Position-Aided Millimeter Wave V2I Beam Alignment: A Learning-to-Rank Approach

Vutha Va*, Takayuki Shimizu†, Gaurav Bansal†, and Robert W. Heath Jr.*
*Wireless Networking and Communications Group, The University of Texas at Austin
†TOYOTA InfoTechnology Center, U.S.A., Inc., Mountain View, CA
Email: {vutha.va, rheath}@utexas.edu, {tshimizu, gbansal}@us.toyota-itc.com

***Abstract*—Millimeter wave (mmWave) could be a key technology to support high data rate demands for automated vehicles. MmWave needs array gain for the best performance, but this requires correctly pointing the beam, known as beam alignment. Dynamic blockages make beam alignment challenging in the vehicular setting. This paper proposes to leverage a vehicle's position along with past beam measurements to rank desirable pointing directions that can reduce the required beam training to a small set of pointing directions. The ranking is conducted using a learning-to-rank approach, which is a popular machine learning method used in recommender systems. The learning uses a kernel based model, and a new metric for evaluating ranked lists of pointing directions tailored to beam alignment is proposed. The proposed method provides a scalable framework for exploiting context information.**

I. Introduction

Millimeter wave (mmWave) is a promising technology to provide high data rate vehicular communications to support advanced Intelligent Transportation System (ITS) applications [1]. Some examples include cooperative perception, infotainment, and cloud-based services [1], [2]. Because of the small antenna aperture at mmWave frequencies, large antenna arrays are required to provide sufficient link quality [2]. These arrays need to be properly configured, which is challenging, especially in the highly dynamic vehicular environments.

This paper proposes a position-aided approach that leverages past beam measurements to provide fast and efficient mmWave Vehicle-to-Infrastructure (V2I) beam alignment. Past beam measurements are collected in a database maintained and used by the roadside unit (RSU) to rank promising beam pointing directions that helps eliminate unnecessary beam training. Unlike our prior work in [3] that builds probabilistic models to produce a ranked list of beam pairs, this paper proposes to use a learning-to-rank (LtR) approach. LtR is a popular machine learning method used in recommender systems (e.g., recommending movies) and information retrieval systems (e.g., ranking web search results). The proposed method learns a kernel-based scoring function that predicts the scores and ranks different pointing directions. Since different types of context information can be included in the feature vector as input to the scoring function, this method provides a natural framework for exploiting other context information beyond position such as time of the day (correlated with traffic density). In this work, we focus on using only the position.

Beam alignment is important for establishing mmWave links. Conventional approaches often use a hierarchical search [4] which works well for short link distances such as indoor applications. More recently, there have been efforts to use side information. Multipath information from a lower frequency band [5] and from radar [6] has been exploited. More related to this paper are those that use context information [3], [7]–[10]. In line-of-sight (LOS) conditions, positions can be used directly to determine the pointing direction as reported in [7], [8]. In non-LOS (NLOS) conditions, it becomes more challenging and past beam measurements along with position are used to determine promising pointing directions in [3], [9], [10]. It was proposed in [9] to record past useful pointing directions indexed by location. No method for ranking the pointing directions was given in [9]. In [10], a heuristic based on observed past channel strengths was proposed. The search is only in azimuth and horn antennas are assumed in [10]. We assume uniform planar arrays (UPAs) at both the transmitter and the receiver and both azimuth and elevation are considered. Our prior work in [3] uses a probabilistic model for ranking while in this work a machine learning approach is used.

II. System Model

This paper considers a V2I setting in an urban street environment with dense traffic where LOS is often unavailable and is challenging for beam alignment. The channels are generated using a ray-tracing simulator, Wireless InSite [11]. A snapshot of the environment is shown in Fig. 1, where the street has two lanes. Two types of vehicles are modeled: cars and trucks with the ratio of 3 to 2. Depending on the locations of the trucks in the adjacent lane the LOS link can be blocked. The channels used in this study consists of 500 snapshots of the environment where in each snapshot the location of the vehicles are placed randomly with their gaps (bumper to bumper distance) drawn from the Erlang distribution. The carrier frequency used in the simulation is 60 GHz.

978-1-5386-3531-5/17/$31.00

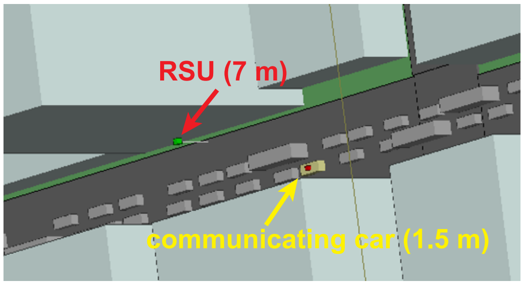

Fig. 1. A snapshot of the ray-tracing simulation environment. Numbers in parenthesis are antenna heights.

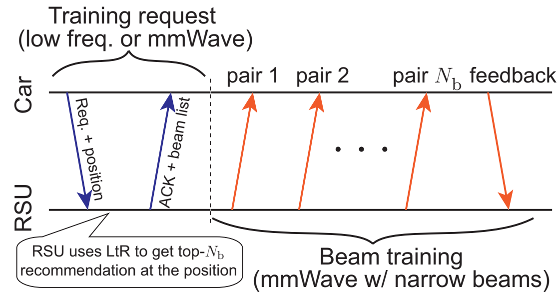

Fig. 2. Timing diagram of the data-driven beam alignment method.

The channel parameters generated from the ray-tracer are post-processed in MATLAB to compute channel strengths for different transmit and receive beam pairs. Denote N_{t} and N_{r} the numbers of transmit and receive antennas, L_{p} the number of paths, α_ℓ the complex channel gain, $\boldsymbol{\Theta}_\ell^{\mathrm{A}} = [\theta_\ell^{\mathrm{A}}, \varphi_\ell^{\mathrm{A}}]^{\mathrm{T}}$ and $\boldsymbol{\Theta}_\ell^{\mathrm{D}} = [\theta_\ell^{\mathrm{D}}, \varphi_\ell^{\mathrm{D}}]^{\mathrm{T}}$ the azimuth and elevation angles of arrival and departure, τ_ℓ the delay of the ℓ-th path, $p(\cdot)$ the combined pulse shaping and lowpass filter response, T the symbol duration, and $\boldsymbol{a}_{\mathrm{t}}$ and $\boldsymbol{a}_{\mathrm{r}}$ the transmit and receive array response, the channel matrix at the n-th tap is given by

$$\boldsymbol{H}[n] = \sqrt{N_{\mathrm{t}} N_{\mathrm{r}}} \sum_{\ell=0}^{L_{\mathrm{p}}-1} \alpha_\ell p(nT - \tau_\ell) \boldsymbol{a}_{\mathrm{r}}(\boldsymbol{\Theta}_\ell^{\mathrm{A}}) \boldsymbol{a}_{\mathrm{t}}^*(\boldsymbol{\Theta}_\ell^{\mathrm{D}}). \tag{1}$$

UPAs are assumed at both the RSU and the car. We assume analog beamforming, where both the transmitter and receiver have only one RF chain and beam steering is controlled by phase-shifters. When using the i-th beam pair consisting of the beamformer $\boldsymbol{f}_{t(i)}$ and combiner $\boldsymbol{w}_{r(i)}$, where $t(i)$ and $r(i)$ are some index mappings, the channel strength is given by

$$\gamma_i = \left\| \boldsymbol{w}_{r(i)}^* \sum_{n=0}^{L} \boldsymbol{H}[n] \boldsymbol{f}_{t(i)} \right\|^2, \tag{2}$$

where $L = 512$ is the channel length. Both $\boldsymbol{f}_{t(i)}$ and $\boldsymbol{w}_{r(i)}$ are subject to the constant modulus constraint of the phase-shifters. We suppose the use of a codebook with beams generated by progressive phase shift between antenna elements. The beams' main lobe directions are separated by their 3 dB beamwidth. Note that the proposed method does not depend on this codebook and any other choices can be used. We neglect the noise here as its effect is small as shown in [3]. The system bandwidth is set to 1760 MHz following IEEE 802.11ad. See [3] for a more detailed description of the system model.

III. Data-Driven Beam Alignment

The main idea of this approach is to use context information (e.g., vehicle's position) and past beam measurements stored in a database (maintained by the RSU) as a hint to determine potential beam pointing directions.

TABLE I
An example of the database. The ℓ-th row corresponds to measurement data contributed by a car at position $\boldsymbol{x}_\ell$. The number on the top of each cell is the beam pair index and the number at the bottom is the channel strength. In our numerical result, $k = 100$ is used.

Position	Best	2nd best	...	k-th best
$\boldsymbol{x}_1$	5 -64.5 dBm	159 -69.2 dBm	... ...	346 -95.8 dBm
$\boldsymbol{x}_2$	159 -70.4 dBm	263 -72.6 dBm	... ...	354 -97.1 dBm
...	...	...	...	...
$\boldsymbol{x}_N$	5 -66.4 dBm	258 -68.1 dBm	... ...	2 -82.6 dBm

With a good method to identify promising beam pairs (i.e., beam pair selection), only a small number of pointing directions need to be searched reducing the alignment overhead. A timing diagram of the method is shown in Fig. 2. First, the car sends a training request with its position information to the RSU. If the RSU is available, it first uses the position information and the learned model to produce a ranked list of promising beam pairs and sends back an ACK with the list. Then, beam training following the list is conducted. After that, the RSU feedbacks the best beam direction and high speed mmWave communication can start.

We now describe our database. We assume an offline learning setting where the database has already been collected before exploitation in the beam alignment. Each vehicle contributing to the data collection conducts beam training over all beam pairs. The RSU records the location of the contributing vehicle and the top-k beam pairs ranked by the measured channel strength. Note that by having the vehicle transmit during the training, there is no need to feedback the channel strength of the top-k directions to the RSU. An example of the database is shown in Table I.

IV. Beam Pair Selection Methods

The main component of the proposed beam alignment is the method to select promising beam pairs using position information (or other contexts if available) and the database. In our prior work [3], position information was quantized into location bin and a probabilistic model for each bin is constructed using the database. While this approach provides a clear interpretation, it does not scale well when different types of context information, such as time of the day (to infer traffic density), becomes available. Assuming there are d contexts and each context is quantized into b bins, then there are b^d models to construct and maintain. This number b^d grows fast with d. A more scalable approach is to use the context information directly without binning. This motivates our contribution in this paper which is to apply a LtR approach for beam selection.

A. Baseline Beam Pair Selection Method

This subsection describes the baseline approach proposed in [3]. In this approach, positions are quantized into bins, and beam pairs are selected to minimize the misalignment probability. The misalignment probability is estimated using the samples available in the database (at the location bin). The misalignment probability for training budget N_{b} is defined as the probability that the optimal beam pair is not within the top-N_{b} positions in the ranked list of beam pairs to be trained. It is shown in [3] that the solution to this problem is to rank beam pairs by their probability of being optimal. The probability of being optimal of the i-th beam pair at the a-th location bin can be estimated using the Monte Carlo method as

$$P^a_{\mathrm{opt}}(i) = \frac{\sum_{\ell=1}^{N} \mathbf{1}\left[\gamma_{i\ell} = \max_{j\in\mathcal{B}} \gamma_{j,\ell}\right] \mathbf{1}[\boldsymbol{x}_\ell \in X_a]}{\sum_{\ell}^{N} \mathbf{1}[\boldsymbol{x}_\ell \in X_a]}, \tag{3}$$

where N is the total number of positions available in the database, $\mathcal{B}$ is the set of all beam pair indices, $\boldsymbol{x}_\ell$ is the position of the ℓ-th data point, $\gamma_{i\ell}$ is the channel strength of the i-th beam pair at $\boldsymbol{x}_\ell$, X_a is the interval defining the a-th location bin, and $\mathbf{1}[\cdot]$ is the indicator function. The denominator is the number of data points in the a-th location bin. This essentially counts how often a given beam pair was observed to be the optimal pair in the database in the location bin, and thus all beam pairs not observed to be optimal in the database have 0 probability. In the simulated database, the number of non-zero probability of being optimal is around 30 and (3) cannot rank beam pairs for $N_{\mathrm{b}} > 30$. In [3], this problem is overcome by combining with the list from a suboptimal version of the method.

B. LtR Beam Pair Selection Method

The idea of this approach is to learn a scoring function $\hat{z}(\cdot)$ that can be used to predict the scores of beam pairs and provides a means to rank them. For this purpose, we need to define a feature vector, which is the input to the scoring function, that distinguishes pointing directions. In particular, the feature vector corresponding to the i-th beam pair and position $\boldsymbol{x}_\ell$ is defined as

$$\boldsymbol{q}_{i\ell} = \left[(\boldsymbol{\Theta}^{\mathrm{rx}}_i)^{\mathrm{T}} \quad (\boldsymbol{\Theta}^{\mathrm{tx}}_i)^{\mathrm{T}} \quad \boldsymbol{x}^{\mathrm{T}}_\ell\right]^{\mathrm{T}}, \tag{4}$$

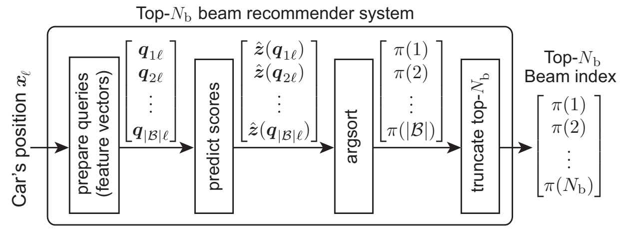

Fig. 3. Top-N_{b} recommendation process at car's position $\boldsymbol{x}_\ell$. $\pi(n)$ denotes the beam index with the n-th highest predicted score.

where $\boldsymbol{\Theta}^{\mathrm{rx}}_i = [\theta^{\mathrm{rx}}_i, \phi^{\mathrm{rx}}_i]^{\mathrm{T}}$ and $\boldsymbol{\Theta}^{\mathrm{tx}}_i = [\theta^{\mathrm{tx}}_i, \phi^{\mathrm{tx}}_i]^{\mathrm{T}}$ are the azimuth and elevation of the main beam direction of the i-th receive and transmit beam pair. Note that should other context information become available, it can be appended to the feature vector in (4). Assuming that $\hat{z}(\cdot)$ is already learned, Fig. 3 shows how the beam recommendation works. In the rest of the section, we describe how to learn $\hat{z}$.

We start with the description of our model for the scoring function $\hat{z}(\cdot)$. Because our feature vector is short compared to typical learning settings, we use a kernel based model which can provide higher model expressibility. Specifically, the scoring function is modeled as

$$\hat{z}(\boldsymbol{q}) = \sum_{\ell=1}^{N} \sum_{i\in\mathcal{I}_\ell} w_{i\ell} k(\boldsymbol{q}_{i\ell}, \boldsymbol{q}), \tag{5}$$

where $\mathcal{I}_\ell$ denotes the set of beam pairs measured at location $\boldsymbol{x}_\ell$ in the training set, $k(\cdot)$ is the kernel function, and $w_{i\ell}$ are the parameters to be learned. For notational convenience, we introduce an index mapping function $u(\cdot)$ that maps (i, ℓ) uniquely to the set $\{1, 2, ..., N_{\mathrm{tot}}\}$, where $N_{\mathrm{tot}} = \sum_{\ell=1}^{N} |\mathcal{I}_\ell|$ is the total number of beam measurements in the database. Now let $\alpha_{u(i,\ell)} = w_{i\ell}$ and $\tilde{\boldsymbol{q}}_{u(i,\ell)} = \boldsymbol{q}_{i\ell}$, we can rewrite $\hat{z}(\boldsymbol{q})$ as

$$\hat{z}(\boldsymbol{q}) = \sum_{n=1}^{N_{\mathrm{tot}}} \alpha_n k(\tilde{\boldsymbol{q}}_n, \boldsymbol{q}). \tag{6}$$

We denote $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_{N_{\mathrm{tot}}}]^{\mathrm{T}}$. In this paper, we use a modified radial basis function (RBF) as the kernel defined as

$$k(\boldsymbol{q}_{i\ell}, \boldsymbol{q}) = \exp\left(-\frac{d^2_{\mathrm{gc}}(\boldsymbol{\Theta}^{\mathrm{rx}}_i, \boldsymbol{\Theta}^{\mathrm{rx}})}{\sigma^2_{\mathrm{rx}}}\right) \exp\left(-\frac{d^2_{\mathrm{gc}}(\boldsymbol{\Theta}^{\mathrm{tx}}_i, \boldsymbol{\Theta}^{\mathrm{tx}})}{\sigma^2_{\mathrm{tx}}}\right) \exp\left(-\frac{\|\boldsymbol{x}_\ell - \boldsymbol{x}\|^2}{\sigma^2_{\mathrm{x}}}\right), \tag{7}$$

where $\sigma_{\mathrm{rx}}, \sigma_{\mathrm{tx}}$, and σ_{x} are the kernel parameters, and $d_{\mathrm{gc}}(\cdot)$ denotes the great circle distance on a unit sphere. We use $d_{\mathrm{gc}}(\cdot)$ because it properly handles the cyclic property of angles. Note that as in any learning methods, the hyperparameters such as the kernel parameters will need to be tuned. The intuition for selecting this kernel is that each component of the feature vector is a different type of physical feature, and thus they should be scaled by different kernel parameters.

We now describe the objective function that will be used for learning the scoring function. Inspired by the Discounted Cumulative Gain (DCG) [12], a popular metric for evaluating ranked lists, we measure the quality of the predicted ranked list at $\boldsymbol{x}_\ell$ location by

$$G_\ell(\boldsymbol{\alpha}) = \frac{1}{|\mathcal{I}_\ell|} \sum_{i\in\mathcal{I}_\ell} \sum_{j\in\mathcal{I}_\ell} S(\delta_{j\ell}) \mathbf{1}[R_{j\ell}(\boldsymbol{\alpha}) \leq R_{i\ell}(\boldsymbol{\alpha})], \tag{8}$$

where $R_{i\ell}(\boldsymbol{\alpha})$ is the predicted rank of the i-th beam pair, $\delta_{j\ell}$ is some measure of goodness of the j-th beam pair at this location, $S(\cdot)$ is some function to transform the raw goodness metric. The inner sum in (8) can be thought of as the overall goodness of the beam pairs with predicted rank from 1 to $R_{i\ell}(\boldsymbol{\alpha})$. Following the concept of power loss introduced in [3], we define $\delta_{j\ell}$ as

$$\delta_{j\ell} = \frac{\gamma_{j\ell}}{\gamma_{\max,\ell}}, \tag{9}$$

where $\gamma_{\max,\ell}$ and $\gamma_{j\ell}$ are the channel strength (linear scale) of the optimal beam pair and the j-th beam pair, respectively, at position $\boldsymbol{x}_\ell$. With this definition, $\delta_{j\ell}$ is close to 1 if the j-th beam pair is "good" and close to 0 if it is "bad." Note that $\delta_{j\ell}$ is not a function of $\boldsymbol{\alpha}$ and is obtained from the training data. The goodness of fit evaluating using the metric in (8) on the training data is the sum over all the locations in the training data, i.e.,

$$G_{\mathrm{tot}}(\boldsymbol{\alpha}) = \sum_{\ell=1}^{N} G_\ell(\boldsymbol{\alpha}). \tag{10}$$

The model parameters $\boldsymbol{\alpha}$ can be learned by maximizing $G_{\mathrm{tot}}(\boldsymbol{\alpha})$ over $\boldsymbol{\alpha}$. To prevent overfitting, we introduce the 2-norm regularization on $\boldsymbol{\alpha}$ and obtain

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^{N_{\mathrm{tot}}}} G_{\mathrm{tot}}(\boldsymbol{\alpha}) - \frac{\lambda}{2}\|\boldsymbol{\alpha}\|^2, \tag{11}$$

where λ is the regularization parameter.

The next step is to solve (11). Since $G_{\mathrm{tot}}(\boldsymbol{\alpha})$ is a sum over $G_\ell(\boldsymbol{\alpha})$, we can apply stochastic gradient descent (SGD) over the location index. We choose SGD because it is a computationally tractable algorithm [13]. Denoting η_t the learning rate at update round t, the algorithm is shown in Algorithm 1. To use Algorithm 1, we need to compute the gradient of $G_\ell(\boldsymbol{\alpha})$ with respect to $\boldsymbol{\alpha}$. Unfortunately, the indicator function in (8) is not continuous and some relaxation is required. Following [14], we approximate the indicator function by

$$\mathbf{1}[R_{j\ell}(\boldsymbol{\alpha}) \leq R_{i\ell}(\boldsymbol{\alpha})] \simeq g(\hat{z}(\boldsymbol{q}_{j\ell}) - \hat{z}(\boldsymbol{q}_{i\ell})), \tag{12}$$

where $g(t) = \frac{1}{1+e^{-t}}$ is the logistic function. With this relaxation, the gradient $\nabla_\alpha G_\ell(\boldsymbol{\alpha})$ can be computed as

$$\nabla_\alpha G_\ell(\boldsymbol{\alpha}) \simeq \frac{1}{|\mathcal{I}_\ell|} \sum_{i\in\mathcal{I}_\ell} \sum_{j\in\mathcal{I}_\ell} S(\delta_{j\ell}) g'(\hat{z}(\boldsymbol{q}_{j\ell}) - \hat{z}(\boldsymbol{q}_{i\ell})) \times (\boldsymbol{k}_{j\ell} - \boldsymbol{k}_{i\ell}), \tag{13}$$

where $g'(t) = e^{-t}/(1+e^{-t})^2$ and

$$\boldsymbol{k}_{i\ell} = \left[k(\tilde{\boldsymbol{q}}_1, \boldsymbol{q}_{i\ell}) \quad k(\tilde{\boldsymbol{q}}_2, \boldsymbol{q}_{i\ell}) \quad \ldots \quad k(\tilde{\boldsymbol{q}}_{N_{\mathrm{tot}}}, \boldsymbol{q}_{i\ell})\right]^{\mathrm{T}}.$$

Algorithm 1 SGD on $G_{\mathrm{tot}}(\boldsymbol{\alpha})$

```
1: while not converge or maximum number of iterations not yet reached do
2:    for each randomly selected training located at index ℓ do
3:       α^{t+1} ← α^t + η_t(∇_α G_ℓ(α^t) − λ‖α^t‖)
4:       t ← t + 1
5:    end for
6: end while
```

V. Numerical Results

This section provides numerical evaluations of the proposed LtR beam pair selection and compares it with the baseline method from [3]. We perform a five-fold cross validation on the 500 channel samples. This means we divide the data into five subsets of size 100, and we select one of the subsets as the test set and use the other four subsets as the training set. This is repeated for five times, each time a different subset is selected as the test set. Each channel sample corresponds to an independent placement of the vehicles in the environment. This also includes the location of the communicating car which is drawn uniformly from the range 27.5 m to 32.5 m from the RSU. The baseline method treats this 5 m range as a location bin, while in the proposed LtR approach actual position is included in the feature vector when computing the prediction as described in Section IV-B. 16×16 UPAs are assumed at both the car and the RSU. The codebook used has 271 beams.

We now present the performance results. We evaluate the method using the misalignment probability. The figures show the misalignment probability as a function of the number of beam pairs trained N_{b}. All the hyperparameters, including the SGD and kernel parameters, are tuned manually. We use a constant learning rate $\eta = 0.01$ and regularization parameter $\lambda = 0.001$, and SGD is run for 20 epochs. The parameters used for our modified RBF kernels are $\sigma_{\mathrm{x}} = 2$ and $\sigma_{\mathrm{rx}} = \sigma_{\mathrm{tx}} = 0.1$. Fig. 4 shows the results when using the modified RBF kernels with different choices of $S(\cdot)$. The choice of $S(\cdot)$ is critical, and here $S(\delta) = \delta^3$ provides the best result. The common choice used in the recommendation context is exponential function [12]. In our case, $S(\delta) = \delta^3$ is the best. This might be because our raw scores $\delta_{j\ell}$ are real numbers in $[0, 1]$ while typical ratings are integers from 1 to 5. Comparing with the baseline, the LtR approach is comparable for N_{b} up to around 34 and outperforms the baseline for larger N_{b}. In fact, there is no misalignment after $N_{\mathrm{b}} = 36$, while the baseline approach struggles.

Fig. 5 shows the results when using different kernels. In all cases, $S(\delta) = \delta^3$ is used. We compare our modified

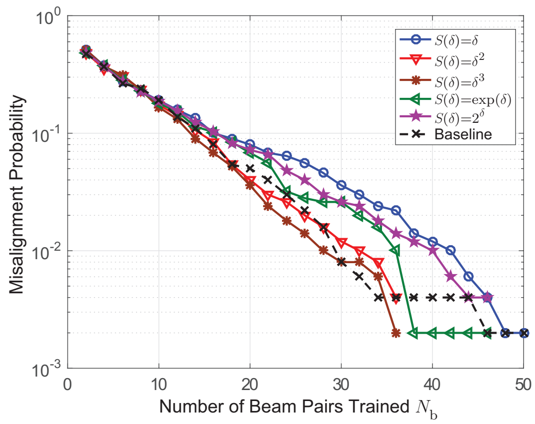

Fig. 4. Misalignment probability with different $S(\cdot)$.

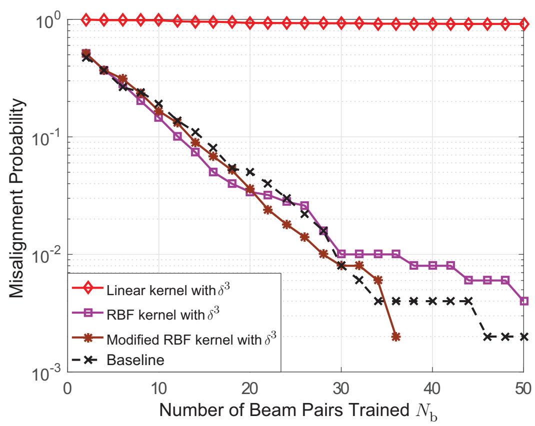

Fig. 5. Misalignment probability with different kernel functions.

RBF kernel and the popular choices of linear and RBF kernels. We perform feature scaling by dividing each component of the feature vector by the maximum possible values that the component can take. Note that feature scaling is not necessary for the modified RBF kernel because it can be absorbed into the kernel parameters. There is no parameter for the linear kernel and there is one length scale parameter for the RBF kernels. After manual tuning, this parameter is set to 0.01. We can see that the linear kernel does not work at all. This is due to the nonlinearity of the problem. The RBF kernel performs slightly better than the modified RBF kernel at small N_b but becomes worse at larger N_b. Overall the proposed modified RBF kernel performs better.

Finally, we note the reduction in training overhead. For a full exhaustive search, $271^2 = 73441$ beam pairs need to be measured while the proposed method trains less than 40 beam pairs. Our method also provides the flexibility to tradeoff beam alignment accuracy for latency by using a smaller N_b [3]. The training overhead follows similar trends as our prior work in [3] but can reduce the number of beam pairs trained by up to 20% depending on the target misalignment probability. We refer to [3] for a comparison in the mobility context with a hierarchical search used in IEEE 802.11ad.

VI. Conclusions

This paper proposes a LtR method for ranking desirable pointing directions. It is shown that leveraging position and past beam measurements, the beam configuration overhead can be reduced. The method outperforms the baseline method that uses probabilistic models which require context binning. Because of the context binning, the number of models increases exponentially with the number of contexts while the LtR approach only needs to increase the length of feature vectors. This provides a promising framework to incorporate other contexts beyond position information in future research.

Acknowledgment

This work is 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.

References

[1] The 5G Infrastructure Public Private Partnership (5GPPP), "5G automotive vision," Oct. 2015.

[2] J. Choi *et al.*, "Millimeter-wave vehicular communication to support massive automotive sensing," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 160–167, Dec. 2016.

[3] V. Va *et al.*, "Inverse fingerprinting for millimeter wave V2I beam alignment," https://arxiv.org/abs/1705.05942.

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

[5] T. Nitsche *et al.*, "Steering with eyes closed: Mm-wave beam steering without in-band measurement," in *Proc. IEEE INFOCOM*, vol. 26, 2015, pp. 2416–2424.

[6] N. Gonzalez-Prelcic *et al.*, "Radar aided beam alignment in mmwave V2I communications supporting antenna diversity," in *Proc. 2016 ITA Workshop*, Feb. 2016.

[7] V. Va *et al.*, "Beam design for beam switching based millimeter wave vehicle-to-infrastructure communications," in *Proc. IEEE ICC*, May 2016, pp. 1–6.

[8] W. B. Abbas and M. Zorzi, "Context information based initial cell search for millimeter wave 5G cellular networks," in *Proc. EuCNC*, June 2016, pp. 111–116.

[9] A. Capone *et al.*, "Obstacle avoidance cell discovery using mm-waves directive antennas in 5G networks," in *Proc. IEEE PIMRC*, Aug. 2015, pp. 2349–2353.

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

[11] Remcom, "Wireless InSite," http://www.remcom.com.

[12] T.-Y. Liu, "Learning to rank for information retrieval," *Found. Trends Inf. Retr.*, vol. 3, no. 3, pp. 225–331, Mar. 2009.

[13] T. J. Hastie, R. J. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning : Data Mining, Inference, and Prediction*, ser. Springer series in statistics. New York: Springer, 2009.

[14] Y Shi *et al.*, "GAPfm: optimal top-n recommendations for graded relevance domains." in *CIKM*. ACM, 2013, pp. 2261–2266.