How to Define the Propagation Environment Semantics and Its Application in Scatterer-Based Beam Prediction

Yutong Sun, Jianhua Zhang, *Senior Member, IEEE*, Li Yu, *Member, IEEE*, Zhen Zhang, and Ping Zhang, *Fellow, IEEE*

***Abstract*—In view of the propagation environment directly determining the channel properties, the applications can also be solved with the aid of environmental information. Inspired by task-oriented semantic communication and machine learning (ML) powered environment-channel mapping methods, we aim to provide a new view of the environment from the semantic level. This letter defines propagation environment semantics (PES) as a limited set of propagation environment semantic symbols (PESS). The PESS is extracted, which is oriented to the application tasks with concerned channel properties as a foundation. For method validation, the PES-aided beam prediction (PESaBP) is investigated and implemented for non-line-of-sight (NLOS) scenarios. Environment features and graph representations are constructed for actions of channel quality evaluation and target scatterer detection with maximum power. It can obtain 0.92 and 0.9 precision, respectively, and save over 87% of the time cost.**

***Index Terms*—Propagation environment semantics, semantic mapping, propagation semantic symbols extraction, beam prediction.**

I. Introduction

With the ever-increasing diverse scenes and communication requirements, predictive 6G network with environment sensing enhancement is becoming promising [1], [2], [3]. Since the physical environment can be reconstructed with advanced sensing techniques, the traditional prediction methods can be improved. Detailed environment information [4], [5], [6] can be introduced and utilized for prediction tasks. However, the demanded new applications with high data rates require simultaneously accurate and online deployment. It is challenging to implement in dynamic and complex environments, especially for non-line-of-sight (NLOS) scenarios [7].

Manuscript received 9 December 2022; revised 12 January 2023; accepted 12 January 2023. Date of publication 18 January 2023; date of current version 11 April 2023. This work was supported in part by the National Science Fund for Distinguished Young Scholars under Grant 61925102; in part by the National Natural Science Foundation of China under Grant 92167202, Grant 62201087, Grant 62101069, and Grant 62031019; and in part by the Beijing University of Posts and Telecommunications-China Mobile Research Institute Joint Innovation Center. The associate editor coordinating the review of this article and approving it for publication was C. Shen. *(Corresponding author: Jianhua Zhang.)*

Yutong Sun, Jianhua Zhang, Li Yu, and Zhen Zhang are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: sun_yutong@bupt.edu.cn; jhzhang@bupt.edu.cn; li.yu@bupt.edu.cn; zhenzhang@bupt.edu.cn).

Ping Zhang is with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China, and also with the Department of Broadband Communication, Peng Cheng Laboratory, Shenzhen 518066, China (e-mail: pzhang@bupt.edu.cn).

Digital Object Identifier 10.1109/LWC.2023.3237827

Powered by natural language processing (NLP) and computer vision (CV) techniques that have lots of potential in processing intelligent tasks, semantic communication (SC) [8], [9] has drawn significant attention. It mainly relies on information conversion at the semantic level to achieve efficient and intelligent interaction [10]. Instead of recovering the raw data at the bit level, the system design focuses on establishing the semantic map between the transmitter (Tx) and receiver (Rx). The difference between the meaning of the transmitted content and the recovered ones is considered.

Similar to the semantic mapping of SC, online tasks do not always perform with every propagation environment changing. Namely, the semantic (one-to-many) mapping that carries the meaning is required rather than the object (one-to-one) mapping without understandable information. In [11], the cluster-nuclei is proposed and deployed by directly mapping the physical environment to the channel. The environment features are proposed in [12], which can assist the efficient channel prediction in dynamic conditions. From the perspective of the radio channel, semantics are implied in the propagation environment. The channel varies with the environment, and the environment impacts the channel properties. Hence, we aim to define the propagation environment semantic (PES). Then, the PES can be utilized for upper task deployment directly.

In this letter, PES is defined as a limited set of propagation environment semantic symbols (PESS). The PESS are deconstructed environment representations that reflect the essential environmental information at a semantic level. Therefore, semantic mapping can be established, and the specific tasks can be implemented by utilizing the related PES. In view of beam prediction in NLOS scenarios, the PES-aided beam prediction (PESaBP) is investigated. The environment features and graph representations are considered the PESS for the two semantic actions: channel quality evaluation and target scatterer detection. Rather than predicting the beam indices, the scatterer that can produce the path with maximum power is predicted to provide the optimal direction option.

II. Problem Formulation

Channel properties are critical information for semantic mapping between the environment and application tasks. In general, environment representations for upper tasks require the analysis of corresponding channel properties first. Then, the tasks can be employed based on the machine learning (ML) methods by utilizing the corresponding PES directly. The architecture of the proposed method is shown in Fig. 1.

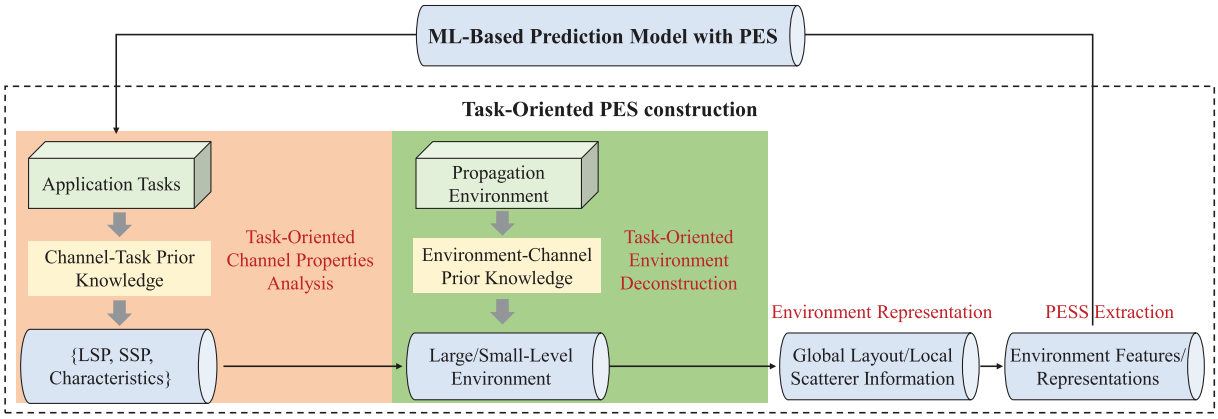

Fig. 1. The semantic mapping establishment: task-oriented PES construction and PES-based task implementation.

A. Task-Oriented Environment Deconstruction

As we know, specific prediction tasks focus on different channel properties. Therefore, although the task can not directly correlate to the environment, the relationship can be built by the corresponding needs of channel properties. In view of the general solution of propagation environment deconstruction, all the channel properties are considered, including large-scale parameters (LSP), small-scale parameters (SSP), and characteristics, as shown in Fig. 1. Furthermore, the propagation environment can be deconstructed following the environment-channel prior knowledge. First, the environment can be roughly represented at large-scale and small-scale levels to meet the essential information requirement of different channel properties. The layout and global environment representations can reflect the environment at the large-scale level, while local and target representations can abstract the environment at the small-scale level. Then, the corresponding PES can be extracted at the semantic level for diverse environment representation requirements.

B. PES Construction With PESS Extraction

In view of the specific task, environmental information is not equally important. Therefore, the task-related PES should be abstracted instead of retaining all the information. Namely, the PES is the variable that reflects the semantics changes of target tasks, and irrelevant environment information should no longer be considered. The environment should be analyzed from the propagation perspective. When radio waves encounter different scatterers, diverse propagation paths between the Tx and Rx are produced. The paths are caused by significant propagation mechanisms, such as line-of-sight (LOS) transmission, reflection, and diffraction. The scatterers that form one specific path can be combined as a group. Thus, the layout of the physical environment is a crucial factor impacting path generation.

As a result, the geometry relationship of scatterers can be regarded as the environment-channel prior knowledge, which can aid the PESS extraction. According to the propagation mechanisms, the essential geometric attributes that affect path production include position, dimensions, and layout [11]. For representing the environment characteristics and global structure, the fundamental PESS can be extracted as features $PESS_{feature}$, linear vectors $PESS_{vector}$, and non-linear graphs $PESS_{graph}$. It is expandable when more representations are considered and contained. The basis PES can be presented as the set of the PESS:

$$PESS = \{PESS_{feature}, PESS_{vector}, PESS_{graph}\}. \quad (1)$$

III. PESABP

A beam prediction case is given for method verification to show how the PES can be extracted and employed. After obtaining the essential PES of the beam prediction, the PESaBP can be implemented. The support vector machine (SVM) and graph neural network (GNN) are deployed for channel quality evaluation and target scatterer detection by leveraging the corresponding PES directly.

A. Task-oriented Channel Properties Analysis

Beam prediction generally aims to improve the communication quantity by switching to a better direction. Two task actions can be considered for PES-based beam prediction: channel quality evaluation and target scatterer prediction, as shown in Fig. 2. Therefore, the respective channel properties can be analyzed according to the requirements. The LSP, including path loss, delay spread (DS), etc., can indicate the overall condition of the current channel. The parameter types can be ignored for semantic mapping. In addition to LSP, the blockage characteristic of LOS is a crucial index that impacts radio wave propagation. It can cause significant power attenuation. Therefore, the critical channel properties of quality evaluation action $A_{evaluation}$ can be expressed as

$$A_{evaluation} = \{LSP, Blockage\}. \quad (2)$$

The optimal beam direction should be found for performance improvement once it is determined that the current channel is unqualified. Instead of predicting with the pre-defined codebook, the target scatterer producing the path of maximum power is detected for the best direction. It is an SSP-related prediction problem. Hence, the channel properties of target scatterer detection $A_{detection}$ can be denoted by

$$A_{detection} = \{P_j, j \in J\}, \quad (3)$$

where P_j is the power of the j-th path and total J paths.

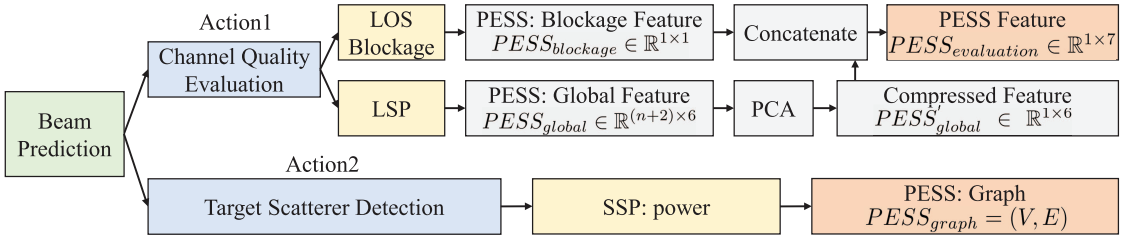

Fig. 2. The process of action decomposition and corresponding PESS extraction.

B. PES-Based Channel Quality Evaluation

According to the channel properties, the environmental information influencing the concerned characteristics and parameters should be represented from global and structural aspects for two different semantic actions. However, for practice applications, a dynamic environment might cause propagation path changes but not significant statistics changes. Hence, the environment is represented for all the concerned properties instead of each parameter or characteristic. The basic environment data are utilized, which are environment-isolated information with no scene constraints, including the position coordinates of Tx (x_t, y_t, z_t), Rx (x_r, y_r, z_r), and internal scatterers (x_i, y_i, z_i). Moreover, as for internal scatterers, the long l_i, width w_i, and height h_i are also exploited, where $i \in n$ for total n internal scatterers.

PESS of Features: For the simultaneous description of the blockage characteristics and statistical LSP, the blockage and global features are extracted and concatenated into an exclusive PESS representation, as shown in Fig. 2. The degree of LOS occlusion is leveraged as a PESS for blockage description. In practice, diverse methods can be utilized for PESS feature calculation according to the geometric relationship between the Tx, Rx, and internal scatterer. There, we use the method mentioned in [12], which obtains the blockage feature for each scatterer by describing the extent to which LOS and scatterers intersect. Specifically, the distance d_i between the center point of the i-th scatterer and LOS is calculated according to the position coordinates. Then b_i can be defined as the ratio of the d_i and width of i-th scatterer w_i. Then the maximum blockage feature is selected as the environment-level PESS, which can be denoted as

$$PESS_{blockage} = max\{b_1, b_2, \ldots, b_n\}, \tag{4}$$

where there are n internal scatterers.

As for the LSP representation, the PESS that contains the general environment information should be extracted. Because of the uncertain parameter requirement, the embedding feature is utilized rather than the particular calculated feature. The matrix of the original global representation $PESS_{global}$ is constructed by the original environmental data. In view of the 3-dimensional coordinates vector of Tx and Rx, while the 6-dimensional vector for scatterers adding the size information, 0 paddings are utilized to fill the Tx and Rx row to deal with the inconsistency of dimensions. The $PESS_{global} \in \mathbb{R}^{(n+2)\times6}$ for the environment sample with n scatterers.

Therefore, the final PESS can be obtained by combining the matrix $PESS_{global}$ and the blockage feature value. In order to concatenate the two features of different dimensional together, the $PESS_{global}$ is first converted into a 1-dimensional vector. The commonly used unsupervised dimensionality reduction algorithm: principal component analysis (PCA), is utilized for compression [13]. Hence, the compressed feature $PESS'_{global} \in \mathbb{R}^{1\times6}$ is obtained. Finally, the blockage and global feature can be concatenated as a whole environment feature, i.e., $PESS_{evaluation} \in \mathbb{R}^{1\times7}$.

SVM-Based Channel Quality Evaluation: As for dividing the channel quality into qualified and unqualified, the quality evaluation can be solved as a binary classification problem, differentiating the unqualified channel as class 0 and the qualified ones as class 1. The quality threshold can be set for diverse requirements according to the cumulative probability of the power. The classic machine learning methods are considered instead of neural network-based methods due to the proposed low-dimensional global features. SVM is employed for efficiency, in which the neural networks have no advantage but require more learning and time costs. The proposed features and corresponding classes are inputs and outputs of SVM model learning.

SVM [14] is a classic ML algorithm to maximize a particular mathematical function for a given collection of data that performs classification by constructing the hyperplane. The kernel function is the crucial calculation that enables the SVM to map the data from a low-dimensional space to a higher-dimensional space, which can be denoted by

$$\langle s_1 \cdot s_2 \rangle \leftarrow K(s_i, s_j) = \big\langle \Phi(s_i) \cdot \Phi(s_j) \big\rangle, \tag{5}$$

where Φ is a nonlinear function that maps the input space into the feature space and K is the kernel function.

Four classical kernel functions are used for nonlinear model learning, including linear, polynomial, sigmoid, and radial basis kernels. Linear and polynomial kernel functions can be described as

$$K(s_i, s_j) = \big\langle s_i, s_j \big\rangle, K(s_i, s_j) = (1 + \langle s_1, s_2 \rangle)^c, \tag{6}$$

where c is the degree of the kernel function. The radial basis kernel can map the primitive features to infinite dimensions, which can be expressed as

$$K(s_i, s_j) = exp(-\frac{\big\| s_i - s_j \big\|}{2\sigma^2}). \tag{7}$$

While the sigmoid kernel function comes from the neural network, which is generally denoted by

$$K(s_i, s_j) = tanh(\gamma \langle s_1, s_2 \rangle + r), \tag{8}$$

where the γ and r are the kernel parameters.

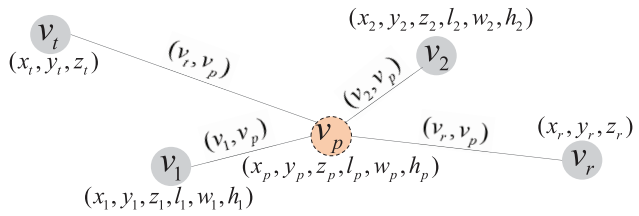

Fig. 3. The $PESS_{graph}$ of three scatterers and pending scatterer v_p.

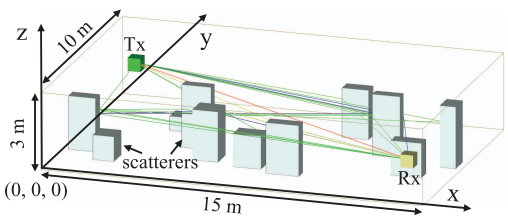

Fig. 4. A simulated sample with a random scatterer layout.

C. PES-Based Target Scatterer Detection

Unlike global environment information extraction, the environment layout demands nonlinear representations to preserve the structure of the propagation environment.

PESS of Environment Graph: The graph data in non-Euclidean space is utilized [15]. Specifically, environment graphs are constructed as the PESS for each internal scatterer to be classified for scatterer detection. In order to mark the pending scatterer, the edges are built between the pending scatterer and other nodes. Let $PESS_{graph} = (V, E)$ denotes the graph with nodes V, edges E, and node feature vectors X. And V consist of Tx node v_t, Rx node v_r, and n scatterer nodes, i.e., $V = \{v_t, v_r, v_1, v_2, \ldots, v_n\}$, as shown in Fig. 3. Meanwhile, E can be expressed as $E = \{(v_t, v_p), (v_r, v_p), \ldots, (v_n, v_p), n \neq p\}$. Feature vectors of Tx, Rx and the i-th scatterer node are $X_t = (x_t, y_t, z_t)$, $X_r = (x_r, y_r, z_r)$ and $X_i = \{(x_i, y_i, z_i, l_i, w_i, h_i), i \in n\}$.

GNN-Based Target Scatterer Detection: In the case of an unqualified channel, the beam should be switched to the better direction, i.e., the target scatterer with maximum power should be detected. Hence, the issue can be considered a scatterer classification mission by classifying the scatterers into two classes, i.e., scatterer with maximum power S_{max} (class: 1) and other scatterers (class: 0). GNN is constructed utilizing the net-architecture in [16] for graph data process and learning. Due to each scatterer corresponding to one graph, the GNN model should be deployed as graph level classification task. The **Aggregate**$(\cdot)$ and **Combine**$(\cdot)$ are the critical operators for modeling, where the former serves as the aggregation function of the neighborhood information, and the latter passes the aggregated node feature to a learnable layer to generate node embedding for the GNN layer.

Let $a_v^{(l)}$ stand for the nodes representing the structural information captured within the l-hop network neighborhood in k iterations of aggregation. Hence, the l-th layer can be denoted by

$$a_v^{(l)} = \mathbf{Aggregate}^{(l)}(\{h_u^{(l-1)} : u \in \mathcal{N}(v)\}). \tag{9}$$

$$h_v^{(l)} = \mathbf{Combine}^{(l)}(h_v^{(l-1)}, a_v^{(l)}), \tag{10}$$

where $h_v^{(l)}$ is the feature vector of node v at the l-th iteration, for $l = 1, 2, \ldots, L$ and $\mathcal{N}(v)$ is a set of nodes adjacent to v. The $h_v^{(0)}$ is initialized with X_v. As for the model, 8 MLPs are constructed. For each MLP, 6 hidden layers are deployed, where 1024 neurons are set for each layer. Finally, scores of two classes can be obtained by a fully-connected network.

However, the classification is independently deployed for each internal scatterer. For the unique detected scatterer of one environment, the scatterer classification results of a specific environment are ranked by the scores. In practice, the scatterer with the top score of class 1 or the one with a minimum score of class 0 is considered the final prediction result.

IV. Simulation and Results

A. Simulation Settings

The environment samples with random scatterer layouts are considered. The 3D modeling software Blender and the ray-tracing tool WirelessInSite are utilized for the traceable environment, and channel generation [12], as shown in Fig. 4. The propagation area's length, width, and height are set at 15, 10, and 3 m, and the Tx and Rx are set on the two sides of the diagonal. Regular scatterers with random numbers, positions, and dimensions are generated. The dataset includes samples with $M \in [3, 12]$ numbers of internal scatterers.

The training and testing data are a random selection of samples with different numbers of scatterers for generalization verification of the prediction method, in which the samples of the testing data consist of 4, 8, and 12 scatterers, and the rest samples are training data. The corresponding channels at 28 GHz are produced using the omnidirectional antenna, and six-order reflections and one-order diffraction are set. After selecting the NLOS samples, there are 1475 and 264 samples in training and testing data.

B. Performance Metrics

As for the binary classification problem, the precision and receiver operating characteristic curve (ROC) are given for performance analysis with the device of one NVIDIA GeForce RTX 2080. The precision *pre* can be calculated as

$$pre = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{TN}} \tag{11}$$

where true-positive (TP) and true-negative (TN) are the true samples classified as positive and negative.

The ROC is wildly used to reflect the identity ability of binary classification problems. It is plotted with a false-positive rate (FPR) and true-positive rate (TPR), which can be expressed as

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}, \mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{12}$$

where the false-positive (FP) and false-negative (FN) denote the false samples that be predicted with positive and negative. The crucial feature of ROC is the area under curve (AUC), where the closer the AUC is to 1, the better the performance.

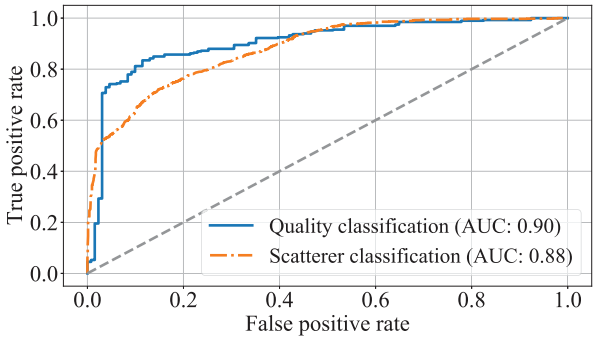

Fig. 5. The ROC curve and AUC of quality and scatterer classification.

TABLE I
THE PRECISION COMPARISIONS

Method	Configuration	Precision
Proposed Target Scatterer Detection	\	0.90
Beam Indices Prediction in [5]	Top-1	0.51
	Top-2	0.72
	Top-3	0.84

TABLE II
THE COMPARISIONS OF TESTING TIME

Action	Testing time (ms)
Proposed Channel Quality Evaluation	4.7
Proposed Target Scatterer Detection	0.33
Beam Indices Prediction in [5]	41

C. Quality and Target Scatterer Classification Results

The quality threshold is set at 60% cumulative distribution function (CDF) of power. The SVM models with linear, polynomial, radial basis, and sigmoid kernel functions get 0.88, 0.92, 0.89, and 0.53 precisions, respectively. The polynomial kernel offers the best result. Then, according to the precision calculation, we can obtain the scatterer classification's accuracy is 0.89. The ROC and AUC of the SVM and GNN-based model are shown in Fig. 5, which indicates that the models can discriminate the classes well.

Instead of predicting the beam index of the pre-defined codebook, the optimal scatterer is detected for the best direction estimation. Based on the scatterer classification results, the target scatterer can be selected for an environment sample by the rank of classification scores. For comparison, the beam indices prediction in [5] is tested by utilizing the code at [17]. The top view images are generated by converting the coordinates of the dataset and setting the scatterers with diverse grayscale according to the different heights. The digital architecture system is employed 8 antennas corresponding to 8 classes. The results are indicated in Table I.

The top-3 precision of the beam indices prediction in [5] is around 0.84, and the proposed scatterer detection gets around 0.9. The target scatterer prediction can obtain higher accuracy under a dynamic environment with low correlations. Moreover, the testing time is compared for cost evaluation in Table II. The results demonstrate that the proposed PESaBP method with two actions together can save over 87% time cost, potentially supporting the online prediction for changing environments in NLOS scenarios.

V. CONCLUSION AND FUTURE WORK

This letter is interested in the PES definition, in which The PES is considered the task-oriented environment representation set. The semantic mapping between the propagation environment and applications is built by analyzing the concerned channel properties. Simulation results of the PESaBP case indicate the efficiency and precision in NLOS scenarios. Considering the environment at a semantic level is a promising solution to support online prediction. With these initial works, further research is required to facilitate PES improvement and clarify the performance of PES utilization in practice.

REFERENCES

[1] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The roadmap to 6G: AI empowered wireless networks," *IEEE Commun. Mag.*, vol. 57, no. 8, pp. 84–90, Aug. 2019.
[2] P. Zhang et al., "Ubiquitous-X: Constructing the future 6G networks," *Scientia Sinica Informationis*, vol. 50, no. 6, pp. 913–930, 2020.
[3] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May/Jun. 2020.
[4] W. Xu, F. Gao, J. Zhang, X. Tao, and A. Alkhateeb, "Deep learning based channel covariance matrix estimation with user location and scene images," *IEEE Trans. Commun.*, vol. 69, no. 12, pp. 8145–8158, Dec. 2021.
[5] M. Alrabeiah, A. Hredzak, and A. Alkhateeb, "Millimeter wave base stations with cameras: Vision-aided beam and blockage prediction," in *Proc. IEEE 91st Veh. Tech. Conf. (VTC2020-Spring)*, 2020, pp. 1–5.
[6] Q. Zhu et al., "Map-based channel modeling and generation for U2V mmWave communication," *IEEE Trans. Veh. Technol.*, vol. 71, no. 8, pp. 8004–8015, Aug. 2022.
[7] G. Nie et al., "A predictive 6G network with environment sensing enhancement: From radio wave propagation perspective," *China Commun.*, vol. 19, no. 6, pp. 105–122, Jun. 2022.
[8] P. Zhang et al., "Toward wisdom-evolutionary and primitive-concise 6G: A new paradigm of semantic communication networks," *Engineering*, vol. 8, pp. 60–73, Jan. 2022.
[9] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning based semantic communications: An initial investigation," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2020, pp. 1–6.
[10] K. Niu et al., "Towards semantic communications: A paradigm shift," 2022, *arXiv:2203.06692*.
[11] J. Zhang, "The interdisciplinary research of big data and wireless channel: A cluster-nuclei based channel model," *China Commun.*, vol. 13, no. 2, pp. 14–26, Jan. 2016.
[12] Y. Sun et al., "Environment features-based model for path loss prediction," *IEEE Wireless Commun. Lett.*, vol. 11, no. 9, pp. 2010–2014, Sep. 2022.
[13] Z. Yuan, J. Zhang, Y. Zhang, P. Tang, and L. Tian, "A novel complex PCA-based wireless MIMO channel modeling methodology," in *Proc. IEEE 92nd Veh. Tech. Conf. (VTC2020-Fall)*, 2020, pp. 1–5.
[14] T. M. Hoang, N. M. Nguyen, and T. Q. Duong, "Detection of eavesdropping attack in UAV-aided wireless systems: Unsupervised learning with one-class SVM and k-means clustering," *IEEE Wireless Commun. Lett.*, vol. 9, no. 2, pp. 139–142, Feb. 2020.
[15] Y. Sun, J. Zhang, Y. Zhang, L. Yu, Q. Wang, and G. Liu, "Environment information-based channel prediction method assisted by graph neural network," *China Commun.*, vol. 19, no. 11, pp. 1–15, Nov. 2022.
[16] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" 2018, *arXiv:1810.00826*.
[17] M. Alrabeiah. Accessed: Mar. 2022. [Online]. Available: https://github.com/malrabeiah/CameraPredictsBeams