

# MACHINE LEARNING FOR RELIABLE MMWAVE SYSTEMS: BLOCKAGE PREDICTION AND PROACTIVE HANDOFF

Ahmed Alkhateeb<sup>†</sup>, Iz Beltagy<sup>‡</sup>, and Sam Alex<sup>\*</sup>

<sup>†</sup> Arizona State University, Email: alkhateeb@asu.edu

<sup>‡</sup> Allen Institute for Artificial Intelligence, Email: beltagy@allenai.org

<sup>\*</sup> Facebook, Email: sampalex@fb.com

## ABSTRACT

The sensitivity of millimeter wave (mmWave) signals to blockages is a fundamental challenge for mobile mmWave communication systems. The sudden blockage of the line-of-sight (LOS) link between the base station and the mobile user normally leads to disconnecting the communication session, which highly impacts the system reliability. Further, reconnecting the user to another LOS base station incurs high beam training overhead and critical latency problem. In this paper, we leverage machine learning tools and propose a novel solution for these reliability and latency challenges in mmWave MIMO systems. In the developed solution, the base stations learn how to predict that a certain link will experience blockage in the next few time frames using their observations of adopted beamforming vectors. This allows the serving base station to proactively hand-over the user to another base station with highly probable LOS link. Simulation results show that the developed deep learning based strategy successfully predicts blockage/hand-off in close to 95% of the times. This reduces the probability of communication session disconnection, which ensures high reliability and low latency in mobile mmWave systems.

**Index Terms**— Millimeter wave, machine learning, beamforming, blockages, hand-off.

## 1. INTRODUCTION

Reliability and latency are two main challenges for millimeter wave (mmWave) wireless systems [1–3]: (i) The high sensitivity of mmWave signal propagation to blockages and the large signal-to-noise ratio gap between LOS and non-LOS links greatly affect the link reliability, and (ii) the frequent search for new base stations (BSs) after link disconnections causes critical latency overhead [4]. This paper leverages machine learning tools to efficiently address these challenges in mobile mmWave systems.

The coordination among multiple BSs to serve the mobile user has been the main approach for enhancing the reliability of mmWave communication links [1–3]. In [1], extensive measurements were done for coordinated multi-point transmission at 73 GHz, and showed that simultaneously serving the user by a number of BSs noticeably improves the network coverage. This coverage performance gain was also confirmed by [2] in heterogeneous mmWave cellular networks using stochastic geometry tools. To overcome the large training overhead and increase the effective achievable rate in coordinated transmissions, especially for highly-mobile applications, [3] proposed to use machine learning tools to predict the beamforming directions at the coordinating BSs from low-overhead features. Despite the interesting coverage gains of

coordinated transmission shown in [1–3], BSs coordination is associated with high cooperation overhead and difficult synchronization challenges.

In this paper, we develop a novel solution that enhances mmWave system reliability in high-mobile applications without requiring the high cooperation overhead of coordinated transmission. In our strategy, the serving BS uses the sequence of beams that it used to serve a mobile user over the past period of time to predict if a hand-off/blockage is going to happen in the next few moments. This allows that user and its serving BS to pro-actively hand-over the communication session to the next BS, which prevents sudden link disconnections due to blockage, improves the system reliability, and reduces the latency overhead. To do that, we develop a machine learning model based on gated recurrent neural networks that are best suited for dealing with variable-length sequences. Simulation results showed that the proposed solution predicts blockages/hand-off with almost 95% success probability and significantly improves the reliability of mmWave large antenna array systems.

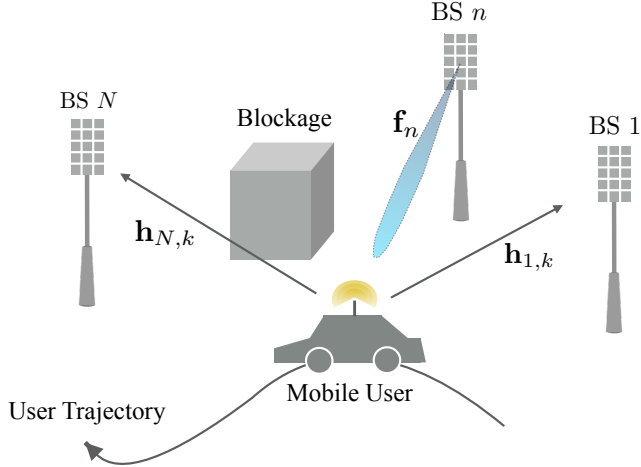
**Notation:** We use the following notation:  $\mathbf{A}$  is a matrix,  $\mathbf{a}$  is a vector,  $a$  is a scalar, and  $\mathcal{A}$  is a set.  $\mathbf{A}^T$ ,  $\mathbf{A}^*$  are the transpose and Hermitian (conjugate transpose) of  $\mathbf{A}$ .  $[\mathbf{a}]_n$  is the  $n$ th entry of  $\mathbf{a}$ .  $\mathbf{A} \circ \mathbf{B}$  is Hadamard product of  $\mathbf{A}$  and  $\mathbf{B}$ .  $\mathcal{N}(\mathbf{m}, \mathbf{R})$  is a complex Gaussian random vector with mean  $\mathbf{m}$  and covariance  $\mathbf{R}$ .

## 2. SYSTEM AND CHANNEL MODELS

In this section, we describe the adopted mmWave system and channel models. Consider the communication setup in Fig. 1, where a mobile user is moving in a trajectory. At every step in this trajectory, the mobile user gets connected to one out of  $N$  candidate base stations (BSs). For simplicity, we assume that the mobile user has a single antenna while the BS is equipped with  $M$  antennas. Extending the results of this paper to the case of multi-antenna users is straight-forward. Let  $\mathbf{h}_{n,k}$  denote the  $M \times 1$  uplink channel vector from the user to the  $n$ th BS at the  $k$ th subcarrier. If the user is connected to the  $n$ th BS, this BS applies a beamforming vector  $\mathbf{f}_n$  to serve this user. In the downlink transmission, the received signal at the mobile user on the  $k$ th subcarrier can then be expressed as

$$y_k = \mathbf{h}_{n,k}^* \mathbf{f}_n s + v, \quad (1)$$

where data symbol  $s \in \mathbb{C}$  satisfies  $\mathbb{E}[|s|^2] = P$ , with  $P$  the total transmit power, and  $v \sim \mathcal{N}_{\mathbb{C}}(0, \sigma^2)$  is the receive noise at the mobile user. Due to the high cost and power consumption of mixed-signal components in mmWave large antenna array systems, beamforming processing is normally done in the analog domain using networks of phase shifters [5]. The constraints on these phase shifters



**Fig. 1.** The system model considers one user moving in a trajectory, and is served by one out of  $N$  candidate BSs at every step in the trajectory.

limit the beamforming vectors to be selected from quantized codebooks. Therefore, we assume that the BS beamforming vector  $\mathbf{f}_n$  is selected from a quantized codebook  $\mathcal{F}$  with size/cardinality  $|\mathcal{F}| = M_{CB}$ . The codewords of this codebook are denoted as  $\mathbf{g}_m, m = 1, 2, \dots, M_{CB}$ . Further, we assume that the beamforming vector  $\mathbf{f}_n$  is selected from the codebook  $\mathcal{F}$  to maximize the received signal power, i.e., according to the criterion

$$\mathbf{f}_n = \arg \max_{\mathbf{g}_m \in \mathcal{F}} \sum_k |\mathbf{h}_{n,k}^* \mathbf{g}_m|^2. \quad (2)$$

We adopt a wideband geometric mmWave channel model [4] with  $L$  clusters. Each cluster  $\ell, \ell = 1, \dots, L$  is assumed to contribute with one ray that has a time delay  $\tau_\ell \in \mathbb{R}$ , and azimuth/elevation angles of arrival (AoA)  $\theta_\ell, \phi_\ell$ . Further, let  $\rho_n$  denote the path-loss between the user and the  $n$ -th BS, and  $p_{rc}(\tau)$  represents a pulse shaping function for  $T_S$ -spaced signaling evaluated at  $\tau$  seconds. With this model, the delay-d channel between the user and the  $n$ th BS follows

$$\mathbf{h}_{n,d} = \sqrt{\frac{M}{\rho_n}} \sum_{\ell=1}^L \alpha_\ell p(dT_S - \tau_\ell) \mathbf{a}_n(\theta_\ell, \phi_\ell), \quad (3)$$

where  $\mathbf{a}_n(\theta_\ell, \phi_\ell)$  is the array response vector of the  $n$ th BS at the AoAs  $\theta_\ell, \phi_\ell$ . Given the delay-d channel in (3), the frequency domain channel vector at subcarrier  $k, \mathbf{h}_{n,k}$ , can be written as

$$\mathbf{h}_{n,k} = \sum_{d=0}^{D-1} \mathbf{h}_{n,d} e^{-j \frac{2\pi k}{K} d}. \quad (4)$$

Considering a block-fading channel model,  $\{\mathbf{h}_{n,k}\}_{k=1}^K$  are assumed to stay constant over the channel coherence time, denoted  $T_C$  [6].

### 3. PROBLEM DEFINITION AND FORMULATION

Maintaining good link reliability is a key challenge for mmWave communication systems, especially with mobility. This is mainly due to the high sensitivity of mmWave signals to blockages, which can frequently cause link disconnections. Further, when the link

to the current BS is blocked, the mobile user incurs critical latency overhead to get connected to another BS. To overcome these challenges, can we predict that a link blockage is going to happen in the next few moments? Successful blockage prediction can be very helpful for mmWave system operation as it allows for proactive hand-off to the next BS. This proactive hand-off enhances the system reliability by ensuring session continuity and avoids the latency overhead that results from link disconnection. In this section, we formulate the mmWave blockage prediction and proactive hand-off problem that we tackle in the next section.

**Beam sequence and hand-off status:** To formulate the problem, we first define two important quantities, namely the beam sequence and the hand-off status. Due to the user mobility, the current/serving BS  $n$  needs to frequently update its beamforming vector  $\mathbf{f}_n \in \mathcal{F}$ . The frequency of updating the beams depends on a number of parameters including the user speed and the beam width. A good approximation for the period every which the BS needs to update its beam is the beam coherence time,  $T_B$ , which is defined as [6]

$$T_B = \frac{D}{v_s \sin(\alpha)} \frac{\Theta_n}{2}, \quad (5)$$

where  $v_s$  denotes the user speed,  $D$  is the distance between the user and the BS,  $\alpha$  is the angle between the direction of travel and the direction of the main scatterer/reflector (or the BS in the case of LOS), and  $\Theta_n$  defines the beam-width of the beams used by BS  $n$ . Now, assuming that the current/serving BS  $n$  updates its beamforming vector every beam-coherence time and calling it a time step, we define  $\mathbf{f}_n^{(t)}$  as the beamforming vector selected by the  $n$ th BS to serve the mobile user in the  $t$ th time step, with  $t = 1$  representing the first time step after the handing-over to the current BS. With this, we define the beam sequence of BS  $n$  until time step  $t$ , denoted  $\mathcal{B}_t$  as

$$\mathcal{B}_t = \{\mathbf{f}_n^{(1)}, \mathbf{f}_n^{(2)}, \dots, \mathbf{f}_n^{(t)}\}. \quad (6)$$

Further, we define  $s_t \in \{1, 2, \dots, N\}$ , as the hand-off status at the  $t$ th time step, with  $s_t = n$  indicating the user will stay connected to the current BS  $n$  in the next time step, and  $s_t \neq n$  indicating that the mobile user will hand-off to another BS in the  $t + 1$  time step. It is important to note here that predicting a hand-off in the next time step is more general than predicting a blockage, as the hand-off can happen due to a sudden blockage or a better SNR. Therefore, we will generally adopt the hand-off prediction that implicitly include link blockage prediction.

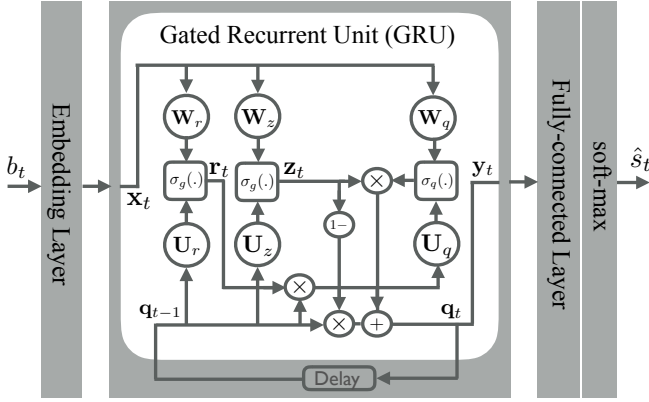
In this paper, we will focus on stationary blockages, which leaves the user mobility as the main factor of the time-varying behavior in the channel. To complete the formulation, we leverage the note that the beamforming vector that maximizes the received signal power, as defined in (2), heavily relies on the user location and the surrounding environment geometry (including blockages) [3]. **Therefore, we formulate the problem as a prediction of the hand-off status at time  $t + 1$ , i.e.,  $\hat{s}_t$ , given the beam-sequence at time  $t, \mathcal{B}_t$ .** Formally, the objective of this paper is to maximize the probability of successful blockage/hand-off prediction defined as

$$\mathbb{P}[\hat{s}_t = s_t | \mathcal{B}_t]. \quad (7)$$

In this next section, we leverage machine learning tools to address this problem.

### 4. DEEP LEARNING BASED PROACTIVE HAND-OFF

In this section, we explain our proposed solution that uses deep learning tools, and more specifically recurrent neural networks, to efficiently predict hand-off/blockages. First, we highlight the key idea



**Fig. 2.** The proposed deep learning model that leverages recurrent neural networks to predict the hand-off BS in the next time step,  $\hat{s}_t$ , given the past beam sequence  $\mathcal{B}_t$

and system operation before delving into the exact machine learning modeling in Section 4.2.

#### 4.1. Main Idea and System Operation

In this subsection, we briefly describe the key idea of the proposed solution as well as the learning/communications system operation. We model the problem of predicting hand-off/blockage as a sequence labeling problem. In summary, given the sequence of previous beams  $\mathcal{B}_t$ , the serving BS predict the most likely base station that the user will connect with in the next time step. If the predicted base station is different from the current one, that indicates that a proactive hand-off needs to happen. As we mentioned in Section 3, by adopting the problem of predicting the hand-off, our system will also predict link blockages if they are going to happen in the next time step as they will require a hand-off.

**System operation:** The proposed learning/communication system operates in two phases. In the first phase (learning), the mmWave communication system operates as normal: At every beam coherence time, the current BS will update its beamforming vector that serves the user. If the link is blocked, the user will follow the initial access process to hand-off to a new BS. During this process, the serving BS will feed the beam sequence  $\mathcal{B}_t$  and the hand-off status  $s_t$  at every time step (beam coherence time) to its machine learning model that will use it for training. It is important to note here that these beam sequences have different lengths depending on the speed of the user, its trajectory, the time period it is spent connected to this BS, etc. As will be explained shortly, We designed our deep learning model to carefully handle this variable sequence length challenge.

After the machine learning model is well-trained, the serving BS will leverage it to predict if the link to the user will face a blockage/hand-off in the next time-step. If a hand-off is predicted, the user and its serving BS will pro-actively initiate the hand-off process to the next BS to ensure session continuity and avoid latency problems.

#### 4.2. Machine Learning Model

Next, we describe the key elements of the proposed machine learning model which is based on recurrent neural networks.

**Input representation:** At every time step  $t$ , the input to our deep learning model is the  $t$ th beam index in the beam sequence  $\mathcal{B}_t$ . Let  $b_t \in \{1, 2, \dots, M_{CB}\}$  denote the index of the  $t$ th beam index, then our model starts with an embedding layer that maps every beam index  $b_t$  to a vector  $\mathbf{x}_t$ .

$$\mathbf{x}_t = \text{embd}(b_t) \quad (8)$$

where  $\text{embd}$  is a lookup table we learn during the training.

**Sequence processing:** The central component of our model is a Gated Recurrent Unit (GRU) [7] — a form of neural networks that is best suited for processing variable length sequences. GRU is a recursive network that runs at every time step of the input, and it maintains a hidden state  $\mathbf{q}_t$  which is a function of the previous state and the current input. In addition, it has a special gating mechanism that helps with long sequences. More formally, GRU is implemented as depicted in Fig. 2, and is described by the following model equations

$$\mathbf{r}_t = \sigma_g(\mathbf{W}_r \mathbf{x}_t + \mathbf{U}_r \mathbf{q}_{t-1} + \mathbf{c}_r) \quad (9)$$

$$\mathbf{z}_t = \sigma_g(\mathbf{W}_z \mathbf{x}_t + \mathbf{U}_z \mathbf{q}_{t-1} + \mathbf{c}_z) \quad (10)$$

$$\mathbf{q}_t = (1 - \mathbf{z}_t) \circ \mathbf{q}_{t-1} + \mathbf{z}_t \circ \sigma_q(\mathbf{W}_q \mathbf{x}_t + \mathbf{U}_q (\mathbf{r}_t \circ \mathbf{q}_{t-1}) + \mathbf{c}_q), \quad (11)$$

where  $\mathbf{x}_t$  is the input vector,  $\mathbf{q}_t$  is the hidden state,  $\mathbf{r}_t$  and  $\mathbf{z}_t$  are "gates" that help the model learn from long distance dependencies if needed. The weight matrices  $\mathbf{W}_r$ ,  $\mathbf{W}_z$ ,  $\mathbf{U}_r$ ,  $\mathbf{U}_z$  and the bias vectors  $\mathbf{c}_r$ ,  $\mathbf{c}_z$ ,  $\mathbf{c}_q$  are model parameters that are learned during the machine learning training phase. Finally,  $\sigma_g$  and  $\sigma_q$  are nonlinear activation functions, the first is sigmoid and the second is tanh. Using non linear activation functions allows the model to represent complex non-linear functions.

**Output:** At every time step  $t$ , the model has a hidden state  $\mathbf{q}_t$  which encompasses what the model learned about the sequence of beams until time step  $t$ , and the next step is to use  $\mathbf{q}_t$  to predict the most likely base station  $\hat{s}_t$  for the next time step. For this, we use a fully connected layer with output size  $N$  equals number of possible base stations, followed by a softmax activation that normalizes the outputs into probabilities (they sum up to 1). The predicted hand-off BS (the final output of the model) can then be expressed as

$$\hat{s}_t = \arg \max_{n \in \{1, 2, \dots, N\}} \text{softmax}(\mathbf{W}_f \mathbf{q}_t + \mathbf{c}_f)_n \quad (12)$$

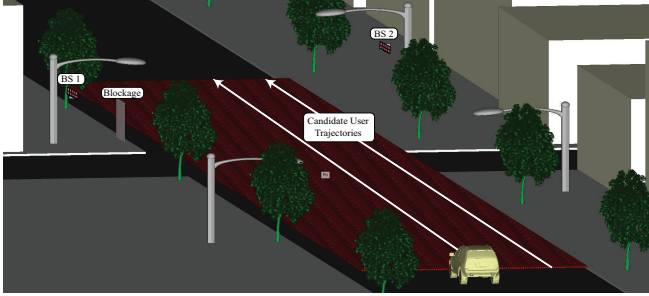
where  $\mathbf{W}_f$  and  $\mathbf{c}_f$  are the weights and biases of the fully-connected layer. The  $\text{softmax}(\cdot)_n$  function takes a vector of length  $N$  and outputs the probability that the  $n$ th BS is going to be the BS of the next time step  $t + 1$ . The  $\text{softmax}(\cdot)_n$  function is defined as follows:

$$\text{softmax}(\mathbf{a})_n = \frac{e^{[\mathbf{a}]_n}}{\sum_{d=1}^N e^{[\mathbf{a}]_d}}. \quad (13)$$

**Training:** The training objective in our supervised learning model is minimizing the cross entropy loss between our model predictions  $\hat{s}_t$  and the actual base station of the following time step  $s_t$ . We compute this loss at every time step  $t$  then sum over all time steps. The cross entropy loss at every time step  $t$  is computed as follows:

$$\text{loss}(\hat{s}_t, s_t) = - \sum_i [\mathbf{p}]_{i,t} \log [\hat{\mathbf{p}}]_{i,t}. \quad (14)$$

where the reference prediction vector  $\mathbf{p}$  has 1 in the entry corresponding to the index of the correct BS in  $s_t$ , and zero otherwise.



**Fig. 3.** This figure illustrates the considered simulation setup where two candidate BSs, each has ULA, serve one vehicle moving in a street.

Further, the model prediction vector  $\hat{\mathbf{p}}$  has the  $d$ th entry equals to  $\text{softmax}(\mathbf{a})_d, \forall d$ .

The final goal of training the model is to find the parameters  $\text{embd}, \mathbf{W}_r, \mathbf{W}_z, \mathbf{W}_h, \mathbf{U}_r, \mathbf{U}_z, \mathbf{U}_h, \mathbf{W}_f, \mathbf{c}_r, \mathbf{c}_z, \mathbf{c}_h, \mathbf{c}_f$  that minimize this loss for all training instances.

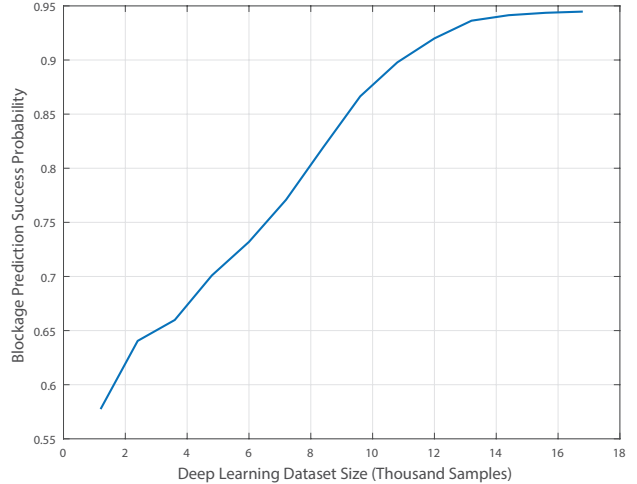
## 5. SIMULATION RESULTS

In this section, we evaluate the performance of the proposed deep-learning based proactive hand-off solution.

**Simulation setup:** We adopt the mmWave system and channel models in Section 2, with two candidate BSs to serve one vehicle/mobile user moving in a street, as depicted in Fig. 3. To generate realistic data for the channel parameters (AoAs/delay/etc.), we use the commercial ray-tracing simulator, Wireless InSite [8], which is widely used in mmWave research [6, 9], and is verified with channel measurements [9]. Each BS is installed on one lamp post at height 4 m, and employs a 32-element uniform linear array (ULA) facing the street. The mobile user is moving in straight-line trajectories in the street, that can be any where of the street width, and with maximum trajectory length of 160 m. The trajectory starting point is randomly selected from the first 40m of the street, and the user speed is randomly selected from  $\{8, 16, 24, 32, 40\}$  km/hr. The BS selects its beamforming vector from a uniformly quantized beamsteering codebook with an oversampling factor of 4, i.e. the codebook size is  $M_{CB} = 128$ . At every beam coherence time, the BS beamforming vector is updated to maximize the receive SNR at the user. During the uplink training, the MS is assumed to use 30dBm transmit power, and the noise variance corresponds to 1GHz system bandwidth. The system is assumed to operate at 60GHz carrier frequency.

We consider the deep learning model described in Section 4.2. The neural network model has an embedding that outputs vectors of length 20 to the GRU unit, with maximum sequence length of 454. Since we have only 2 BSs in our experiment, the fully-connected layer has only two outputs that go to the softmax function. We use the Adam optimizer [10]. In the deep learning experimental work, we used the *Keras* libraries [11] with a *TensorFlow* backend.

**Hand-off/Blockage prediction:** To evaluate the performance of the proposed deep-learning based proactive hand-off solution, Fig. 4 plots the blockage/hand-off successful prediction probability, defined in (7), versus the training size. Fig. 4 shows that with sufficient dataset size (larger than 12 thousand samples), the machine learning model successfully predicts the hand-off with more than 90% probability, given only the sequence of past beams  $\mathbf{B}_t$ . This illustrates the potential of the proposed solution in enhancing the



**Fig. 4.** This figure shows that the proposed deep-learning proactive hand-off solution successfully predicts blockages/hand-off with high probabilities when the machine learning model is well-trained.

reliability of next-generation mmWave systems.

## 6. CONCLUSION

In this paper, we proposed a novel solution for the reliability and latency problem in mobile mmWave systems. Our solution leveraged deep learning tools to efficiently predict blockage and the need for hand-off. This allows the mobile user to proactively hand-off to the next BS without disconnecting the session or suffering from high latency overhead due to sudden link disconnections. The simulation results showed that the developed proactive hand-off strategy can successfully predict blockage/hand-off with high probability when the machine learning model is trained with reasonable dataset sizes. In the future, it is interesting to extend the developed solution to multi-user systems and account for mobile blockages.

## 7. REFERENCES

- [1] G. R. MacCartney, T. S. Rappaport, and A. Ghosh, "Base station diversity propagation measurements at 73 GHz millimeter-wave for 5G coordinated multipoint (CoMP) analysis," in *2017 IEEE Globecom Workshops (GC Wkshps)*, Dec 2017, pp. 1–7.
- [2] D. Maamari, N. Devroye, and D. Tuninetti, "Coverage in mmwave cellular networks with base station co-operation," *IEEE Transactions on Wireless Communications*, vol. 15, no. 4, pp. 2981–2994, April 2016.
- [3] 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*, pp. 1–1, 2018.
- [4] R. W. Heath, N. Gonzalez-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, "An overview of signal processing techniques for millimeter wave MIMO systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 3, pp. 436–453, April 2016.
- [5] A. Alkhateeb, Jianhua Mo, N. Gonzalez-Prelcic, and R.W. Heath, "MIMO precoding and combining solutions for

- millimeter-wave systems,” *IEEE Communications Magazine*, vol. 52, no. 12, pp. 122–131, Dec. 2014.
- [6] V. Va, J. Choi, T. Shimizu, G. Bansal, and R. W. Heath, “Inverse multipath fingerprinting for millimeter wave v2i beam alignment,” *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2017.
- [7] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” in *Proc. of SSST*, 2014.
- [8] Remcom, “Wireless insite,” <http://www.remcom.com/wireless-insite>.
- [9] Q. Li, H. Shirani-Mehr, T. Balercia, A. Papathanassiou, G. Wu, S. Sun, M. K. Samimi, and T. S. Rappaport, “Validation of a geometry-based statistical mmwave channel model using ray-tracing simulation,” in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, May 2015, pp. 1–5.
- [10] Diederik P Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [11] Frdric Branchaud-Charron, Fariz Rahman, and Taehoon Lee, “Keras,” .