

Geo2ComMap: Deep Learning-Based MIMO Throughput Prediction Using Geographic Data

Fan-Hao Lin, Tzu-Hao Huang, Chao-Kai Wen, *Fellow, IEEE*, Trung Q. Duong, *Fellow, IEEE*

Abstract—Accurate communication performance prediction is crucial for wireless applications such as network deployment and resource management. Unlike conventional systems with a single transmit and receive antenna, throughput (Tput) estimation in antenna array-based multiple-output multiple-input (MIMO) systems is computationally intensive, i.e., requiring analysis of channel matrices, rank conditions, and spatial channel quality. These calculations impose significant computational and time burdens. This paper introduces Geo2ComMap, a deep learning-based framework that leverages geographic databases to efficiently estimate multiple communication metrics across an entire area in MIMO systems using only sparse measurements. To mitigate extreme prediction errors, we propose a sparse sampling strategy. Extensive evaluations demonstrate that Geo2ComMap accurately predicts full-area communication metrics, achieving a median absolute error of 27.35 Mbps for Tput values ranging from 0 to 1900 Mbps.

I. INTRODUCTION

Radiomaps play a crucial role in future intelligent communication systems, as they enable the integration of environmental knowledge into system design to support high-throughput and reliable communication. High-precision radiomaps are essential for applications such as cellular network planning, fingerprint-based localization, and UAV path planning [1]–[3]. Deterministic simulations using propagation models, such as ray tracing, are widely used for radiomap generation. However, the high computational complexity of ray tracing limits its practicality. Recent advances in deep learning (DL) and the availability of large-scale datasets have facilitated DL-based radiomap estimation [2], [4], [5] (see [6] for a survey), achieving high accuracy while significantly reducing computational costs and processing time. Notably, [5] introduced Geo2SigMap, a framework that reconstructs dense path gain maps using geographic data and sparse site measurements with excellent accuracy.

Despite these advancements, path gain alone is insufficient for comprehensive wireless communication analysis. In conventional systems with a single transmit and receive antenna,

path gain directly correlates with throughput (Tput). However, in antenna array-based MIMO systems—prevalent in 5G and emerging 6G networks—Tput estimation requires modeling spatial signal interactions, channel matrices, and key indicators such as the Rank Indicator (RI) and Channel Quality Indicator (CQI) [7]. Consequently, Tput estimation is significantly more resource-intensive than path gain mapping. Building on the foundation of Geo2SigMap [5], this paper introduces Geo2ComMap, a comprehensive framework that leverages geographic databases to estimate multiple communication metrics in MIMO-OFDM systems. Our key contributions include:

Dataset Generation: Traditional radiomaps do not include datasets linking geographic maps to MIMO-OFDM Tput, posing a challenge for deep learning-based estimation. To address this, we integrate a MIMO-OFDM Tput simulation program with geographic databases [5] to generate a comprehensive training dataset, enabling effective DL model development.

Deep Learning Model: We propose U-Net-TP, the first U-Net-based architecture designed to predict multiple communication metric maps using geographic data and sparse measurements. U-Net-TP generalizes across different locations without requiring retraining. We extensively evaluate its performance across various input configurations, providing key insights into geographic-to-communication mappings.

Sampling Selection Strategy: To mitigate high-error regions, we introduce a sparse map design strategy that reduces extreme prediction errors without increasing system costs.

II. SYSTEM DESIGN

This section outlines the methodology of **Geo2ComMap**, which comprises three primary modules. The first module converts a geographical Building Map into corresponding channel matrices. The second module computes system Tput based on these channel matrices. The final module leverages a deep learning method to enable efficient mapping from Building Map to communication metrics.

A. Building Map to Channel Matrix

To build a comprehensive training dataset for the deep learning model, we adopt a systematic approach inspired by [5]. This involves generating a Building Map and conducting ray-tracing simulations.

For a given geographical area of dimensions $L_x \times L_y$ (in square meters), we first generate a Building Map, $\mathbf{B} \in \mathbb{R}^{N_x \times N_y}$, with a resolution of r , where $L_x/N_x = L_y/N_y = r$ meters. This map is accompanied by a 3D mesh model, generated using:

- **OpenStreetMap (OSM)** [8], an open-source geographic database, to extract building data from a selected area.

Manuscript received Mar. 31, 2025; accepted Apr. 2, 2025. The work of F.-H. Lin, T.-H. Huang, and C.-K. Wen was supported in part by the National Science and Technology Council of Taiwan (NSTC 113-2218-E-110-009), by a Google award, and by the Higher Education SPROUT Project of the Ministry of Education, Taiwan, funding the Sixth Generation Communication and Sensing Research Center. The work of T. Q. Duong was supported in part by the Canada Excellence Research Chair (CERC) Program CERC-2022-00109. (*Corresponding author: Chao-Kai Wen.*)

F.-H. Lin and C.-K. Wen are with the Institute of Communications Engineering, National Sun Yat-sen University, Kaohsiung 80424, Taiwan, Email: 22135lin@gmail.com, chaokai.wen@mail.nsysu.edu.tw.

T.-H. Huang is with the Institute of Electrical Control Engineering, National Yang Ming Chiao Tung University, Hsinchu 300, Taiwan, Email: peter94135@gmail.com.

T. Q. Duong is with the Faculty of Engineering and Applied Science, Memorial University, St. John's, NL A1C 5S7, Canada, and with the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, U.K, E-mail: tduong@mun.ca.

- **Blender** [9], an open-source 3D graphics tool, to convert the extracted data into detailed 3D building models.

Using these 3D models, we conduct ray-tracing simulations with **Sionna** [10] to compute key propagation characteristics. The simulations consider both isotropic and directional antennas operating at a carrier frequency of 3.5 GHz. The isotropic antenna has a gain of 0 dBi, while the directional antenna has a boresight gain of 6.3 dBi and a horizontal/vertical half-power beamwidth of $65^\circ/8^\circ$, typical for cellular antennas. The directional antenna's orientation is randomly set within the azimuth range of $[0, 2\pi)$.

The base station (BS), serving as the transmitter, is randomly positioned within the x-y plane at a height of 5 meters above the tallest structure in the area. The user equipment (UE), acting as the receiver, is uniformly distributed across outdoor locations at a height of 1.5 meters. Using **Sionna**, we generate a Path Gain (PG) Map, $\mathbf{P}_{\text{iso}} \in \mathbb{R}^{N_x \times N_y}$, for each area, assuming an isotropic antenna at the BS, with values ranging from -160 dBm to 0 dBm. Alternatively, the PG Map can be estimated using DL-based methods such as **Geo2SigMap** [5].

While PG Maps can be efficiently generated without detailed system parameters, calculating MIMO system T_{put} requires a joint consideration of channel propagation characteristics and system-specific factors. Key propagation characteristics include channel coefficients α_l , propagation delay τ_l , and the zenith/azimuth angles of departure (AoD) (ϕ_l^t, θ_l^t) and arrival (AoA) (ϕ_l^r, θ_l^r) . Combining these elements within a MIMO-OFDM system with N_t transmit and N_r receive antennas, the channel matrix at subcarrier n is expressed as:

$$\mathbf{H}_n = \sum_{l=1}^L \alpha_l e^{-j2\pi f_n \tau_l} \mathbf{a}_r(\phi_l^r, \theta_l^r) \mathbf{a}_t^H(\phi_l^t, \theta_l^t), \quad (1)$$

where L is the number of multipath components, f_n is the corresponding subcarrier frequency, and $\mathbf{a}_r(\cdot) \in \mathbb{C}^{N_r \times 1}$ and $\mathbf{a}_t(\cdot) \in \mathbb{C}^{N_t \times 1}$ are the array response vectors of the UE and BS, respectively.

Using ray-tracing simulations, (1) is applied to generate channel matrices for both isotropic and directional antennas. The differences between the two antenna types are inherently captured through the ray-tracing process, which accounts for variations in channel coefficients.

B. Channel Matrix to System T_{put}

T_{put} estimation involves evaluating the RI and CQI, which determine the number of spatial streams and the modulation and coding scheme (MCS), respectively. According to 3GPP TS 38.214 [11], CQI levels range from 1 to 15, each corresponding to a specific spectral efficiency. The process of determining RI and CQI follows these steps:

- 1) Compute the autocorrelation matrix over a group of subcarriers from the channel matrices in (1), and perform eigenvalue decomposition to obtain spatial precoders and their corresponding equivalent channel gains.
- 2) Evaluate performance metrics for all possible RI and CQI combinations based on the equivalent channel gains.

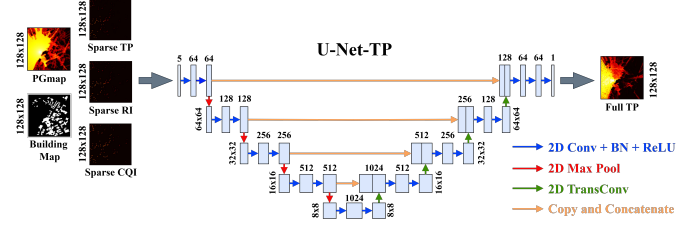


Fig. 1. Architecture of Geo2ComMap, which is a 5-input channel U-Net designed to achieve efficient and precise T_{put} prediction.

- 3) Select the RI and CQI combination that maximizes spectral efficiency under current channel conditions.

Once the RI and CQI are determined, the system T_{put} is calculated according to 3GPP TS 38.306 [7] as:

$$TP = 10^{-6} \cdot v \cdot \log_2 Q \cdot R \cdot \frac{N_{\text{RE}}}{T_s}, \quad (\text{Mbps}) \quad (2)$$

where $1 \leq v \leq \min(N_t, N_r)$ represents the RI; Q and R denote the MCS associated with the CQI; $N_{\text{RE}} = 135 \times 12 = 1620$ represents the number of resource elements per OFDM symbol; $T_s = 10^{-3}$ seconds is the duration of one OFDM symbol; the factor 10^{-6} converts the unit from bps to Mbps. The resulting T_{put} Map, $\mathbf{TP} \in \mathbb{R}^{N_x \times N_y}$, represents the spatial distribution of T_{put} values, ranging from 0 to 1900 Mbps, across different locations within the Building Map, \mathbf{B} .

C. Deep Learning Model for T_{put} Prediction

The previous subsections have demonstrated that generating a T_{put} Map involves a multi-step process. Each step requires complex computations and substantial processing time. To address this challenge, the core of **Geo2ComMap** is a deep learning model designed to efficiently predict a complete T_{put} Map for a given area using a Building Map and sparse T_{put} -related measurements. The proposed model, named **U-Net-TP**, is illustrated in Fig. 1. U-Net-TP takes a 5-channel input image: $[\mathbf{B}, \mathbf{P}_{\text{iso}}, \mathbf{R}\mathbf{I}_s, \mathbf{C}\mathbf{Q}\mathbf{I}_s, \mathbf{T}\mathbf{P}_s] \in \mathbb{R}^{N_x \times N_y \times 5}$, where \mathbf{B} represents the Building Map, \mathbf{P}_{iso} is the isotropic PG Map, and $\mathbf{R}\mathbf{I}_s, \mathbf{C}\mathbf{Q}\mathbf{I}_s, \mathbf{T}\mathbf{P}_s$ are sparse maps containing RI, CQI, and T_{put} measurements from a limited number of UE locations. The model outputs a complete T_{put} Map, \mathbf{TP} .

U-Net-TP consists of nine convolutional blocks and four downsampling/upsampling layers. Each convolutional block includes two 2D convolutional layers with a kernel size of 3×3 , followed by batch normalization and ReLU activation. Downsampling is achieved through 2×2 max pooling, while upsampling is performed via 2D transposed convolution layers. In the contracting path, the number of channels doubles from 64 to 1024, while the spatial dimensions are halved at each stage. Skip connections link corresponding layers in the contracting and expanding paths, helping to retain spatial information and mitigate gradient vanishing or explosion during training.

Notably, U-Net-TP shares the same underlying structure as the model used for PG Map generation [5], which is a simpler task. However, the key differences lie in the input and output feature sets. In the next section, we will discuss the relevance

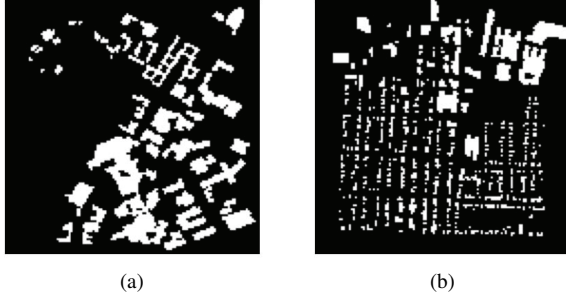


Fig. 2. Building Maps used in the experiments. (a) BM1, characterized by a prominent open space. (b) BM2, featuring densely clustered buildings.

of input features and the feasibility of using U-Net-TP to generate both Tput and PG Maps simultaneously.

III. DISCUSSIONS

A. Dataset Generation and U-Net Training

To train U-Net-TP in Geo2ComMap, we generate Building Maps and ray tracing datasets following the procedure described in Section II-A. The dataset consists of 32 distinct, non-overlapping areas across North America. Data generation is conducted using Sionna 0.15.1 and Blender 3.6. For each area, 100 BS locations are simulated with five antenna power levels ranging from 40 to 50 dBm, yielding a total of 15,920 datasets (excluding erroneous data). To generate complete ground truth communication-metric maps, we consider a MIMO-OFDM system with $N_t = N_r = 4$ and a bandwidth of 100 MHz. The channel matrix \mathbf{H} is derived from propagation paths using (1). A constant noise level of -93.98 dBm is assumed in all cases. To generate sparse RI, CQI, and Tput maps, we sample 200, 300, 400, 500, and 600 points from the full RI, CQI, and Tput Maps obtained through the MIMO Tput simulation described in Section II-B. These correspond to 1.22%, 1.83%, 2.44%, 3.05%, and 3.66% of the total area, respectively. Detailed analyses of the impact of varying sampling densities and distributions are provided in the subsequent subsections.

The dataset is divided into training, validation, and testing sets with an 84%, 10%, and 6% split, respectively. To ensure robust evaluation, this division is performed at the Building Map level, ensuring no overlap between the Building Maps used for training, validation, and testing. Data augmentation techniques—such as rotations (0° , 90° , 180° , and 270°) and mirroring of both input and output maps—are applied, expanding the dataset size fivefold. The model is trained using the mean squared error (MSE) loss function on the target communication metrics for 50 epochs on an NVIDIA GeForce GTX 1080 Ti GPU. For testing, 1,000 data samples are utilized, equally distributed across two scenarios: 500 samples for Building Map 1 (BM1) and 500 samples for Building Map 2 (BM2), as shown in Fig. 2. BM1 features a large open space, whereas BM2 consists of densely clustered buildings.

B. Effect of RI and CQI Information

The Tput in (2) is highly dependent on RI and CQI values. Therefore, it is reasonable to use sparse RI and CQI maps,

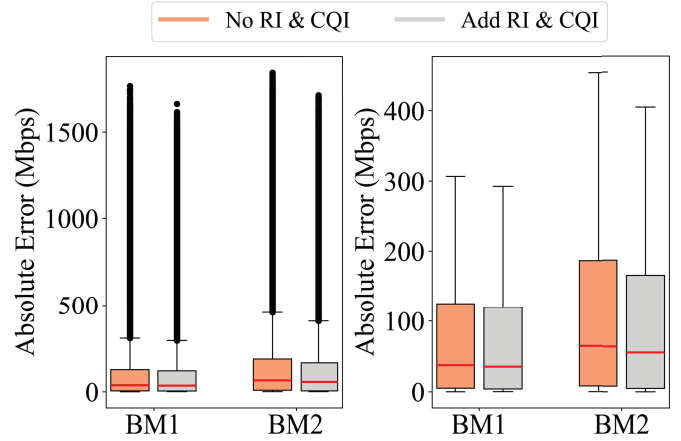


Fig. 3. Absolute error comparison of different input configurations in testing BM1 and testing BM2. The left box plot includes outliers (represented as black dots beyond the whiskers), while the right box plot excludes them. Each box represents the IQR of the absolute error, with the red line inside indicating the median value. The whiskers extend to the furthest data points within the range defined as $Q1 - 1.5 \cdot IQR$ to $Q3 + 1.5 \cdot IQR$, where $Q1$ and $Q3$ denote the first and third quartiles, respectively. Horizontal lines outside the boxes represent the outlier boundaries.

($\mathbf{RI}_s, \mathbf{CQI}_s$), as inputs for U-Net-TP. To assess the impact of these input features on Tput prediction, we compared two U-Net-TP architectures: one utilizing a 5-channel input image, as described in Sec. II-C, and another with a 3-channel input image, given by $[\mathbf{B}, \mathbf{P}_{iso}, \mathbf{TP}_s] \in \mathbb{R}^{N_x \times N_y \times 3}$.

The comparison results, obtained using testing data with 200 sampled sparse points, are presented in Fig. 3. The figure illustrates the absolute error distribution between the two U-Net-TP models. The findings clearly indicate that incorporating RI and CQI information significantly reduces the interquartile range (IQR), median, and maximum error magnitude. Specifically, with the 5-channel input configuration, the median absolute error of the predicted Tput Map is 35.37 Mbps and 55.98 Mbps for BM1 and BM2, respectively, while the IQR of the absolute error is 115.37 Mbps and 159.97 Mbps. In contrast, with the 3-channel input configuration, the median absolute error increases to 37.36 Mbps and 64.59 Mbps for BM1 and BM2, respectively, with an IQR of 120.68 Mbps and 178.55 Mbps. The inclusion of RI and CQI information results in substantial error reductions. This result highlights that incorporating detailed Tput-related features enables U-Net-TP to achieve more accurate predictions, particularly in complex environments such as BM2.

C. Effect of Different Numbers of Sampling Points

Next, we evaluate the impact of varying the number of sparse sampling points on model performance by training and testing models with different sparse point densities per sparse map. Fig. 4 presents the root MSE (RMSE) for various combinations of sparse points used in training and testing. The results show that increasing the number of sparse points during testing consistently reduces RMSE, regardless of the number of sparse points used in training. This improvement is attributed to the enhanced prediction accuracy afforded by a higher density of sparse points in the test set. Notably, while increasing the number of sparse points during

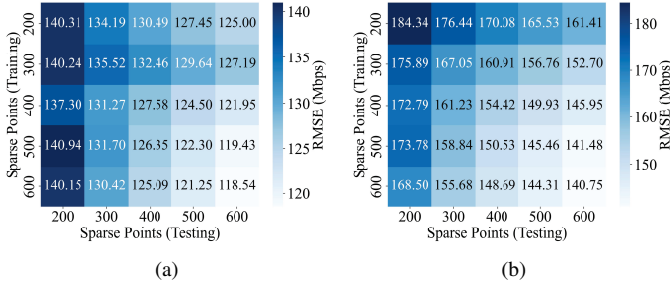


Fig. 4. RMSE values for different combinations of sparse points used in training and testing. (a) RMSE in BM1. (b) RMSE in BM2.

TABLE I
RMSE COMPARISON WITH 600 SAMPLED SPARSE POINTS

Method	BM1 RMSE	BM2 RMSE
Linear Interpolation	200.35 (Mbps)	204.06 (Mbps)
KNN Regression	162.33 (Mbps)	177.06 (Mbps)
U-Net-TP	118.54 (Mbps)	140.75 (Mbps)

training generally leads to a reduction in RMSE, this trend does not consistently hold across all combinations of training and testing configurations. In certain cases, a higher training point density does not necessarily translate to improved performance. For comparison with conventional approaches, we also present results from linear interpolation and KNN regression using uniform grid samples. Table I summarizes the RMSE values, demonstrating that U-Net-TP achieves superior prediction accuracy.

Fig. 5 presents the absolute error for models trained and tested with identical sparse point densities. The results show a consistent reduction in the IQR of absolute errors as the number of sparse points increases, reflecting improved predictive accuracy. Additionally, the median absolute error decreases with higher sparse point densities, reinforcing the benefits of increased sampling. Specifically, the model achieves median absolute errors of 35.36, 35.65, 31.03, 26.20, and 28.62 Mbps for BM1, and 55.98, 53.98, 47.73, 41.21, and 43.24 Mbps for BM2. However, the highest errors (outliers) remain largely unchanged despite the increase in sparse points, suggesting that a higher sampling density alone does not effectively mitigate extreme errors. The following section presents a method designed to address these high-error outliers more effectively.

D. Dealing with High-Error Outliers

In this subsection, we analyze the causes of high-error outliers and propose a solution to mitigate them. Fig. 6 highlights the locations of the top 5% highest errors in BM1, with the left and right plots corresponding to different BS locations. The results show that these high-error regions predominantly occur near building edges or in the spaces between buildings. These errors arise due to sudden T_{put} variations at building boundaries, especially in areas near the BS where T_{put} values are higher. Such abrupt transitions challenge the model’s ability to accurately predict T_{put} , leading to increased errors. This issue is further exacerbated in BM2, where densely clustered buildings create more complex T_{put} patterns and sharper value changes.

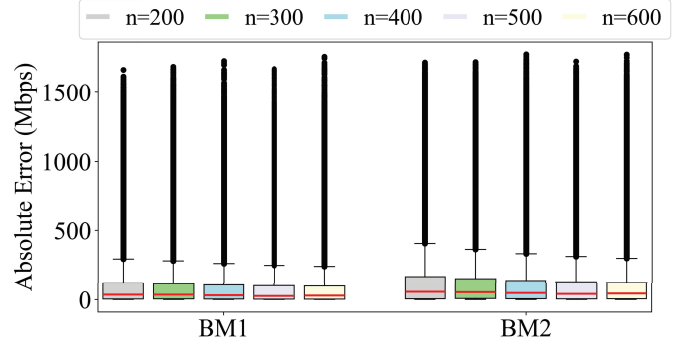


Fig. 5. Absolute error comparison for different numbers of sampled sparse points, where n denotes the number of sparse points.

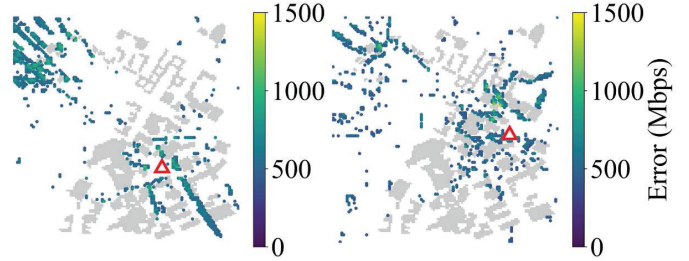


Fig. 6. Locations of the top 5% highest errors in BM1 with 200 sampled sparse points. Error locations are color-coded based on their magnitude, with darker colors indicating higher errors. The red triangle denotes the BS.

To address high-error regions, we introduce a sparse sampling strategy in which half of the samples are placed at locations with significant throughput variation, and the other half are selected randomly. High-gradient points are identified by computing the gradient of the isotropic PG map, \mathbf{P}_{iso} , using NumPy’s gradient function. The total gradient at each point is computed by summing the absolute gradients along both axes, enabling the selection of regions with large throughput changes. This method, referred to as special sampling, deterministically selects points where prediction errors are more likely to occur.

Fig. 7 compares absolute errors before and after applying the special sampling method across five different sampling densities. The results show significant reductions in the highest error values, with improvements of 142.53, 213.91, 142.47, 223.76, and 245.94 Mbps for BM1, and 35.54, 44.02, 135.21, 100.08, and 139.82 Mbps for BM2. These results confirm that the special sampling method effectively mitigates extreme errors and enhances predictive accuracy in both Building Maps.

E. Multi-Output U-Net-TP and Advanced Architectures

The multi-output U-Net-TP architecture (MO U-Net-TP) predicts multiple performance metrics in a single forward pass, offering improved computational efficiency compared to training separate models for each output. Here, we explore a MO U-Net-TP architecture with a 3-channel output image: $[\mathbf{P}_{\text{dir}}, \mathbf{TP}, \mathbf{RI}] \in \mathbb{R}^{N_x \times N_y \times 3}$, where \mathbf{P}_{dir} represents the directional PG Map, and \mathbf{TP} and \mathbf{RI} correspond to the

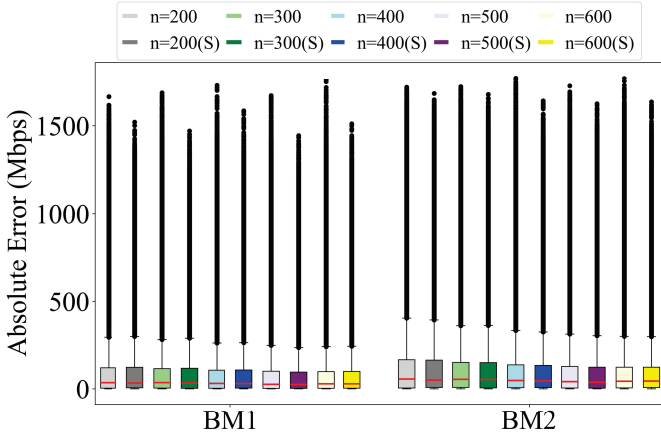


Fig. 7. Comparison of absolute errors with and without special sampling in BM1 and BM2. (S) represents special sampling.

complete Tput Map and RI Map, respectively. The model is trained and tested using 600 sparse points sampled via the special sampling method.

Fig. 8a compares the performance of U-Net-TP and MO U-Net-TP. The results show that adding additional output channels reduces the IQR. However, predicting multiple performance metrics within a single forward path introduces additional complexity, leading to a slight increase in the median absolute error compared to the single output (SO) configuration.

Although not explicitly depicted in the figures, MO U-Net-TP achieves a median absolute error of 0.15 for the predicted **RI** in BM1 and 0.18 in BM2. For \mathbf{P}_{dir} prediction, the median absolute error is 2.10 dB for BM1 and 2.18 dB for BM2, which is comparable to the performance of Geo2SigMap [5], a framework specifically designed for PG Map prediction. These results demonstrate that multi-task prediction using the MO framework is a viable and effective approach.

To address the observed increase in median absolute error, we investigate advanced architectural enhancements, such as attention mechanisms. [12] introduced attention gates (AGs), which can be seamlessly integrated into conventional CNNs with minimal computational overhead. AGs leverage coarse-scale contextual information to gate skip connections, allowing only relevant feature activations to be propagated during concatenation, thereby improving the network’s focus on salient regions. We incorporate AGs into the skip connections of the MO U-Net-TP, resulting in the AG U-Net-TP architecture. Fig. 8b presents the cumulative distribution function (CDF) of the absolute error for U-Net-TP, MO U-Net-TP, and AG U-Net-TP. The results show that AG U-Net-TP further reduces the median absolute error, achieving 27.35 Mbps for BM1 and 38.62 Mbps for BM2, with an average inference time of 0.23 seconds over 100 trials—representing the best Tput prediction performance among all evaluated configurations.

IV. CONCLUSION

We introduced Geo2ComMap, an efficient DL framework for Tput prediction in MIMO-OFDM systems. By leveraging Building Maps, isotropic PG Maps, and approximately 3%

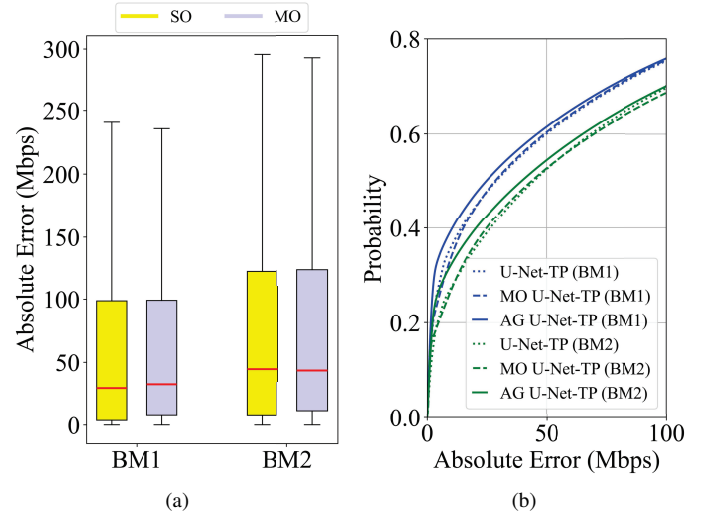


Fig. 8. Absolute error comparison of different U-Net architecture testing in BM1 and BM2. (a) U-Net-TP compared with MO U-Net-TP. (b) CDF of absolute error of U-Net-TP, MO U-Net-TP, and AG U-Net-TP.

sparse communication measurements, Geo2ComMap accurately reconstructs complete Tput maps and other key communication metrics for a given area. Through extensive experiments, we identified critical input features for the proposed DL model, U-Net-TP, and evaluated its predictive performance. To address the impact of high-error outliers, we introduced a specialized sampling strategy that significantly reduces both extreme errors and RMSE when compared to random sampling. Furthermore, we explored advanced DL architectures: the MO U-Net-TP improves the IQR of absolute errors, while the AG U-Net-TP further reduces overall prediction error. The source code and datasets used in this study are publicly available at [13].

REFERENCES

- [1] S. Zhang *et al.*, “Physics-inspired machine learning for radiomap estimation: Integration of radio propagation models and artificial intelligence,” *IEEE Commun. Mag.*, vol. 62, no. 8, pp. 155–161, 2024.
- [2] Y. Zhou *et al.*, “Efficient transmission of radiomaps via physics-enhanced semantic communications,” *arXiv preprint arXiv:2501.10654*.
- [3] E. Elsayed, “Investigations on OFDM UAV-based free-space optical transmission system with scintillation mitigation for optical wireless communication-to-ground links in atmospheric turbulence,” *Opt Quant Electron*, vol. 56, p. 837, 2024.
- [4] R. Levie *et al.*, “RadioUNet: Fast radio map estimation with convolutional neural networks,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 4001–4015, 2021.
- [5] Y. Li *et al.*, “Geo2SigMap: High-fidelity RF signal mapping using geographic databases,” in *2024 IEEE Int. Symp. Dynamic Spectrum Access Networks (DySPAN)*, 2024, pp. 277–285.
- [6] Ç. Yapar *et al.*, “Overview of the first pathloss radio map prediction challenge,” *IEEE Open J. Signal Process.*, vol. 5, pp. 948–963, 2024.
- [7] 3GPP, “User Equipment (UE) radio access capabilities,” 3GPP, Tech. Rep. TS 38.306 V16.3.0 (2021-01), 2021.
- [8] “OpenStreetMap (OSM),” 2023. [Online]. Available: <https://www.openstreetmap.org/>
- [9] “Blender,” 2023. [Online]. Available: <https://www.blender.org/>
- [10] J. Hoydis *et al.*, “Sionna,” 2022. [Online]. Available: <https://nvlabs.github.io/sionna/>
- [11] 3GPP, “3GPP TS 38.214,” 3GPP, Tech. Rep. TS 38.214 V17.1.0 (2022-04), 2022.
- [12] O. Oktay *et al.*, “Attention U-Net: Learning where to look for the pancreas,” *arXiv preprint arXiv:1804.03999*.
- [13] “Geo2ComMap source code,” 2025. [Online]. Available: <https://github.com/geo2commap/Geo2ComMap>