IEEE Trans. Signal Process.*, vol. 67, no. 23, pp. 5893–5908, 2019.
- [23] Y. Chen, Y. Xiong, D. Chen, T. Jiang, S. X. Ng, and L. Hanzo, "Hybrid precoding for wideband millimeter wave mimo systems in the face of beam squint," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2020.
- [24] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing, "Deep kernel learning," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 51. Cadiz, Spain: PMLR, 09–11 May 2016, pp. 370–378.
- [25] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- [26] P. Alquier, J. Ridgway, and N. Chopin, "On the properties of variational approximations of gibbs posteriors," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 8374–8414, 2015.
- [27] O. El Ayach, S. Rajagopal, S. Abu-Surra, Z. Pi, and R. Heath, "Spatially sparse precoding in millimeter wave MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 13, no. 3, pp. 1499–1513, March 2014.
- [28] 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.
- [29] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [30] D. Duvenaud, "Automatic model construction with gaussian processes," Ph.D. dissertation, University of Cambridge, 2014.
- [31] O. Catoni, "Pac-Bayesian supervised classification: The thermodynamics of statistical learning," *Lecture Notes-Monograph Series*, vol. 56, pp. 1–163, 2007.
- [32] R. Amit and R. Meir, "Meta-learning by adjusting priors based on extended PAC-Bayes theory," *2018 ICML*, 2018.



Jianjun Zhang (S'16-M'18) 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. He is currently a Research Fellow of the electrical and electronics engineering with University College London (UCL), U.K. He was the recipient of the Best Paper Award in the IEEE Globecom 2019. His current research interests include machine learning and optimization, intelligent communications, and probability theory and its applications.



Christos Masouros (M'06-SM'14) 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 in 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 Communication Engineering research group, Department of Electrical and Electronic Engineering, and affiliated with the Institute for Communications and Connected Systems, 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, Communications and Radar Co-existence, interference mitigation techniques for MIMO and multi-carrier communications. He was the recipient of the Best Paper Awards in the IEEE Globecom 2015 and IEEE WCNC 2019 conferences, and has been recognized 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, IEEE Transactions on Wireless Communications, the IEEE Open Journal of Signal Processing, and Editor-at-Large for IEEE Open Journal of the Communications Society. 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", "Hybrid Analog / Digital Signal Processing for Hardware-Efficient Large Scale Antenna Arrays" and "Joint Communication and Radar Sensing for Emerging Applications". He is currently an elected member of the EURASIP SAT Committee on Signal Processing for Communications and Networking.