arXiv:2509.11725v1 [eess.SP] 15 Sep 2025

ATTENTION-ENHANCED LEARNING FOR SENSING-ASSISTED LONG-TERM BEAM TRACKING IN MMWAVE COMMUNICATIONS

Mengyuan Ma[*], *Nhan Thanh Nguyen*[*], *Nir Shlezinger*[†], *Yonina C. Eldar*[§], *and Markku Juntti*[*]

[*]Centre for Wireless Communications (CWC), University of Oulu, Finland
[†]School of ECE, Ben-Gurion University of the Negev, Beer-Sheva, Israel
[§]Faculty of Math and CS, Weizmann Institute of Science, Rehovot, Israel
{mengyuan.ma, nhan.nguyen, markku.juntti}@oulu.fi; nirshl@bgu.ac.il; yonina.eldar@weizmann.ac.il

ABSTRACT

Beam training and prediction in millimeter-wave communications are highly challenging due to fast time-varying channels and sensitivity to blockages and mobility. In this context, infrastructure-mounted cameras can capture rich environmental information that can facilitate beam tracking design. In this work, we develop an efficient attention-enhanced machine learning model for long-term beam tracking built upon convolutional neural networks and gated recurrent units to predict both current and future beams from past observed images. The integrated temporal attention mechanism substantially improves its predictive performance. Numerical results demonstrate that the proposed design achieves Top-5 beam prediction accuracies exceeding 90% across both current and six future time slots, significantly reducing overhead arising from sensing and processing for beam training. It further attains 97% of state-of-the-art performance with only 3% of the computational complexity.

***Index Terms*—** Long-term beam tracking, vision-aided communications, machine learning, mmWave communications.

1. INTRODUCTION

In millimeter wave (mmWave) communications, beam training requires significant overhead in terms of signaling, latency, and power consumption [1]. As a solution, various sensory information have recently been explored for beam prediction, including the user position captured by Global Navigation Satellite System (GNSS) [2, 3], the point cloud scanned by LiDAR [4, 5], radar-sensed data [6, 7], and vision information from cameras [1, 8–10]. Such approaches fall under the category of sensing-assisted communications, which is among the main use cases in Integrated Sensing and Communications (ISAC).

The joint exploitation of multiple sensing modalities for beam training have been investigated in [9, 11–17]. However, effectively extracting features of different sensing modalities and performing efficient feature fusion typically requires high complexity machine learning (ML) models, posing a challenge for practical implementation. For example, multiple Transformer architectures [18] were adopted in [11–13, 15]. Moreover, most existing works on sensing-aided beam management focus on predicting only the current beam based on the current and/or past sensory data. Such methods result in frequent inference at each time step causing high overhead for sensing and processing in terms of power consumption, latency, and signaling. Long-term beam prediction can alleviate this by jointly predicting multiple future time steps, but this idea has remained largely unexplored. Existing work [4, 19] has considered long-term beam prediction based on vision and LiDAR data, respectively, achieving encouraging results. However, LiDAR is costly for widespread usage while a large model is required in [19] to extract features from raw images.

To develop low-complexity and low-latency beam training, in this paper, we study efficient sensing-assisted long-term beam tracking designs by leveraging vision data to predict optimal beams for current and multiple future time slots. Specifically, given a predefined beamformer codebook, we formulate the beam tracking problem as an ML classification task, which is addressed using a sequence-to-sequence model. Distinct from [19], we adopt an efficient image preprocessing method followed by a dedicated convolutional neural network (CNN) to efficiently extract compact image feature. We further integrate an attention mechanism with gated recurrent unit (GRU) networks to capture the temporal dependencies of features across time slots, enabling robust long-term beam prediction. Numerical results on a real-world dataset demonstrate the superiority of the proposed design over the scheme in [19] in terms of achieving comparable beam prediction accuracies across both current and six future time slots while reducing computational complexity by more than 97%.

2. SYSTEM MODEL AND PROBLEM FORMULATION

System Model: We consider a downlink mmWave communications system, where the BS serves a single-antenna mobile user equipment (UE), as illustrated in Fig. 1. The BS is equipped with a uniform linear array and an RGB camera (visual data sensor). At time step t, the BS transmits symbol $s[t] \in \mathbb{C}$, with $\mathbb{E}\left[|s|^2\right] = 1$, to the UE. We assume a block fading channel between time slots. Let $\mathbf{v}[t]$ denote the beamforming vector at time step t. Then, the received signal $y[t]$

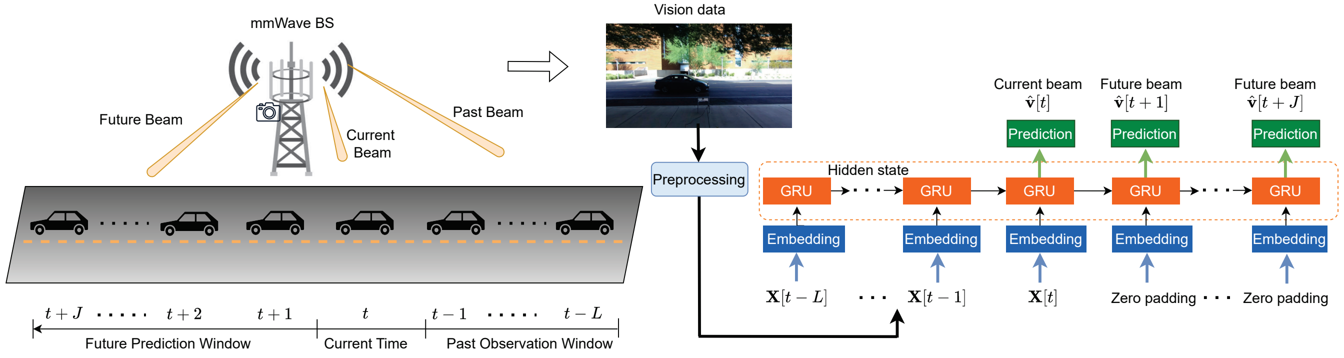


Fig. 1: Illustration of the considered system model. The BS senses the environment and the moving UE with an RGB camera. The sensory data are collected and cached for beam tracking using the designed ML model.

can be written as

$$y[t] = \mathbf{h}[t]^{\mathsf{H}}\mathbf{v}[t]s[t] + n[t], \tag{1}$$

where $\mathbf{h}[t]$ denotes the channel between the BS and the UE at time step t, and $n[t] \sim \mathcal{CN}(0, \sigma_{\mathrm{n}}^2)$ models the additive white Gaussian noise (AWGN) with noise power σ_{n}^2. The signal-to-noise ratio (SNR) at time slot t is $\mathrm{SNR}[t] = \frac{\left|\mathbf{h}[t]^{\mathsf{H}}\mathbf{v}[t]\right|^2}{\sigma_{\mathrm{n}}^2}$.

Problem Formulation: At the current time slot t, our goal is to determine the transmit beamforming vectors at the BS for current and J future time slots, i.e., $\{t, t+1, \dots, t+J\}$. Let $\mathcal{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_C\}$ and $\mathcal{I}_{\mathcal{V}} = \{1, \dots, C\}$ denote the beamforming codebook and its associated index set with $C \triangleq |\mathcal{V}|$. In the considered beam tracking problem, we aim to find $\mathbf{v}[\tau] \in \mathcal{V}, \forall \tau$, to maximize the overall spectral efficiency over these $J+1$ time slots, which is expressed as $R_J = \sum_{\tau=t}^{t+J} \log(1 + \mathrm{SNR}[\tau])$. For low SNR scenarios, we can formulate the beamforming problem as [1, 4, 20]

$$\underset{\mathbf{v}[\tau]\in\mathcal{V},\forall\tau}{\text{maximize}} \quad \sum_{\tau=t}^{t+J} \left|\mathbf{h}[\tau]^{\mathsf{H}}\mathbf{v}[\tau]\right|^2. \tag{2}$$

Let $\mathbf{b}^\star[t] = \left[b^\star[t], \dots, b^\star[t+J]\right]^{\mathsf{T}}$ be the vector of beam indices corresponding to the optimal solution of (2), i.e.,

$$\mathbf{b}^\star[t] = \underset{b[\tau]\in\mathcal{I}_{\mathcal{V}},\forall\tau}{\arg\max} \quad \sum_{\tau=t}^{t+J} \left|\mathbf{h}[\tau]^{\mathsf{H}}\mathbf{v}_{b[\tau]}\right|^2. \tag{3}$$

The solution to (3) can be obtained by decoupling it into $J+1$ subproblems with each solved via an exhaustive search over the C candidate beams. However, the complexity of such a method scales as $J \cdot C$, which incurs high latency, especially with the large codebooks used in massive MIMO. Moreover, this approach requires perfect channel state information (CSI) at not only the current time slot, but also the J future time slots, which is generally unavailable in practice.

In this work, we consider CSI-free beam tracking, where instead of aiming to recover $\mathbf{b}[t]$ based on knowledge of $\mathbf{h}[t]$, we utilize sensed visual data, denoted by $\mathbf{Z}[t]$. Accordingly, our aim is to design a CSI-free mapping from $\mathbf{Z}[t]$ into $\mathbf{b}[t]$, such that the results remains effective with respect to the CSI-based performance measure in (3). Unlike [1, 6, 8–17] which address problem (3) by decoupling it into $J+1$ subproblems, we propose an efficient learning framework that directly solves problem (3) for long-term beam tracking design, as will be elaborated below.

3. VISION-BASED LONG-TERM BEAM TRACKING

Learning Task: Let $\mathbf{Z}[t] \in \mathbb{R}^{3\times d_{\mathrm{H}}\times d_{\mathrm{W}}}$ denote the RGB image obtained at time slot t, where the dimension 3 corresponds to the number of RGB color channels, and d_{H} and d_{W} respectively represent the image height, and image width in pixels. Let $\mathcal{Z}[t]$ denote the sequence of sensory data, i.e., RGB images, from L past time slots to the current time t, given by $\mathcal{Z}[t] = \{\mathbf{Z}[t-L], \mathbf{Z}[t-L+1], \dots, \mathbf{Z}[t]\}$. We aim to develop an efficient ML model to solve (2), i.e., to predict the optimal beams (equivalently the optimal beam indices in$\mathcal{I}_{\mathcal{V}}$) for current time slot t and future J time slots $t+1, \dots, t+J$. Denote the data preprocessing operations by $\mathcal{X}[t] = g(\mathcal{Z}[t])$, mapping the input sequence to the ML model. Let $f(\mathcal{X}[t]; \Theta)$ denote the ML model with learnable parameters Θ. The ML model outputs the probabilities of all possible beams for beamforming at $J+1$ (current and future) time slots. Let $p_c[t+j]$ denote the probability of selecting the c-th beam in the codebook at time slot $t+j$, and define $\mathbf{p}[t+j] = [p_1[t+j], \dots, p_C[t+j]]^{\mathsf{T}} \in \mathbb{R}^C, j = 0, \dots, J$. The predicted beam index is obtained as

$$\hat{b}[\tau] = \underset{c\in\mathcal{I}_{\mathcal{V}}}{\arg\max}\ p_c[\tau],\ \tau = t, \dots, t+J. \tag{4}$$

The desired ML model for vision-aided beam tracking can be mathematically written as

$$f^\star(;\Theta^\star) = \underset{f(;\Theta)}{\arg\max} \quad \sum_{\tau=t}^{t+J} \mathbb{P}\{\hat{b}[\tau] = b^\star[\tau]\}, \tag{5}$$

where $\mathbb{P}\{\cdot\}$ denotes the probability. We note that J and L are hyperparameters, which are determined empirically.

ML Model Structure: As shown in Fig. 1, the preprocessed input is firstly fed into the embedding block to extract semantic low-dimensional feature representation. The GRU block receives the feature vectors and captures their temporal dependencies by hidden states, which are updated step by step to produce a context vector that summarizes the information of the entire input $\{\mathbf{X}[\tau]\}$. Given this context

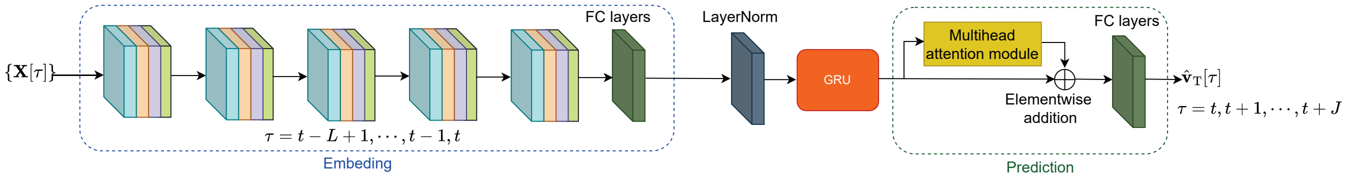

$\{\mathbf{X}[\tau]\}$
FC layers
$\tau = t-L+1, \cdots, t-1, t$
Embeding
LayerNorm
GRU
Multihead attention module
Elementwise addition
FC layers
$\hat{\mathbf{v}}_{\mathrm{T}}[\tau]$
$\tau = t, t+1, \cdots, t+J$
Prediction

Fig. 2: Illustration of the ML model structure.

vector and current data $\mathbf{X}[t]$, the current and future beams are predicted with the prediction block. We adopt the standard GRU structure and focus on the design of the embedding and prediction blocks, which are illustrated in Fig. 2. A pretrained ResNet for image feature extraction [11–15] would result in a high cost of computational complexity and memory usage. To improve efficiency, we design a dedicated five-layer CNN network to extract spatial features of the images in the input sequence. Each layer consists of four standard Pytorch modules, including Cov2d, BatchNorm2d, ReLU, and MaxPool2d. The output of the GRUs is further processed by a residual multihead attention (MHA) module, which can strengthen the model's ability to capture temporal relations among the input sequence.

The MHA aims to provide a self-attention mechanism that allows each feature in a sequence to attend to all others and simultaneously learn different contextual information components. For the mathematical foundations of MHA, we refer the reader to [18]. Using MHA self-attention after the GRU combines the strengths of both sequential and attention-based modeling. Specifically, while GRUs effectively capture temporal dependencies in a step-by-step manner, they focus mainly on local dependencies. In contrast, MHA self-attention allows the model to directly attend to all positions in the sequence, enhancing its ability to capture global features. Furthermore, multiple attention heads enable the extraction of diverse features from the GRU outputs, leading to richer and more informative representations that can improve the model's expressive ability and boost performance. The effectiveness of the MHA module will be justified using numerical experiments in Section 4.

Training of the ML Model: Let $\mathcal{D} = \{\{\mathcal{Z}[t], \mathbf{b}^\star[t]\}, t = 0, \ldots, T\}$ denote the set of data sequences from the source dataset, where T denotes the number of time slots over which vision data are collected, $\mathcal{Z}[t]$ and $\mathbf{b}^\star[t]$ are the ML model's input and label, respectively. The received signal $y[t]$ in (1) is leveraged to obtain $\mathbf{b}^\star[t]$, which in real-world mmWave communications is generally non-uniformly distributed among the C candidate beams. Such class imbalance among the datasets can lead to poor performance for the minority class. During the training, we use the focal loss [21], which is a modification of the standard cross-entropy loss designed to address the class imbalance problem.

Given a sample $\mathbf{Z}[\tau]$, the output of the ML model is the predicted probability vector $\mathbf{p}[\tau]$ for the C classes of beams. Let $\mathbf{z}[\tau] = [z_1[\tau], \ldots, z_C[\tau]]$ denote the vector of the output logits. The c-th element of $\mathbf{p}[\tau]$ is obtained as $p_c[\tau] = \sigma_c(\mathbf{z}[\tau])$, where $\sigma_c(\mathbf{z}[\tau]) = \frac{\exp(z_c[\tau])}{\sum_{k=1}^{C}\exp(z_k[\tau])}$ denotes the softmax function. The focal loss for a single sample $\mathbf{Z}[\tau]$ is given by

Algorithm 1: Learning Procedure for Problem (5).

```
   Input: Training and validation datasets $\mathcal{D}_{\mathrm{tr}}$, $\mathcal{D}_{\mathrm{evl}}$
   Output: Model parameters $\Theta$
1  Initialize $\Theta^\star$, $\Theta = \Theta^\star$, and learning rate.
2  for $e = 1, \ldots, E$ do
3      Randomly divide $\mathcal{D}_{\mathrm{tr}}$ into $N_{\mathrm{b}}$ batches $\{\mathcal{D}_{\mathrm{tr}}^{(n)}\}_{n=1}^{N_{\mathrm{b}}}$ with
         batch size $B$.
4      for $n = 1, \ldots, N_{\mathrm{b}}$ do
5          Perform data preprocessing $\mathcal{X}[t] = g(\mathcal{Z}[t])$ with
             $t \in \mathcal{T}^{(n)} = \{t_1^{(n)}, \ldots, t_B^{(n)}\}$
6          Feed the sequences $\mathcal{X}[t_q^{(n)}], q = 1, \ldots, B$ into the
             the ML model and compute
             $\mathbf{p}_q[t+j], j = 0, 1, \ldots, J, q = 1, \ldots, B$.
7          Compute average task loss over the batch in $J+1$
             time slots: $\mathcal{L}_{\mathrm{task}} = \frac{1}{B(J+1)}\sum_{t\in\mathcal{T}^{(n)}}\mathcal{L}_{\mathrm{task}}[t]$.
8          Update $\Theta$ with an optimizer
9      end
10     Compute validation loss based on $f(;\Theta)$ and $\mathcal{D}_{\mathrm{evl}}$.
11     Update the best model $\Theta^\star = \Theta$ if a lower validation
         loss is found.
12 end
13 Return $\Theta^\star$.
```

$$l_{\mathrm{focal}}[\tau] = -(1 - p_{b^\star}[\tau])^\gamma \log\left(p_{b^\star}[\tau]\right), \tag{6}$$

where $p_{b^\star}[\tau]$ denotes the predicted probability of selecting the ground-truth beam index $b^\star[\tau]$ at time slot τ. The hyperparameter γ is the focusing parameter which down-weights easy examples. A large γ leads to small loss for well-classified samples (those with high output probabilities), helping the model concentrate on difficult or misclassified samples that are more informative. On the contrary, $\gamma = 0$ leads to the conventional cross-entropy loss which treats all samples with equal importance. The overall task loss for the input sequence $\mathcal{Z}[t]$ is expressed as

$$\mathcal{L}_{\mathrm{task}}[t] = \sum\nolimits_{\tau=t}^{t+J} l_{\mathrm{focal}}[\tau]. \tag{7}$$

The overall training procedure is summarized in Algorithm 1, where E, N_{b}, and B represents the number of total epochs, the number of batches in each epoch, and batch size, respectively. Here, $\mathcal{D}_{\mathrm{tr}}$ and $\mathcal{D}_{\mathrm{evl}}$ represents the training and validation datasets, respectively. With random initialization, the model is updated over E epochs. For each epoch, N_{b} batches are randomly generated for batch training in steps 4–9, where $\mathcal{T}^{(n)} = \{t_i^{(n)} | \mathcal{Z}[t_i^{(n)}] \in \mathcal{D}_{\mathrm{tr}}^{(n)}, \forall i\} = \{t_1^{(n)}, \ldots, t_B^{(n)}\}$ denotes the set of time stamps in the n-th batch. The best model is updated if a lower validation loss is found after batch training.

Table 1: Overall generalization performance of the ML model in $\%$ for ATop-k accuracy and ADBA score.

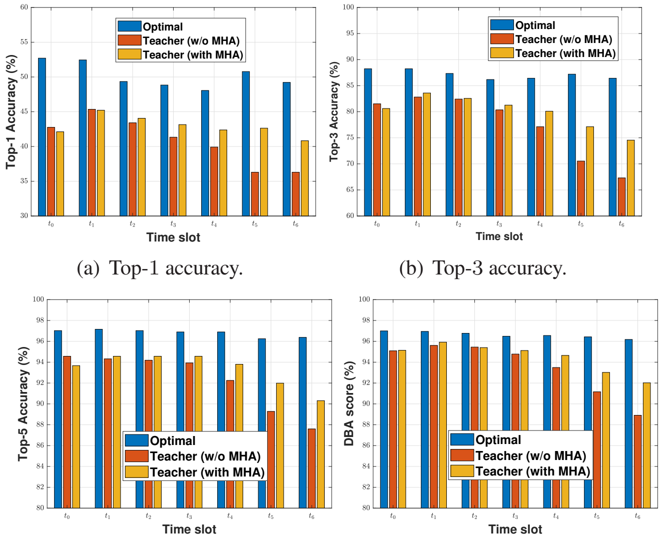
Metric	W/o MHA	With MHA	Optimal [19]
Test loss	1.141	1.050	0.8158
ATop-1	40.77	42.91	50.20
ATop-3	77.44	79.97	87.15
ATop-5	92.31	93.35	96.81
ADBA	93.50	94.47	96.63

4. NUMERICAL RESULTS

In this section, we evaluate the performance of the proposed ML model for beam tracking.[1] Experiments are based on Scenario 9 of the DeepSense 6G dataset [22], which provides sensory data and optimal beams for real-world mmWave communications. For any time step t, the maximum number of past images is set to $L = 8$ in each sequence sample $\mathcal{Z}[t]$, while the number of future time steps for beam prediction is set to $J = 6$. The overall dataset contains a total of $T = 4060$ samples $\{\mathcal{Z}[t], \mathbf{b}^{\star}[t]\}$ with 80% training samples and 20% validation samples. In the training stage, the initial learning rate is 10^{-4} with a cyclic cosine annealing scheduler used. We set $\gamma = 2$ for the loss function. In the MHA module, we adopt 8 heads for diverse attentions. As a result, the ML model has approximately a total of 1.8×10^6 trainable parameters.

We use the task loss, Top-k accuracy, and distance-based accuracy (DBA) [23] for performance evaluation. The task loss at the inference stage reflects the overall generalization performance, while the other two metrics targets for specific time slots. Specifically, the Top-k accuracy measures whether the ground-truth label is among the model's Top-k predicted labels. In contrast, the DBA metric computes the distance of the predicted beams from the ground-truth beams and assigns scores based on how far the predicted beam is from the ground-truth beam. The DBA score is computed using Top-3 accuracy, and we refer readers to [23] for a detailed explanation. Note that both the Top-k and DBA scores target one time slot. To reflect the overall performance across all $J + 1$ time slots, we further consider average Top-k (ATop-k) and average DBA (ADBA) over all time slots.

Table 1 shows the overall generalization performance of the designed ML model, where "W/o MHA" and "With MHA" represent the architecture without and with the MHA module. Unlike [19], which relies on YOLOv4 [24] and manual selection to eliminate the interference to the sensing target, we suppress background interference through adjacent-frame subtraction and highlight the sensing target with motion masks. Together with the designed CNNs, we extract compact image features at significantly lower complexity than YOLOv4 (approximately 6.4×10^7 parameters). It is seen that the MHA module can effectively enhance the generalization of the ML model. Furthermore, the proposed ML model (with MHA) can achieve over 97% of the optimal ADBA with only 3% complexity of the model in [19] (including the YOLOv4).



(a) Top-1 accuracy.

(b) Top-3 accuracy.

(c) Top-5 accuracy.

(d) DBA score.

Fig. 3: Performance of the ML model.

Figs. 3(a)–3(d) show the Top-1, Top-3, and Top-5 accuracy and DBA score of the considered schemes versus current and six future time slots, respectively. All schemes show lower accuracy and DBA scores for distant time slots, as beam prediction becomes more difficult further into the future. Furthermore, we observe that the integrated attention mechanism primarily improves the prediction accuracy of far future time slots, i.e., t_4–t_6, especially for Top-1 and Top-3 accuracy. For instance, percentage-point gains of 6.34 in Top-1 accuracy, 6.59 in Top-3 accuracy, 2.7 in Top-5 accuracy, and 1.86 in DBA score at time slot t_5 are attained by the attention-enhanced ML model compared to the architecture without MHA. Such results verify that the MHA can effectively capture the time dependency relation of the input sequence, extracting useful information for more challenging future beam predictions. Although the Top-3 accuracy of the proposed ML model is no more than 85%, the corresponding DBA scores can reach up to 95%, as seen in Fig. 3(d).

5. CONCLUSIONS

This paper proposed an attention-enhanced end-to-end learning framework for long-term beam tracking based on past sensing images. We designed an efficient ML model consisting of CNNs, GRUs, and MHA for extracting features and processing temporal dependencies. Numerical results demonstrate the superiority of the proposed ML model over the state-of-the-art scheme in terms of achieving comparable beam prediction accuracies across both current and six future time slots while requiring only 3% its complexity. The designed long-term beam tracking significantly reduces signaling overhead, latency, and power consumption associated with sensing and processing for beam training, advancing the practical implementation.

[1] The code source is available online at https://github.com/WillysMa/Sensing-Assisted-Beam-Tracking.git.

6. REFERENCES

[1] Shoaib Imran, Gouranga Charan, and Ahmed Alkhateeb, "Environment semantic communication: Enabling distributed sensing aided networks," vol. 5, pp. 7767–7786, 2024.

[2] Sajad Rezaie, Elisabeth de Carvalho, and Carles Navarro Manchón, "A deep learning approach to location- and orientation-aided 3D beam selection for mmWave communications," *IEEE Trans. Wireless Commun.*, vol. 21, no. 12, pp. 11110–11124, 2022.

[3] João Morais, Arash Bchboodi, Hamed Pezeshki, and Ahmed Alkhateeb, "Position-aided beam prediction in the real world: How useful GPS locations actually are?," in *Proc. IEEE Int. Conf. Commun.*, 2023.

[4] Shuaifeng Jiang, Gouranga Charan, and Ahmed Alkhateeb, "LiDAR aided future beam prediction in real-world millimeter wave V2I communications," *IEEE Wireless Commun. Lett.*, vol. 12, no. 2, pp. 212–216, 2023.

[5] Aldebaro Klautau, Nuria González-Prelcic, and Robert W. Heath, "LIDAR data for deep learning-based mmWave beam-selection," *IEEE Wireless Commun. Lett.*, vol. 8, no. 3, pp. 909–912, 2019.

[6] Umut Demirhan and Ahmed Alkhateeb, "Radar aided 6G beam prediction: Deep learning algorithms and real-world demonstration," in *Proc. IEEE Wireless Commun. and Networking Conf.*, 2022.

[7] Hao Luo, Umut Demirhan, and Ahmed Alkhateeb, "Millimeter wave V2V beam tracking using radar: Algorithms and real-world demonstration," in *Proc. European Sign. Proc. Conf.*, 2023.

[8] Yuwen Yang, Feifei Gao, Xiaoming Tao, Guangyi Liu, and Chengkang Pan, "Environment semantics aided wireless communications: A case study of mmWave beam prediction and blockage prediction," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 7, pp. 2025–2040, 2023.

[9] Gouranga Charan, Tawfik Osman, Andrew Hredzak, Ngwe Thawdar, and Ahmed Alkhateeb, "Vision-position multi-modal beam prediction using real millimeter wave datasets," in *Proc. IEEE Wireless Commun. and Networking Conf.*, 2022.

[10] Weihua Xu, Feifei Gao, Shi Jin, and Ahmed Alkhateeb, "3D scene-based beam selection for mmWave communications," *IEEE Wireless Commun. Lett.*, vol. 9, no. 11, pp. 1850–1854, 2020.

[11] Yuanhao Cui, Jiali Nie, Xiaowen Cao, Tiankuo Yu, Jiaqi Zou, Junsheng Mu, and Xiaojun Jing, "Sensing-assisted high reliable communication: A transformer-based beamforming approach," *IEEE J. Sel. Topics Signal Process.*, vol. 18, no. 5, pp. 782–795, 2024.

[12] Shehbaz Tariq, Brian Estadimas Arfeto, Uman Khalid, Sunghwan Kim, Trung Q. Duong, and Hyundong Shin, "Deep quantum-transformer networks for multimodal beam prediction in ISAC systems," *IEEE Internet Things J.*, vol. 11, no. 18, pp. 29387–29401, 2024.

[13] Yu Tian, Qiyang Zhao, Fouzi Boukhalfa, Kebin Wu, Faouzi Bader, et al., "Multimodal transformers for wireless communications: A case study in beam prediction," *arXiv preprint arXiv:2309.11811*, 2023.

[14] Binpu Shi, Min Li, Ming-Min Zhao, Ming Lei, and Liyan Li, "Multimodal deep learning empowered millimeter-wave beam prediction," in *Proc. IEEE Veh. Technol. Conf.*, 2024.

[15] Yu Min Park, Yan Kyaw Tun, Walid Saad, and Choong Seon Hong, "Resource-efficient beam prediction in mmwave communications with multimodal realistic simulation framework," *arXiv preprint arXiv:2504.05187*, 2025.

[16] Qihao Zhu, Yu Wang, Wenmei Li, Hao Huang, and Guan Gui, "Advancing multi-modal beam prediction with cross-modal feature enhancement and dynamic fusion mechanism," *IEEE Trans. Commun.*, Early access, 2025.

[17] Kai Zhang, Wentao Yu, Hengtao He, Shenghui Song, Jun Zhang, and Khaled B Letaief, "Multimodal deep learning-empowered beam prediction in future THz ISAC systems," *arXiv preprint arXiv:2505.02381*, 2025.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[19] Shuaifeng Jiang and Ahmed Alkhateeb, "Computer vision aided beam tracking in a real-world millimeter wave deployment," in *Proc. IEEE Global Commun. Conf. Workshop*, 2022.

[20] MTCAJ Thomas and A Thomas Joy, *Elements of information theory*, Wiley-Interscience, 2006.

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[22] Ahmed Alkhateeb, Gouranga Charan, Tawfik Osman, Andrew Hredzak, Joao Morais, Umut Demirhan, and Nikhil Srinivas, "Deepsense 6G: A large-scale real-world multi-modal sensing and communication dataset," *IEEE Commun. Mag.*, vol. 61, no. 9, pp. 122–128, 2023.

[23] Gouranga Charan, Umut Demirhan, João Morais, Arash Behboodi, Hamed Pezeshki, and Ahmed Alkhateeb, "Multi-modal beam prediction challenge 2022: Towards generalization," *arXiv preprint arXiv:2209.07519*, 2022.

[24] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.