Online Learning for Position-Aided Millimeter Wave Beam Training

Vutha Va, *Student Member, IEEE,* Takayuki Shimizu, *Member, IEEE,* Gaurav Bansal, *Member, IEEE,* and Robert W. Heath, Jr., *Fellow, IEEE,*

***Abstract*—Accurate beam alignment is essential for beam-based millimeter wave communications. Conventional beam sweeping solutions often have large overhead, which is unacceptable for mobile applications like vehicle-to-everything. Learning-based solutions that leverage sensor data like position to identify good beam directions are one approach to reduce the overhead. Most existing solutions, though, are supervised-learning where the training data is collected beforehand. In this paper, we use a multi-armed bandit framework to develop online learning algorithms for beam pair selection and refinement. The beam pair selection algorithm learns coarse beam directions in some predefined beam codebook, e.g., in discrete angles separated by the 3dB beamwidths. The beam refinement fine-tunes the identified directions to match the peak of the power angular spectrum at that position. The beam pair selection uses the upper confidence bound (UCB) with a newly proposed risk-aware feature, while the beam refinement uses a modified optimistic optimization algorithm. The proposed algorithms learn to recommend good beam pairs quickly. When using 16x16 arrays at both the transmitter and receiver, it can achieve on average 1dB gain over the exhaustive search (over 271x271 beam pairs) on the unrefined codebook within 100 time-steps with a training budget of only 30 beam pairs.**

***Index Terms*—Millimeter wave, beam alignment, beam refinement, position-aided, online learning**

I. Introduction

Position information may be leveraged for fast beam alignment in millimeter wave (mmWave) systems [1]–[3]. Such side information is widely available in vehicular applications of mmWave [4]. Inverse fingerprinting is one approach to exploit position information [1], which works in non-line-of-sight (NLOS) channels. The key idea is that machine learning (ML) is used to make recommendations of promising beam pairs based on the location of the target vehicle relative to the base station (BS) and past beam measurements. The intuition here is drawn from the recommender system analogy where the user ID corresponds to the relative location and the past user ratings correspond to past beam measurement results. Given the location, past beam measurements can be input into a learning algorithm that learns to rank promising beam directions. By prioritizing beam training in top-ranked directions, the training overhead can be reduced. In other words, the location information is used to condition the environment, and past beam measurements in the location are used to learn which directions are promising in that environment.

Although efficient, the inverse fingerprinting method has some limitations. First, the approach is offline, which means its use is delayed until the database is collected. Second, also due to being offline, its performance depends entirely on the accuracy of the collected database, which may become stale over time. Online approaches keep collecting new observations during operation, making it possible to improve the database. Third, without any knowledge of the power angular spectrum (PAS), the codebook must uniformly cover the antenna array's field of view (e.g., beams are spaced by the 3dB beamwidths). At a given location, depending on the scatterers in the environment, the PAS will have peaks at some specific angles. By adapting the beams such that their main beam directions match those peaks, we expect gains beyond the generic good-for-all-cases codebook. That is, position-based learning opens up an opportunity to also adapt the beam codebook to the environment. In this paper, we propose an online beam pair selection and refinement algorithm to address the limitations of a completely offline approach.

Our contributions are summarized as follows.

- We propose an online algorithm to learn to select beam pairs with risk-awareness to reduce the probability of severe beam misalignment during the learning. This is done by designing the algorithm to select high-risk beam pairs less often. The proposed solution balances the learning burden on early-stage users and the learning speed.
- We provide regret analyses of the proposed algorithms, which provide insights into the cost of the learning due to the introduction of risk-awareness.
- We formulate the beam pair refinement problem as a continuum-armed bandit (CAB) problem. Our solution is based on the hierarchical optimistic optimization (HOO) [5] with modifications to suit the beam alignment context.
- We integrate the two algorithms into a two-layer online learning solution that learns to select and refine the beam pairs at the same time. The beam pair selection part learns coarse beam directions and the refinement part learns to refine them. This hierarchy is more efficient than learning the refined beam directions directly since now the refinement learning focuses only in promising

Vutha Va and Robert W. Heath, Jr. are with the Wireless Networking and Communications Group, The University of Texas at Austin, TX 78712-1687 USA (e-mail: vutha.va@utexas.edu, rheath@utexas.edu).

Takayuki Shimizu and Gaurav Bansal are with TOYOTA InfoTechnology Center, U.S.A., Inc., Mountain View, CA 94043 USA (e-mail: tshimizu@us.toyota-itc.com, gbansal@us.toyota-itc.com).

V. Va and R. W. Heath, Jr. were supported in part by the U.S. Department of Transportation through the Data-Supported Transportation Operations and Planning (D-STOP) Tier 1 University Transportation Center and by the Texas Department of Transportation under Project 0-6877 entitled "Communications and Radar-Supported Transportation Operations and Planning (CAR-STOP)" and by a gift from TOYOTA InfoTechnology Center, U.S.A., Inc.

directions selected by the beam selection part.

Our numerical result in a vehicle-to-infrastructure (V2I) scenario shows that the integrated solution learns quickly. Using 16×16 arrays at both the transmitter and receiver and a training budget of 30 beam pairs, it achieves an average gain of 1dB over the exhaustive beam search over 271^2 beam pairs in the unrefined codebook within the first 100 time steps. The gain can reach up to about 1.5dB over time. Unlike prior work [6]–[8] that uses simplistic abstract models that match exactly with the underlying statistical assumptions of the problem formulation, we use realistic channels generated by ray-tracing to evaluate our algorithms.

Beam alignment has been investigated intensively in the literature. Several directions have been pursued such as approaches based on beam sweeping [9], [10], angle of arrival and departure (AoA/AoD) estimation [11], [12], black-box function optimization [13], [14], and the use of side-information [1]–[3], [15]. We refer to [1] for a summary of the differences of these approaches for beam alignment. The last category is most related to our work, especially those that use position information.

There are different approaches to position-aided beam alignment. One line of research assumes LOS channels and determines the pointing direction directly from the transmitter and receiver position [16]–[19]. That approach does not require beam training when the position information is accurate. Position error can be translated into uncertainty in the AoA/AoD. A small amount of beam training covering the AoA/AoD uncertainty range can counter the position error [17], [19]. While efficient, the LOS assumption is not always valid. For example, in our ray-tracing simulations, the channel is NLOS about 38% of the time.

In another line of research, a database of past beam measurements indexed by location is used to identify both NLOS and LOS beamforming directions [1]–[3]. The work in [2] proposed to store the most recent successful antenna configuration (having received power larger than some threshold) at each location (defined as grid points). The objective here is to find a direction that can support a link while our objective is to find the beam pair that provides the strongest received power. Also, omni-directional users are assumed in [2], which limits the achievable communication range. A hierarchical search was proposed in [3] assuming users are equipped with horn antennas with different beamwidths. A hierarchical search requires multiple feedbacks, which could be the bottleneck as they are sent via a slow link in the control plane. Unlike [2], [3], our prior work in [1] assumed directional users and performs the beam training at the beam level without hierarchical search requiring only one feedback. Also, the method in [1] uses the probability of being optimal (i.e., alignment probability) as the selection metric. This method can be shown to be optimal in maximizing the probability of finding the strongest beam direction [1]. In this paper, we propose an online version of this optimal selection method from [1]. We also develop a new beam pair refinement method to adapt the beam codebook to the environment to further maximize the beamforming gain.

Another line of research uses position information of surrounding scatterers to further increase the beam alignment efficiency [20], [21]. In [20], a decision-tree algorithm was proposed for a mmWave V2I beam alignment using positions of neighboring vehicles as the features. The training dataset consists of past beam measurement and feature pairs. In a more traditional (i.e., not a data-driven approach leveraging past beam measurements) and abstract setting, the positions of dominant scatterers are assumed known with some error in [21], and a distributed beam alignment framework was proposed. While more information will be helpful, additional sensors and/or procedures are required to obtain those positions and they may be sensitive to the level of knowledge. Incorporating more side informaiton is an interesting direction for future work. In this paper, we will limit to using only the transmitter/receiver position, i.e., this work belongs in the category described in the previous paragraph.

The recent progress in ML has revived interest in applying ML techniques to communications [22]–[24]. Related work that applies ML to beam alignment includes [20], [25], [26]. Position-aided beam prediction was proposed in [20], [25]. Decision tree learning was used in [20], and a learning-to-rank method was used in [25]. A coordinated beamforming solution using deep learning was proposed in [26]. Here, the received training signals via omni reception at a set of coordinating BSs are used as the input to a deep learning model that predicts the beamforming vectors at those BSs to serve a single user. The work in [20], [25], [26] shows that machine learning is valuable for mmWave beam prediction. Unfortunately, the proposed methods are all supervised learning techniques, which assume an offline learning setting and require a separate training data collection phase. In this paper, we propose online learning algorithms using the multi-armed bandit (MAB) framework, which is a special class of reinforcement learning (RL).

Recent applications of RL/MAB for beam training include [6], [7] which uses a partially observable Markov decision process (POMDP) framework, and [8], [27] which uses an MAB framework. The work in [6], [7] addresses tracking problems where the POMDP with known state transition models provides a means to predict the state of the channel enabling an informed choice of the probing beams for good performance. The state transition models, however, are not easily obtained in a practical setting. In [8], the beam alignment problem is solved using an MAB framework with the assumption that the success probability (the received power is larger than some threshold) is a unimodal function of the pointing direction. The efficiency of that solution depends on this unimodal property, which cannot be guaranteed in our setting with random blockage. The work in [27] implemented the UCB1 algorithm and its variants from [28] for antenna state selection for a 2.4GHz IEEE 802.11 system. Note that both [27] and [8] proposed single-play MAB solution, while this paper proposes a multiple-play MAB. The two types of MAB will be described next.

MAB is a useful tool for solving online learning (also called sequential decision-making) problems [29], [30]. The most common form of MAB is the single-play MAB with

a finite number of arms, where only one arm is selected in each time step. The proposed method trains multiple beam pairs (up to the training budget) in each beam alignment attempt. Thus, our beam pair selection problem can be cast as a multiple-play MAB problem (also known as combinatorial bandit) [31], where multiple arms may be tried in each round or time step. In an MAB setting, in each round, the player must decide between using the knowledge obtained so far to select the best arm or explore lesser-known arms, which is called the explore-exploit dilemma. The optimism in the face of uncertainty is a core design idea for balancing the explore-exploit tradeoff, which results in a widely successful family of algorithms known as the upper confidence bound (UCB). Our solution employs the UCB in a multiple-play setting. Most related to our solution is the cascading bandit [32], which performs the same selection procedure as our Algorithm 1 but with a different model to collect the reward measurements. Another important difference is that the reduction to the greedy selection is based on the independent arms assumption in [32], while in our case it is based on the modularity property (i.e., additivity) of the reward signal as a function of the subset of selected arms. Also, we extend beyond the greedy UCB selection by introducing risk-awareness designed to avoid severe beam misalignment during the learning.

We cast our beam pair refinement as a stochastic CAB problem, which has infinitely many arms. CAB assumes the reward function has some smoothness property (e.g., Lipschitz continuous). There are different approaches to solve CAB such as Bayesian optimization (BO) [33], the zooming algorithm [34], and optimistic optimization (OO) [35]. BO does not discretize the arm space but has high complexity. It is more suitable when sampling is expensive or the learning horizon is short. The zooming algorithm and OO rely on smart discretization of the arm space. The zooming algorithm uses an adaptive approach that applies finer discretization in promising regions. This is done using an activation rule that is assumed given to the algorithm, but this rule is a non-trivial problem itself. OO approaches discretize the arm space using a tree and exploit the hierarchy for an efficient search for the best arm. OO approaches designed for stochastic settings include Stochastic Simultaneous OO (StoSOO) and HOO [35]. StoSOO is an explore-first algorithm where the task is to find the best arm given an exploration budget. This does not fit our setting where there is no separate explore and exploit phases. HOO is designed for maximizing the cumulative reward and suits our setting well. Applying HOO in its original form does not work well. We propose three modifications to suit our beam refinement problem.

The rest of the paper is structured as follows. Section II describes our system model and how we generate the data. Section III presents our beam alignment framework and reviews offline beam pair selection methods from [1], which are the basis for our online learning solution. Section IV describes the proposed two-layer online learning algorithm with the beam pair selection in the first layer and beam pair refinement in the second. Section V and Section VI provide the details of the two layers along with some analysis. Numerical evaluations are given in Section VII followed by the conclusions in Section VIII.

II. System and data model

In this section, we describe our model of the communication system and how we generate the data for evaluating the learning algorithms.

A. System model

Our system model is comprised of the channel model, the received signal model, and the codebook. We assume a wideband geometric channel model that is widely used in mmWave simulations [36]. We denote N_{t} and N_{r} the numbers of transmit and receive antennas, L_{p} the number of rays, T the symbol period, $g(\cdot)$ the combined response of matched and lowpass filtering, $\boldsymbol{a}_{\mathrm{t}}$ and $\boldsymbol{a}_{\mathrm{r}}$ the normalized transmit and receive array steering vectors, α_ℓ the complex channel gain, τ_ℓ the delay, ϕ_ℓ^{A} and ϕ_ℓ^{D} the azimuth AoA and AoD, θ_ℓ^{A} and θ_ℓ^{D} the elevation AoA and AoD of the ℓ-th path, and $(\cdot)^*$ the conjugate transpose operator. The channel at the delay tap m is given by

$$\boldsymbol{H}[m] = \sqrt{N_{\mathrm{r}}N_{\mathrm{t}}} \sum_{\ell=0}^{L_{\mathrm{p}}-1} \alpha_\ell g(mT-\tau_\ell)\boldsymbol{a}_{\mathrm{r}}(\theta_\ell^{\mathrm{A}},\phi_\ell^{\mathrm{A}})\boldsymbol{a}_{\mathrm{t}}^*(\theta_\ell^{\mathrm{D}},\phi_\ell^{\mathrm{D}}). \tag{1}$$

We parametrize this channel using ray-tracing assuming single antennas at both the transmitter and receiver. This means α_ℓ is the channel gain of a single antenna, and the factor $\sqrt{N_{\mathrm{r}}N_{\mathrm{t}}}$ is needed to express the array gain (since $\boldsymbol{a}_{\mathrm{t}}$ and $\boldsymbol{a}_{\mathrm{r}}$ are normalized).

We assume an analog beamforming architecture with one RF chain. Denote L the channel length, P_{t} the transmit power, $s[k]$ the known training signal, $\boldsymbol{H}_\ell = \sqrt{N_{\mathrm{r}}N_{\mathrm{t}}}\alpha_\ell \boldsymbol{a}_{\mathrm{r}}(\theta_\ell^{\mathrm{A}},\phi_\ell^{\mathrm{A}})\boldsymbol{a}_{\mathrm{t}}^*(\theta_\ell^{\mathrm{D}},\phi_\ell^{\mathrm{D}})$ the channel matrix corresponding to the ℓ-th path, $v_i[k]$ the complex Gaussian noise $\mathcal{CN}(0,\sigma_v^2)$, and $r(i)$ and $t(i)$ the mappings from the beam pair index i to the combiner $\boldsymbol{w}$ and beamformer $\boldsymbol{f}$ vector indices, the received signal is given by

$$y_i[k] = \sqrt{P_{\mathrm{t}}} \sum_{m=0}^{L-1} s[k-m] \underbrace{\sum_{\ell=0}^{L_{\mathrm{p}}-1} g(mT+\tau_0-\tau_\ell)\boldsymbol{w}_{r(i)}^* \boldsymbol{H}_\ell \boldsymbol{f}_{t(i)}}_{h_i[m]} + v_i[k]. \tag{2}$$

The channel strength is defined as the squared norm of $\boldsymbol{h}_i = [h_i[0],\ldots,h_i[L-1]]^{\mathrm{T}}$, i.e.,

$$\gamma_i = \|\boldsymbol{h}_i\|^2. \tag{3}$$

The effective channel $\boldsymbol{h}_i$ can be estimated from noisy signals, for example using a least-squared estimator [37]. The impact of noise on the beam alignment has been studied in [1], [38], where it is shown that the impact is small for a wideband system. For a clear exposition of the learning algorithms, we assume noise-free γ_i in this paper.

The beam codebook used in this paper is generated using progressive phase-shift [39]. We note that this choice is not

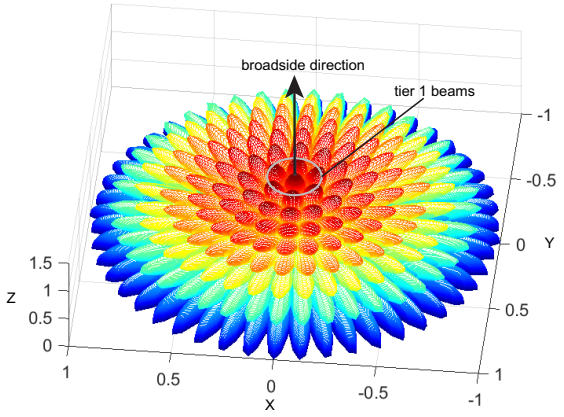


Fig. 1. Beam patterns in our codebook for an 8×8 array. The array is assumed to face upward in the $+z$ direction. The codebook covers the directions in the $+z$ half-space (i.e., assuming no radiation in the backplane).

important, and other codebooks such as the DFT codebook could be used. Uniform planar arrays (UPA) are assumed at both the transmitter and the receiver. With a UPA, each beam is defined by its azimuth ϕ and elevation θ main beam direction. Let $G_{\text{ant}}(\cdot)$ be the antenna element radiation pattern, $\Omega_{\text{y}} = kd_{\text{y}}\sin(\theta)\sin(\phi)$, $\Omega_{\text{x}} = kd_{\text{x}}\sin(\theta)\cos(\phi)$, $k = 2\pi/\lambda$ be the wave number, $\otimes$ denote the Kronecker product, N_{x} and N_{y} be the numbers of elements along the x- and y-axis, and d_{x} and d_{y} be the element spacing in the x- and y-direction, a beam pointing in (θ,ϕ) direction is given by [39]

$$\boldsymbol{a}(\theta,\phi) = \frac{G_{\text{ant}}(\theta,\phi)}{\sqrt{N_{\text{x}}N_{\text{y}}}} \begin{bmatrix} 1 \\ e^{\text{j}\Omega_{\text{y}}} \\ \vdots \\ e^{\text{j}(N_{\text{y}}-1)\Omega_{\text{y}}} \end{bmatrix} \otimes \begin{bmatrix} 1 \\ e^{\text{j}\Omega_{\text{x}}} \\ \vdots \\ e^{\text{j}(N_{\text{x}}-1)\Omega_{\text{x}}} \end{bmatrix}. \quad (4)$$

We assume $d_{\text{x}} = d_{\text{y}} = \lambda/2$ in this work. We assume no backplane radiation and set

$$G_{\text{ant}}(\theta,\phi) = \begin{cases} 1 & \text{if } \theta < 90^\circ \\ 0 & \text{otherwise} \end{cases}. \quad (5)$$

We use (5) for simplicity, but we can replace it with a more sophisticated pattern like that of a patch antenna.

The beams are generated such that they are separated by their 3dB beamwidth in the azimuth and elevation. Fig. 1 shows the beams for an 8×8 array. The procedure starts from the broadside direction. First, fixing the azimuth angle at 0°, the elevation beam direction that crosses the broadside beam at the 3dB point is determined numerically. We then do the same procedure in the azimuth while fixing the elevation angle until all 360° are covered (call these the tier 1 beams). Next, we compute the elevation beam direction that crosses the tier 1 beam at the 3dB beamwidth and determine all the azimuth directions until all 360° are covered. This is repeated until the main beam direction in elevation exceeds 90°, i.e., reaching the backplane direction.

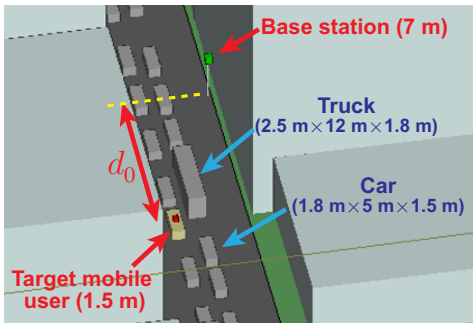


Fig. 2. A snapshot of the ray-tracing simulation in an urban street. The street has two lanes, and two types of vehicles (cars and trucks) are simulated. The BS's antenna is at 7m and the MU's antenna is at 1.5m from the ground.

B. Data model

To generate the data, we parametrize the channel model (1) using a commercial ray-tracing simulator, called Wireless InSite [40]. Ray-tracing ensures spatial consistency in the channels, which is essential to our location-based learning problems. The ray-tracing simulation is shown in Fig. 2. We assume a V2I setting in an urban street canyon. The street has two lanes, and there are two types of vehicles represented by metal boxes: cars ($1.8\text{m} \times 5\text{m} \times 1.5\text{m}$) and trucks ($2.5\text{m}\times12\text{m}\times3.8\text{m}$). The mobile user (MU) is a car on the far side lane from the BS. Because it is larger, a truck can cause blockage to the MU. We assume roof-mounted MU antenna at 1.5m and BS antenna at 7m. We generate the channel in a per snapshot basis. In each snapshot, the MU is placed uniformly at random within a location bin $[d_0 - \sigma_{\text{d}}, d_0 + \sigma_{\text{d}}]$ in the far side lane, and all other vehicles are placed randomly with their gaps following an Erlang distribution. We refer to [1] for a detailed description including the material properties used for the ray-tracing. We set the carrier frequency to 60GHz, $d_0 = 30\text{m}$, and $\sigma_{\text{d}} = 2.5\text{m}$, which corresponds to a location bin of 5m length. Note that our method only requires that the position be accurate enough to identify the location bin. The system bandwidth is set to 1.76GHz, which is used in IEEE 802.11ad. The system bandwidth is used to compute the symbol period used in the channel model (1). Using these parameters, we generate 10,000 channel samples. We assume 16×16 UPA at both the MU and the BS when computing the channel strengths. The codebook described in Section II-A has 271 beams.

III. Position-aided beam alignment

Our beam alignment method relies on the premise that context information (e.g., position) can be used to reduce the training overhead. Using context, the method only needs to train the most promising beam directions. Fig. 3 illustrates the intuition behind the proposed method. Consider the vehicle at position A. In this case, the geometry of the environment allows only two pointing directions: the LOS path and the building reflection path. If the system can learn from past beam alignment experience to identify these two directions, the beam alignment overhead can be reduced to training only these two directions. Now, in practice there will be position error. The

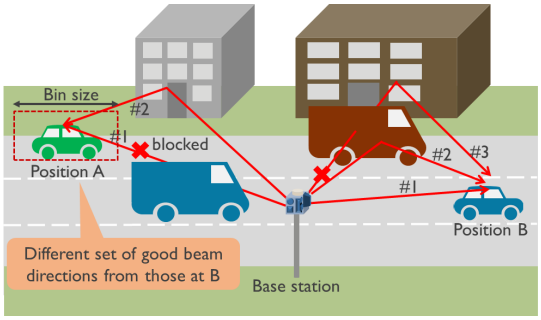


Fig. 3. Illustration of the intuition of the proposed position-aided beam alignment. Consider a vehicle at position A. The geometry of the environment only allows two possible pointing directions: the LOS and the building-reflection path. If the system can learn from past beam measurement results at position A to identify the two beam directions, then beam training can be reduced to just train these two directions. In an actual setting there will be position error. In our proposed solution, we use location bin that allows position inaccuracy to be in the range of the bin size.

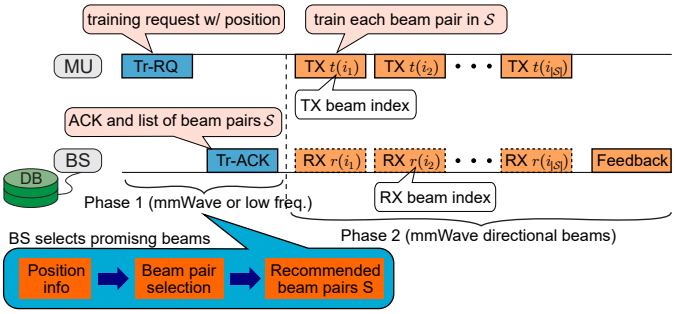


Fig. 4. An illustration of position-aided beam alignment in the uplink. It consists of two phases. Phase 1 is for the training request where the MU position is sent to the BS. The BS uses the position and its learned database to determine a list of promising beam pairs $\mathcal{S}$. In Phase 2, the beam pairs in the list are trained, and a feedback indicating the best beam index is sent at the end. The database used for the beam pair selection is stored and maintained at the BS without any burden on the MU.

larger the error, the more uncertain the pointing directions, and the required beam training overhead will increase. In our simulations, we allow position inaccuracy to be within the 5m location bin, and a training budget of only 30 beam pairs is enough to achieve negligible performance loss compared to the exhaustive search when using 16×16 arrays. (See [1, Section VI.C] for a detailed evaluation of the effect of the location bin size.) Note that the edge effect at bin boundaries can be mitigated by defining overlapping bins. The small number of possible pointing direction given a location has also been observed in measurements. For example, [41, p. 19] reported that in a 28GHz indoor measurement setting with moving pedestrians the received powers were concentrated in no more than three dominant directions.

In this section, we first describe the position-aided beam alignment framework. Then, we explain the beam alignment accuracy metric, based on which the core of the framework, the beam pair selection method, is developed. Lastly, we review the offline beam pair selection method from [1] which will be used in our online solution.

A. Overview of the proposed position-aided beam alignment

The idea of our proposed approach is to learn from past beam measurements experienced in a given discretized location. The measurements themselves and/or learning parameters are stored and maintained at the BS. In the offline learning case as in our prior work [1], a sufficient number of beam measurements are assumed collected before they are used to recommend beam pairs. In the online learning case considered in this paper, the learning parameters of a location bin are updated every time a beam alignment attempt is made and a new set of beam measurements becomes available at that location bin.

Assuming the database is available for making recommendation, we describe how the beam alignment procedure works. Fig. 4 illustrates the position-aided beam alignment, which consists of two phases. We start with the uplink. In Phase 1, the MU sends a training request along with its context information to the BS. In this work, we use position as the context. The BS uses the position and the database it maintains to determine a subset of promising beam directions, denoted by $\mathcal{S}$. The size of the set $\mathcal{S}$ is a system parameter chosen to balance the training overhead and the alignment accuracy. The BS then responds with an acknowledgment and the beam pair subset $\mathcal{S}$ to the MU. Since the beams are not aligned in this phase, a lower frequency control channel or mmWave with a large spreading factor can be used. In Phase 2, the beam pairs in $\mathcal{S}$ are trained and the best beam index is fed back at the end. MmWave with directional beams is used during this phase.

In the downlink, Phase 1 changes slightly. The process starts with the BS sending a training request to the MU, which then responds with an acknowledgment including its position. Next, the BS sends the list of promising beam pairs $\mathcal{S}$. The beam training in Phase 2 is kept the same. This is possible because of the reciprocity in the AoAs/AoDs, where the AoAs become the AoDs and vice versa when reversing the transmitter and receiver role. This AoA/AoD reciprocity only depends on the reciprocity property of electromagnetic waves, which holds when they propagate in passive medium like wireless channels (excluding the device's circuits) [39].

Finally, we emphasize that this beam alignment framework is flexible and different beam pair selection methods can be used for the "Beam pair selection" block. In particular, we proposed an offline learning approach in [1] (briefly described in Section III-C) that assumes the past beam measurements are already collected. In this paper, we propose online algorithms to select the beam pairs, which will be detailed in Section IV.

B. Quantifying beam alignment accuracy

There are many ways to quantify the accuracy of beam alignment. Here, we use the power loss and the power loss probability. The power loss can capture the severity of the misalignment and differentiate whether the current alignment is 3dB or 10dB away from the optimal alignment. This is important because while both cases are misaligned, the former likely still provides a good link while the latter likely cannot.

We start with the definition of the power loss. It is the ratio of the channel strength between the selected beam pair indexed

TABLE I
AN EXAMPLE OF THE DATABASE AT A LOCATION BIN. EACH ROW CONSISTS OF THE MEASURED CHANNEL STRENGTHS OF ALL BEAM PAIR COMBINATIONS FOR A GIVEN CHANNEL REALIZATION. HERE, WE DENOTE I THE BEAM PAIR INDEX AND γ THE CHANNEL STRENGTHS IN dB.

Obsv. No.	Best		2nd best		...	$\lvert\mathcal{B}\rvert$-th best	
	I	γ	I	γ	...	I	γ
1	5	-64.5	159	-69.2	...	346	-95.8
2	159	-70.4	263	-72.6	...	354	-97.1
...	...	...	...	...	...	...	...
N	5	-66.4	258	-68.1	...	2	-82.6

by s and the optimal channel strength given by

$$\xi = \frac{\max_{i\in\mathcal{B}} \gamma_i}{\gamma_s}, \tag{6}$$

where $\mathcal{B}$ denotes the set of all possible beam pairs in the codebook. The beam pair s is selected from the selection set $\mathcal{S}$, and with accurate beam training $s = \arg\max_{i\in\mathcal{S}} \gamma_i$. If the codebook is used without any modification, then $\mathcal{S} \subset \mathcal{B}$ and $\xi \geq 1$ always holds. The proposed online learning method also includes a component to refine the beam pairs to adapt the codebook to the environment, in which case $\xi < 1$ is possible.

We quantify the beam alignment accuracy by the power loss probability defined by

$$P_{\rm pl}(c, \mathcal{S}) = \mathbb{P}\left[\xi > c\right], \tag{7}$$

for some constant $c \geq 1$. We call the case when $c = 1$ the misalignment probability. A related concept to the misalignment probability is the probability of being optimal, given by

$$P_{\rm opt}(\mathcal{S}) = \mathbb{P}[i^\star \in \mathcal{S}], \tag{8}$$

where $i^\star = \arg\max_{i\in\mathcal{B}} \gamma_i$ denotes the index of the optimal beam pair. We note that

$$\begin{aligned} P_{\rm opt}(\mathcal{S}) &= \mathbb{P}[\xi = 1] &(9)\\ &= 1 - \mathbb{P}[\xi > 1] &(10)\\ &= 1 - P_{\rm pl}(1, \mathcal{S}), &(11)\end{aligned}$$

where (10) follows because $\xi \geq 1$ (without refinement). $P_{\rm opt}(\mathcal{S})$ is used in the optimal beam selection method described in the next subsection.

C. Offline inverse fingerprinting beam pair selection

In this subsection, we review two offline beam pair selection methods from [1] which form the basis for our online solution. We begin by describing the offline database. It is assumed that this database has already been collected before it is used for beam pair selection. The database has N observations, where each observation consists of the channel strengths of all beam pairs combinations in $\mathcal{B}$ measured in a location bin. It is assumed that each observation is measured within a beam coherence time so that the spatial channel has negligible change [42]. An example of the database is shown in Table I. We note that some tradeoff between the performance and the cost for collecting and storing the database is possible (see [1] for details).

We now describe the two beam pair selection methods from [1], namely AvgPow and MinMisProb. AvgPow is a heuristic that selects the beam pairs by their average channel strengths. Denote $\bar{\gamma}_i$ the sample average of the channel strength of the i-th beam pair and $\arg\max_{i\in\mathcal{B};M}\{\cdot\}$ the operator that returns the top-M indices, the selection set of size $|\mathcal{S}_{\rm AP}| = M$ can be written as

$$\mathcal{S}_{\rm AP} = \arg\max_{i\in\mathcal{B};M} \left\{\bar{\gamma}_i\right\}. \tag{12}$$

MinMisProb is an optimal selection method that minimizes the misalignment probability. It can be shown that this optimal selection reduces to a greedy selection using the probability of being optimal [1]. Let $|\mathcal{S}_{\rm MMP}| = M$ and denote $\hat{P}_{\rm opt}(i)$ the probability of being optimal of the beam pair i estimated from the database, then

$$\mathcal{S}_{\rm MMP} = \arg\max_{i\in\mathcal{B};M} \left\{\hat{P}_{\rm opt}(i)\right\}. \tag{13}$$

In this paper, we develop a risk-aware online learning version of MinMisProb, where beam pairs selected by the UCB indices are rejected with a probability reflecting their risk of low received powers. When rejected, the replacement pair is selected using both MinMisProb and AvgPow.

IV. PROPOSED TWO-LAYER ONLINE LEARNING

Our aim in this paper is to develop an online learning algorithm for fast and efficient beam alignment. We propose a two-layer online solution to achieve this goal. The idea here is to learn coarse beam directions (quantized by the 3dB beamwidths) that are promising in the first layer and conduct a refinement of those promising directions in the second layer. This kind of hierarchy is efficient because the refinement is only done in promising directions.

An overview of the proposed online learning solution is illustrated in Fig. 5. Note that this is a solution for the "Beam pair selection" block in Fig. 4. The learning happens at the BS as explained earlier. The algorithm runs in an infinite loop, where in each iteration, it recommends a list of beam pairs and updates the learning parameters recorded in the database upon receiving the beam measurements of those pairs. As mentioned in Section III-C, by having the MU transmit, there is no extra feedback overhead to collect the beam measurements. We highlight the groups of blocks that correspond to the learning agent and the environment in Fig. 5. This shows a typical RL setting where the agent optimizes its action through direct interaction with the environment [43]. In our problem, the action is the subset of beam pairs selected for the training, and the environmental response is the beam measurement results. The algorithm starts by running a detection loop for a request for beam training from the user. If a request is detected, the position (other context can also be used, but we focus on position) is extracted from the training request packet and input to the beam pair selection procedure. Then, the procedure produces a list of beam pairs using the learning parameters corresponding to the location bin stored in the database. If the beam pair refinement is enabled, the refinement parameters of the selected beam pairs are picked by the beam pair refinement procedure. The resulting subset of beam pairs is then sent to the user with an acknowledgment to allow the

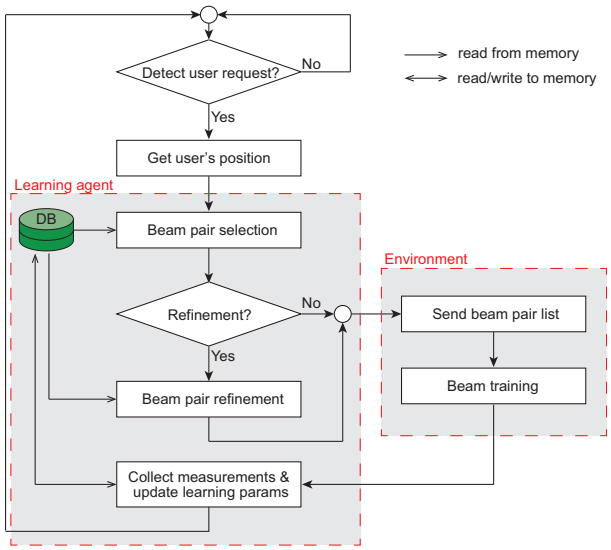


Fig. 5. A flowchart of the two-layer online learning. The algorithm starts with a training request detection loop. When it detects a request, the algorithm decodes the user's position and input to the beam selection procedure, which then reads the learning parameters corresponding to the position and determines a subset of promising beam pairs. If beam refinement is enabled, the refinement parameters of those selected pairs are selected. The beam subset is then sent to the user and the subset of beam pairs are trained. The beam measurements are used to update the learning parameters and the algorithm returns to the training request detection loop.

beam training. After beam training, the measurements of the selected beam pairs are used to update the learning parameters in the database. Then, the algorithm goes back to the detection loop to wait for the next training request.

V. Online Beam Pair Selection

In this section, we describe the first layer of the two-layer solution. We start with the problem statement. Then, we develop online beam pair selection algorithms, first without and then with risk-awareness. We conclude the section with regret analyses of the proposed algorithms and some discussion.

A. Problem statement

Our goal here is to develop an online version of the optimal beam pair selection method, MinMisProb. Specifically, the algorithm needs to solve the following optimization problem in an online setting:

$$\begin{aligned} \underset{\mathcal{S}}{\text{minimize}} \quad & P_{\text{pl}}(1,\mathcal{S}) \\ \text{subject to} \quad & \mathcal{S} \subset \mathcal{B},\ |\mathcal{S}| \leq B_{\text{tr}}, \end{aligned} \tag{14}$$

where B_{tr} is the desired subset size (the training budget). In an online learning setting, $P_{\text{pl}}(1,\mathcal{S})$ is not known, and it has to be estimated on the fly. To gain accurate knowledge of the beam pairs, each of them must be sampled multiple times, which means the learning can be very slow when $\mathcal{B}$ is a large set (i.e., when using large arrays with narrow beams). To remedy this problem, we propose to apply a heuristic to screen the beam pairs using a small offline database (of size N) to obtain a smaller set $\hat{\mathcal{B}}$ to apply the learning algorithm on. $\hat{\mathcal{B}}$ is obtained as the set of the unique beam pairs among the NC entries of the first C columns of Table I. In our simulations, the offline database size $N = 5$ and $C = 200$ seem to be good enough for this purpose.

B. Greedy UCB algorithm

We first propose a solution to (14) without risk-awareness. A subset $\mathcal{S}$ can be treated as a super-arm, and a single-play MAB algorithm can be used. Such an approach is not efficient because it treats each super-arm as independent and the number of super-arms is large due to the combinatorial nature of the number of all possible subsets.

A more efficient approach to solve (14) is to leverage the structure of $P_{\text{pl}}(1,\mathcal{S})$ to take advantage of the dependence between the subsets. Specifically, we make use of the modularity property of the probability of being optimal [1]. We note that by the relationship in (11), the problem (14) is equivalent to a maximization of $P_{\text{opt}}(\mathcal{S})$ with the same constraints, i.e.,

$$\begin{aligned} \underset{\mathcal{S}}{\text{maximize}} \quad & P_{\text{opt}}(\mathcal{S}) \\ \text{subject to} \quad & \mathcal{S} \subset \hat{\mathcal{B}},\ |\mathcal{S}| \leq B_{\text{tr}}, \end{aligned} \tag{15}$$

where we also replace $\mathcal{B}$ by $\hat{\mathcal{B}}$ as explained earlier. Since $P_{\text{opt}}(\mathcal{S})$ is modular [1], it can be decomposed as

$$P_{\text{opt}}(\mathcal{S}) = \sum_{i \in \mathcal{S}} P_{\text{opt}}(i). \tag{16}$$

This property is due to the exclusive nature of the events that the i-th beam pair is optimal (i.e., having the highest channel strength). Recall that the probability of a union of exclusive events is the sum of the probability of each individual event [44]. The main implication of (16) is that the reward of $\mathcal{S}$ can be computed from the individual rewards of each of the beam pairs in $\mathcal{S}$. This means the optimal beam pair subset can be obtained by a greedy approach, where one beam pair is selected at a time. Observing this property, we propose to use a greedy UCB algorithm as shown in Algorithm 1, that selects the beam pairs greedily using their UCB indices. The UCB index of an arm is a high confidence bound of the expected reward, which consists of the expected reward seen so far and the uncertainty (the confidence margin) [28].

An important component of Algorithm 1 is the reward signal. Since the expected reward is the probability of being optimal, an ideal choice for the reward signal is

$$x_{i,t} = \begin{cases} 1 & \text{if } i \text{ was best in } \hat{\mathcal{B}} \\ 0 & \text{otherwise} \end{cases}, \tag{17}$$

which takes the value 1 if the pair i is best and 0 otherwise. In an actual setting, it is not known if a pair is the best in $\hat{\mathcal{B}}$ since only the beam measurements for the beam pairs in the subset $\mathcal{S} \subset \hat{\mathcal{B}}$ are available. The best guess would be the strongest pair among the beam pairs trained. Considering this limitation, we propose to use an alternative and practical reward signal,

$$x_{i,t} = \begin{cases} 1 & \text{if } i \text{ was best in } \mathcal{S} \\ 0 & \text{all other pairs in } \mathcal{S} \end{cases}, \tag{18}$$

which takes the value 1 for the pair with the strongest beam measurement in $\mathcal{S}$ and 0 for all other pairs in $\mathcal{S}$. Denoting

Algorithm 1 Greedy UCB

1: // initialize arms' parameters using a small offline database
2: $X_{\rm tot}[i] \leftarrow 0$, for $\forall i \in \hat{\mathcal{B}}$
3: $X_{\rm tot}[\arg\max_{i\in\hat{\mathcal{B}}} \bar{\gamma}_i^{\rm init}] \leftarrow 1$
4: $T_i \leftarrow 1$, for $\forall i \in \hat{\mathcal{B}}$
5: **for** $n = 1, 2, \ldots$ **do**
6: // Compute UCB values
7: $\mathsf{UCB}_i \leftarrow \frac{X_{\rm tot}[i]}{T_i} + \sqrt{\frac{2\log(n)}{T_i}}$, for $\forall i \in \hat{\mathcal{B}}$
8: // Greedy selection using UCB values
9: $\mathcal{S} \leftarrow \emptyset$
10: **for** $k = 1, 2, \ldots, B_{\rm tr}$ **do**
11: $\mathcal{S} \leftarrow \mathcal{S} \cup \arg\max_{i\in\hat{\mathcal{B}}\setminus\mathcal{S}} \mathsf{UCB}_i$
12: **end for**
13: Train the selected $B_{\rm tr}$ beam pairs to get $\gamma_{i,n}$ for $\forall i \in \mathcal{S}$
14: // Update the learning parameters
15: $T_i \leftarrow T_i + 1$, for $\forall i \in \mathcal{S}$
16: $X_{\rm tot}[\arg\max_{k\in\mathcal{S}} \gamma_{k,n}] \leftarrow X_{\rm tot}[\arg\max_{k\in\mathcal{S}} \gamma_{k,n}] + 1$
17: **end for**

$X_{\rm tot}[i] = \sum_{t=1}^{n} x_{i,t}$, the expected reward of beam pair i at time n is estimated by $\hat{P}_{\rm opt}(i) = X_{\rm tot}[i]/T_i$, where T_i is the number of times that the pair i was selected up to time n.

An intuitive understanding of this alternative reward definition can be drawn from an analogy to a sport tournament. In each round, the winners from each subgroup from the previous round play against each other to decide who will proceed to the next round, which eventually will reach the championship. We, thus, expect that over time only strong beam pairs will receive rewards of 1. We believe that under certain assumption on the underlying reward statistics of the beam pairs, it is possible to provide some guarantee that $\hat{P}_{\rm opt}(i)$ will converge to the true $P_{\rm opt}(i)$ as $T_i \to \infty$. This is outside the focus of this paper and is left for future work.

As will be seen in Section VII-A, Algorithm 1 does not perform well. The main reason for this is because it only tries to minimize the cumulative regret and is oblivious to the multiple-play setting in the beam alignment problem. Since multiple beam pairs are trained, the subset $\mathcal{S}$ can be divided into two parts. One part is for exploitation that uses the knowledge obtained so far to select the beam pairs and the other part is for exploration that aims at improving the accuracy of the learning parameters. By balancing these two parts, it is possible to reduce the risk (large power loss events) at any given round. In other words, in the multiple-play setting, the risk of large losses can be traded off with the speed of learning (time to get accurate statistics of the arms). Another point for improvement in Algorithm 1 is that it throws away the magnitude information because the reward signal is binary. Recall that the binary reward signal is needed because we make use of the modularity of $P_{\rm opt}(\mathcal{S})$ that allows the greedy selection using the UCB indices. To remedy these weaknesses, we propose a risk-aware version of Algorithm 1.

C. Risk-aware greedy UCB algorithm

We first start with the definition of risk. A possible choice for the risk is the power loss, which measures the misalignment severity. Since only the beam pairs in $\mathcal{S}$ are trained, the channel strength of the optimal beam pair is not necessarily known (especially, during the early stage of the learning) and the power loss cannot be computed directly. Another important point is that this risk needs to be estimated. Therefore, it is crucial to quantify the uncertainty of the risk estimate for it to be useful for the beam pair selection. For these two reasons, we propose to use a binary risk signal defined in terms of the ratio of the channel strength of the beam pair and the best beam pair in $\mathcal{S}$, i.e., the risk signal of the beam pair i at time t is given by

$$z_{i,t} = \begin{cases} 1 & \text{if } \frac{\max_{k\in\mathcal{S}} \gamma_{k,t}}{\gamma_{i,t}} > \Gamma_{\rm risk} \\ 0 & \text{otherwise} \end{cases}, \tag{19}$$

where $\Gamma_{\rm risk}$ is a risk threshold. The choice of $\Gamma_{\rm risk}$ will be discussed in Section VII-B.

A way to capture the uncertainty is to put a prior distribution on the risk based on the observations seen so far (i.e., a Bayesian approach). By the definition (19), $z_{i,t}$ is Bernoulli distributed with some unknown parameter ζ. It is well-known that the Beta distribution is the conjugate prior to the Bernoulli distribution [45]. This means that the belief on the risk of the beam pair i upon seeing the measurements up to time n can be updated conveniently by updating the parameters of the Beta distribution. Specifically, denoting $Z_{\rm tot}[i] = \sum_{t=1}^{n} z_{i,t}$ the prior is updated as

$$\tilde{Z}_n \sim \mathsf{Beta}\left(1 + Z_{\rm tot}[i], 1 + T_i - Z_{\rm tot}[i]\right). \tag{20}$$

Here, we assume that at time 0 without any observation, $\tilde{Z}_0 \sim \mathsf{Beta}(1,1)$, which is the uniform distribution over $[0,1]$. This is a reasonable assumption since no information on the beam pair i is available at time 0.

We next explain how the risk estimate along with the prior are used in the rejection mechanism to reduce the probability of large power loss events during the learning. The new algorithm is shown in Algorithm 2, which we call the risk-aware greedy UCB algorithm. The new addition to Algorithm 1 is the risk-aware feature that rejects a beam pair selected by the greedy UCB with a probability reflecting its risk. The rejection probability is determined using the risk drawn from the prior distribution given in (20) and the confidence margin in two steps. First, a random variable $\tilde{Z}_n$ is drawn from this prior (line 13). Then, it is multiplied by the confidence margin for those beam pairs with $X_{\rm tot}[\ell] > 0$. The obtained Z_n is the rejection probability. The second step is needed because any beam pair is subject to blockage and their risks are not zero. This means that if $\tilde{Z}_n$ is used directly as the rejection probability, even good beam pairs will be rejected with non-zero probability even when $n \to \infty$. The second step ensures the algorithm accepts the UCB selection for "good" beam pairs with increasing probability over time.

The proposed rejection mechanism is a random method that rejects the beam pair with a probability Z_n. First, the algorithm draws a Bernoulli random variable Rej with parameter Z_n. If $\mathsf{Rej} = 0$, the algorithm accepts the beam pair, otherwise it rejects the pair. In that case, the replacement beam pair is selected using $\hat{P}_{\rm opt}(i)$ when there are pairs with $X_{\rm tot}[i] > 0$,

Algorithm 2 Risk-Aware Greedy UCB

// initialize arms' parameters using a small offline database
$X_{\text{tot}}[i] \leftarrow 0$, for $\forall i \in \hat{\mathcal{B}}$
$X_{\text{tot}}[\arg\max_{i\in\hat{\mathcal{B}}} \bar{\gamma}_i^{\text{init}}] \leftarrow 1$
$Z_{\text{tot}}[i] \leftarrow 0$, for $\forall i \in \hat{\mathcal{B}}$
$T_i \leftarrow 1$, for $\forall i \in \hat{\mathcal{B}}$
for $n = 1, 2, \ldots$ **do**
  // Compute UCB values
  $\mathsf{UCB}_i \leftarrow \frac{X_{\text{tot}}[i]}{T_i} + \sqrt{\frac{2\log(n)}{T_i}}$, for $\forall i \in \hat{\mathcal{B}}$
  // Greedy selection using UCB values
  $\mathcal{S} \leftarrow \emptyset$
  for $k = 1, 2, \ldots, B_{\text{tr}}$ **do**
    $\ell \leftarrow \arg\max_{i\in\hat{\mathcal{B}}\setminus\mathcal{S}} \mathsf{UCB}_i$
    $\tilde{Z}_n \sim \mathsf{Beta}\left(1 + Z_{\text{tot}}[\ell], 1 + T_\ell - Z_{\text{tot}}[\ell]\right)$
    $Z_n \leftarrow \tilde{Z}_n \times \sqrt{\frac{2\log(n)}{T_\ell}}$ if $X_{\text{tot}}[\ell] > 0$, else $Z_n \leftarrow \tilde{Z}_n$
    $\text{Rej} \sim \mathsf{Ber}(Z_n)$
    if $\text{Rej} = 0$ **then**
      $\mathcal{S} \leftarrow \mathcal{S} \cup \{\ell\}$
    else
      if $\exists i \in \hat{\mathcal{B}} \setminus \mathcal{S}$ with $X_{\text{tot}}[i] > 0$ **then**
        $\mathcal{S} \leftarrow \mathcal{S} \cup \arg\max_{i\in\hat{\mathcal{B}}\setminus\mathcal{S}} \hat{P}_{\text{opt}}(i)$
      else
        $\mathcal{S} \leftarrow \mathcal{S} \cup \arg\max_{i\in\hat{\mathcal{B}}\setminus\mathcal{S}} \bar{\gamma}_i$
      end if
    end if
  end for
  Train the selected B_{tr} beam pairs to get $\gamma_{i,n}$ for $\forall i \in \mathcal{S}$
  // Update the learning parameters
  $T_i \leftarrow T_i + 1$, for $\forall i \in \mathcal{S}$
  $X_{\text{tot}}[\arg\max_{k\in\mathcal{S}} \gamma_{k,n}] \leftarrow X_{\text{tot}}[\arg\max_{k\in\mathcal{S}} \gamma_{k,n}] + 1$
  $Z_{\text{tot}}[i] \leftarrow Z_{\text{tot}}[i] + 1$ if $(\max_{k\in\mathcal{S}} \gamma_{k,n})/\gamma_{i,n} > \Gamma_{\text{risk}}$, $\forall i \in \mathcal{S}$
end for

and using the average channel strength $\bar{\gamma}_i$ when all remaining pairs have $X_{\text{tot}}[i] = 0$. Unlike Algorithm 1, which does not use the amplitudes of the beam measurements $\gamma_{i,n}$, here they are used to update the risk parameters and also used for the replacement selection. This new algorithm makes a fuller use of the measurement information as compared to Algorithm 1.

D. Regret analysis

In this subsection, we derive regret bounds of the two algorithms that will provide insights on the effect of the rejection mechanism we introduced in Algorithm 2. We make a few simplifications to the problems to allow tractable analysis which we will describe in detail when presenting the results. Proofs are provided in Appendix A and B.

Before stating the results, we first describe the metric used for the evaluation. For this type of online learning problem, a widely used metric is the cumulative regret. It is defined as *the cumulative performance loss as compared to the performance of an oracle that always plays the best arm* [29]. Translating this to the beam alignment problem, *the regret incurred in a time step is non-zero when the algorithm does not select the best subset of beam pairs* $\mathcal{S}^\star$. Assuming $|\mathcal{S}^\star| = B_{\text{tr}}$, the optimal selection in (13) tells us that $\mathcal{S}^\star$ contains the top-B_{tr} beam pairs with the highest probability of being optimal $P_{\text{opt}}(\cdot)$. Now, we call the beam pairs with the B_{tr}-highest $P_{\text{opt}}(\cdot)$ as optimal and the rest of beam pairs as suboptimal. Then, the cumulative regret increases whenever one or more suboptimal beam pairs are selected in the selection set $\mathcal{S}$.

In the following, we present what is called a problem-dependent bound on cumulative regret (we drop 'cumulative' from now on for convenience), which quantifies the regret in terms of the optimality gap. The optimality gap is defined as the difference in the probability of being optimal for an optimal pair $i^\star$ and a suboptimal pair ℓ, i.e.,

$$\Delta_{\ell,i^\star} = P_{\text{opt}}(i^\star) - P_{\text{opt}}(\ell). \tag{21}$$

By definition, $0 < \Delta_{\ell,i^\star} < 1$ if $P_{\text{opt}}(i^\star) > 0$. Note that $\Delta_{\ell,i^\star}$ measures the difficulty in discriminating the suboptimal pair ℓ from the optimal pair $i^\star$ for the particular problem at hand; thus, the name problem-dependent when the regret bound is expressed using optimality gaps.

For Algorithm 1, we assume the reward signal during the learning is the ideal reward and not the alternative one, i.e., we assume the reward signal is given by (17) instead of (18). We make this assumption because it is intractable to deal directly with the dynamics of the alternative reward signal in (18). The main step in deriving the regret bound is the application of the Chernoff-Hoeffding inequality to bound the probability that the sample average of the reward is within the UCB value. To apply the Chernoff-Hoeffding inequality, it is required that the sample rewards are IID, which cannot be guaranteed when using the alternative reward definition because its distribution depends on the history of the selection done so far. This, however, is a reasonable assumption, since we expect that (18) will approach (17) for large n, which is the domain where the regret bound is meaningful.

Theorem 1: Assuming that the ideal reward signal (17) is accessible, the expected regret at time n of the greedy UCB algorithm is upper bounded by

$$R_1[n] \leq 8\log(n) \sum_{\ell\in\mathcal{B}\setminus\mathcal{S}^\star} \sum_{i^\star\in\mathcal{S}^\star} \frac{1}{\Delta_{\ell,i^\star}} + \left(1 + \frac{\pi^2}{3}\right) \sum_{\ell\in\mathcal{B}\setminus\mathcal{S}^\star} \sum_{i^\star\in\mathcal{S}^\star} \Delta_{\ell,i^\star}. \tag{22}$$

Theorem 1 shows a regret bound for Algorithm 1. The first term, which increases with n, dominates the bound. It increases as $\Delta_{\ell,i^\star}$ decreases. This makes sense because a small $\Delta_{\ell,i^\star}$ means more samples are needed to differentiate the suboptimal pair ℓ from the optimal pair $i^\star$ with confidence. The regret bound is $\mathcal{O}(\log(n))$, which is known to be optimal up to the constant coefficient in front of $\log(n)$ [29]. This confirms that the algorithm is a reasonable solution.

For Algorithm 2, we make two additional assumptions besides the accessibility of the ideal reward signal. The first assumption is that the rejection probability of any beam pair ℓ is constant, denoted by $1 - \zeta_\ell$. This is used because the rejection probability of the algorithm is dynamic (depending

on the observation so far) and is not tractable. With a large enough n, we expect the risk estimate to stabilize, and thus this is not an unreasonable assumption. The second assumption is that when rejected the replacement selection has an optimality gap $\tilde{\Delta}_{\ell,i^\star}$.

Theorem 2: Assuming that the ideal reward signal is available, the rejection probability of beam pair ℓ is $1-\zeta_\ell$, and that when rejected the optimality gap of the replacement selection is $\tilde{\Delta}_{\ell,i^\star}$, then the expected regret at time n of the risk-aware greedy UCB algorithm is bounded by

$$R_2[n] \leq \frac{8\log(n)}{\delta^2} \sum_{\ell\in\mathcal{B}\setminus\mathcal{S}^\star} \sum_{i^\star\in\mathcal{S}^\star} \frac{1}{\Delta_{\ell,i^\star}} + \frac{8\log(n)}{\delta^2} \sum_{\ell\in\mathcal{B}\setminus\mathcal{S}^\star} \sum_{i^\star\in\mathcal{S}^\star} \frac{(1-\zeta_\ell)\tilde{\Delta}_{\ell,i^\star}}{\zeta_\ell \Delta_{\ell,i^\star}^2} + \left(1+\frac{\pi^2}{2}\right) \sum_{\ell\in\mathcal{B}\setminus\mathcal{S}^\star} \sum_{i^\star\in\mathcal{S}^\star} (\zeta_\ell \Delta_{\ell,i^\star} + (1-\zeta_\ell)\tilde{\Delta}_{\ell,i^\star}), \quad (23)$$

where $\delta = (\sqrt{5}-1)/2$.

Theorem 2 shows a regret bound for Algorithm 2. The algorithm still has $\mathcal{O}(\log(n))$ regret but with a larger constant. This shows that introducing risk-awareness increases the learning time in the sense that $R_2[n] > R_1[n]$. This is because by rejecting a high-risk beam pair, the algorithm loses the chance to get information on that beam pair. The idea of Algorithm 2 is to distribute the learning of these high-risk beam pairs (which has high cost) more evenly among the users by rejecting them with some probability. In other words, Algorithm 2 tradeoffs the learning speed to balance the risk of severe misalignment endured by each user at different stages of the learning.

VI. Online beam pair refinement

In this section, we describe our beam refinement solution, which is the second layer of the two-layer online learning algorithm. We start with the problem formulation and then describe our modified HOO solution.

A. Problem statement

We formulate our beam pair refinement as a stochastic CAB problem. The beams are generated by progressive phase-shift and are defined by their main beam directions. The goal is to find the pointing direction of a beam pair to maximize the average channel strength of that beam pair in an online setting. Specifically, denoting $\phi_i^{\mathrm{t}}, \theta_i^{\mathrm{t}}, \phi_i^{\mathrm{r}}$, and θ_i^{r} the transmit and receive main beam directions in the azimuth and elevation of the beam pair i defined in the codebook, and $\Phi_i^{\mathrm{t}}, \Theta_i^{\mathrm{t}}, \Phi_i^{\mathrm{r}}$, and Θ_i^{r} the corresponding 3dB beamwidths, the problem of refining the beam pair i can be written as

$$\begin{aligned} \underset{\phi^{\mathrm{t}},\theta^{\mathrm{t}},\phi^{\mathrm{r}},\theta^{\mathrm{r}}}{\text{maximize}} \quad & \mathbb{E}[\gamma_i(\phi^{\mathrm{t}},\theta^{\mathrm{t}},\phi^{\mathrm{r}},\theta^{\mathrm{r}})] \\ \text{subject to} \quad & \phi^{\mathrm{t}} \in [\phi_i^{\mathrm{t}} - \Phi_i^{\mathrm{t}}/2, \phi_i^{\mathrm{t}} + \Phi_i^{\mathrm{t}}/2], \\ & \phi^{\mathrm{r}} \in [\phi_i^{\mathrm{r}} - \Phi_i^{\mathrm{r}}/2, \phi_i^{\mathrm{r}} + \Phi_i^{\mathrm{r}}/2], \\ & \theta^{\mathrm{t}} \in [\theta_i^{\mathrm{t}} - \Theta_i^{\mathrm{t}}/2, \theta_i^{\mathrm{t}} + \Theta_i^{\mathrm{t}}/2], \\ & \theta^{\mathrm{r}} \in [\theta_i^{\mathrm{r}} - \Theta_i^{\mathrm{r}}/2, \theta_i^{\mathrm{r}} + \Theta_i^{\mathrm{r}}/2]. \end{aligned} \quad (24)$$

Any pointing direction $(\phi^{\mathrm{t}}, \theta^{\mathrm{t}}, \phi^{\mathrm{r}}, \theta^{\mathrm{r}})$ satisfying the constraints is an arm in this problem. The space to search for the best arm is the hyperrectangle defined by the constraints, which is a continuous space. This means the directions are fine-tuned within the 3dB beamwidths of the original beam pair i defined by the pointing direction $(\phi_i^{\mathrm{t}}, \theta_i^{\mathrm{t}}, \phi_i^{\mathrm{r}}, \theta_i^{\mathrm{r}})$. The coarse search to within the 3dB beamwidth is supposed to be done by the beam pair selection algorithm.

B. Modified HOO for beam pair refinement

HOO is a CAB algorithm that runs on a tree. We start by describing the search tree. Then, we describe the flow of HOO. Finally, we explain the modifications made to the original algorithm to fit the beam refinement task. We describe the algorithm for refining a beam pair i. Since all the description is in the context of this beam pair i, we drop explicit references to beam pair i here to avoid notational clutter.

We now define the search tree $\mathcal{T}$ which HOO runs on. Each node in the tree is a pair of transmit and receive pointing directions $(\phi^{\mathrm{t}}, \theta^{\mathrm{t}}, \phi^{\mathrm{r}}, \theta^{\mathrm{r}})$ satisfying the constraints in (24). The root of the tree is the original pointing direction of the beam pair i $(\phi_i^{\mathrm{t}}, \theta_i^{\mathrm{t}}, \phi_i^{\mathrm{r}}, \theta_i^{\mathrm{r}})$ defined in the codebook. Each node in the tree at depth $\ell < \ell_{\max}$ has 16 children which correspond to all possible combinations of transmit and receive beam directions perturbed by $1/2^\ell$ of the beamwidths in the four variables. Denote $(\phi_{\ell,k}^{\mathrm{t}}, \theta_{\ell,k}^{\mathrm{t}}, \phi_{\ell,k}^{\mathrm{r}}, \theta_{\ell,k}^{\mathrm{r}})$ the parameters of the k-th node at depth ℓ in $\mathcal{T}$, its set of 16 children nodes can be written using a Cartesian product as

$$\left\{ \begin{bmatrix} [\phi_{\ell,k}^{\mathrm{t}} + \Phi_i^{\mathrm{t}}/2^\ell, \ \theta_{\ell,k}^{\mathrm{t}}]^{\mathrm{T}}, \\ [\phi_{\ell,k}^{\mathrm{t}} - \Phi_i^{\mathrm{t}}/2^\ell, \ \theta_{\ell,k}^{\mathrm{t}}]^{\mathrm{T}}, \\ [\phi_{\ell,k}^{\mathrm{t}}, \ \theta_{\ell,k}^{\mathrm{t}} + \Theta_i^{\mathrm{t}}/2^\ell]^{\mathrm{T}}, \\ [\phi_{\ell,k}^{\mathrm{t}}, \ \theta_{\ell,k}^{\mathrm{t}} - \Theta_i^{\mathrm{t}}/2^\ell]^{\mathrm{T}} \end{bmatrix} \times \begin{bmatrix} [\phi_{\ell,k}^{\mathrm{r}} + \Phi_i^{\mathrm{r}}/2^\ell, \ \theta_{\ell,k}^{\mathrm{r}}]^{\mathrm{T}}, \\ [\phi_{\ell,k}^{\mathrm{r}} - \Phi_i^{\mathrm{r}}/2^\ell, \ \theta_{\ell,k}^{\mathrm{r}}]^{\mathrm{T}}, \\ [\phi_{\ell,k}^{\mathrm{r}}, \ \theta_{\ell,k}^{\mathrm{r}} + \Theta_i^{\mathrm{r}}/2^\ell]^{\mathrm{T}}, \\ [\phi_{\ell,k}^{\mathrm{r}}, \ \theta_{\ell,k}^{\mathrm{r}} - \Theta_i^{\mathrm{r}}/2^\ell]^{\mathrm{T}} \end{bmatrix} \right\}. \quad (25)$$

Using this node expansion rule, a node at depth $\ell+1$ deviates from its parent node in the pointing direction by $\mathrm{beamwidth}/2^\ell$ and depends only on ℓ. Each depth in the tree can be thought of as a grid partitioning the search space defined by the constraints in (24). The grid becomes finer deeper in the tree (i.e., as ℓ increases).

We now describe how the modified HOO works. A pseudo-code is shown in Algorithm 3. It runs on a finite tree with a maximum depth of $\ell_{\max}$. The nodes in the tree are activated on the fly, and only the root node and its children are active at time $n=0$. Thus, the initial tree is $\mathcal{T} = \{(1,1)\} \cup \mathcal{C}_{1,1}$, where $\mathcal{C}_{\ell,k}$ denotes the set of the indices of the children of the node (ℓ,k). In each iteration, there are three main parts. First, a node is selected by traversing the active tree starting from the root following the path through nodes that have the largest B-values (line 6-13), which is the best optimistic estimate of the average rewards of the nodes. The second part is the beam measurement for the selected node (line 14). Lastly, after obtaining the measurement, the learning parameters are updated. If the condition is met, a node in the tree is expanded, i.e., activating its 16 children nodes. Note that to lower the risk of expanding a suboptimal node, it is enforced that a node can be expanded only after it is sampled K_{exd} times. The last part

Algorithm 3 Modified HOO for Beam Pair Refinement

1: // Initialization
2: $\mathcal{T} \leftarrow \{(0,1)\} \cup \mathcal{C}_{1,1}$
3: $(B_{2,j}, T_{2,j}, \hat{\mu}_{2,j}, \mathsf{Sq}_{2,j}) \leftarrow (\infty, 0, 0, 0)$ for $\forall j \in \mathcal{C}_{1,1}$
4: **for** $n = 1, 2, \ldots$ **do**
5: // Select a node in the tree to sample
6: $(\ell, k) \leftarrow (1,1)$ // Start from the root node
7: $\mathcal{P} \leftarrow \{(\ell, k)\}$
8: **for** $\ell = 1, \ldots, \min\{\text{depth}(\mathcal{T}), \ell_{\max} - 1\}$ **do**
9: $k^\star \leftarrow \arg\max_{j \in \mathcal{C}_{\ell,k}} B_{\ell+1,j}$
10: $(\ell, k) \leftarrow (\ell + 1, k^\star)$
11: $\mathcal{P} \leftarrow \mathcal{P} \cup \{(\ell, k)\}$
12: **end for**
13: $(\ell_{\rm s}, k_{\rm s}) \leftarrow (\ell, k)$
14: Obtain the beam measurement for node $(\ell_{\rm s}, k_{\rm s})$ denoted by γ
15: // Update the learning parameters
16: **for** $(\ell, k) \in \mathcal{P}$ **do** // Update sample averages
17: $T_{\ell,k} \leftarrow T_{\ell,k} + 1$
18: $\hat{\mu}_{\ell,k} \leftarrow (1 - \frac{1}{T_{\ell,k}})\hat{\mu}_{\ell,k} + \gamma / T_{\ell,k}$
19: $\mathsf{Sq}_{\ell,k} \leftarrow \mathsf{Sq}_{\ell,k} + \gamma^2$
20: $\sigma^2_{\ell,k} \leftarrow (\mathsf{Sq}_{\ell,k} - \hat{\mu}^2_{\ell,k} T_{\ell,k}) / T_{\ell,k}$
21: **end for**
22: **for all** $(\ell, k) \in \mathcal{T}$ **do** // Update U-values
23: $U_{\ell,k} \leftarrow \left(\hat{\mu}_{\ell,k} + \sqrt{16 \hat{\sigma}^2_{\ell,k} \frac{\log(n)}{T_{\ell,k}}}\right) \nu(\ell)$
24: $U_{\ell,k} \leftarrow \infty$ if $T_{\ell,k} < \lceil \alpha_{\rm norm} \log(n) \rceil$ or $T_{\ell,k} < K_{\min}$ // Forced exploration
25: **end for**
26: // Expand a node if conditions are met
27: **if** $\ell_{\rm s} < \ell_{\max} \wedge T_{\ell_{\rm s},k_{\rm s}} > K_{\rm exd} \wedge (\ell_{\rm s}, k_{\rm s})$ is a leaf **then**
28: $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{C}_{\ell_{\rm s},k_{\rm s}}$
29: $(B_{\ell_{\rm s}+1,j}, T_{\ell_{\rm s}+1,j}, \hat{\mu}_{\ell_{\rm s}+1,j}, \mathsf{Sq}_{\ell_{\rm s}+1,j}) \leftarrow (\infty, 0, 0, 0)$ for $\forall j \in \mathcal{C}_{\ell_{\rm s},j}$
30: **end if**
31: // Update B-values
32: **for** $\ell = \ell_{\rm s}, \ell_{\rm s} - 1, \ldots, 2$ **do**
33: $B_{\ell,k} \leftarrow \min\{U_{\ell,k}, \max_{j \in \mathcal{C}_{\ell,k}} B_{\ell+1,j}\}, \forall$ nodes at depth ℓ in $\mathcal{T}$
34: **end for**
35: **end for**

of the parameter update is the B-values. They are computed by back calculation from the sampled node back to the root (line 33). The B-value of node (ℓ, k) is the minimum between its own U-value and the maximum B-value of its children nodes, i.e.,

$$B_{\ell,k} \leftarrow \min\left\{U_{\ell,k}, \max_{j \in \mathcal{C}_{\ell,k}} B_{\ell+1,j}\right\}. \tag{26}$$

The U-value is similar to the UCB value, but it also accounts for the smoothness property (line 23). The U-value provides an optimistic estimate of its average reward using the parameter of the node, and the maximum B-value among its children nodes provides another optimistic estimate of its reward. By taking the minimum between the two, the obtained B-value provides a refined optimistic estimate of the average reward of the node.

We introduce three main modifications to the original HOO tailored to the beam refinement setting. The first one is the use of a finite tree. The original HOO assumes an infinite tree to represent the arm space. Since small adjustments (e.g., 1/8 of the beamwidth) have a small impact on the gain, a finite tree of maximum depth $\ell_{\max}$ is used instead to save computation and storage for the learning parameters. The second one is the smoothness bound. The original HOO assumes an additive offset. Due to the multiplicative nature of the antenna gains, a multiplicative smoothness coefficient $\nu(\ell)$ as shown in line 23 is more suitable. The coefficient is computed using Lemma 1 which will be detailed in the last part of this subsection.

The third modification is the confidence margin. Because the original margin $\sqrt{2\log(n)/T_{\ell,k}}$ is too loose in our setting, we propose to use that of the norm-UCB (line 23) [28]. The margin $\sqrt{2\log(n)/T_{\ell,k}}$ is derived from the Chernoff-Hoeffding inequality, which is applicable to any distribution with the support in $[0,1]$. While normalizing the channel strengths by a large enough number will approximately guarantee that the support is within $[0,1]$, the average typically takes a value much less than 1 and the margin $\sqrt{2\log(n)/T_{\ell,k}}$ is too loose for reasonable learning horizons. The reason that the average is much smaller than 1 is the small scale fading nature of the wireless channel. Fading is due to the multipath effect and can cause the maximum instantaneous channel strength to be much larger than the average [37]. A good property of the new margin is that the sample variance is also used. Note that to enable regret analysis, the norm-UCB algorithm requires each arm be sampled at least $\lceil \alpha_{\rm norm} \log(n) \rceil$ at time n with $\alpha_{\rm norm} = 8$ [28]. This is enforced by setting the U-values of the nodes that need to be explored to infinity (see line 24). Note that we also introduce the condition $T_{\ell,k} < K_{\min}$, which is used to ensure that there are at least $K_{\min}$ samples of the node for computing the sample variance. This is needed when using a small $\alpha_{\rm norm}$.

We next state a lemma defining the smoothness property of the objective function in (24). The lemma is useful for computing the smoothness coefficient $\nu(\ell)$.

Lemma 1: Assume a single-path *azimuth* PAS with the optimal beam direction $\phi^\star$ with isotropic transmit antenna, $G(\cdot; \phi_0)$ the normalized gain of the beam pattern pointing at ϕ_0 assumed to be decreasing and concave in $[\phi_0, \phi_0 + \Phi/2]$ with Φ denoting the 3 dB beamwidth (e.g., true for a uniform planar array), for a receive pointing direction ϕ_0 such that $|\phi^\star - \phi_0| \le \Delta\phi \le \Phi/2$,

$$\bar{\gamma}(\phi_0)/G(\phi_0 + \Delta\phi; \phi_0) \ge \bar{\gamma}(\phi^\star). \tag{27}$$

Moreover, for a general PAS with the support within $[\phi_0 - \Psi, \phi_0 + \Psi]$ with $\Psi \le \Phi/2$,

$$\bar{\gamma}(\phi_0)/G(\phi_0 + \Delta\phi; \phi_0) \ge \bar{\gamma}(\phi^\star) - \mathsf{Err}, \tag{28}$$

where $\mathsf{Err} \ge 0$ is a residual term that depends on the shape of the PAS and $\mathsf{Err} \to 0$ as $\Delta\phi \to 0$.

We now explain how to determine the smoothness coefficient $\nu(\ell)$ based on Lemma 1. While we state Lemma 1 assuming an isotropic transmit antenna to avoid tedious

notations, the same argument applies when we also include the transmit beam pattern. In particular, denoting $\phi^{\mathrm{r}\star}, \phi^{\mathrm{t}\star}$ the optimal receive and transmit beam direction,

$$\frac{\bar{\gamma}(\phi_0^{\mathrm{r}}, \phi_0^{\mathrm{t}})}{G_{\mathrm{r}}(\phi_0^{\mathrm{r}} + \Delta\phi^{\mathrm{r}}; \phi_0^{\mathrm{r}}) G_{\mathrm{t}}(\phi_0^{\mathrm{t}} + \Delta\phi^{\mathrm{t}}; \phi_0^{\mathrm{t}})} \geq \bar{\gamma}(\phi^{\mathrm{r}\star}, \phi^{\mathrm{t}\star}) - \mathsf{Err}. \tag{29}$$

When steering the elevation only, we get the same relation as (29). If we assume square UPAs, the beam pattern in the azimuth and elevation will be the same. Since we only change the azimuth or elevation but not both per (25), the smoothness coefficient is given by

$$\nu(\ell) = a/(G_{\mathrm{r}}(\phi_\ell^{\mathrm{r}} + \Phi/2^\ell; \phi_\ell^{\mathrm{r}}) G_{\mathrm{t}}(\phi_\ell^{\mathrm{t}} + \Phi/2^\ell; \phi_\ell^{\mathrm{t}})) \tag{30}$$

$$\simeq a/g^2(\mathrm{beamwidth}/2^\ell), \tag{31}$$

where $a > 1$ is a correction coefficient to account for Err if deemed necessary. For convenience, we approximate the gain by $g(\cdot)$ the beam pattern at broadside as a function of the deviation from the broadside direction. Note that we will need a only for large $\Delta\phi$. Deeper in the tree, the change in the angle is small and thus Err will become negligible. Also, for the sake of clear argument, we restrict $\Psi \leq \Phi/2$, but with a more elaborate choice of the coefficient of Err in the proof, we can allow Ψ to be larger. This, however, is not a big concern in our setting because $\Delta\phi$ will be $\Phi/4$ or less and Err is restricted to a small value already.

VII. Numerical results

We start with the general setting of our numerical evaluations. As described in Section II-B, our codebook for 16×16 UPA has 271 beams and thus there are 271^2 beam pairs. Using the heuristic screening to get $\hat{\mathcal{B}}$ as explained in Section V-A with the initial database size of $N = 5$ and $C = 200$, the size of the set of beam pairs to be learned $|\hat{\mathcal{B}}|$ is typically between 400 and 600 depending on the simulation run. As mentioned in Section II-B, we generated 10,000 channel samples using ray-tracing. To eliminate the effect of the ordering of the channel samples on the learning performance, the evaluation metrics are averaged over 100 simulation runs, where in each run we randomly permute these 10,000 channel samples. We apply a moving average with a window size of 50 time steps to better show the trends.

As an evaluation metric, we use the 3 dB power loss probability and the gain defined as the inverse of the power loss in (6). The 3 dB power loss probability (i.e., $c = 2$ in (7)) measures how often the selected beam pair has a loss larger than 3 dB as compared to the best beam pair selected by exhaustive search, and thus capturing the beam alignment accuracy. This metric, however, is not suitable for evaluating the beam pair refinement because it cannot capture the improvement over the exhaustive search in the original codebook. Allowing the refinement, a beam pair better than the best in the original codebook can be selected resulting in power loss taking a value less than one, or equivalently, a positive gain in dB.

The rest of the section is divided into three parts. Section VII-A evaluates the beam pair selection alone without the refinement option. Section VII-B assesses the performance of the beam pair refinement assuming an offline learning for the beam pair selection. Section VII-C provides evaluations of the integrated solution incorporating both components.

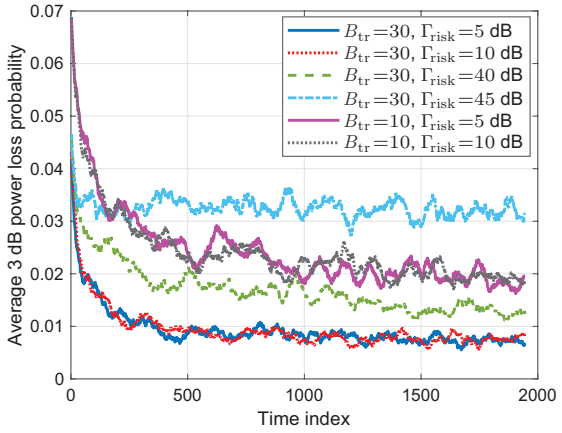


Fig. 6. Average 3dB power loss probability using the proposed risk-aware greedy UCB algorithm with different training budgets B_{tr} and risk thresholds Γ_{risk}. For both $B_{\mathrm{tr}} = 10$ and 30, the plots show similar learning behavior. A smaller training budget $B_{\mathrm{tr}} = 10$ provides less accuracy beam alignment. The plots using different Γ_{risk} show that the performance is not sensitive to Γ_{risk} as long as it is not too large.

A. Online Beam Pair Selection

This subsection evaluates the performance of the proposed risk-aware greedy UCB algorithm without the beam refinement option. There are two parameters to be decided when running Algorithm 2: the training budget B_{tr} and the risk threshold Γ_{risk} in (19). We note that our solution does not require that B_{tr} be fixed, but for simplicity, we assume that the same B_{tr} is used during the entire learning horizon. Fig. 6 shows the average 3dB power loss probability versus time for $B_{\mathrm{tr}} = 10$ and 30 with different Γ_{risk}. We can confirm from the figure that using a larger training budget B_{tr} leads to lower 3dB power loss probability, i.e., more accurate beam alignment. The learning seems to have two phases: the fast improvement phase in the early time steps and the slower improvement phase after that. For $B_{\mathrm{tr}} = 30$ and $\Gamma_{\mathrm{risk}} = 5$dB, this phase change happens at around time index 500. The slower learning phase starts when the algorithm has identified high-risk beam pairs (with some certainty) and learns those beam pairs at a slow pace due to the rejection mechanism. Regarding the risk threshold, the results show that the algorithm is not sensitive to the choice of Γ_{risk}. As long as Γ_{risk} is not too large (e.g., less than 40dB), it performs well. The main reason for this behavior is due to the effect of the replacement selection method (line 19-23 in Algorithm 2) that selects beam pairs to replace those rejected; even if a good beam pair gets rejected due to risk overestimation (when using a small Γ_{risk}), it will likely be picked up by the replacement selection.

Fig. 7 shows a performance comparison of the greedy UCB algorithm with and without the risk-awareness. The performance without risk-awareness is an order of magnitude worse than that with risk-aware. This might seem a bit

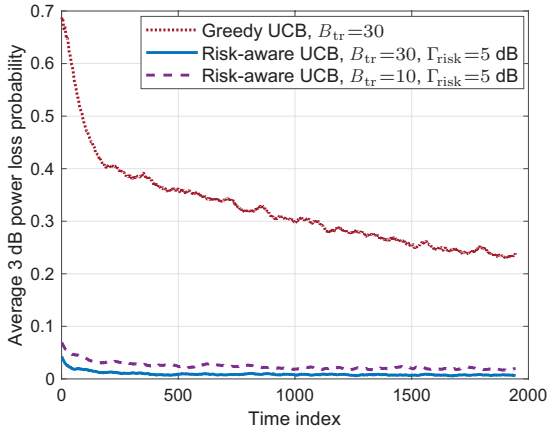


Fig. 7. Performance comparison of greedy UCB with and without risk-awareness. The performance is an order of magnitude worse without risk-awareness. This is because the risk-aware greedy UCB uses the risk estimates to control the number of high-risk beam pairs selected in the subset $\mathcal{S}$ reducing the probability of severe misalignment.

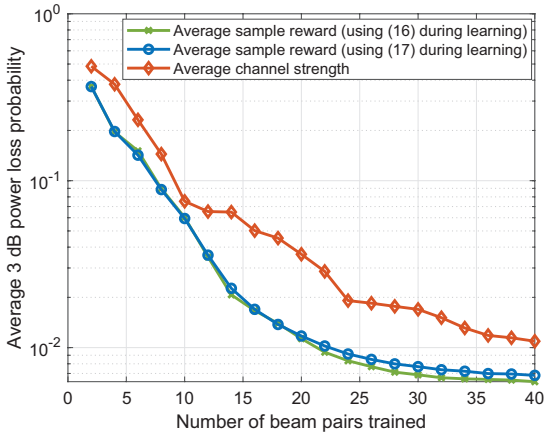


Fig. 8. A comparison of the accuracy of the selection set produced by the average of the proposed reward signal ($\hat{P}_{\rm opt}(i)$) and the more intuitive choice of average channel strength. The performance when using $\hat{P}_{\rm opt}(i)$ is consistently better for all training budgets. The comparison when using the proposed practical reward signal (18) as opposed to the ideal reward signal (17) shows negligible performance loss.

counterintuitive because the regret bound of the risk-aware algorithm is higher. One way to understand this behavior is this. The goal of the UCB selection is to reach a state where we can ensure that a suboptimal arm is *not* selected with high probability (call this the optimal state). To reach the optimal state, each arm has to be sampled enough times ($T^{(0)}_{\ell,i^\star}$ in the derivation in Appendix A). Algorithm 1 samples the arms to reach this state fast, but it will expose early stage users to select more high-risk arms. Algorithm 2 balances the number of high-risk arms at any round by the rejection mechanism which results in a slower speed to reach the optimal state, i.e., a slower learning speed. By not exposing a user to too many high-risk arms, Algorithm 2 can ensure that the regret each user has to endure is not too large. In other words, although the cumulative regret is smaller (at a large enough time), users in early stages of Algorithm 1 have to sacrifice. Algorithm 2 distributes the regret more evenly among the users at different learning stages. We note that because of the large number of arms (400 to 600 as noted earlier), the time to reach the optimal state is large and Algorithm 1 is not practical as an online solution as shown in Fig. 7.

The last part of this subsection shows the effectiveness of our choice of the reward signal in (18). Specifically, we compare the accuracy of the beam selection using the average sample rewards ($\hat{P}_{\rm opt}(i)$) versus the more intuitive choice of average channel strength $\bar{\gamma}_i$. We also compare it with the case where we assume that the ideal reward defined in (17) is available to the algorithm during the learning. To evaluate this, we let the online learning run for 2000 time steps. We, then, use $\hat{P}_{\rm opt}(i)$ and $\bar{\gamma}_i$ estimated at time step 2000 to get two sets of beam selections and evaluate the two sets over 500 channel samples. We use $B_{\rm tr} = 30$ and $\Gamma_{\rm risk} = 5$dB for the online learning. Fig. 8 shows the 3dB power loss probability against the number of beam pairs trained. We can see that the beam pair selection using $\hat{P}_{\rm opt}(i)$ is more accurate than using the average channel strengths. Also, the plots show that the degradation due to the use of the proposed alternative and practical reward signal in (18) during the learning results in negligible loss. These results confirm the effectiveness of our choice of the reward signal in (18).

Before moving on to the beam refinement part, we provide some comments regarding the overhead and the requirement on the position accuracy. Regarding the accuracy of the alignment, as reported in [1], there is negligible loss in the data rate as compared to the exhaustive search when the average 3dB power loss probability is less than 1%. We can see in Fig. 6 that the average 3dB power loss falls below 2% quickly and reaches 1% within about 300 time steps with $B_{\rm tr} = 30$ and $\Gamma_{\rm risk} = 5$dB. The overhead per beam alignment attempt, which affects the instantaneous performance, is the same as the offline method, and the detailed study has been reported in our prior work [1]. We thus only provide a summary of the main results here for completeness.

The overhead per beam alignment attempt of the proposed method is determined by the training budget $B_{\rm tr}$. When using 16×16 arrays at both the transmitter and receiver, $B_{\rm tr} = 30$ was shown in [1] to provide negligible performance loss compared to the exhaustive search. This is the reason we chose $B_{\rm tr} = 30$ to run our online beam pair selection algorithm so that the overhead in each beam training attempt is kept the same as in the offline approach from [1]. In [1], we compared the performance of the offline approach with two existing methods: IEEE 802.11ad and the method that uses the position only without the past beam measurements. The training overhead of our approach is less than a few percents of that of the IEEE 802.11ad method (the larger the array the smaller the percentage). Detailed comparison in the mobility context shows that our proposed beam alignment can support large arrays at high vehicular speed, while the IEEE 802.11ad struggles and its beam training time can eat up all the communication time before realignment is needed. It was also shown that using only the position results in severe performance loss when the blockage probability is not

negligible (e.g., in a dense traffic). In other words, the value of past beam measurements increases with the LOS blockage probability.

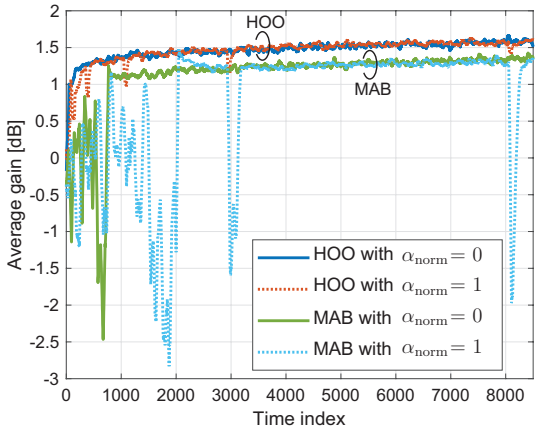
Regarding the position accuracy, our method only requires that the position is accurate enough to identify the location bin. To investigate the sensitivity to position error, in [1] we evaluated the performance for bin sizes ranging from 2m to 5m. The results show that for 16×16 arrays, there was no performance difference. Thus, we use the 5m location bin in our evaluation in this paper. When using a larger array such as 32×32, the position accuracy requirement increases and the results show that a smaller bin size such as 2m yields better performance. While the results show that larger arrays require more accurate position information, even for the large 16×16 array considered in this paper, only a few meters of accuracy is required.

B. Online beam pair refinement

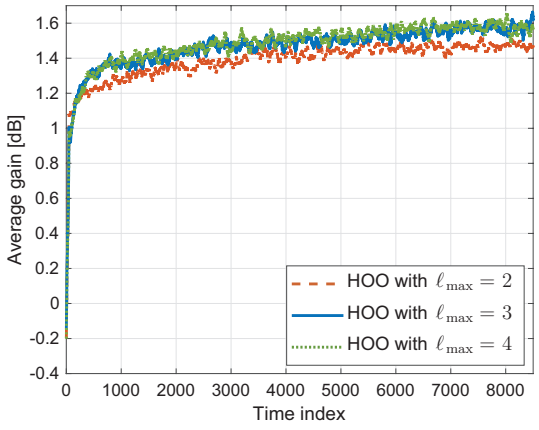
To evaluate the performance of the beam refinement on its own, we perform an offline beam pair selection using the MinMisProb method from [1] before running the beam refinement. In each simulation run, we use the first 300 channel samples to determine the selection set $\mathcal{S}$, and then we run the beam pair refinement on each of the beam pairs in $\mathcal{S}$ where we set the training budget to $B_{\rm tr} = 30$. For a baseline comparison, we implement an MAB solution using the norm-UCB algorithm from [28]. The MAB solution is run on the leaves of the search tree, and thus the number of arms is $16^{\ell_{\max}-1}$. Besides $B_{\rm tr}$, we also need to specify the maximum tree depth $\ell_{\max}$ and the forced exploration parameter $\alpha_{\rm norm}$. We use $K_{\min} = 3$ and $K_{\rm exd} = 10$.

We start by comparing the performance of MAB and our modified HOO solution in Fig. 9a. We can see that HOO learns much faster by leveraging the tree structure. We can see the cost of exploration of MAB in the initial stage, where each arm has to be tried $K_{\min}$ times. Using the search tree, starting from the root, HOO will first explore the nodes at depth 2. At depth 3, it explores only the children nodes of promising nodes at depth 2, and this goes on until reaching $\ell_{\max}$. This way, HOO does not have to sample all the leaves uniformly to explore the whole arm space leading to more efficient exploration than MAB.

We next show the effect of $\alpha_{\rm norm}$ and $\ell_{\max}$ on the performance. We noted earlier that $\alpha_{\rm norm} = 8$ is required to derive a regret bound in [28]. Forcing exploration this way with $\alpha_{\rm norm}$ turns out to result in bad performance for our applications as shown in Fig. 9a. The dips in the gains are due to this forced exploration, and the intervals between dips decrease as $\alpha_{\rm norm}$ increases. Note that even with $\alpha_{\rm norm} = 0$, both MAB and HOO still explore because of the confidence margin of the norm-UCB $\sqrt{16\hat{\sigma}^2_{\ell,k}\log(n)/T_{\ell,k}}$. Fig. 9a shows that for both MAB and HOO, $\alpha_{\rm norm} = 0$ provides the best performance. Fig. 9b compares the performance of HOO for $\ell_{\max} = 2, 3$ and 4 with $\alpha_{\rm norm} = 0$. We can see that a larger $\ell_{\max}$ improves the gains, which is expected since it allows a more refined search. Remarkably, thanks to the structure of the search tree, a larger $\ell_{\max}$ does not require more cost in the exploration. Since the performance improvement is quite small and the number of nodes in the tree increases quickly, we use $\ell_{\max} = 3$ from now on.



(a) HOO vs. MAB with $\ell_{\max} = 3$.



(b) HOO with different $\ell_{\max}$ and $\alpha_{\rm norm} = 0$.

Fig. 9. A comparison of HOO and MAB with different $\alpha_{\rm norm}$ and $\ell_{\max}$. Fig. (a) compares the performance when $\ell_{\max} = 3$. MAB does not use the hierarchical structure of the search tree as HOO and suffers a larger exploration penalty. The penalty is even more severe as $\alpha_{\rm norm}$ increases. The results show that the forced exploration is not needed and $\alpha_{\rm norm} = 0$ should be used. Fig. (b) compares the performance of HOO when using different $\ell_{\max}$. There is negligible gain for setting $\ell_{\max}$ beyond 3. We also see that HOO does not have extra degradation due to exploration when we increase $\ell_{\max}$.

Fig. 10 compares the HOO beam refinement with and without the smoothness coefficient $\nu(\ell)$. The performance difference is negligible. This is likely because the search region in our problem is already confined to a small local region (within the 3dB beamwidths of the selected beam pair) so that the constraint derived from the smoothness property does not have much value. This has a welcoming implication. The algorithm can be expected to be robust to small irregularity in the detailed shape of the beam patterns (thus affecting the exact smoothness property), which can be expected with real hardware.

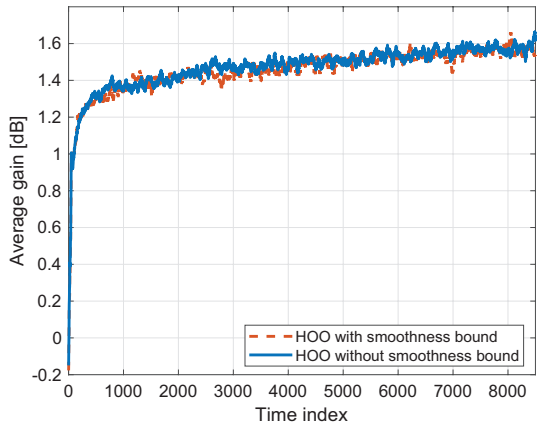


Fig. 10. A comparison of HOO with and without the smoothness coefficient $\nu(\ell)$ (for computing the U-values). The smoothness coefficient shows negligible effect. This is likely because the refinement problem searches locally within the 3dB beamwidth. Since this is in the vicinity of the optimal point, it is not possible to eliminate search regions using the smoothness bound.

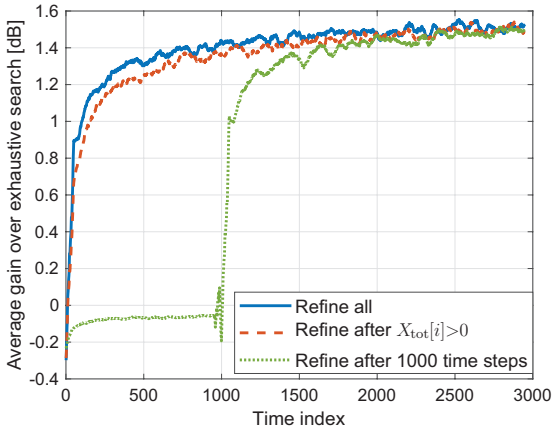


Fig. 11. A comparison of average gain of the integrated solution with the three options for when to start the beam refinement. The plots show no negative impact of the beam pair refinement on the online learning for beam pair selection. It is best to start the refinement simultaneously with the online beam pair selection.

C. Integrated online learning solution

This subsection evaluates the performance when combining the beam pair selection and refinement together. One thing that needs to be specified when combining the two is when to start the refinement for a selected beam pair. We consider the following three variations to start the beam refinement:

1) Refine all: The beam pair refinement is started for any beam pair from the first time it is selected by the online beam pair selection algorithm. This is the most straightforward way to combine the two components.
2) Refine after $X_{\rm tot}[i] > 0$: The refinement of the beam pair i starts from the time step that the beam pair i receives a reward, i.e., when $X_{\rm tot}[i]$ becomes positive. The point for this option is that the algorithm only refines those beam pairs deemed to be most promising.
3) Refine after n_0 time steps: The beam refinement of all selected beam pairs starts after running the online beam pair selection for n_0 time steps. The rationale for this option is to prevent the beam pair refinement algorithm from affecting the learning of the beam pair selection algorithm. This option allows the beam pair selection to run for a while so that it stabilizes to some extent before starting the beam pair refinement.

While it seems more efficient to focus the refinement on promising beam pairs only as in Option 2, refining suboptimal beam pairs as well will maximize their average received signal and could reduce the risk of large power loss. Thus, it is not obvious which option provides the best performance.

Fig. 11 compares the average gains over the exhaustive search (on the original codebook) of the three options. Here, $B_{\rm tr} = 30$, $\Gamma_{\rm risk} = 5$dB, $\ell_{\max} = 3$, $\alpha_{\rm norm} = 0$, and no smoothness coefficient is used (i.e., $\nu(\ell) = 1$). We can see that the first option, which is also the most straightforward one, provides the best performance. Focusing just on promising beam directions as in the second option performs quite well but is slightly worse than the first option. The results show that there is no benefit in waiting for some time before enabling the beam refinement as in the third option.

VIII. Conclusions

In this paper, we proposed position-based online learning algorithms for beam pair selection and refinement. We used the MAB framework to develop a risk-aware greedy UCB algorithm for beam pair selection and a modified HOO for the beam pair refinement. Combining the two solutions together, we can gain up to about 1.5dB over the received power obtained by exhaustive search over the original beam codebook before refinement. The learning is fast and it achieves an average gain of about 1dB within the first 100 time steps. While we only use position in this paper, more side information from sensors on devices or the BS about the environment will help further reduce the beam training overhead. As shown in this work, even efficient learning algorithms can be impractical without risk-awareness because the focus is on cumulative rather than instantaneous performance. Therefore, we believe risk-awareness is a key to developing practical online learning solutions to take full advantage of these sensors to enable fast and efficient mmWave communications.

Appendix A
Proof of Theorem 1

The regret is non-zero when one or more suboptimal beam pairs are selected. Thus, the total expected regret can be bounded by the average number of times suboptimal pairs are selected. We note that this derivation follows the steps of the UCB1 derivation from [28, Theorem 1] with the exception of the multiple-play setting. We provide the full details for completeness and readability. Denote ℓ and $i^\star$ the indices of a suboptimal and optimal pair. Denote $T_{\ell,i^\star}[n]$ the number of times ℓ is selected instead of $i^\star$ up to time n, the expected regret is

$$R_{\ell,i^\star}[n] = \mathbb{E}\left[T_{\ell,i^\star}[n]\right]\Delta_{\ell,i^\star}, \tag{32}$$

where $\Delta_{\ell,i^\star}$ is the optimality gap defined in (21). This is because whether ℓ is selected or not at time n depends on the rewards up to time $n-1$ and the loss depends only on the rewards at n. Thus, the two are independent by the assumption of independent reward signals across time.

We will now compute a bound for $\mathbb{E}\left[T_{\ell,i^\star}[n]\right]$. A *necessary* condition for the pair ℓ to be selected instead of the pair $i^\star$ is that $\mathsf{UCB}_\ell > \mathsf{UCB}_{i^\star}$. After the pair ℓ has been selected $T^{(0)}_{\ell,i^\star}$ times, the number of times ℓ is selected instead of $i^\star$ up to time n can be bounded by

$$T_{\ell,i^\star}[n] \le T^{(0)}_{\ell,i^\star} + \sum_{t=t_0}^{n} \mathbf{1}\left\{\mathsf{UCB}_\ell \ge \mathsf{UCB}_{i^\star}, T_\ell[t-1] \ge T^{(0)}_{\ell,i^\star}\right\} \tag{33}$$

$$= T^{(0)}_{\ell,i^\star} + \sum_{t=t_0}^{n} \mathbf{1}\left\{\hat{P}_{\mathrm{opt}}(\ell) + c_{t-1,T_\ell[t-1]} \ge \hat{P}_{\mathrm{opt}}(i^\star) + c_{t-1,T_{i^\star}[t-1]}, T_\ell[t-1] \ge T^{(0)}_{\ell,i^\star}\right\} \tag{34}$$

$$\le T^{(0)}_{\ell,i^\star} + \sum_{t=t_0}^{n} \mathbf{1}\left\{\max_{T^{(0)}_{\ell,i^\star} < u_\ell < t} \left\{\hat{P}_{\mathrm{opt}}(\ell) + c_{t-1,u_\ell}\right\} \ge \min_{0<u<t} \left\{\hat{P}_{\mathrm{opt}}(i^\star) + c_{t-1,u}\right\}\right\} \tag{35}$$

$$\le T^{(0)}_{\ell,i^\star} + \sum_{t=1}^{n-1}\sum_{u=1}^{t-1}\sum_{u_\ell=T^{(0)}_{\ell,i^\star}}^{t-1} \mathbf{1}\left\{\hat{P}_{\mathrm{opt}}(\ell) + c_{t,u_\ell} \ge \hat{P}_{\mathrm{opt}}(i^\star) + c_{t,u}\right\}. \tag{36}$$

where $t_0 \ge T^{(0)}_{\ell,i^\star}$ and $c_{t,u} = \sqrt{2\log(t)/u}$ is the confidence margin. For $\{\hat{P}_{\mathrm{opt}}(\ell) + c_{t,u_\ell} \ge \hat{P}_{\mathrm{opt}}(i^\star) + c_{t,u}\}$ to be true, at least one of the followings must hold

$$\hat{P}_{\mathrm{opt}}(i^\star) \le P_{\mathrm{opt}}(i^\star) - c_{t,u} \tag{37}$$

$$\hat{P}_{\mathrm{opt}}(\ell) \ge P_{\mathrm{opt}}(\ell) + c_{t,u_\ell} \tag{38}$$

$$P_{\mathrm{opt}}(i^\star) < P_{\mathrm{opt}}(\ell) + 2c_{t,u_\ell}. \tag{39}$$

Note that (37) means the UCB value underestimates the true reward of pair $i^\star$, and (38) means the UCB value overestimates the true reward of pair ℓ by larger than the corresponding confidence margins. Setting $T^{(0)}_{\ell,i^\star} = \lceil 8\log(n)/\Delta^2_{\ell,i^\star}\rceil$, it can be shown that (39) is impossible [28, p. 243], and we can bound $\mathbb{E}[T_{\ell,i^\star}[n]]$ by

$$\mathbb{E}[T_{\ell,i^\star}[n]] \le \left\lceil \frac{8\log(n)}{\Delta^2_{\ell,i^\star}} \right\rceil + \sum_{t=1}^{n-1}\sum_{u=1}^{t-1}\sum_{u_\ell=T^{(0)}_{\ell,i^\star}}^{t-1} \mathbb{P}[(37) \text{ is true}] + \mathbb{P}[(38) \text{ is true}] \tag{40}$$

$$\le \left\lceil \frac{8\log(n)}{\Delta^2_{\ell,i^\star}} \right\rceil + \sum_{t=1}^{\infty}\sum_{u=1}^{t-1}\sum_{u_\ell=T^{(0)}_{\ell,i^\star}}^{t-1} (t^{-4} + t^{-4}) \tag{41}$$

$$\le \left\lceil \frac{8\log(n)}{\Delta^2_{\ell,i^\star}} \right\rceil + 2\sum_{t=1}^{\infty}\sum_{u=1}^{t}\sum_{u_\ell=1}^{t} t^{-4} \tag{42}$$

$$\le \left\lceil \frac{8\log(n)}{\Delta^2_{\ell,i^\star}} \right\rceil + 2\frac{\pi^2}{6}. \tag{43}$$

The second line in (41) follows because the probability terms can be shown to be bounded by t^{-4} using the Chernoff-Hoeffding inequality [28].

The total regret bound follows by summing all pairs of optimal and suboptimal beam pairs:

$$R_1[n] \le \sum_{\ell\in\hat{\mathcal{B}}\setminus\mathcal{S}^\star}\sum_{i^\star\in\mathcal{S}^\star} \mathbb{E}[T_{\ell,i^\star}[n]]\Delta_{\ell,i^\star}. \tag{44}$$

Substituting (43) in and after some algebra, we obtain (22).

Appendix B
Proof of Theorem 2

The derivation follows similarly to that of Theorem 1, but we need to be careful about the rejection mechanism. Even if a pair is selected by the greedy UCB selection, it will not be used for the training if it is rejected. Let $\tilde{T}_{\ell,i^\star}$ and $T_{\ell,i^\star}$ be the number of times the pair ℓ is selected instead of the pair $i^\star$ and the number of times it is accepted for beam training, respectively. We proceed similarly to obtain a bound similar to (36) given by

$$\tilde{T}_{\ell,i^\star}[n] \le \tilde{T}^{(0)}_{\ell,i^\star} + \sum_{t=1}^{n-1}\sum_{u=1}^{t-1}\sum_{\tilde{u}_\ell=\tilde{T}^{(0)}_{\ell,i^\star}}^{t-1} \mathbf{1}\{\hat{P}_{\mathrm{opt}}(\ell) + c_{t,u_\ell} \ge \hat{P}_{\mathrm{opt}}(i^\star) + c_{t,u}\}. \tag{45}$$

To compute the bound on $\mathbb{E}\left[\tilde{T}_{\ell,i^\star}[n]\right]$, we again use (37)-(39). The probability bounds on (37) and (38) are still applicable. Because of the rejection, we cannot guarantee that (39) is impossible, but we can bound its probability. Note that $\tilde{u}_\ell$ is the number of times the pair ℓ is selected, and u_ℓ is the number of times it is accepted for beam training. It can be shown that (39) is impossible if $u_\ell > \frac{8\log(t)}{\Delta^2_{\ell,i^\star}}$ [28, p. 243]. Thus, we can bound the probability that (39) holds by $\mathbb{P}\left[u_\ell \le \frac{8\log(t)}{\Delta^2_{\ell,i^\star}}\right]$. With the acceptance probability ζ_ℓ, we have $\mathbb{E}[u_\ell] = \mathbb{E}[\tilde{u}_\ell]\zeta_\ell$. Setting $\tilde{T}^{(0)}_{\ell,i^\star} = \lceil 8\log(n)/(\zeta_\ell\delta^2\Delta^2_{\ell,i^\star})\rceil$, we get the following bound

$$\mathbb{P}\left[u_\ell \le \frac{8\log(t)}{\Delta^2_{\ell,i^\star}}\middle|\, \tilde{u}_\ell = t\right] \le \mathbb{P}\left[u_\ell \le \frac{8\log(t)}{\Delta^2_{\ell,i^\star}}\middle|\, \tilde{u}_\ell = \tilde{T}^{(0)}_{\ell,i^\star}\right] \tag{46}$$

$$\le \mathbb{P}\left[u_\ell \le \frac{8\log(n)}{\Delta^2_{\ell,i^\star}}\middle|\, \tilde{u}_\ell = \tilde{T}^{(0)}_{\ell,i^\star}\right] \tag{47}$$

$$\le n^{-4}. \tag{48}$$

Here, (46) follows because $t \ge t_0 \ge \tilde{T}^{(0)}_{\ell,i^\star}$, (47) holds because $t < n$, and (48) is the application of the lower tail of the Chernoff bound with $\delta = (\sqrt{5}-1)/2$ for the Bernoulli distribution [46, Theorem 4]. Taking the expectation of (45)

and substitute the probability bounds for (37)-(39) to hold, we get

$$\mathbb{E}\left[\tilde{T}_{\ell,i^\star}[n]\right] \leq \tilde{T}^{(0)}_{\ell,i^\star} + \sum_{t=1}^{n-1}\sum_{u=1}^{t-1}\sum_{\tilde{u}_\ell=\tilde{T}^{(0)}_{\ell,i^\star}}^{t-1}(2t^{-4}+n^{-4}) \quad (49)$$

$$\leq \tilde{T}^{(0)}_{\ell,i^\star} + 3\sum_{t=1}^{\infty}\sum_{u=1}^{t}\sum_{\tilde{u}_\ell=1}^{t} t^{-4} \quad (50)$$

$$= \tilde{T}^{(0)}_{\ell,i^\star} + \pi^2/2. \quad (51)$$

To obtain the regret, we note that when the pair ℓ is selected the regret incurred is $\zeta_\ell\Delta_{\ell,i^\star} + (1-\zeta_\ell)\tilde{\Delta}_{\ell,i^\star}$ because when rejected (with probability $1-\zeta_\ell$), the regret is $\tilde{\Delta}_{\ell,i^\star}$. The total expected regret is then

$$R_2[n] \leq \sum_{\ell\in\hat{\mathcal{B}}\setminus\mathcal{S}^\star}\sum_{i^\star\in\mathcal{S}^\star}\mathbb{E}[\tilde{T}_{\ell,i^\star}[n]](\zeta_\ell\Delta_{\ell,i^\star} + (1-\zeta_\ell)\tilde{\Delta}_{\ell,i^\star}), \quad (52)$$

which after rearranging terms will result in (23).

Appendix C
Proof of Lemma 1

Assuming a normalized PAS, then the single-path PAS can be represented by the delta function $\delta(\phi-\phi^\star)$. The average received power can be written as

$$\bar{\gamma}(\phi_0) = \int_{\phi_0-\Delta\phi}^{\phi_0+\Delta\phi}\delta(\phi-\phi^\star)G(\phi;\phi_0)\mathrm{d}\phi. \quad (53)$$

Since the gain $G(\phi;\phi_0)$ is decreasing in $[\phi_0, \phi_0+\Phi/2]$ and $|\phi^\star-\phi_0| \leq \Delta\phi \leq \Phi/2$ by the assumption of the Lemma,

$$\frac{G(\phi;\phi_0)}{G(\phi_0+\Delta\phi;\phi_0)} \geq 1 \geq G(\phi;\phi^\star), \quad \forall\phi\in[\phi_0-\Delta\phi,\phi_0]\cup[\phi_0,\phi_0+\Delta\phi]. \quad (54)$$

Multiply both sides by $\delta(\phi-\phi^\star)$ and integrate to get

$$\int_{\phi_0-\Delta\phi}^{\phi_0+\Delta\phi}\delta(\phi-\phi^\star)G(\phi;\phi_0)/G(\phi_0+\Delta\phi;\phi_0)\mathrm{d}\phi \geq \int_{\phi_0-\Delta\phi}^{\phi_0+\Delta\phi}\delta(\phi-\phi^\star)G(\phi;\phi^\star)\mathrm{d}\phi \quad (55)$$

$$\bar{\gamma}(\phi_0)/G(\phi_0+\Delta\phi;\phi_0) \geq \bar{\gamma}(\phi^\star). \quad (56)$$

Now, for a more general PAS $\mathcal{P}(\phi)$ with a bounded support in $[\phi_0-\Psi, \phi_0+\Psi]$, the average received power can be written as

$$\bar{\gamma}(\phi_0) = \int_{\phi_0-\Psi}^{\phi+\Psi}\mathcal{P}(\phi)G(\phi;\phi_0)\mathrm{d}\phi \quad (57)$$

$$= \int_{\phi_0-\Psi}^{\phi_0-\Delta\phi}\mathcal{P}(\phi)G(\phi;\phi_0)\mathrm{d}\phi + \int_{\phi_0-\Delta\phi}^{\phi_0+\Delta\phi}\mathcal{P}(\phi)G(\phi;\phi_0)\mathrm{d}\phi + \int_{\phi_0+\Delta\phi}^{\phi_0+\Psi}\mathcal{P}(\phi)G(\phi;\phi_0)\mathrm{d}\phi. \quad (58)$$

By the same argument as in the single-path PAS case, we have

$$\mathcal{P}(\phi)G(\phi;\phi_0)\frac{G(\phi_0;\phi_0)}{G(\phi_0+\Delta\phi;\phi_0)} \geq \mathcal{P}(\phi)G(\phi;\phi^\star), \quad \forall\phi\in[\phi_0-\Delta\phi,\phi_0+\Delta\phi] \quad (59)$$

$$\mathcal{P}(\phi)G(\phi;\phi_0)\frac{G(\phi_0+\Psi-\Delta\phi;\phi_0)}{G(\phi_0+\Psi;\phi_0)} \geq \mathcal{P}(\phi)G(\phi;\phi^\star), \quad \forall\phi\in[\phi_0-\Psi,\phi_0-\Delta\phi]\cup[\phi_0+\Delta\phi,\phi_0+\Psi]. \quad (60)$$

Taking the integral of (59) and (60), we have

$$\bar{\gamma}(\phi_0)/G(\phi_0+\Delta\phi) \geq \bar{\gamma}(\phi^\star) - \mathsf{Err} \quad (61)$$

where

$$\mathsf{Err} = \left(\frac{G(\phi_0+\Psi-\Delta\phi;\phi_0)}{G(\phi_0+\Psi;\phi_0)} - \frac{1}{G(\phi_0+\Delta\phi;\phi_0)}\right)\times \left(\int_{\phi_0-\Psi}^{\phi_0-\Delta\phi}\mathcal{P}(\phi)G(\phi;\phi_0)\mathrm{d}\phi + \int_{\phi_0+\Delta\phi}^{\phi_0+\Psi}\mathcal{P}(\phi)G(\phi;\phi_0)\mathrm{d}\phi\right). \quad (62)$$

Because $G(\phi;\phi_0)$ is decreasing and concave for $\phi\in[\phi_0,\phi_0+\Phi/2]$ by the assumptions of the Lemma (e.g., true for a uniform planar array), the coefficient is positive and decreasing as $\Delta\phi$ decreases. Since the integrands are positive by definition, Err is positive. Further, because the sum of the integrals in (62) is less than $\bar{\gamma}(\phi_0)$ (thus, finite), we have $\mathsf{Err}\to 0$ as $\Delta\phi\to 0$. Also, for small Ψ the integration intervals decrease and when $\Psi\leq\Delta\phi$ they disappear, i.e., $\mathsf{Err}=0$.

References

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

[2] F. Devoti, I. Filippini, and A. Capone, "Facing the millimeter-wave cell discovery challenge in 5G networks with context-awareness," *IEEE Access*, vol. 4, pp. 8019–8034, Nov. 2016.

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

[4] V. Va, T. Shimizu, G. Bansal, and R. W. Heath Jr., "Millimeter wave vehicular communications: A survey," *Found. Trends Netw.*, vol. 10, no. 1, pp. 1–113, 2016.

[5] S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvári, "X-armed bandits," *J. Mach. Learn. Res.*, vol. 12, pp. 1655–1695, Jul. 2011.

[6] J. Seo, Y. Sung, G. Lee, and D. Kim, "Training beam sequence design for millimeter-wave MIMO systems: A POMDP framework," *IEEE Trans. Signal Process.*, vol. 64, no. 5, pp. 1228–1242, Mar. 2016.

[7] M. Scalabrin, N. Michelusi, and M. Rossi, "Beam training and data transmission optimization in millimeter-wave vehicular networks," https://arxiv.org/abs/1804.10181, Apr. 2018.

[8] M. Hashemi, A. Sabharwal, C. E. Koksal, and N. B. Shroff, "Efficient beam alignment in millimeter wave systems using contextual bandits," https://arxiv.org/abs/1712.00702, Apr. 2018, *IEEE INFOCOM*.

[9] J. Wang *et al.*, "Beam codebook based beamforming protocol for multi-Gbps millimeter-wave WPAN systems," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 8, pp. 1390–1399, Oct. 2009.

[10] K. Hosoya *et al.*, "Multiple sector ID capture (MIDC): A novel beamforming technique for 60-GHz band multi-Gbps WLAN/PAN systems," *IEEE Trans. Antennas Propag.*, vol. 63, no. 1, pp. 81–96, Jan. 2015.

[11] Q. Duan, T. Kim, H. Huang, K. Liu, and G. Wang, "AoD and AoA tracking with directional sounding beam design for millimeter wave MIMO systems," in *Proc. IEEE Int. Symp. on Personal, Indoor and Mobile Radio Commun.*, Aug. 2015, pp. 2271–2276.

[12] Z. Marzi, D. Ramasamy, and U. Madhow, "Compressive channel estimation and tracking for large arrays in mm-wave picocells," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 514–527, 2016.

[13] T. Kadur, H.-l. Chiang, and G. Fettweis, "Effective beam alignment algorithm for low cost millimeter wave communication," in *Proc. IEEE Veh. Technol. Conf.*, 2016, pp. 1–5.
[14] B. Li, Z. Zhou, W. Zou, X. Sun, and G. Du, "On the efficient beam-forming training for 60 GHz wireless personal area networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 2, pp. 504–515, 2013.
[15] T. Nitsche, A. B. Flores, E. W. Knightly, and J. Widmer, "Steering with eyes closed: Mm-wave beam steering without in-band measurement," in *Proc. IEEE Int. Conf. on Comp. and Comm. (INFOCOM)*, vol. 26, 2015, pp. 2416–2424.
[16] V. Va, T. Shimizu, G. Bansal, and R. W. Heath Jr., "Beam design for beam switching based millimeter wave vehicle-to-infrastructure communications," in *Proc. IEEE Int. Conf. on Commun.*, May 2016, pp. 1–6.
[17] N. Garcia, H. Wymeersch, E. G. Strom, and D. Slock, "Location-aided mm-wave channel estimation for vehicular communication," in *Proc. IEEE Workshop on Signal Process. Adv. in Wireless Commun.*, no. 5. IEEE, Jul. 2016, pp. 1–5.
[18] W. B. Abbas and M. Zorzi, "Context information based initial cell search for millimeter wave 5G cellular networks," in *Proc.of the EuCNC*, Jun. 2016, pp. 111–116.
[19] J. Choi, V. Va, N. Gonzalez-Prelcic, R. C. Daniels, C. R. Bhat, and R. W. Heath Jr., "Millimeter-wave vehicular communication to support massive automotive sensing," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 160–167, Dec. 2016.
[20] Y. Wang, M. Narasimha, and R. W. Heath Jr, "MmWave beam prediction with situational awareness: A machine learning approach," https://arxiv.org/abs/1805.08912, Jun. 2018, in *Proc. of IEEE SPAWC*.
[21] F. Maschietti, D. Gesbert, P. de Kerret, and H. Wymeersch, "Robust location-aided beam alignment in millimeter wave massive MIMO," in *Proc. IEEE Global Commun. Conf.*, Dec. 2017, pp. 1–6.
[22] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 98–105, Apr. 2017.
[23] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks," https://arxiv.org/abs/1710.02913, Oct. 2017.
[24] H. Ye, L. Liang, G. Y. Li, J. Kim, L. Lu, and M. Wu, "Machine learning for vehicular networks: Recent advances and application examples," *IEEE Veh. Technol. Mag.*, vol. 13, no. 2, pp. 94–101, Jun. 2018.
[25] V. Va, T. Shimizu, G. Bansal, and R. W. Heath Jr., "Position-aided millimeter wave V2I beam alignment: A learning-to-rank approach," in *Proc. IEEE Int. Symp. on Personal, Indoor and Mobile Radio Commun.*, Oct. 2017, pp. 1–5.
[26] 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, Jun. 2018.
[27] N. Gulati and K. R. Dandekar, "Learning state selection for reconfigurable antennas: A multi-armed bandit approach," *IEEE Trans. Antennas Propag.*, vol. 62, no. 3, pp. 1027–1038, Mar. 2014.
[28] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, pp. 235–256, 2002.
[29] S. Bubeck and N. Cesa-Bianchi, *Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems*, ser. Found. Trends Mach. Learn. Now Publishers, 2012.
[30] G. Burtini, J. Loeppky, and R. Lawrence, "A survey of online experiment design with the stochastic multi-armed bandit," https://arxiv.org/abs/1510.00757, Nov. 2015.
[31] W. Chen, Y. Wang, and Y. Yuan, "Combinatorial multi-armed bandit: General framework and applications," in *Proc. Int. Conf. on Mach. Learn.*, Jun. 2013, pp. 151–159.
[32] B. Kveton, C. Szepesvári, Z. Wen, and A. Ashkan, "Cascading bandits: Learning to rank in the cascade model," in *Proc. Int. Conf. on Mach. Learn.*, 2015, pp. 767–776.
[33] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," in *Proc. Int. Conf. on Mach. Learn.*, 2010, pp. 1015–1022.
[34] R. Kleinberg, A. Slivkins, and E. Upfal, "Multi-armed bandits in metric spaces," in *Proc. 40th Annu. ACM Symp. on Theory of Comput.*, 2008, pp. 681–690.
[35] R. Munos, "From bandits to Monte-Carlo tree search: The optimistic principle applied to optimization and planning," *Found. Trends Mach. Learn.*, vol. 7, no. 1, pp. 1–129, 2014.
[36] R. W. Heath Jr., N. Gonzalez-Prelcic, S. Rangan, W. Roh, and A. 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, Apr. 2016.
[37] R. W. Heath Jr., *Introduction to Wireless Digital Communication: A Signal Processing Perspective*, 1st ed. Prentice Hall, Mar. 2017.
[38] V. Va, J. Choi, T. Shimizu, G. Bansal, and R. W. Heath Jr., "Impact of measurement noise on millimeter wave beam alignment using beam subsets," *IEEE Wireless Comm. Lett.*, vol. 7, no. 5, pp. 1–4, Oct. 2018.
[39] C. A. Balanis, *Antenna Theory: Analysis and Design*, 3rd ed. Wiley-Interscience, 2005.
[40] Remcom, "Wireless InSite," http://www.remcom.com/wireless-insite.
[41] D. Chizhik, J. Du, R. Feick, M. Rodriguez, G. Castro, and R. A. Valenzuela, "Path loss, beamforming gain and time dynamics measurements at 28 GHz for 90% indoor coverage," https://arxiv.org/abs/1712.06580, Bell Laboratories, Nokia, Tech. Rep., Dec. 2017.
[42] V. Va, J. Choi, and R. W. Heath Jr., "The impact of beamwidth on temporal channel variation in vehicular channels and its implications," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 5014–5029, Jun. 2017.
[43] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.
[44] G. Grimmett and D. Stirzaker, *Probability and Random Processes*, 3rd ed. Oxford University Press, 2001.
[45] M. Jordan, "The exponential family: Conjugate priors," https://people.eecs.berkeley.edu/∼jordan/courses/260-spring10/other-readings/chapter9.pdf, 2009, lecture Note.
[46] M. Goemans, "Chernoff bounds, and some applications," http://math.mit.edu/∼goemans/18310S15/chernoff-notes.pdf, Feb. 2015, lecture Note.