# A Scalable and Generalizable Pathloss Map Prediction

Ju-Hyung Lee, *Member, IEEE*, and Andreas F. Molisch, *Fellow, IEEE*

***Abstract*— Large-scale channel prediction, *i.e.*, estimation of the pathloss from geographical/morphological/building maps, is an essential component of wireless network planning. Ray tracing (RT)-based methods have been widely used for many years, but they require significant computational effort that may become prohibitive with the increased network densification and/or use of higher frequencies in B5G/6G systems. In this paper, we propose a data-driven, model-free pathloss map prediction (PMP) method, called PMNet. PMNet uses a supervised learning approach: it is trained on a limited amount of RT data and map data. Once trained, PMNet can predict pathloss over location with high accuracy (an RMSE level of $10^{-2}$) in a few milliseconds. We further extend PMNet by employing transfer learning (TL). TL allows PMNet to learn a new network scenario quickly ($\times 5.6$ faster training) and efficiently (using $\times 4.5$ less data) by transferring knowledge from a pre-trained model, while retaining accuracy. Our results demonstrate that PMNet is a scalable and generalizable ML-based PMP method, showing its potential to be used in several network optimization applications.**

***Index Terms*— Pathloss map prediction, ray tracing, machine learning, computer vision, transfer learning, network optimization, digital twin, 6G.**

## I. Introduction

DIGITAL twin (DT) network is emerging as a key enabler for the artificial intelligence (AI) and machine learning (ML)-driven design, simulation, and optimization of 6G systems [3], [4]. A DT network is a dynamic, digital replica of a real-world network environment, providing real-time, accurate reflections of physical network scenarios. It can be used for a variety of applications, including dynamic resource allocation, beam management, and localization using ML-based PMP, which demands quick adjustments for new scenarios. However, implementing DT is challenging in 6G networks, which are characterized by increased deployment density, complex distributed architectures, and high-frequency operation in millimeter wave (mmWave) and terahertz (THz) bands.

Individually and taken together, these developments necessitate dramatically faster large-scale channel prediction methods.[1] Since traditional ray tracing (RT) tools are too slow for the repeated runs required in such DT implementation processes, there is a strong need for new, accurate, and fast methods for channel prediction over a large-scale area (*e.g.*, campus or city-map scale).

Several works have addressed this need by channel prediction using powerful ML techniques. These works use ground-truth channel data (from RT simulations or real channel measurements/soundings campaigns) to train neural networks (NNs). This eventually provides an accurate and fast prediction of channel information (*e.g.*, received power, delay, angles, and so on) for a certain area, a technique called ML-based site-specific radio propagation modeling.

Still, these ML-based approaches use supervised learning, meaning they are trained to solve a specific network scenario with a certain labeled dataset. In other words, the models may need to be rebuilt for a new network scenario, *e.g.*, different map scales, environmental aspects, and/or network configuration - a process that can be time-consuming and expensive. This creates a need for a method that can furthermore transfer knowledge of propagation channels across different network scenarios and environments.

### A. Related Works

Due to the high cost and complexity of field measurements with channel sounders, most cellular deployment planning has long replaced channel measurements with *electromagnetic (EM) simulation-based* approaches, such as RT [5], [6] and ray launching [7] simulation.[2] Over the past 30 years, the efficiency and accuracy of RT have improved significantly [9], thanks to the prevalence of GPUs (graphic processing units) that efficiently facilitate RT tasks.

However, due to the factors mentioned above (such as the need for more detailed environmental consideration at higher

Manuscript received 6 December 2023; revised 20 February 2024 and 3 July 2024; accepted 27 August 2024. This work was supported by NSF under Project 2133655 and Project 2008443. An earlier version of this paper was presented in part at the 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023) [DOI: 10.1109/ICASSP49357.2023.10095257] and in part at the 2023 Global Communications Conference (Globecom 2023) [DOI: 10.1109/GLOBECOM54140.2023.10437562]. The associate editor coordinating the review of this article and approving it for publication was C. Han. *(Corresponding author: Ju-Hyung Lee.)*

Ju-Hyung Lee was with Ming Hsieh Department of Electrical and Computer Engineering, University of Southern California (USC), Los Angeles, CA 90007 USA. He is now with Nokia, Sunnyvale, CA 94085 USA (e-mail: juhyung.lee@outlook.com).

Andreas F. Molisch is with the Ming Hsieh Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA 90089 USA (e-mail: molisch@usc.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TWC.2024.3457431.

Digital Object Identifier 10.1109/TWC.2024.3457431

[1]The word "channel prediction" is often used for two different problems: (i) computation of the propagation channel at a particular location based on maps of the environment, and (ii) temporal prediction of the channel (often for a mobile device moving on a trajectory), based on measurements in the immediate past. This paper only considers the former case.

[2]Our research utilizes RT simulations via Wireless Insite to investigate path loss. This simulation program has been validated against real-world measurements, e.g., in [8] showing an acceptable error margin.

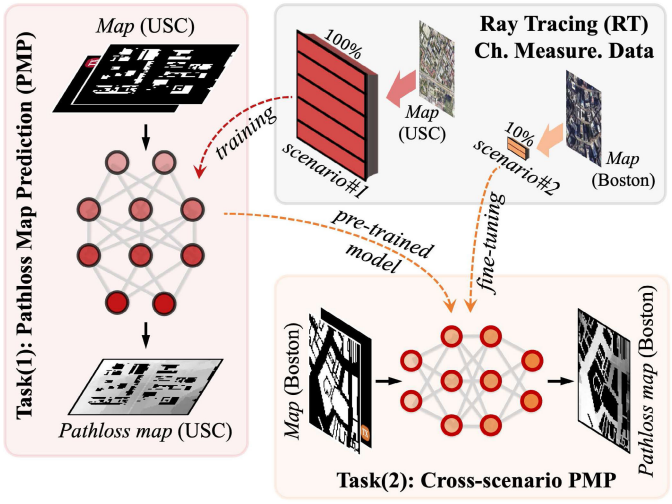



Fig. 1. Overview of the pathloss map prediction (PMP) task and the cross-scenario PMP. The input Map feature includes the transmitter (TX) location.

frequencies and the need for fast simulations with higher deployment density), RT simulations are too computationally intensive for large-scale network deployment in 6G systems. Consequently, simplified *model-based* approaches like the dominant path model [10], or fine-tuning of generic pathloss models (*e.g.*, 3GPP path gain model) with limited measurement data [11], [12] have been proposed over the years. Nevertheless, these approaches have gained limited acceptance by network operators due to their insufficient accuracy in predicting the propagation characteristics of signals in complex environments.

In recent years, supervised ML has been applied to solve a variety of challenging problems in wireless communication, including channel measurement/prediction for 6G networks. Such an *ML-based* approach can be trained on a map of the environment (topology/morphology) and a relatively small set of measurement data to learn how to provide a virtual replica (*e.g.*, DT) of a large-scale network environment in real-time while accurately modeling the behavior of channel characteristics.

On the one hand, models like WiNeRT [13] and NeRF2 [14] are specifically developed to predict detailed channel information (*e.g.*, power, delay, and angle information) of each multi-path component (MPC) between TX and receiver (RX) with the input of detailed information, including spatial configuration and wireless configuration parameters. These models are particularly well-suited for applications in small-scale indoor areas, where high-detailed channel prediction is required (*e.g.*, indoor sensing).

On the other hand, models like RadioUNet [15] and FadeNet [16] aim to predict the path gain, received power, or coverage for TX-RX in a given area with the input of a building map. These models are designed for large-scale channel prediction, where fast operation is essential (*e.g.*, network optimization).

In particular, several state-of-the-art works, such as Agile [17], PPNet [18], and PMNet [1], are pushing the boundaries of predictive accuracy and computational efficiency for large-scale channel prediction (*e.g.*, radio environment map estimation), as evidenced by their performance in ML competitions such as the *RadioMap Prediction Challenge* (see details in [19]). This highlights the applicability and importance of large-scale channel prediction in evolving wireless network optimization, which aligns with our research direction.

### B. Contributions

This paper proposes a scalable and generalizable channel prediction approach specifically designed for large-scale channel prediction, called PMP task. Our contributions can be summarized as follows:

- We design a PMP-oriented NN architecture, called PMNet, by leveraging computer-vision techniques, generating highly accurate channel prediction results for a given map in few milliseconds. PMNet achieves the best channel prediction accuracy compared to two baselines: a model-based scheme (*3GPP-UMi* model [20]) and another ML-based scheme (*RadioUNet* [15]) (see **Table V** in Sec. IV) and also in different PMP datasets. PMNet achieved *1st*-rank in the ICASSP 2023 Radio Map Prediction Challenge [19].[3]
- We build three sets of real-world channel datasets using a RT simulation tool, *i.e.*, *Wireless Insite*, for training and evaluation, which reflects different network scenarios (*e.g.*, different map scale, environment, and network configuration) in two different light urban environments (the USC and UCLA campuses) and a metropolitan area (the Boston area), see **Table I** in Sec. III.
- We propose a method of predicting pathloss in unseen network scenarios by using transfer learning (TL) with a pre-trained model. We prepare three pre-trained models for TL: VGG16 [22] and two pre-trained PMNet models trained with 3GPP prediction results and RT simulation results, respectively, and quantitatively and qualitatively evaluate their accuracy (see **Table IX** and **Fig. 8** in Sec. V).
- We empirically demonstrate that our PMNet pre-trained model has generalization capability for different network scenarios, adjusting to new network scenarios $\times 5.6$ faster and using $\times 4.5$ less data than a baseline model without TL, while still achieving high accuracy of an RMSE of $10^{-2}$ level (see **Fig. 6** and **Table. VIII** in Sec. V).
- We release source code for the experiments to promote reproducible ML research in wireless communication.[4]

### C. Paper Organization

The rest of the paper is organized as follows: Sec. II presents the background on two important concepts: (1) *ray tracing simulation*, which is used to generate ground-truth channel information for training and evaluation; and (2) *transfer learning*, which enables us to transfer the knowledge learned

[3]In this competition, PMNet demonstrated its high accuracy in the PMP task on a different dataset [21], which featured a different map scale, network configuration, and was generated by a different RT simulation tool, *i.e.*, *WinProp*, highlighting PMNet's generalization capability.

[4]https://github.com/abman23/PMNet

from a source task/dataset to a new task/dataset (*e.g.*, unseen network scenario). After introducing our dataset based on real geographical maps in Sec. III, Sec. IV introduces the PMP task and our proposed NN architecture (PMNet) for this channel prediction task. We also present the training and evaluation process, as well as simulation results. Then, Sec. V presents our approach for efficiently learning and predicting channels in unseen network environments by transferring the pre-trained knowledge from other networks. We provide extensive experimental results and quantitative and qualitative performance analysis, followed by concluding remarks in Sec. VI.

*Notation:* Throughout this paper, we use the normal-face font to denote scalars and the boldface font to denote vectors. We use $P(\cdot)$ and $P(\cdot|\cdot)$ to represent a marginal probability distribution and conditional distribution, respectively. We also use $\|\cdot\|$ to denote the $L^2$-norm, which is an Euclidean norm. $\mathcal{N}(\mu, \sigma)$ denotes the normal distribution with mean $\mu$ and standard deviation $\sigma$.

## II. Background

To provide a comprehensive understanding of our work, it is essential to cover Pathloss, Ray Tracing Simulation, and Transfer Learning, as these areas are integral to our methodology and analysis.

### A. Pathloss

The link gain between a TX at location $\boldsymbol{q}_{\mathrm{TX}}$ and an RX at location $\boldsymbol{q}_{\mathrm{RX}}$ at time $t$ and frequency $f$ can be expressed as follows:

$$|h(t, f, \boldsymbol{q}_{\mathrm{TX}}, \boldsymbol{q}_{\mathrm{RX}})|^2 = \frac{P_{\mathrm{RX}}(t, f, \boldsymbol{q}_{\mathrm{RX}})}{P_{\mathrm{TX}}(t, f, \boldsymbol{q}_{\mathrm{TX}})} \tag{1}$$

where $P_{\mathrm{RX}}$ and $P_{\mathrm{TX}}$ are received and transmitted power, respectively. This link gain includes the effects of antenna gains at TX and RX; when isotropic antennas are used, it becomes identical to the channel gain. It exhibits variations in time and/or location due to small-scale fading, shadowing, and large-scale distance changes. Averaging over small-scale fading removes (under certain circumstances, see [23, Ch. 7]) the dependence on frequency and time, providing the *path gain* (PG) that can be written as a function of only the large-scale distance changes:

$$\mathrm{PG}(\boldsymbol{q}_{\mathrm{TX}}, \boldsymbol{q}_{\mathrm{RX}}) = \frac{1}{T_{\mathrm{S}}} \frac{1}{B_{\mathrm{S}}} \int_{T_{\mathrm{S}}} \int_{B_{\mathrm{S}}} |h(t, f, \boldsymbol{q}_{\mathrm{TX}}, \boldsymbol{q}_{\mathrm{RX}})|^2 \, df \, dt. \tag{2}$$

Here, $T_{\mathrm{S}}$ and $B_{\mathrm{S}}$ denote the stationary-time and -bandwidth, respectively. The path gain can be represented as the sum of the powers of the $N$ MPCs, as discussed further in Sec. III-A. For later reference, we note that the pathloss is the inverse of the path gain (or the sign-flipped value when expressed in dB).

### B. Ray Tracing (RT) Simulation

RT is an approximate method for modeling the propagation of electromagnetic waves in wireless communication scenarios. It works by tracing the paths of individual rays as they propagate through the environment, whose features are represented in a geographical database. The rays are reflected, deflected, and scattered by the objects in the environment, with the various interaction processes computed according to high-frequency approximations, namely (most commonly) Snell's laws for specular reflection and transmission, uniform theory of diffraction (UTD) for diffraction, and Kirchhoff scattering theory for diffuse scattering [23, Ch. 4].[5] The RT tool simulates radio wave propagation deterministically based on physical laws, offering site-specific radio propagation modeling, in contrast to stochastic wireless channel models (*e.g.*, 3GPP standardized channel model).

In this paper, we employ a commercial RT tool, *Wireless Insite* from Remcom [7] for all RT simulations, both because of its user-friendliness and the fact that its accuracy has been compared against a number of channel sounder measurements [6], [24], [25]. RT can be used to predict channel information, such as received signal strength, delay, and angles, in a variety of wireless environments, both indoor and outdoor. The accuracy of RT simulations depends on various factors, such as the complexity of the environment, the accuracy of the geographical database, and the carrier frequency. The channel information obtained from the RT can be utilized, *inter alia*, for various network optimization tasks, including base station (BS) deployment planning, BS parameter optimization, as well as beam management and localization.

### C. Transfer Learning (TL)

TL is a machine learning technique that leverages a pre-trained model on a new task, significantly reducing the amount of data and training time required for new scenarios. This approach is particularly advantageous when there is limited data available for the new task or when the new task shares similarities with a previously learned task. By utilizing knowledge from a related task, TL can enhance model performance, expedite training processes, and mitigate overfitting, especially in data-constrained environments.

For instance, a model pre-trained on image classification tasks can be effectively repurposed for object detection or semantic segmentation. This reuse is possible because the model has already learned useful *feature representations* from a large and diverse dataset, enabling it to adapt more efficiently to new, related tasks.

One of the most popular pre-trained models is VGG16 [22], which is trained on more than a million images from the ImageNet database for image classification. VGG16 has demonstrated its versatility by being reused to improve performance in various tasks, including semantic segmentation [26] and object detection [27]. These applications showcase the model's ability to transfer learned features, thus enhancing performance in new domains with minimal additional training.

However, it is important to note that the effectiveness of TL depends on the *similarity* between the pre-trained task and the

[5] RT can be implemented via image-theory-based RT, or as ray launching. We will henceforth use the expression RT for both those methods.

target task.[6] The transferability of deep feature representations decreases as the discrepancy between the pre-trained task and the target task increases [28]. In other words, the further apart the task is, the less transferable the knowledge. One example is catastrophic forgetting, which is a phenomenon that can occur when fine-tuning a pre-trained model on a new task, resulting in a loss of previously acquired knowledge [29].

Research has shown that well-generalized models, particularly those with excellent pre-training performance [30], have the potential to require minimal fine-tuning or even none at all (*e.g.*, zero-shot learning) for new tasks [31]. These suggest the importance of selecting a pre-trained model suitable for the target task.

## III. Dataset

In this section, we discuss the dataset preparation process for our pathloss map datasets, reflecting real-world network scenarios in USC, UCLA, and Boston areas.

We obtained the ground-truth channel data using the commercial RT tool *Wireless Insite* [7], which takes into account the geographical and morphological features of the propagation environment. We then pre-processed the data (*e.g.*, interpolation and data augmentation) to prepare the ground-truth pathloss map.

### A. Channel Data

*1) RT Simulation:* As discussed in Sec. II-A, RT emulates the behavior of each MPC between TX and RX, following physical principles including the free-space power loss and interaction with different interacting objects (IOs). This allows us to compute for each MPC the information of complex amplitude $a$, directions of departure $\Omega$ and arrival $\Psi$, and delay $\tau$. The contribution of $m$-th MPC can be expressed as [32]:

$$h_m(t,\tau,\Omega,\Psi) = a_m \delta(\tau-\tau_m)\delta(\Omega-\Omega_m)\delta(\Psi-\Psi_m), \quad (3)$$

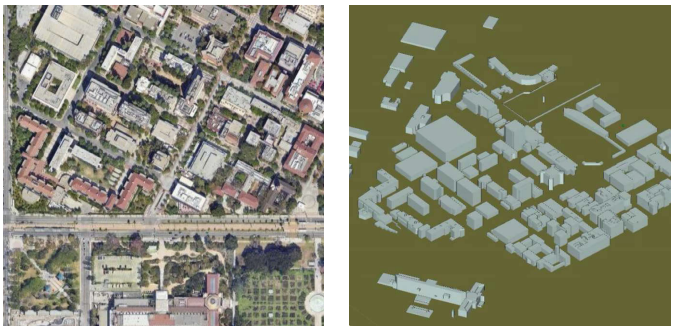
where the dependence of $\Omega, \Psi, \tau, a$ on $t$ is not written explicitly on the r.h.s. The sum of contributions from all MPCs is given by

$$h(t,\tau,\Omega,\Psi) = \sum_{m=1}^{N} h_m(t,\tau,\Omega,\Psi). \quad (4)$$

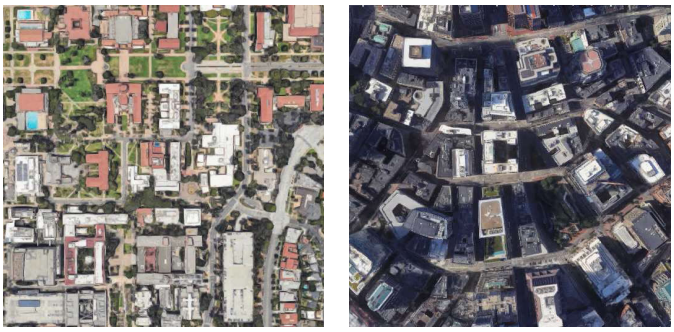
Since $\Omega, \Psi, \tau, |a|$ are constant over a stationarity-time and bandwidth, while $\arg(a)$ varies over many periods of $2\pi$, and assuming isotropic antennas at TX and RX (so that $\Omega, \Psi$ do not matter), the path gain averaged over the small-scale fading can be computed from (2) as

$$\mathrm{PG} = \sum_{m=1}^{N} |h_m(\tau,\Omega,\Psi)|^2 = \sum_{m=1}^{N} |a_m|^2. \quad (5)$$

[6]The wireless community has long classified environments—rural, suburban, urban, and metropolitan—based on their channel characteristics, a practice dating back to the COST 207 models of the mid-1980s. Despite the subjective nature of these categories and the lack of specific numerical criteria, their differentiation by factors like population density and infrastructure complexity is widely accepted for assessing wireless signal propagation and network performance and its *similarity*.



(a) USC campus (Map) (b) USC campus (Geometry in *Wireless Insite*)



(c) UCLA campus (Map) (d) Boston (Map)

Fig. 2. Map of USC, UCLA, and Boston used in RT simulation. Fig. 2a is imported and converted to Fig. 2b. The ground-truth pathloss map over the USC campus is then obtained using *Wireless Insite* RT simulation and pre-processing (*e.g.*, interpolation, gray conversion, and data augmentation).

Note that our pathloss map uses the information of path gain (in [dB]) while other information on angles and delay is not needed (though this information can be used for further applications, *e.g.*, beamforming algorithms).

Thus, $P_{\mathrm{RX}}$ (in [dBm]) can be expressed as a function of $P_{\mathrm{TX}}$ (in [dBm]) as follows:

$$P_{\mathrm{RX}} = P_{\mathrm{TX}} + \mathrm{PG}. \quad (6)$$

Note that we set $P_{\mathrm{TX}} = 0$ [dBm] in our RT dataset to simplify the analysis, which makes $P_{\mathrm{RX}}$ in [dBm] equal to PG in [dB].

To generate a ground-truth (labeled) dataset that simulates real-world network scenarios, we conduct *Wireless Insite* RT simulations on the geographical and morphological maps of the University of Southern California (USC) campus, the University of California, Los Angeles (UCLA) campus, and the Boston area. Both campus areas are in Los Angeles, CA, and exhibit a (light) urban build-up, with most buildings being five stories or less (with a few high-rises interspersed), gaps between buildings along the street canyons, and some open squares. The Boston area is in downtown of Boston, MA. It is a metropolitan area with multiple high-rises; its streets are *not* arranged along a rectangular grid. Each dataset has different network configurations and environmental characteristics (*e.g.*, map scale, and geographical features, such as vegetation). See Fig. 2 and Table I for more details.[7]

[7]It is worth noting that the simulations are performed at the sub-6 GHz band, which is the most widely used cellular band. Similar simulations can be performed in other frequency bands, such as the mmWave and THz bands, with minor adjustments to the parameters. However, at those high frequency bands, geographical data bases with higher resolution might be required for comparable accuracy.

TABLE I
PARAMETERS OF USC, UCLA, AND BOSTON DATASETS

| *Parameter* | *Dataset* | | |
|---|---|---|---|
| | USC | UCLA | Boston |
| Map scale | $880 \times 880$ [m$^2$] | $760 \times 760$ [m$^2$] | $553 \times 553$ [m$^2$] |
| Cropped map scale (per pixel) | $221 \times 221$ [m$^2$] ($0.86 \times 0.86$ [m$^2$]) | $225 \times 225$ [m$^2$] ($0.88 \times 0.88$ [m$^2$]) | $187 \times 187$ [m$^2$] ($0.73 \times 0.73$ [m$^2$]) |
| Terrain | ✓ | ✓ | ✓ |
| Buildings | ✓ | ✓ | ✓ |
| Foliage[a] | ✗ | ✗ | ✓ |
| Carrier frequency | 2.5 [GHz] | 3.0 [GHz] | 3.0 [GHz] |
| Transmit power | 0 [dBm] | 0 [dBm] | 0 [dBm] |
| TX antenna type[b] | Isotropic (vertical) | Half-wave dipole (vertical) | Half-wave dipole (vertical) |
| Total # of data/scene | 4754 | 3776 | 3143 |

[a]Our study assesses PMP accuracy across datasets varying in map scale, network configurations, and geography, focusing on how factors like foliage impact path loss predictions.

[b]Isotropic and half-wave dipole antennas provide almost identical radiation patterns within a certain angular extent. MPC induced outside of the angular extent does not contribute significantly to the link.

We stress that the goal of our work is the correct prediction of "ground-truth" pathloss by ML techniques. The pathloss obtained from the RT simulations might deviate from measured values due to inaccuracies of the database or inherent approximations of RTs. However, such deviations are irrelevant to the assessment of our ML methods, since they only impact what is used as "ground-truth" and not the prediction process itself. In other words, if the ground-truth is more accurate (similar to measurement results), our prediction inherently becomes more accurate as well.[8]

*2) 3GPP Model:* The 3GPP 38.901 channel model [20] (henceforth simply called the "3GPP model" for conciseness) is a widely used model for wireless system standardization that claims validity for frequencies spanning from 0.5 to 100 [GHz].

For the purposes of this paper, we only consider the 3GPP modeling of the pathloss, which follows the classical $\alpha - \beta$ model

$$\mathrm{PL}_{\alpha-\beta}(d) = 10\alpha \log_{10}(d) + \beta + S, \tag{7}$$

where $S \sim \mathcal{N}(0, \sigma_S)$ is a lognormally distributed random variable (with variance $\sigma_S$) representing the shadow fading, and $\alpha$, $\beta$, and $\sigma$ are parameters of the model that are based on measurement campaigns and that are different in different environments. Important for our later discussions, those parameters are also different depending on whether an unobstructed optical line of sight (LoS) exists between TX and RX or not.

Specifically, for urban environments, the following describes the path gain:

$$\mathrm{PG}_{\mathrm{UMi-LoS}} = \begin{cases} \mathrm{PL}_1, & (10[\mathrm{m}] \le d_{\mathrm{2D}} \le d_{\mathrm{BP}}) \\ \mathrm{PL}_2, & (d_{\mathrm{BP}} \le d_{\mathrm{2D}} \le 5[\mathrm{km}]) \end{cases} \tag{8}$$

$$\mathrm{PG}_{\mathrm{UMi-NLoS}} = \max(\mathrm{PG}_{\mathrm{UMi-LoS}}, \mathrm{PL}_3), \quad (10[\mathrm{m}] \le d_{\mathrm{2D}} \le 5[\mathrm{km}]) \tag{9}$$

[8]Thus, if our proposed PMNet can accurately predict/reproduce RT results when trained with RT data, it will also be able to do so for measurement data when trained with measurement data.

where the two-dimensional $xy$-distance is $d_{\mathrm{2D}}$ and the three-dimensional $xyz$-distance is $d_{\mathrm{3D}}$,

$$\begin{aligned} \mathrm{PL}_1 &= 32.4 + 21 \log_{10}(d_{\mathrm{3D}}) + 20 \log_{10}(f_c), \\ \mathrm{PL}_2 &= 32.4 + 40 \log_{10}(d_{\mathrm{3D}}) + 20 \log_{10}(f_c) \\ &\quad - 9.5 \log_{10}((d_{\mathrm{BP}})^2 + (h_{\mathrm{BS}} - h_{\mathrm{UT}})^2), \\ \mathrm{PL}_3 &= 22.4 + 35.3 \log_{10}(d_{\mathrm{3D}}) + 21.3 \log_{10}(f_c) \\ &\quad - 0.6(h_{\mathrm{UT}} - 1.5). \end{aligned} \tag{10}$$

Here, the breakpoint distance is $d_{\mathrm{BP}} = 2\pi h_{\mathrm{BS}} h_{\mathrm{UT}} f_c / c$ where $f_c$ is the center frequency in [Hz] and $c = 3.0 \times 10^8$[m/s] is the speed of light. The antenna heights at the TX (*e.g.*, base station), $h_{\mathrm{BS}}$, and the RX (*e.g.*, user terminal), $h_{\mathrm{UT}}$, are set to 1.5 [m] and 10 [m], respectively. Note that the model differs for LoS and non-LoS (NLoS) situations.

This model is employed as one of our baselines for the prediction (see Sec. IV-E). While the 3GPP model also models shadowing, it incorporates it as *stochastic* variations that cannot be related to particular map features; we therefore omit them for the purposes of this paper.

### B. Pre-Processing

The raw numeric data from the RT simulation is pre-processed using *gray conversion* and *interpolation* methods to generate the ground-truth pathloss map, *data augmentation* methods to create an increased amount of labeled data, and *sampling* methods to divide them into training and testing sets.

*1) Gray Conversion:* To generate the pathloss map, we begin by converting the received power $P_{\mathrm{RX}}$ (in [dBm]) (or the path gain PG in [dB]) into grayscale between 1 and 255 using *Min-Max* normalization, with the minimum value of $-254$ [dBm] and the maximum value of 0 [dBm]. While the upper value is higher than physically reasonable, this pair of values was chosen for convenience to have a 1 [dBm] per gray value step mapping. A smaller (or larger) step size does not have a significant impact on the prediction performance.

The gray value 0 is filled at pixels of building area, which is not our region-of-interest (RoI), while, for our RoI, each pixel is filled with gray values between 1 and 255, which corresponds to $P_{\mathrm{RX}}$. Then, the pathloss map is generated after

scaling the considered map scale into a $256 \times 256$ gray image. Note that the image size ($256 \times 256$) has nothing to do with the grayscale ($0 - 255$).

*2) Interpolation:* Since the RT simulations are carried out over a discrete set of RX locations, and it is computationally challenging to gather the channel information for every available RX location, there is missing channel information in a few pixel locations. To fill the missing part of the pathloss map, we utilize *bilinear interpolation*, which approximates the missing value with a weighted sum of the gray values of the adjacent locations.

*3) Data Augmentation:* Typically, a larger dataset leads to improved performance of NN training. In other words, the larger the data set, the better the outcome. We thus use two augmentation methods - cropping and rotation - to increase the size of our data set.

The entire map data is cropped into images of about a quarter of the size, taking TX as an anchor point. This augments the size of the dataset by a factor of 96. The image is first cropped as a $64 \times 64$ size image and then upsampled to a $256 \times 256$ size image. Note that some cropped images, not including any TX, are skipped since the TX location will be used as our second input feature. After cropping, the image sets are rotated by $90°$, $180°$, and $270°$, thus increasing the size of the dataset by a further factor of $4$.

*4) Sampling:* In the training and testing of PMNet on the pathloss map dataset, we employ an exclusive division scheme. Specifically, $90\%$ and $10\%$ of images are randomly split into the training and validation set, while the images from the same geographical map belong exclusively to either the training or the validation set. This approach is taken to enhance the generalization performance of PMNet.

## IV. Pathloss Map Prediction

### A. Task (1): Pathloss Map Prediction

We now formulate the prediction task in ML nomenclature. A domain (*i.e.*, wireless channel prediction) is composed of a feature space $\mathcal{X}$, where $x \in \mathcal{X}$. Given the domain, a PMP task is defined as $\mathcal{T} = \{\mathcal{Y}, P(y|\mathbf{x})\}$, which is composed of a label space $\mathcal{Y}$, where $y \in \mathcal{Y}$. Given the task, a dataset is defined as $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$, which is a collection of $|\mathcal{D}| = \mathcal{N}$ channel data that belong to a domain with a task $\mathcal{T}$.

For the PMP task, $\mathcal{X}$ consists of (1) a building map (including terrain, building, and/or foliage) and (2) a TX location and $\mathcal{Y}$ is a Pathloss map. The goal of the PMP task $\mathcal{T}$ is to find a predictive function $f(\cdot)$, which accurately predicts $\mathcal{Y}$ for a given $\mathcal{X}$. It is worth noting that integrating RoI (denoted as $\mathcal{A}^\star$) segmentation with path gain prediction simplifies the PMP task and eliminates the need for separate pre- or post-processing steps for the RoI segmentation for each map. Additionally, this integration helps NN better understand the different IOs in a given building map.

In a nutshell, the PMP task is to predict the pathloss/path gain (and received power $P_{\mathrm{RX}}$ using simple normalization) at RX locations $\boldsymbol{q}_{\mathrm{RX}}$ given TX location $\boldsymbol{q}_{\mathrm{TX}}$ in RoI $\mathcal{A}^\star$. This channel prediction task exploits *site-specific* geographical information, focusing on the large-scale effects in the channel.

We employ a supervised ML method for the PMP task. We train the model on a dataset of RT channel for an area of $\mathcal{A}$, such as the USC dataset in Table I; see Fig. 3 for an overview of the ML-based PMP approach.

### B. Network Architecture

In this subsection, we present the design process of our proposed PMP-oriented NN architecture, referred to as *PMNet*. Our design principles are summarized as follows: (1) several state-of-the-art techniques in the field of image processing are carefully selected and tested, (2) some essential techniques are selected following the concept of ablation study, and (3) the NN with selected techniques is optimized with extensive trials.

*1) Design Choices:* In the PMP task, the NN is required to perform image segmentation to identify the RoI and predict received power within the RoI, while accounting for complex wireless propagation physics. To accomplish this, our proposed PMNet is designed based on such methods, *Encoder-Decoder* and *Atrous convolution*.

*2) Encoder-Decoder:* Encoder-Decoder networks are a widely applied architecture for many computer vision tasks, *e.g.*, object detection [33], human pose estimation [34], and semantic segmentation [35], [36], [37]. The encoder-decoder architecture allows to learn a lower-dimensional representation from a higher-dimensional dataset and utilize the learned representation for various tasks. However, as the encoder shrinks the input feature maps, it may lose essential information, leading to a *bottleneck problem*. Several architectures, including U-Net [38], address the bottleneck problem by adding *skip connections* between the encoder and the decoder parts. Skip connections allow the decoder to access feature maps from the encoder, which helps to propagate context information to higher-resolution layers.

*3) Atrous Convolution: Receptive field* of a convolutional layer is the region of the input feature map that contributes to the output feature map at a given location. The size of the receptive field is determined by the resolution of the input feature map and the field-of-view (FoV) of the filter. There is a logarithmic relationship between the localization accuracy of a model and the size of its receptive field. This means the receptive field size should be sufficient if the given dataset and task are observed with wide FoV. A standard convolutional filter detects a particular feature by sliding over the input feature map, resulting in the output feature map seeing only the adjacent part of the input feature map. In terms of computational complexity, having a wide receptive field with the standard convolutional filter is expensive. Thus, broadly speaking, the receptive field of the standard convolution filter is somewhat narrow, seeing only little context.

Atrous convolution, also known as dilated convolution, is a technique that addresses this limitation [39]. It allows capturing a larger context with a wider FoV by modifying the standard convolution operation. For the two-dimensional case, atrous convolution is applied over the input feature map $f$ to produce the output feature map $g$ at location $\{i, j\}$ using the convolution filter $w$. This operation can be expressed as

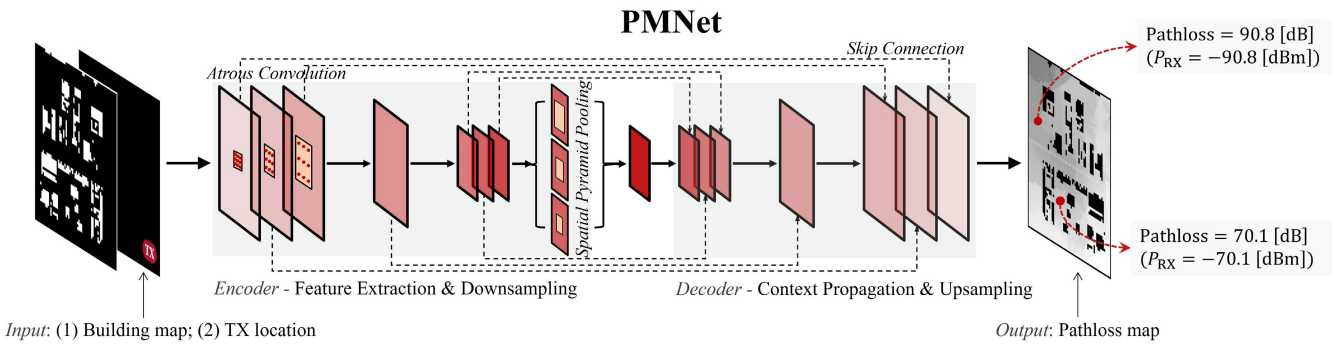



Fig. 3. Overview of the PMP task and the PMNet architecture.

follows:

$$g_{\{i,j\}} = \sum_{m=1}^{k}\sum_{n=1}^{k} f_{\{i+rm,j+rn\}} w_{\{m,n\}}. \tag{11}$$

Here, $k$ represents the kernel size, and $r$ is the atrous rate, which determines the stride level. Notably, the atrous rate $r$ allows to adaptively control the FoV of the filter. For example, an atrous rate of $r = 2$ doubles the FoV of the filter, while an atrous rate of $r = 3$ triples it. The standard convolution can be seen as a special case of (11) where $r = 1$.

In the context of the PMP task, the encoder-decoder architecture of PMNet facilitates efficient context propagation from the encoder to the decoder, while atrous convolution enables it to handle scale variations and capture broader context in map data, setting it apart from other UNet-based networks [15], [16], [17], [18]. The combination of these two features enables PMNet to efficiently and accurately predict pathloss maps, while also accounting for complex wireless propagation physics.

*4) Design Parameters:* PMNet architectures are composed of a stack of *ResLayers*, each containing multiple residual blocks [40]. These ResLayers can be configured with varying numbers of blocks, atrous rates, multi-grids, and output strides. These elements are summarized as follows:

- *Number of blocks*: The number of residual blocks in a ResLayer controls the complexity and depth of the network. Increasing the number of blocks may improve the accuracy of the model, but it also increases the computational cost.
- *Atrous rates*: Atrous rates control the spacing between the convolutions in a ResLayer. Larger atrous rates allow the network to capture more larger spatial contexts in the PMP task.
- *Multi-grids*: Multi-grids allow the network to capture multi-scale information from different levels of the CNN architecture.
- *Output stride*: The output stride of a ResLayer controls the ratio between the resolution of the input image and the output image's resolution. A higher output stride results in a lower-resolution output image. This can be useful to strike a balance between accuracy and speed.

Parameters are optimized through thorough simulations. Note that the impact of output stride in the PMP task is shown in Table IV in Sec. IV-E (*e.g.*, the case of $\frac{H}{8} \times \frac{W}{8}$), demonstrating the most substantial effect compared to other parameters. With these design choices and parameters, PMNet effectively predicts pathloss maps even for different channel datasets (*e.g.*, *RadioMapSeer* [21]). For an architectural overview, please refer to Fig. 3 and Table II. For more details, please see our source code repository.

### C. Training

Table III lists the hyper-parameters that are used for the training of PMNet. We implement the PMNet using PyTorch and use an NVIDIA GeForce RTX 3080 Ti GPU. For more stable training, we normalize the input values into $[0, 1]$ via scaling. During the training, we evaluate the PMNet by mean squared error (MSE) on the validation set at the end of every epoch. For testing, we use the parameters of PMNet with the best MSE score on the validation set. Consequently, the pathloss map for a given map can be generated within a few milliseconds after training.

### D. Evaluation

*1) Root Mean Square Error (RMSE):* RMSE is a widely used loss function in regression analysis and is used as the primary evaluation metric for this task. It measures the overall difference between the prediction $\hat{\boldsymbol{y}}$ and ground-truth $\boldsymbol{y}$ and quantifies the overall accuracy of the model. The formula for RMSE is:

$$\mathrm{RMSE}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\hat{y}_n - y_n)^2}, \tag{12}$$

where $\hat{y}_n \in \hat{\boldsymbol{y}}$ and $y_n \in \boldsymbol{y}$ denote predicted and ground-truth gray value (corresponding $P_{\mathrm{RX}}$) at the $n$-th pixel, respectively, and $N$ is the number of pixels in a pathloss map, *i.e.*, $256 \times 256$. The RMSE averaged over all samples is the primary evaluation metric for the PMP task.

*2) RoI Segmentation Error:* The RoI segmentation error, calculated using the intersection over union (IoU) metric, quantifies the accuracy of RoI and non-RoI area segmentation for all pixels in the ground-truth ($\{i, j\}$) and prediction ($\{\hat{i}, \hat{j}\}$) - that is calculated as follows:

$$\text{RoI Segmentation Err.} = \frac{\sum_i \sum_j \mathrm{Err}^{\mathrm{B}}{}_{\{i,j\}}}{\sum_i \sum_j \mathrm{Bld}_{\{i,j\}}}. \tag{13}$$

TABLE II
PMNET ARCHITECTURES AND PARAMETERS. ↓ AND ↑ REPRESENT THE DOWNSAMPLING AND UPSAMPLING LAYERS, RESPECTIVELY

| PMNet | | | | | |
|---|---|---|---|---|---|
| **Encoder** | | | **Decoder** | | |
| *#* | *Type* | *Output Size* | *#* | *Type* | *Output Size* |
| Input | Image | $2 \times 256 \times 256$ | Output | Image | $1 \times 256 \times 256$ |
| 1(↓) | Conv2d, MaxPool2d | $64 \times 65 \times 65$ | 1(↑) | Conv2d | $(128+2) \times 256 \times 256$ |
| 2 | ResLayer | $256 \times 65 \times 65$ | 2 | Conv2d | $(256+64) \times 65 \times 65$ |
| 3(↓) | ResLayer | $512 \times 33 \times 33$ | 3 | Conv2d | $(256+256) \times 65 \times 65$ |
| 4(↓) | ResLayer | $512 \times 17 \times 17$ | 4(↑) | ConvTranspose2d | $(256+256) \times 65 \times 65$ |
| 5 | ResLayer | $1024 \times 17 \times 17$ | 5(↑) | ConvTranspose2d | $(512+512) \times 33 \times 33$ |
| 6 | Conv2d, AdaptiveAvgPool2d | $512 \times 17 \times 17$ | 6 | Conv2d | $(512+512) \times 17 \times 17$ |

TABLE III
TRAINING CONFIGURATION AND HYPER-PARAMETERS FOR PMNET TRAINING

| ***Model*** | PMNet |
|---|---|
| ***Dataset*** (USC) | |
| Map | USC campus |
| Split for training (test) set | 90% (10%) of dataset |
| ***Hyper-parameter*** | |
| Learning rate (LR) | $10^{-3} \sim 5 \times 10^{-4}$ |
| LR gamma, step size | 0.5, 10 |
| Batch size | $16 \sim 32$ |
| Optimizer | Adam |
| # of of epochs | 50 |

Here, $\mathrm{Err}^{\mathrm{B}}{}_{\{i,j\}}$ and $\mathrm{Bld}_{\{i,j\}}$ are defined as:

$$\mathrm{Err}^{\mathrm{B}}{}_{\{i,j\}} = \begin{cases} 1, & \{i,j\} \in \mathcal{B} \text{ and } \{\hat{i},\hat{j}\} \in \mathcal{A}^\star \\ 1, & \{i,j\} \in \mathcal{A}^\star \text{ and } \{\hat{i},\hat{j}\} \in \mathcal{B} \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

$$\mathrm{Bld}_{\{i,j\}} = \begin{cases} 1, & \{i,j\} \in \mathcal{B} \\ 0. & \text{otherwise} \end{cases} \tag{15}$$

Within a given map, the non-RoI area, denoted as black (gray value 0), is represented by $\mathcal{B}$, while the RoI area, denoted as non-black (grayscale $1-255$), is represented by $\mathcal{A}^\star$. $\mathcal{B}$ and $\mathcal{A}^\star$ are complementary set within $\mathcal{A}$. $\mathcal{B}$ can include buildings, foliage, and/or small objects.

*3) Channel Prediction Error:* Channel prediction error directly evaluates path gain accuracy for pixels within the RoI area, evaluating power in [dBm] (or path gain in [dB]) unlike RMSE, which quantifies differences based on gray values.

To calculate channel prediction error, gray values within the RoI area of both the predicted and ground-truth pathloss maps are converted into corresponding received power values. The RMSE formula is then applied to these power values:

$$\mathrm{RMSE}(\hat{\boldsymbol{p}}, \boldsymbol{p}) = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\hat{p}_n - p_n)^2}, \tag{16}$$

where $\hat{p}_n \in \hat{\boldsymbol{p}}$ and $p_n \in \boldsymbol{p}$ represent the predicted and ground-truth $P_{\mathrm{RX}}$ at the $n$-th pixel, respectively. Channel Prediction Error is then computed by averaging $\mathrm{RMSE}(\hat{\boldsymbol{p}}, \boldsymbol{p})$ across all given samples.

## E. Simulation Result

*1) Training Optimization:* Table IV presents an ablation study to identify the factors that significantly contribute to PMNet's performance in the PMP task, such as *data augmentation* and *feature map size*.[9]

*2) Impact of Data Augmentation:* For the data augmentation, we do horizontal, vertical and diagonal flips. In other words, including the original images, we use the $\times 4$ number of images for training. Note that data augmentation has several advantages in general: first, it enhances the diversity of the training data by generating additional examples that capture various variations of the original data. Second, it reduces overfitting by exposing the model to a wider range of input patterns. Finally, data augmentation helps to make the model more robust to noise and variability in the input data. As shown in Table IV, it improves the performance of PMNet by 15.7% in terms of RMSE.

*3) Impact of Feature Map Size:* We analyze the performances of PMNet according to the size of the feature map, which is the output of the encoder. Table IV compares the results with the feature map sizes $\frac{H}{8} \times \frac{W}{8}$ and $\frac{H}{16} \times \frac{W}{16}$, where $H$ and $W$ are the height and width of an input image, respectively. To adjust the feature map size, we modify the strides of the convolution layers in the encoder. We employ the feature map size of $\frac{H}{8} \times \frac{W}{8}$ as the default option, because PMNet yields better performance with the feature map size of $\frac{H}{8} \times \frac{W}{8}$ than that of $\frac{H}{16} \times \frac{W}{16}$.

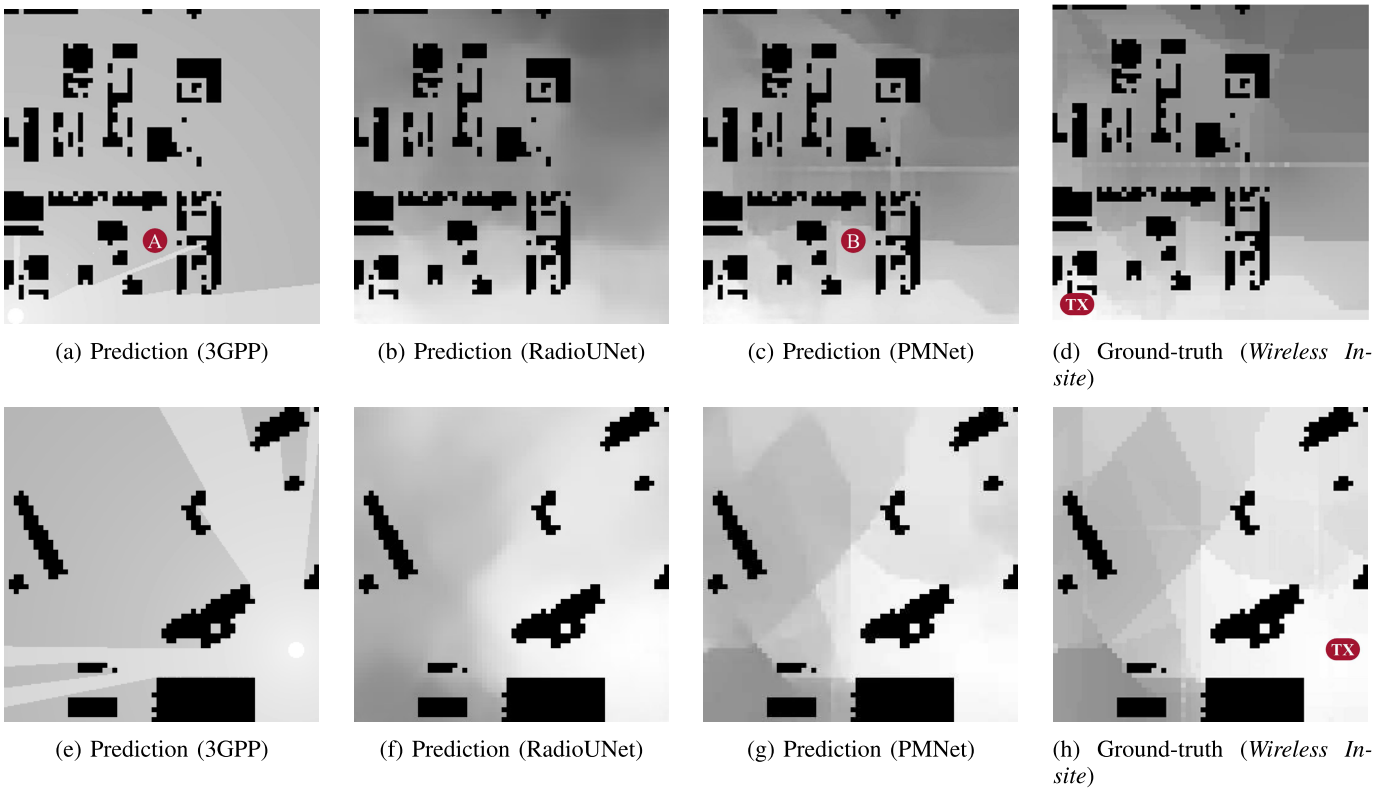
*4) Accuracy:* We compare the ML-based PMP with our proposed PMNet model to two other methods for the PMP task: a model-based approach, *3GPP*, and an ML-based approach, *RadioUNet*. All three methods produce a single-channel $256 \times 256$ image of the pathloss map as the output, given the input of a two-channel $256 \times 256$ image containing the geographical map and the TX location. Here are the details of these baseline methods:

1) **3GPP (with map info.)** As discussed in Sec. III-A.2, the 3GPP model determines the pathloss at a particular location based on the Euclidean distance and whether the link between the TX and RX is in LoS or NLoS. To ensure a fair comparison with other baselines, we utilize map information to determine the LoS or NLoS condition of

[9]Our extensive experiments tested other factors, such as different sampling methods, training loss functions, and additional input features (*e.g.*, TX distance heatmap), but these factors did not show a meaningful improvement to justify the additional complexity.

TABLE IV
ABLATION STUDY FOR PMNET TRAINING OPTIMIZATION. LOWER VALUES INDICATE BETTER PERFORMANCE

| *Case* | *Data Aug. (×4)* | *Feature Size* | *RMSE*↓ | *RoI Segmentation Err.*↓ | *Channel Prediction Err.*↓ |
|---|---|---|---|---|---|
| w/o Data-Aug. | ✗ | $\frac{H}{16} \times \frac{W}{16}$ | 0.01637 | 0.00263 | 0.01860 |
| w/ Data-Aug. | ✓ | $\frac{H}{16} \times \frac{W}{16}$ | 0.01259 | 0.00025 | 0.01403 |
| $\frac{H}{8} \times \frac{W}{8}$ | ✓ | $\frac{H}{8} \times \frac{W}{8}$ | 0.01057 | 0.00096 | 0.01175 |




(a) Prediction (3GPP) (b) Prediction (RadioUNet) (c) Prediction (PMNet) (d) Ground-truth (*Wireless Insite*)

(e) Prediction (3GPP) (f) Prediction (RadioUNet) (g) Prediction (PMNet) (h) Ground-truth (*Wireless Insite*)

Fig. 4. Comparison of the predicted pathloss map of 3GPP, RadioUNet, and PMNet. TX in ground-truth represents the TX location. The scenes are randomly selected, not cherry-picked.

specific pixels to the TX.[10] Note that it does not require any NN training as it is a model-based approach.

2) **RadioUNet** [15] is an ML-based PMP method that extends the UNet architecture by employing two UNets. Each UNet comprises 8 encoder layers with convolution, ReLU, and Maxpool layers, followed by 8 decoder layers with transposed convolution and ReLU layers. The encoders and decoders are concatenated, as in the original UNet architecture. Here, RadioUNet employs curriculum training to enhance training: in the first stage, the first UNet is trained for a specific number of epochs, with the second UNet frozen. In the second stage, the second UNet is trained using the two-channel input features and the output of the first UNet, effectively making it a three-channel input network.

3) **PMNet (Proposed)** is our proposed ML-based PMP method. This network employs several parallel atrous convolutions with different rates and the encoder-decoder network. The encoder consists of 6 ResNet-based layers. Each ResNet layer comprises several bottleneck layers consisting of convolution, batch normalization, max pooling, and ReLU. The decoder consists of 6 layers consisting of convolution, adaptive average pooling, ReLU, transposed convolution, and ReLU. Skip connections are used between encoders and decoders.

[10]The original 3GPP pathloss model uses a probabilistic model to determine LoS/NLoS condition at a particular distance. However, to ensure a fair comparison, we use here the deterministic LoS/NLoS condition determined from the map information in calculating the pathloss gain.

*5) Qualitative Analysis:* Fig. 4 shows the prediction results of the baselines. Recall that each pixel in the RoI corresponds to the predicted received power $P_{\mathrm{RX}}$ (or the path gain PG). Note that some pixel values in the ground-truth data appear noisy due to interpolation during the gray conversion process after RT simulation.

3GPP exhibits a substantial deviation from ground-truth obtained through RT simulation, highlighting the differences between how RT simulation and 3GPP model calculate a pathloss. Specifically, for RX locations with LoS conditions close to the TX, the results obtained using the 3GPP model approximately match the ground-truth data obtained from *Wireless Insite*. However, for RX locations farther from the TX

TABLE V
COMPARISON STUDY FOR PMP SCHEMES (3GPP, RADIOUNET, AND PMNET). LOWER VALUES INDICATE BETTER PERFORMANCE, AND THE LOWEST ERRORS ARE HIGHLIGHTED

| *Scheme* | *ML-based* | *RMSE*↓ | *RoI Segmentation Err.*↓ | *Channel Prediction Err.*↓ |
|---|---|---|---|---|
| 3GPP (with map info.) [20] | ✗ | 15.9451 | - | 17.5973 |
| RadioUNet [15] | ✓ | 0.02634 | 0.00840 | 0.01249 |
| PMNet | ✓ | **0.01057** | **0.00096** | **0.01175** |

TABLE VI
NUMERICAL RESULTS OF PMNET ON AN UNSEEN NETWORK SCENARIO. PMNET WAS TRAINED ON THE USC DATASET AND EVALUATED ON THE UCLA AND BOSTON DATASET

| *Case* | *Model* | *Train Data* | *Eval. Data* | *RMSE*↓ | *RoI Segmentation Err.*↓ | *Channel Prediction Err.*↓ |
|---|---|---|---|---|---|---|
| Vanilla | PMNet | USC | USC | 0.01057 | 0.00096 | 0.01175 |
| Cross-scenario (UCLA) | PMNet | USC | UCLA | 0.19146 | 0.03925 | 0.21700 |
| Cross-scenario (Boston) | PMNet | USC | Boston | 0.25842 | 0.04602 | 0.32436 |

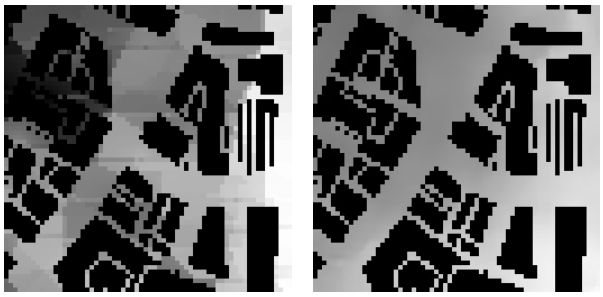
or under NLoS conditions, the 3GPP model exhibits significant discrepancy from the ground-truth data. It is worth noting that the 3GPP pathloss model does not provide results for near-field within a link distance of 10 meters; so, we arbitrarily set the power in the near-field area to gray value 255, which does not introduce significant errors. The 3GPP pathloss model is a simplified model that does not account for the complex wireless propagation physics of reflection, diffraction, and scattering (highlighted in A). Instead, it relies solely on two models for LoS and NLoS locations, respectively, and only considers link distance and carrier frequency. This simplified approach inevitably leads to significant inaccuracies in the pathloss prediction.

RadioUNet demonstrates impressive RoI segmentation results, while its channel prediction outputs appear somewhat blurry. It is worth noting that RadioUNet conducts curriculum-based training with 50 epochs each in the first and second stages, utilizing the same training/validation set as PMNet, which is trained with a total of 50 epochs.

PMNet, on the other hand, achieves notable results for both RoI segmentation and channel prediction. As highlighted in B, PMNet effectively captures the intricate wireless propagation physics of reflection, diffraction, and scattering. This can be attributed to PMNet's ability to incorporate a broader contextual understanding of the environment, enabling it to capture the representation of wireless propagation physics in the surrounding environment.

*6) Quantitative Analysis:* Table V compares our proposed PMNet model to the model-based 3GPP method and the ML-based RadioUNet method in terms of three accuracy metrics for the PMP task: RMSE, RoI segmentation error, and channel prediction error. Note that the ground-truth dataset is made by RT simulation; therefore, the error shows the difference between a scheme and the RT simulation.

The model-based 3GPP method has inferior results compared to ML-based methods, which can be explained by the oversimplifications inherent in this model, as discussed above. While our proposed PMNet model achieves the best score on all three metrics, another ML-based PMP method, RadioUNet, also achieves high accuracy ($RMSE \leq 0.03$). This result highlights the capability of ML-based PMP approaches to learn a representation of the wireless propagation physics implicit in the ground-truth RT channel data.



(a) Ground-truth (*Wireless Insite*) (b) Prediction (PMNet)

Fig. 5. Prediction results of PMNet on an unseen network scenario (*i.e.*, cross-scenario evaluation). The model is trained on the USC dataset and evaluated on the Boston dataset.

## V. TRANSFERABLE PATHLOSS MAP PREDICTION

### A. Challenge: PMP for Unseen Network Scenario

As demonstrated in the previous section, PMNet exhibits high accuracy of the PMP task for a given dataset. However, minimizing re-training efforts for new network scenarios remains a challenge. To evaluate PMNet's generalizability across different scenarios, we conducted a cross-scenario evaluation, testing PMNet trained on USC data on the Boston dataset.

As shown in Fig. 5 and Table VI, the PMNet achieves the RoI segmentation error on the order of $10^{-2}$ and the channel prediction error on the order of $10^{-1}$ in a new scenario. Such deterioration is due to differences in network configuration and environmental characteristics between the two scenarios (*e.g.*, different map scales and geographical features). This highlights the need for further development to improve PMNet's performance across different network scenarios, a task we refer to as *cross-scenario PMP*.

### B. Task (2): Cross-Scenario PMP

To enable better performance, we now allow cross-scenario PMP to *improve* the model trained on a different network scenario through training with a *reduced-size* training in the new scenario. This will allow the network to adapt to the new scenario with less time and resource effort, while maintaining high accuracy. To address this challenge, we leverage *transfer learning* (TL).

*1) Approach: Transfer Learning:* TL is an ML technique that allows knowledge transfer from one task or dataset to another, reducing the amount of data and training time required for new scenarios. In the context of cross-scenario PMP, we can transfer the knowledge from the source scenario, which learns a predictive function $f_S(\cdot)$ from a source dataset $\mathcal{D}_S$ (*e.g.*, USC), to the target scenario, which learns a predictive function $f_T(\cdot)$ from a target dataset $\mathcal{D}_T$ (*e.g.*, UCLA and Boston).

There are two main ways to use TL for the cross-scenario PMP.

- *Feature extraction*: We can train a feature extractor on a source scenario and then use that feature extractor to extract features from data from a target scenario. Once we have extracted the features, we can train a simple model (*e.g.*, a linear regressor) to predict the pathloss map for the target scenario.
- *Fine-tuning*: We can fine-tune a pre-trained model on the target scenario. This can be done by unfreezing some or all of the layers of the pre-trained model and training the model on data from the target scenario.

The choice between those two methods depends on a number of factors, including the size and complexity of the pre-trained model, the availability of training data for the target dataset, and the computational resources available.

In this work, we focus on the fine-tuning TL approach with all of the layers of the pre-trained model unfrozen.[11] This approach is simple yet effective, achieving higher accuracy on various cross-scenario PMP tasks with less training data and shorter training time, as elaborated in the following subsection.

We prepare and use the following pre-trained models in our experiments:

(i) **VGG16$_{\mathbf{ImgNet}}$** is the pre-trained CNN model trained on the ImageNet dataset, which contains 140k images belonging to 22k categories. It is a powerful image classification model that has been used to achieve state-of-the-art results on a variety of image classification benchmarks.

(ii) **PMNet$_{\mathbf{3gpp}}$** is the pre-trained PMNet model trained on the 3GPP pathloss map dataset. The 3GPP pathloss map dataset is prepared with the 3GPP pathloss model in [20] (see 3GPP in Sec. IV-E.4, Fig. 4, and Table V).

(iii) **PMNet$_{\mathbf{usc}}$** is the pre-trained PMNet model trained on the USC RT dataset. It is similar to PMNet$_{\mathbf{3gpp}}$ but is trained on a different dataset. This is our main pre-trained model.

Each pre-trained model is available on our GitHub page.

[11] While we have performed sample experiments with unfreezing certain layers, such as the encoder-frozen and decoder-unfrozen, performance did not improve significantly. A more comprehensive investigation of this topic is, however, beyond the scope of this paper.

TABLE VII
TRAINING CONFIGURATION AND HYPER-PARAMETERS IN CROSS-SCENARIO PMP

| | |
|---|---|
| ***Model*** | |
| Backbone | PMNet, VGG16 |
| Pre-trained model | PMNet$_{\text{usc}}$, PMNet$_{\text{3gpp}}$, VGG16$_{\text{ImgNet}}$ |
| ***Dataset*** (UCLA, Boston) | |
| Map | UCLA campus, Boston |
| Split for training (test) set | $10\% \sim 90\%$ ($10\%$) of dataset |
| ***Hyper-parameter*** | |
| LR | $10^{-3} \sim 5 \times 10^{-4}$ |
| LR gamma, step size | 0.5, 10 |
| Batch size | 16 |
| Optimizer | Adam |
| # of of epochs | 50 |

TABLE VIII
IMPACT OF TL ON TRAINING SPEED ($= \frac{1}{\text{steps}}$). PMNET MODELS WITH OR WITHOUT PMNET$_{\text{usc}}$ PRE-TRAINED MODEL ARE TRAINED AND EVALUATED ON THE BOSTON DATASET

| ***Case*** | ***# of Required Step (Training Speed)*** | |
|---|---|---|
| | ***RMSE*** $\leq 0.1$ | ***RMSE*** $\approx 0.03$ |
| Vanilla (90% Data) | 5841 ($\times\mathbf{1.0}$) | 6195 ($\times\mathbf{1.0}$) |
| PMNet$_{\text{usc}}$ (20% Data) | 1040 ($\times\mathbf{5.6}$) | 1520 ($\times\mathbf{4.1}$) |

### C. Simulation Results

As demonstrated in the cross-scenario evaluation results (in Fig. 5 and Table VI), there is a need for further development to make PMNet adapt to different network scenarios. To this end, our approach is fine-tuning a pre-trained model with down-sized data for the new scenario. Here, the main questions in performing cross-scenario PMP are: (1) How quickly and with how minimal data PMNet can effectively adapt to new scenarios; and (2) Which pre-trained model should be utilized for optimal performance in cross-scenario PMP.

*1) Efficiency:* For cross-scenario PMP, rapidly adapting PMNet models to new network scenarios using limited data is essential due to the time-consuming and expensive nature of channel measurement using RT simulation or channel sounder. This is particularly critical for applications like beam management and localization using ML-based PMP, which demand quick adjustments for new scenarios.

*2) Impact of TL:* TL can significantly improve the training speed of PMNet models for cross-scenario PMP. As shown in Fig. 6 and Table VIII, the TL case with the PMNet$_{\text{usc}}$ pre-trained model achieves a given level of accuracy much faster even with much less amount of training data. In particular, PMNet$_{\text{usc}}$ achieves the same level of accuracy (*RMSE* $\leq$ 0.1 and *RMSE* $\approx$ 0.03) $\times 5.6$ and $\times 4.1$ faster, respectively, as the Vanilla case (highlighted in C), where we define as "Vanilla" the training from scratch in a particular environment.

Furthermore, the TL can also significantly save the required amount of data for cross-scenario PMP. As shown in Fig. 7, the TL (PMNet$_{\text{usc}}$) trained with about $20\%$ of the Boston dataset achieves equivalent results to the Vanilla case trained with about $90\%$ of the dataset.

It is worth noting that limited training data can easily induce overfitting, as observed in the Vanilla case with $20\%$ Data (highlighted in D). For the same amount of new scenario

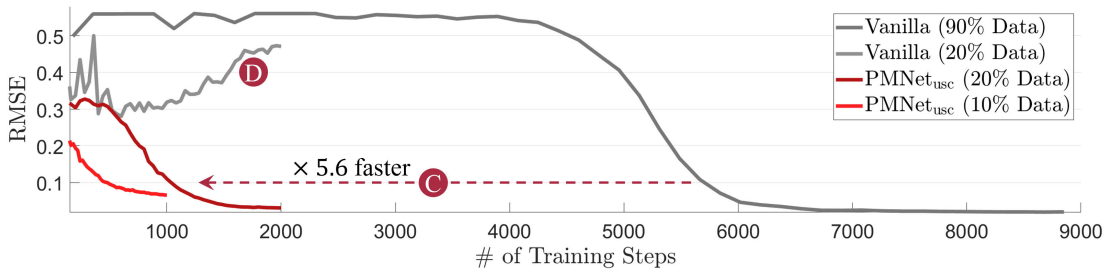



Fig. 6. Comparison of the training efficiency of PMNet models with and without TL. PMNet models are trained for 50 epochs and evaluated on the Boston dataset.

TABLE IX

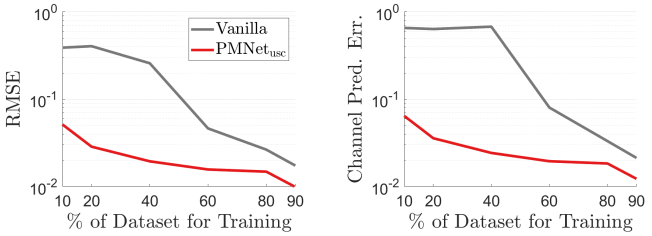
COMPARISON OF PRE-TRAINED MODELS (VGG16$_{\text{ImgNet}}$, PMNET$_{\text{3gpp}}$, AND PMNET$_{\text{usc}}$) IN TERMS OF ACCURACY. MODELS ARE EVALUATED ON THE UCLA AND BOSTON DATASETS, USING 90% OF THE DATA FOR TRAINING AND 10% OF THE DATA FOR VALIDATION. 50 EPOCHS ARE USED FOR TRAINING. LOWER VALUES INDICATE BETTER PERFORMANCE, AND THE LOWEST ERRORS ARE HIGHLIGHTED

| *Case* | *Pre-training* | *Model* | *RMSE*↓ | *RoI Segmentation Err.*↓ | *Channel Prediction Err.*↓ |
|---|---|---|---|---|---|
| Vanilla | ✗ | PMNet | 0.03415 | 0.02935 | 0.03844 |
| TL (ImageNet) | ✓ (ImageNet) | VGG16 | 0.04528 | 0.01814 | 0.05108 (E) |
| TL (3GPP) | ✓ (3GPP) | PMNet | 0.02809 | **0.00655** | 0.03238 |
| TL (USC) | ✓ (USC) | PMNet | **0.02792** | 0.01666 | **0.03145** |

(a) UCLA

| *Case* | *Pre-training* | *Model* | *RMSE*↓ | *RoI Segmentation Err.*↓ | *Channel Prediction Err.*↓ |
|---|---|---|---|---|---|
| Vanilla | ✗ | PMNet | 0.01736 | 0.02417 | 0.02125 |
| TL (ImageNet) | ✓ (ImageNet) | VGG16 | 0.01999 | **0.02040** | 0.02512 |
| TL (3GPP) | ✓ (3GPP) | PMNet | 0.01762 | 0.04030 | 0.02187 |
| TL (USC) | ✓ (USC) | PMNet | **0.00987** | 0.03530 | **0.01225** |

(b) Boston




(a) RMSE (b) Channel prediction error

Fig. 7. Impact of TL on training data requirements. PMNet models with or without PMNet$_{\text{usc}}$ pre-trained model are trained with 50 epochs and evaluated on the Boston dataset.

data, the TL case (PMNet$_{\text{usc}}$ (20%)) does not experience the overfitting issue. This suggests that TL also enhances training stability (less overfitting issue with limited data) in cross-scenario PMP.

Our findings demonstrate that the pre-trained PMNet$_{\text{usc}}$ model efficiently accelerates the training process by leveraging its knowledge of PMP tasks, including the physics of wireless channel propagation and RoI segmentation, and this model can be readily adapted to new scenarios with minimal data and training steps.

Consequently, we confirm that fine-tuning with a *suitable* pre-trained model is an effective cross-PMP task method. Another key question is which pre-trained model is suitable and which is not, which is discussed further in the following.

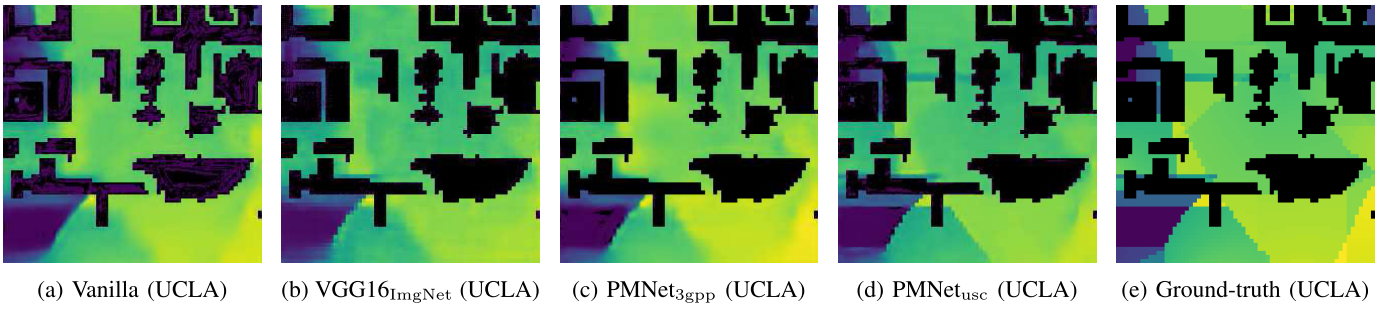
*3) Accuracy:* As discussed in Sec. V, the source and target scenario (task or domain) should be sufficiently similar for effective TL to occur. For instance, to successfully apply TL to the target task of predicting wireless communication channels, the NN should extract relevant features of wireless propagation physics from the source task.

*4) "Suitable" Pre-Trained Model:* Table IX compares the performance of the PMNet model with and without TL. The baseline model, referred to as Vanilla, is trained without any TL (without any pre-trained model). Additionally, we compare the performance of TL using a pre-trained model trained on an unrelated source scenario (*i.e.*, VGG16 trained on ImageNet) with TL using a pre-trained model trained on a related source scenario (*i.e.*, PMNet trained on USC or 3GPP datasets).

As shown in Table IX, both PMNet models trained on PMNet$_{\text{usc}}$ and PMNet$_{\text{3gpp}}$ outperform the Vanilla case on all performance metrics, suggesting that using a pre-trained model trained on a related source task can significantly improve accuracy.

Interestingly, while the VGG16 model trained on ImageNet (VGG16$_{\text{ImgNet}}$) outperforms the Vanilla for RoI segmentation, it fails to do so for channel prediction (highlighted in E). This discrepancy stems from the VGG16 pre-trained model, which has an inherent understanding of segmentation and image representation from its source task; however, does not have any knowledge of the physics of wireless propagation.

Fig. 8 visually confirms the findings from Table IX. All models achieve high accuracy for RoI segmentation, while



(a) Vanilla (UCLA) (b) $\text{VGG16}_{\text{ImgNet}}$ (UCLA) (c) $\text{PMNet}_{\text{3gpp}}$ (UCLA) (d) $\text{PMNet}_{\text{usc}}$ (UCLA) (e) Ground-truth (UCLA)

(f) Vanilla (Boston) (g) $\text{VGG16}_{\text{ImgNet}}$ (Boston) (h) $\text{PMNet}_{\text{3gpp}}$ (Boston) (i) $\text{PMNet}_{\text{usc}}$ (Boston) (j) Ground-truth (Boston)

Fig. 8. Comparison of the prediction results of pre-trained models ($\text{VGG16}_{\text{ImgNet}}$, $\text{PMNet}_{\text{3gpp}}$, and $\text{PMNet}_{\text{usc}}$). 50 epochs are used for training. Brighter colors indicate higher PG. Note that pixels with non-zero gray value are converted to color to highlight differences between results.

only the TL case using a pre-trained model trained on a related source scenario (*e.g.*, $\text{PMNet}_{\text{3gpp}}$ and $\text{PMNet}_{\text{usc}}$) achieves high accuracy for channel prediction, capturing subtle details of the wireless propagation physics. This suggests that our PMNet pre-trained model is generalizable to different scenarios due to its inherent knowledge of channel propagation representation, and that TL can further improve accuracy.

These results empirically demonstrate that pre-trained model's source dataset (task or domain) should be similar to the target dataset (task or domain) to transfer useful information during TL. Specifically, for cross-scenario PMP, it is important to use a pre-trained model that has been trained extensively on data related to wireless propagation physics.

Therefore, we conclude that the suggested TL approach, fine-tuning with a stable and closely related pre-trained model (such as $\text{PMNet}_{\text{usc}}$), is a simple yet effective way to address the cross-scenario PMP task, which is important for practical applications.

## VI. Conclusion

This work introduces an ML-based large-scale channel prediction framework, PMNet, which can create highly accurate pathloss predictions for a given map in a few milliseconds. Utilizing an RT channel dataset of real-world scenarios (*e.g.*, USC, UCLA, and Boston area), PMNet is verified for its accuracy and training efficiency. In particular, TL with our PMNet pre-trained model, which has generalization capability for different network scenarios, enables the PMNet to adapt itself quickly and efficiently to a new network scenario, while achieving an RMSE of $10^{-2}$ level.

The high accuracy and low runtime of the PMNet framework make it suitable for deployment planning in dense networks as well as online optimization of network parameters.

Still, it remains an open question whether the knowledge of wireless propagation physics in our PMNet pre-trained model can be transferred to other downstream tasks beyond the PMP task; this question will be the topic of our future research.

## Acknowledgment

The help of Dr. Zheda Li in the creation of the USC data set is gratefully acknowledged.

## References

[1] J.-H. Lee, J. Lee, S.-H. Lee, and A. F. Molisch, "PMNet: Large-scale channel prediction system for ICASSP 2023 first pathloss radio map prediction challenge," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–2.

[2] J.-H. Lee, O. G. Serbetci, D. P. Selvam, and A. F. Molisch, "PMNet: Robust pathloss map prediction via supervised learning," in *Proc. IEEE Global Commun. Conf.*, Dec. 2023, pp. 1–20.

[3] X. Lin, L. Kundu, C. Dick, E. Obiodu, T. Mostak, and M. Flaxman, "6G digital twin networks: From theory to practice," *IEEE Commun. Mag.*, vol. 1, no. 1, pp. 1–7, Aug. 2023.

[4] A. Alkhateeb, S. Jiang, and G. Charan, "Real-time digital twins: Vision and research directions for 6G and beyond," *IEEE Commun. Mag.*, vol. 61, no. 11, pp. 128–134, Nov. 2023.

[5] R. Valenzuela, "A ray tracing approach to predicting indoor wireless transmission," in *Proc. IEEE 43rd Veh. Technol. Conf.*, Apr. 1993, pp. 214–218.

[6] S.-C. Kim et al., "Radio propagation measurements and prediction using three-dimensional ray tracing in urban environments at 908 MHz and 1.9 GHz," *IEEE Trans. Veh. Technol.*, vol. 48, no. 3, pp. 931–946, May 1999.

[7] *Wireless Insite*. Accessed: Sep. 2024. [Online]. Available: https://www.remcom.com/wirelessinsite

[8] (2024). *Path Loss Comparison in Rosslyn*. [Online]. Available: https://www.remcom.com/examples/path-loss-comparison-in-rosslyn-virginia.html

[9] V. Degli-Esposti, F. Fuschini, E. M. Vitucci, and G. Falciasecca, "Measurement and modelling of scattering from buildings," *IEEE Trans. Antennas Propag.*, vol. 55, no. 1, pp. 143–153, Jan. 2007.

[10] R. Wahl, G. Wolfle, P. Wildbolz, and F. Landstorfer, "Dominant path prediction model for urban scenarios," in *Proc. 14th IST Mobile Wireless Commun. Summit*, 2005, pp. 1–26.

[11] M. Lee and D. Han, "Voronoi tessellation based interpolation method for Wi-Fi radio map construction," *IEEE Commun. Lett.*, vol. 16, no. 3, pp. 404–407, Mar. 2012.

[12] D. Romero and S.-J. Kim, "Radio map estimation: A data-driven approach to spectrum cartography," 2022, *arXiv:2202.03269*.

[13] T. Orekondy, P. Kumar, S. Kadambi, H. Ye, J. Soriaga, and A. Behboodi, "WiNeRT: Towards neural ray tracing for wireless channel modelling and differentiable simulations," in *Proc. Int. Conf. Learn. Represent.*, 2023, pp. 1–20.

[14] X. Zhao, Z. An, Q. Pan, and L. Yang, "NeRF2: Neural radio-frequency radiance fields," in *Proc. 29th Annu. Int. Conf. Mobile Comput. Netw.*, vol. 33, Oct. 2023, pp. 1–15.

[15] R. Levie, Ç. Yapar, G. Kutyniok, and G. Caire, "RadioUNet: Fast radio map estimation with convolutional neural networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 4001–4015, Jun. 2021.

[16] V. V. Ratnam et al., "FadeNet: Deep learning-based mm-Wave large-scale channel fading prediction and its applications," *IEEE Access*, vol. 9, pp. 3278–3290, 2021.

[17] E. Krijestorac, H. Sallouha, S. Sarkar, and D. Cabric, "Agile radio map prediction using deep learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–2.

[18] K. Qiu, S. Bakirtzis, H. Song, I. Wassell, and J. Zhang, "Deep learning-based path loss prediction for outdoor wireless communication systems," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–2.

[19] Ç. Yapar, F. Jaensch, R. Levie, G. Kutyniok, and G. Caire, "The first pathloss radio map prediction challenge," 2023, *arXiv:2310.07658*.

[20] *Study on Channel Model for Frequencies From 0.5 to 100 GHz*, Standard TR 38.901, 2018.

[21] Ç. Yapar, R. Levie, G. Kutyniok, and G. Caire, "Dataset of pathloss and ToA radio maps with localization application," 2022, *arXiv:2212.11777*.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[23] A. F. Molisch, *Wireless Communications: From Fundamentals to Beyond 5G*, 3rd ed., Hoboken, NJ, USA: Wiley, 2023.

[24] F. Fuschini, H. El-Sallabi, V. Degli-Esposti, L. Vuokko, D. Guiducci, and P. Vainikainen, "Analysis of multipath propagation in urban environment through multidimensional measurements and advanced ray tracing simulation," *IEEE Trans. Antenna Propag.*, vol. 56, no. 3, pp. 848–857, Mar. 2008.

[25] T. Rautiainen, G. Wolfle, and R. Hoppe, "Verifying path loss and delay spread predictions of a 3D ray tracing propagation model in urban environment," in *Proc. IEEE 56th Veh. Technol. Conf.*, vol. 4, Jun. 2002, pp. 2470–2474.

[26] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[27] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–20.

[28] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" 2014, *arXiv:1411.1792*.

[29] K. James et al., "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.

[30] S. Kornblith, J. Shlens, and Q. V. Le, "Do better ImageNet models transfer better?" 2018, *arXiv:1805.08974*.

[31] M. Huh, P. Agrawal, and A. A. Efros, "What makes ImageNet good for transfer learning?" 2016, *arXiv:1608.08614*.

[32] M. Steinbauer, A. F. Molisch, and E. Bonek, "The double-directional radio channel," *IEEE Antennas Propag. Mag.*, vol. 43, no. 4, pp. 51–63, Aug. 2001, doi: 10.1109/74.951559.

[33] R. Araki, T. Onishi, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "MT-DSSD: Deconvolutional single shot detector using multi task learning for object detection, segmentation, and grasping detection," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 10487–10493.

[34] T. Xu and W. Takano, "Graph stacked hourglass networks for 3D human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1–23.

[35] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with atrous separable convolution for semantic image segmentation," in *Proc. ECCV*, 2018, pp. 1–27.

[36] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5168–5177.

[37] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters—Improve semantic segmentation by global convolutional network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1743–1751.

[38] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*. Cham, Switzerland: Springer, 2015, pp. 234–241.

[39] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. ICLR*, 2016, pp. 1–29.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.


**Ju-Hyung Lee** (Member, IEEE) received the Ph.D. degree from Korea University, Seoul, South Korea. He was a Post-Doctoral Researcher in electrical and computer engineering with the University of Southern California (USC), Los Angeles, CA, USA, where he worked under the supervision of Prof. Andreas F. Molisch. Before that, he was a Research Professor with Korea University. He is currently a Principal Researcher with Nokia, Sunnyvale, CA, USA. His research interests include generative AI solutions for wireless communication systems and on-device AI/LLM. He has been recognized with several prestigious awards, including first rank in a machine learning (ML) competition, best paper awards at key IEEE conferences, and the Grand Prize for Research Excellence from Korea University. For more information visit the link (https://juhyung-lee.com/) and (https://juhyung-lee.com/).

**Andreas F. Molisch** (Fellow, IEEE) received the Dipl. (Ing.), Ph.D., and Habilitation degrees from Technical University Vienna, Austria, in 1990, 1994, and 1999, respectively.

He spent the next ten years in industry, FTW, AT&T (Bell) Laboratories, and Mitsubishi Electric Research Laboratories (where he rose to the Chief Wireless Standards Architect). In 2009, he joined the University of Southern California (USC), Los Angeles, CA, USA, as a Professor and founded the Wireless Devices and Systems (WiDeS) Group. In 2017, he was appointed to the Solomon Golomb—Andrew and Erna Viterbi Chair. His research interests revolve around wireless propagation channels, wireless systems design and their interaction, wireless channel measurement and modeling for 5G and 6G systems, joint communication-caching-computation, hybrid beamforming, UWB/TOA-based localization, and novel modulation/multiple access methods. Overall, he has published five books [among them the textbook *Wireless Communications* (third edition in 2023)], 22 book chapters, more than 300 journal articles, and more than 400 conference papers. He is also the inventor of 80 patents, and the co-author of some 70 standards contributions. His work has been cited more than 73,000 times, his H-index is 113, and he is a Clarivate Highly Cited Researcher.

Dr. Molisch is a fellow of the National Academy of Inventors, AAAS, and IET; an IEEE Distinguished Lecturer, and a member of Austrian Academy of Sciences. He has received numerous awards, among them the IET Achievement Medal, the Technical Achievement Awards of IEEE Vehicular Technology Society (Evans Avant-Garde Award) and the IEEE Communications Society (Edwin Howard Armstrong Award), and the Technical Field Award of the IEEE for Communications (Eric Sumner Award). He has been an editor of a number of journals and special issues, the general chair, the technical program committee chair, or the symposium chair of multiple international conferences; and the Chairperson of various international standardization groups.

# A Scalable and Generalizable Pathloss Map Prediction

Ju-Hyung Lee, *Member, IEEE*, and Andreas F. Molisch, *Fellow, IEEE*

***Abstract*— Large-scale channel prediction, *i.e.*, estimation of the pathloss from geographical/morphological/building maps, is an essential component of wireless network planning. Ray tracing (RT)-based methods have been widely used for many years, but they require significant computational effort that may become prohibitive with the increased network densification and/or use of higher frequencies in B5G/6G systems. In this paper, we propose a data-driven, model-free pathloss map prediction (PMP) method, called PMNet. PMNet uses a supervised learning approach: it is trained on a limited amount of RT data and map data. Once trained, PMNet can predict pathloss over location with high accuracy (an RMSE level of $10^{-2}$) in a few milliseconds. We further extend PMNet by employing transfer learning (TL). TL allows PMNet to learn a new network scenario quickly ($\times 5.6$ faster training) and efficiently (using $\times 4.5$ less data) by transferring knowledge from a pre-trained model, while retaining accuracy. Our results demonstrate that PMNet is a scalable and generalizable ML-based PMP method, showing its potential to be used in several network optimization applications.**

***Index Terms*— Pathloss map prediction, ray tracing, machine learning, computer vision, transfer learning, network optimization, digital twin, 6G.**

## I. Introduction

DIGITAL twin (DT) network is emerging as a key enabler for the artificial intelligence (AI) and machine learning (ML)-driven design, simulation, and optimization of 6G systems [3], [4]. A DT network is a dynamic, digital replica of a real-world network environment, providing real-time, accurate reflections of physical network scenarios. It can be used for a variety of applications, including dynamic resource allocation, beam management, and localization using ML-based PMP, which demands quick adjustments for new scenarios. However, implementing DT is challenging in 6G networks, which are characterized by increased deployment density, complex distributed architectures, and high-frequency operation in millimeter wave (mmWave) and terahertz (THz) bands.

Individually and taken together, these developments necessitate dramatically faster large-scale channel prediction methods.[1] Since traditional ray tracing (RT) tools are too slow for the repeated runs required in such DT implementation processes, there is a strong need for new, accurate, and fast methods for channel prediction over a large-scale area (*e.g.*, campus or city-map scale).

Several works have addressed this need by channel prediction using powerful ML techniques. These works use ground-truth channel data (from RT simulations or real channel measurements/soundings campaigns) to train neural networks (NNs). This eventually provides an accurate and fast prediction of channel information (*e.g.*, received power, delay, angles, and so on) for a certain area, a technique called ML-based site-specific radio propagation modeling.

Still, these ML-based approaches use supervised learning, meaning they are trained to solve a specific network scenario with a certain labeled dataset. In other words, the models may need to be rebuilt for a new network scenario, *e.g.*, different map scales, environmental aspects, and/or network configuration - a process that can be time-consuming and expensive. This creates a need for a method that can furthermore transfer knowledge of propagation channels across different network scenarios and environments.

### A. Related Works

Due to the high cost and complexity of field measurements with channel sounders, most cellular deployment planning has long replaced channel measurements with *electromagnetic (EM) simulation-based* approaches, such as RT [5], [6] and ray launching [7] simulation.[2] Over the past 30 years, the efficiency and accuracy of RT have improved significantly [9], thanks to the prevalence of GPUs (graphic processing units) that efficiently facilitate RT tasks.

However, due to the factors mentioned above (such as the need for more detailed environmental consideration at higher

Manuscript received 6 December 2023; revised 20 February 2024 and 3 July 2024; accepted 27 August 2024. This work was supported by NSF under Project 2133655 and Project 2008443. An earlier version of this paper was presented in part at the 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023) [DOI: 10.1109/ICASSP49357.2023.10095257] and in part at the 2023 Global Communications Conference (Globecom 2023) [DOI: 10.1109/GLOBECOM54140.2023.10437562]. The associate editor coordinating the review of this article and approving it for publication was C. Han. *(Corresponding author: Ju-Hyung Lee.)*

Ju-Hyung Lee was with Ming Hsieh Department of Electrical and Computer Engineering, University of Southern California (USC), Los Angeles, CA 90007 USA. He is now with Nokia, Sunnyvale, CA 94085 USA (e-mail: juhyung.lee@outlook.com).

Andreas F. Molisch is with the Ming Hsieh Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA 90089 USA (e-mail: molisch@usc.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TWC.2024.3457431.

Digital Object Identifier 10.1109/TWC.2024.3457431

[1]The word "channel prediction" is often used for two different problems: (i) computation of the propagation channel at a particular location based on maps of the environment, and (ii) temporal prediction of the channel (often for a mobile device moving on a trajectory), based on measurements in the immediate past. This paper only considers the former case.

[2]Our research utilizes RT simulations via Wireless Insite to investigate path loss. This simulation program has been validated against real-world measurements, e.g., in [8] showing an acceptable error margin.

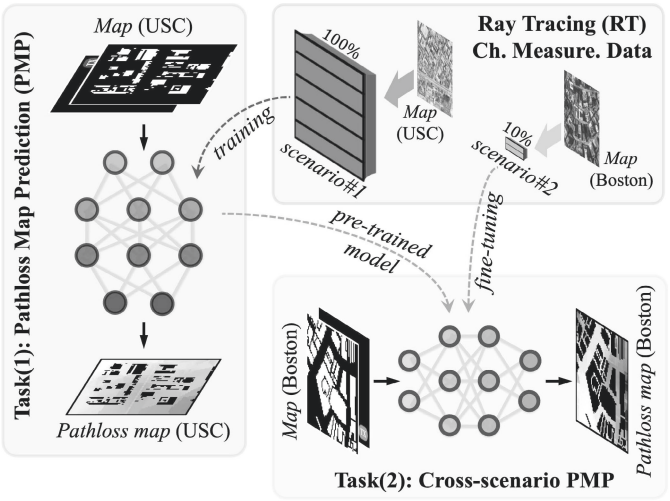



Fig. 1. Overview of the pathloss map prediction (PMP) task and the cross-scenario PMP. The input Map feature includes the transmitter (TX) location.

frequencies and the need for fast simulations with higher deployment density), RT simulations are too computationally intensive for large-scale network deployment in 6G systems. Consequently, simplified *model-based* approaches like the dominant path model [10], or fine-tuning of generic pathloss models (*e.g.*, 3GPP path gain model) with limited measurement data [11], [12] have been proposed over the years. Nevertheless, these approaches have gained limited acceptance by network operators due to their insufficient accuracy in predicting the propagation characteristics of signals in complex environments.

In recent years, supervised ML has been applied to solve a variety of challenging problems in wireless communication, including channel measurement/prediction for 6G networks. Such an *ML-based* approach can be trained on a map of the environment (topology/morphology) and a relatively small set of measurement data to learn how to provide a virtual replica (*e.g.*, DT) of a large-scale network environment in real-time while accurately modeling the behavior of channel characteristics.

On the one hand, models like WiNeRT [13] and NeRF2 [14] are specifically developed to predict detailed channel information (*e.g.*, power, delay, and angle information) of each multi-path component (MPC) between TX and receiver (RX) with the input of detailed information, including spatial configuration and wireless configuration parameters. These models are particularly well-suited for applications in small-scale indoor areas, where high-detailed channel prediction is required (*e.g.*, indoor sensing).

On the other hand, models like RadioUNet [15] and FadeNet [16] aim to predict the path gain, received power, or coverage for TX-RX in a given area with the input of a building map. These models are designed for large-scale channel prediction, where fast operation is essential (*e.g.*, network optimization).

In particular, several state-of-the-art works, such as Agile [17], PPNet [18], and PMNet [1], are pushing the boundaries of predictive accuracy and computational efficiency for large-scale channel prediction (*e.g.*, radio environment map estimation), as evidenced by their performance in ML competitions such as the *RadioMap Prediction Challenge* (see details in [19]). This highlights the applicability and importance of large-scale channel prediction in evolving wireless network optimization, which aligns with our research direction.

### B. Contributions

This paper proposes a scalable and generalizable channel prediction approach specifically designed for large-scale channel prediction, called PMP task. Our contributions can be summarized as follows:

- We design a PMP-oriented NN architecture, called PMNet, by leveraging computer-vision techniques, generating highly accurate channel prediction results for a given map in few milliseconds. PMNet achieves the best channel prediction accuracy compared to two baselines: a model-based scheme (*3GPP-UMi* model [20]) and another ML-based scheme (*RadioUNet* [15]) (see **Table V** in Sec. IV) and also in different PMP datasets. PMNet achieved *1st*-rank in the ICASSP 2023 Radio Map Prediction Challenge [19].[3]
- We build three sets of real-world channel datasets using a RT simulation tool, *i.e.*, *Wireless Insite*, for training and evaluation, which reflects different network scenarios (*e.g.*, different map scale, environment, and network configuration) in two different light urban environments (the USC and UCLA campuses) and a metropolitan area (the Boston area), see **Table I** in Sec. III.
- We propose a method of predicting pathloss in unseen network scenarios by using transfer learning (TL) with a pre-trained model. We prepare three pre-trained models for TL: VGG16 [22] and two pre-trained PMNet models trained with 3GPP prediction results and RT simulation results, respectively, and quantitatively and qualitatively evaluate their accuracy (see **Table IX** and **Fig. 8** in Sec. V).
- We empirically demonstrate that our PMNet pre-trained model has generalization capability for different network scenarios, adjusting to new network scenarios $\times 5.6$ faster and using $\times 4.5$ less data than a baseline model without TL, while still achieving high accuracy of an RMSE of $10^{-2}$ level (see **Fig. 6** and **Table. VIII** in Sec. V).
- We release source code for the experiments to promote reproducible ML research in wireless communication.[4]

### C. Paper Organization

The rest of the paper is organized as follows: Sec. II presents the background on two important concepts: (1) *ray tracing simulation*, which is used to generate ground-truth channel information for training and evaluation; and (2) *transfer learning*, which enables us to transfer the knowledge learned

[3]In this competition, PMNet demonstrated its high accuracy in the PMP task on a different dataset [21], which featured a different map scale, network configuration, and was generated by a different RT simulation tool, *i.e.*, *WinProp*, highlighting PMNet's generalization capability.

[4]https://github.com/abman23/PMNet

from a source task/dataset to a new task/dataset (*e.g.*, unseen network scenario). After introducing our dataset based on real geographical maps in Sec. III, Sec. IV introduces the PMP task and our proposed NN architecture (PMNet) for this channel prediction task. We also present the training and evaluation process, as well as simulation results. Then, Sec. V presents our approach for efficiently learning and predicting channels in unseen network environments by transferring the pre-trained knowledge from other networks. We provide extensive experimental results and quantitative and qualitative performance analysis, followed by concluding remarks in Sec. VI.

*Notation:* Throughout this paper, we use the normal-face font to denote scalars and the boldface font to denote vectors. We use $P(\cdot)$ and $P(\cdot|\cdot)$ to represent a marginal probability distribution and conditional distribution, respectively. We also use $\|\cdot\|$ to denote the $L^2$-norm, which is an Euclidean norm. $\mathcal{N}(\mu, \sigma)$ denotes the normal distribution with mean $\mu$ and standard deviation $\sigma$.

## II. Background

To provide a comprehensive understanding of our work, it is essential to cover Pathloss, Ray Tracing Simulation, and Transfer Learning, as these areas are integral to our methodology and analysis.

### A. Pathloss

The link gain between a TX at location $\boldsymbol{q}_{\mathrm{TX}}$ and an RX at location $\boldsymbol{q}_{\mathrm{RX}}$ at time $t$ and frequency $f$ can be expressed as follows:

$$|h(t, f, \boldsymbol{q}_{\mathrm{TX}}, \boldsymbol{q}_{\mathrm{RX}})|^2 = \frac{P_{\mathrm{RX}}(t, f, \boldsymbol{q}_{\mathrm{RX}})}{P_{\mathrm{TX}}(t, f, \boldsymbol{q}_{\mathrm{TX}})} \tag{1}$$

where $P_{\mathrm{RX}}$ and $P_{\mathrm{TX}}$ are received and transmitted power, respectively. This link gain includes the effects of antenna gains at TX and RX; when isotropic antennas are used, it becomes identical to the channel gain. It exhibits variations in time and/or location due to small-scale fading, shadowing, and large-scale distance changes. Averaging over small-scale fading removes (under certain circumstances, see [23, Ch. 7]) the dependence on frequency and time, providing the *path gain* (PG) that can be written as a function of only the large-scale distance changes:

$$\mathrm{PG}(\boldsymbol{q}_{\mathrm{TX}}, \boldsymbol{q}_{\mathrm{RX}}) = \frac{1}{T_{\mathrm{S}}} \frac{1}{B_{\mathrm{S}}} \int_{T_{\mathrm{S}}} \int_{B_{\mathrm{S}}} |h(t, f, \boldsymbol{q}_{\mathrm{TX}}, \boldsymbol{q}_{\mathrm{RX}})|^2 \, df \, dt. \tag{2}$$

Here, $T_{\mathrm{S}}$ and $B_{\mathrm{S}}$ denote the stationary-time and -bandwidth, respectively. The path gain can be represented as the sum of the powers of the $N$ MPCs, as discussed further in Sec. III-A. For later reference, we note that the pathloss is the inverse of the path gain (or the sign-flipped value when expressed in dB).

### B. Ray Tracing (RT) Simulation

RT is an approximate method for modeling the propagation of electromagnetic waves in wireless communication scenarios. It works by tracing the paths of individual rays as they propagate through the environment, whose features are represented in a geographical database. The rays are reflected, deflected, and scattered by the objects in the environment, with the various interaction processes computed according to high-frequency approximations, namely (most commonly) Snell's laws for specular reflection and transmission, uniform theory of diffraction (UTD) for diffraction, and Kirchhoff scattering theory for diffuse scattering [23, Ch. 4].[5] The RT tool simulates radio wave propagation deterministically based on physical laws, offering site-specific radio propagation modeling, in contrast to stochastic wireless channel models (*e.g.*, 3GPP standardized channel model).

In this paper, we employ a commercial RT tool, *Wireless Insite* from Remcom [7] for all RT simulations, both because of its user-friendliness and the fact that its accuracy has been compared against a number of channel sounder measurements [6], [24], [25]. RT can be used to predict channel information, such as received signal strength, delay, and angles, in a variety of wireless environments, both indoor and outdoor. The accuracy of RT simulations depends on various factors, such as the complexity of the environment, the accuracy of the geographical database, and the carrier frequency. The channel information obtained from the RT can be utilized, *inter alia*, for various network optimization tasks, including base station (BS) deployment planning, BS parameter optimization, as well as beam management and localization.

### C. Transfer Learning (TL)

TL is a machine learning technique that leverages a pre-trained model on a new task, significantly reducing the amount of data and training time required for new scenarios. This approach is particularly advantageous when there is limited data available for the new task or when the new task shares similarities with a previously learned task. By utilizing knowledge from a related task, TL can enhance model performance, expedite training processes, and mitigate overfitting, especially in data-constrained environments.

For instance, a model pre-trained on image classification tasks can be effectively repurposed for object detection or semantic segmentation. This reuse is possible because the model has already learned useful *feature representations* from a large and diverse dataset, enabling it to adapt more efficiently to new, related tasks.

One of the most popular pre-trained models is VGG16 [22], which is trained on more than a million images from the ImageNet database for image classification. VGG16 has demonstrated its versatility by being reused to improve performance in various tasks, including semantic segmentation [26] and object detection [27]. These applications showcase the model's ability to transfer learned features, thus enhancing performance in new domains with minimal additional training.

However, it is important to note that the effectiveness of TL depends on the *similarity* between the pre-trained task and the

[5] RT can be implemented via image-theory-based RT, or as ray launching. We will henceforth use the expression RT for both those methods.

target task.[6] The transferability of deep feature representations decreases as the discrepancy between the pre-trained task and the target task increases [28]. In other words, the further apart the task is, the less transferable the knowledge. One example is catastrophic forgetting, which is a phenomenon that can occur when fine-tuning a pre-trained model on a new task, resulting in a loss of previously acquired knowledge [29].

Research has shown that well-generalized models, particularly those with excellent pre-training performance [30], have the potential to require minimal fine-tuning or even none at all (*e.g.*, zero-shot learning) for new tasks [31]. These suggest the importance of selecting a pre-trained model suitable for the target task.

## III. Dataset

In this section, we discuss the dataset preparation process for our pathloss map datasets, reflecting real-world network scenarios in USC, UCLA, and Boston areas.

We obtained the ground-truth channel data using the commercial RT tool *Wireless Insite* [7], which takes into account the geographical and morphological features of the propagation environment. We then pre-processed the data (*e.g.*, interpolation and data augmentation) to prepare the ground-truth pathloss map.

### A. Channel Data

*1) RT Simulation:* As discussed in Sec. II-A, RT emulates the behavior of each MPC between TX and RX, following physical principles including the free-space power loss and interaction with different interacting objects (IOs). This allows us to compute for each MPC the information of complex amplitude $a$, directions of departure $\Omega$ and arrival $\Psi$, and delay $\tau$. The contribution of $m$-th MPC can be expressed as [32]:

$$h_m(t,\tau,\Omega,\Psi) = a_m\delta(\tau-\tau_m)\delta(\Omega-\Omega_m)\delta(\Psi-\Psi_m), \quad (3)$$

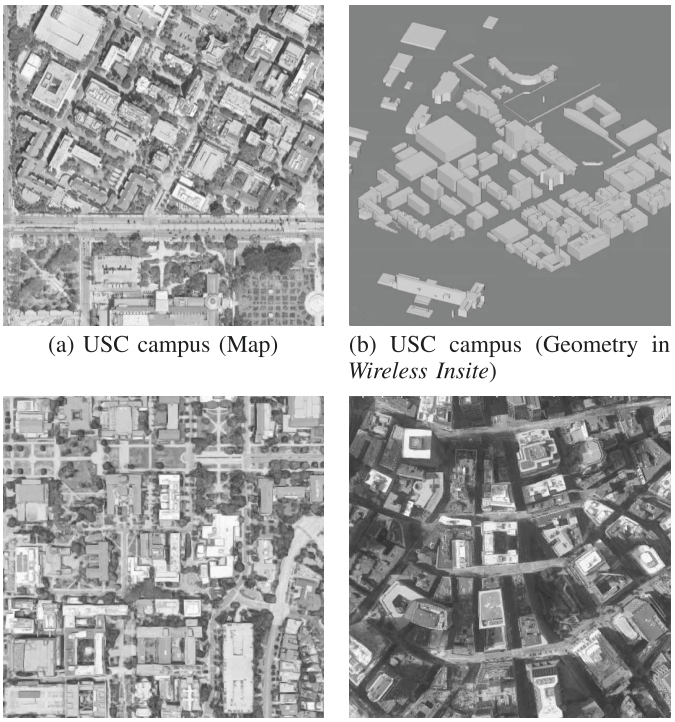
where the dependence of $\Omega, \Psi, \tau, a$ on $t$ is not written explicitly on the r.h.s. The sum of contributions from all MPCs is given by

$$h(t,\tau,\Omega,\Psi) = \sum_{m=1}^{N} h_m(t,\tau,\Omega,\Psi). \quad (4)$$

Since $\Omega, \Psi, \tau, |a|$ are constant over a stationarity-time and bandwidth, while $\arg(a)$ varies over many periods of $2\pi$, and assuming isotropic antennas at TX and RX (so that $\Omega, \Psi$ do not matter), the path gain averaged over the small-scale fading can be computed from (2) as

$$\mathrm{PG} = \sum_{m=1}^{N} |h_m(\tau,\Omega,\Psi)|^2 = \sum_{m=1}^{N} |a_m|^2. \quad (5)$$

[6]The wireless community has long classified environments—rural, suburban, urban, and metropolitan—based on their channel characteristics, a practice dating back to the COST 207 models of the mid-1980s. Despite the subjective nature of these categories and the lack of specific numerical criteria, their differentiation by factors like population density and infrastructure complexity is widely accepted for assessing wireless signal propagation and network performance and its *similarity*.



(a) USC campus (Map) (b) USC campus (Geometry in *Wireless Insite*)

(c) UCLA campus (Map) (d) Boston (Map)

Fig. 2. Map of USC, UCLA, and Boston used in RT simulation. Fig. 2a is imported and converted to Fig. 2b. The ground-truth pathloss map over the USC campus is then obtained using *Wireless Insite* RT simulation and pre-processing (*e.g.*, interpolation, gray conversion, and data augmentation).

Note that our pathloss map uses the information of path gain (in [dB]) while other information on angles and delay is not needed (though this information can be used for further applications, *e.g.*, beamforming algorithms).

Thus, $P_{\mathrm{RX}}$ (in [dBm]) can be expressed as a function of $P_{\mathrm{TX}}$ (in [dBm]) as follows:

$$P_{\mathrm{RX}} = P_{\mathrm{TX}} + \mathrm{PG}. \quad (6)$$

Note that we set $P_{\mathrm{TX}} = 0$ [dBm] in our RT dataset to simplify the analysis, which makes $P_{\mathrm{RX}}$ in [dBm] equal to PG in [dB].

To generate a ground-truth (labeled) dataset that simulates real-world network scenarios, we conduct *Wireless Insite* RT simulations on the geographical and morphological maps of the University of Southern California (USC) campus, the University of California, Los Angeles (UCLA) campus, and the Boston area. Both campus areas are in Los Angeles, CA, and exhibit a (light) urban build-up, with most buildings being five stories or less (with a few high-rises interspersed), gaps between buildings along the street canyons, and some open squares. The Boston area is in downtown of Boston, MA. It is a metropolitan area with multiple high-rises; its streets are *not* arranged along a rectangular grid. Each dataset has different network configurations and environmental characteristics (*e.g.*, map scale, and geographical features, such as vegetation). See Fig. 2 and Table I for more details.[7]

[7]It is worth noting that the simulations are performed at the sub-6 GHz band, which is the most widely used cellular band. Similar simulations can be performed in other frequency bands, such as the mmWave and THz bands, with minor adjustments to the parameters. However, at those high frequency bands, geographical data bases with higher resolution might be required for comparable accuracy.

TABLE I
PARAMETERS OF USC, UCLA, AND BOSTON DATASETS

| *Parameter* | *Dataset* | | |
|---|---|---|---|
| | USC | UCLA | Boston |
| Map scale | $880 \times 880$ [m$^2$] | $760 \times 760$ [m$^2$] | $553 \times 553$ [m$^2$] |
| Cropped map scale (per pixel) | $221 \times 221$ [m$^2$] ($0.86 \times 0.86$ [m$^2$]) | $225 \times 225$ [m$^2$] ($0.88 \times 0.88$ [m$^2$]) | $187 \times 187$ [m$^2$] ($0.73 \times 0.73$ [m$^2$]) |
| Terrain | ✓ | ✓ | ✓ |
| Buildings | ✓ | ✓ | ✓ |
| Foliage[a] | ✗ | ✗ | ✓ |
| Carrier frequency | 2.5 [GHz] | 3.0 [GHz] | 3.0 [GHz] |
| Transmit power | 0 [dBm] | 0 [dBm] | 0 [dBm] |
| TX antenna type[b] | Isotropic (vertical) | Half-wave dipole (vertical) | Half-wave dipole (vertical) |
| Total # of data/scene | 4754 | 3776 | 3143 |

[a]Our study assesses PMP accuracy across datasets varying in map scale, network configurations, and geography, focusing on how factors like foliage impact path loss predictions.

[b]Isotropic and half-wave dipole antennas provide almost identical radiation patterns within a certain angular extent. MPC induced outside of the angular extent does not contribute significantly to the link.

We stress that the goal of our work is the correct prediction of "ground-truth" pathloss by ML techniques. The pathloss obtained from the RT simulations might deviate from measured values due to inaccuracies of the database or inherent approximations of RTs. However, such deviations are irrelevant to the assessment of our ML methods, since they only impact what is used as "ground-truth" and not the prediction process itself. In other words, if the ground-truth is more accurate (similar to measurement results), our prediction inherently becomes more accurate as well.[8]

*2) 3GPP Model:* The 3GPP 38.901 channel model [20] (henceforth simply called the "3GPP model" for conciseness) is a widely used model for wireless system standardization that claims validity for frequencies spanning from 0.5 to 100 [GHz].

For the purposes of this paper, we only consider the 3GPP modeling of the pathloss, which follows the classical $\alpha - \beta$ model

$$\mathrm{PL}_{\alpha-\beta}(d) = 10\alpha \log_{10}(d) + \beta + S, \tag{7}$$

where $S \sim \mathcal{N}(0, \sigma_S)$ is a lognormally distributed random variable (with variance $\sigma_S$) representing the shadow fading, and $\alpha$, $\beta$, and $\sigma$ are parameters of the model that are based on measurement campaigns and that are different in different environments. Important for our later discussions, those parameters are also different depending on whether an unobstructed optical line of sight (LoS) exists between TX and RX or not.

Specifically, for urban environments, the following describes the path gain:

$$\mathrm{PG}_{\mathrm{UMi-LoS}} = \begin{cases} \mathrm{PL}_1, & (10[\mathrm{m}] \le d_{\mathrm{2D}} \le d_{\mathrm{BP}}) \\ \mathrm{PL}_2, & (d_{\mathrm{BP}} \le d_{\mathrm{2D}} \le 5[\mathrm{km}]) \end{cases} \tag{8}$$

$$\mathrm{PG}_{\mathrm{UMi-NLoS}} = \max(\mathrm{PG}_{\mathrm{UMi-LoS}}, \mathrm{PL}_3), \quad (10[\mathrm{m}] \le d_{\mathrm{2D}} \le 5[\mathrm{km}]) \tag{9}$$

[8]Thus, if our proposed PMNet can accurately predict/reproduce RT results when trained with RT data, it will also be able to do so for measurement data when trained with measurement data.

where the two-dimensional $xy$-distance is $d_{\mathrm{2D}}$ and the three-dimensional $xyz$-distance is $d_{\mathrm{3D}}$,

$$\begin{aligned} \mathrm{PL}_1 &= 32.4 + 21 \log_{10}(d_{\mathrm{3D}}) + 20 \log_{10}(f_c), \\ \mathrm{PL}_2 &= 32.4 + 40 \log_{10}(d_{\mathrm{3D}}) + 20 \log_{10}(f_c) \\ &\quad - 9.5 \log_{10}((d_{\mathrm{BP}})^2 + (h_{\mathrm{BS}} - h_{\mathrm{UT}})^2), \\ \mathrm{PL}_3 &= 22.4 + 35.3 \log_{10}(d_{\mathrm{3D}}) + 21.3 \log_{10}(f_c) \\ &\quad - 0.6(h_{\mathrm{UT}} - 1.5). \end{aligned} \tag{10}$$

Here, the breakpoint distance is $d_{\mathrm{BP}} = 2\pi h_{\mathrm{BS}} h_{\mathrm{UT}} f_c / c$ where $f_c$ is the center frequency in [Hz] and $c = 3.0 \times 10^8$[m/s] is the speed of light. The antenna heights at the TX (*e.g.*, base station), $h_{\mathrm{BS}}$, and the RX (*e.g.*, user terminal), $h_{\mathrm{UT}}$, are set to 1.5 [m] and 10 [m], respectively. Note that the model differs for LoS and non-LoS (NLoS) situations.

This model is employed as one of our baselines for the prediction (see Sec. IV-E). While the 3GPP model also models shadowing, it incorporates it as *stochastic* variations that cannot be related to particular map features; we therefore omit them for the purposes of this paper.

### *B. Pre-Processing*

The raw numeric data from the RT simulation is pre-processed using *gray conversion* and *interpolation* methods to generate the ground-truth pathloss map, *data augmentation* methods to create an increased amount of labeled data, and *sampling* methods to divide them into training and testing sets.

*1) Gray Conversion:* To generate the pathloss map, we begin by converting the received power $P_{\mathrm{RX}}$ (in [dBm]) (or the path gain PG in [dB]) into grayscale between 1 and 255 using *Min-Max* normalization, with the minimum value of $-254$ [dBm] and the maximum value of 0 [dBm]. While the upper value is higher than physically reasonable, this pair of values was chosen for convenience to have a 1 [dBm] per gray value step mapping. A smaller (or larger) step size does not have a significant impact on the prediction performance.

The gray value 0 is filled at pixels of building area, which is not our region-of-interest (RoI), while, for our RoI, each pixel is filled with gray values between 1 and 255, which corresponds to $P_{\mathrm{RX}}$. Then, the pathloss map is generated after

scaling the considered map scale into a $256 \times 256$ gray image. Note that the image size ($256 \times 256$) has nothing to do with the grayscale ($0 - 255$).

*2) Interpolation:* Since the RT simulations are carried out over a discrete set of RX locations, and it is computationally challenging to gather the channel information for every available RX location, there is missing channel information in a few pixel locations. To fill the missing part of the pathloss map, we utilize *bilinear interpolation*, which approximates the missing value with a weighted sum of the gray values of the adjacent locations.

*3) Data Augmentation:* Typically, a larger dataset leads to improved performance of NN training. In other words, the larger the data set, the better the outcome. We thus use two augmentation methods - cropping and rotation - to increase the size of our data set.

The entire map data is cropped into images of about a quarter of the size, taking TX as an anchor point. This augments the size of the dataset by a factor of $96$. The image is first cropped as a $64 \times 64$ size image and then upsampled to a $256 \times 256$ size image. Note that some cropped images, not including any TX, are skipped since the TX location will be used as our second input feature. After cropping, the image sets are rotated by $90°$, $180°$, and $270°$, thus increasing the size of the dataset by a further factor of $4$.

*4) Sampling:* In the training and testing of PMNet on the pathloss map dataset, we employ an exclusive division scheme. Specifically, $90\%$ and $10\%$ of images are randomly split into the training and validation set, while the images from the same geographical map belong exclusively to either the training or the validation set. This approach is taken to enhance the generalization performance of PMNet.

## IV. Pathloss Map Prediction

### A. Task (1): Pathloss Map Prediction

We now formulate the prediction task in ML nomenclature. A domain (*i.e.*, wireless channel prediction) is composed of a feature space $\mathcal{X}$, where $x \in \mathcal{X}$. Given the domain, a PMP task is defined as $\mathcal{T} = \{\mathcal{Y}, P(y|\mathbf{x})\}$, which is composed of a label space $\mathcal{Y}$, where $y \in \mathcal{Y}$. Given the task, a dataset is defined as $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$, which is a collection of $|\mathcal{D}| = \mathcal{N}$ channel data that belong to a domain with a task $\mathcal{T}$.

For the PMP task, $\mathcal{X}$ consists of (1) a building map (including terrain, building, and/or foliage) and (2) a TX location and $\mathcal{Y}$ is a Pathloss map. The goal of the PMP task $\mathcal{T}$ is to find a predictive function $f(\cdot)$, which accurately predicts $\mathcal{Y}$ for a given $\mathcal{X}$. It is worth noting that integrating RoI (denoted as $\mathcal{A}^\star$) segmentation with path gain prediction simplifies the PMP task and eliminates the need for separate pre- or post-processing steps for the RoI segmentation for each map. Additionally, this integration helps NN better understand the different IOs in a given building map.

In a nutshell, the PMP task is to predict the pathloss/path gain (and received power $P_{\mathrm{RX}}$ using simple normalization) at RX locations $\boldsymbol{q}_{\mathrm{RX}}$ given TX location $\boldsymbol{q}_{\mathrm{TX}}$ in RoI $\mathcal{A}^\star$. This channel prediction task exploits *site-specific* geographical information, focusing on the large-scale effects in the channel.

We employ a supervised ML method for the PMP task. We train the model on a dataset of RT channel for an area of $\mathcal{A}$, such as the USC dataset in Table I; see Fig. 3 for an overview of the ML-based PMP approach.

### B. Network Architecture

In this subsection, we present the design process of our proposed PMP-oriented NN architecture, referred to as *PMNet*. Our design principles are summarized as follows: (1) several state-of-the-art techniques in the field of image processing are carefully selected and tested, (2) some essential techniques are selected following the concept of ablation study, and (3) the NN with selected techniques is optimized with extensive trials.

*1) Design Choices:* In the PMP task, the NN is required to perform image segmentation to identify the RoI and predict received power within the RoI, while accounting for complex wireless propagation physics. To accomplish this, our proposed PMNet is designed based on such methods, *Encoder-Decoder* and *Atrous convolution*.

*2) Encoder-Decoder:* Encoder-Decoder networks are a widely applied architecture for many computer vision tasks, *e.g.*, object detection [33], human pose estimation [34], and semantic segmentation [35], [36], [37]. The encoder-decoder architecture allows to learn a lower-dimensional representation from a higher-dimensional dataset and utilize the learned representation for various tasks. However, as the encoder shrinks the input feature maps, it may lose essential information, leading to a *bottleneck problem*. Several architectures, including U-Net [38], address the bottleneck problem by adding *skip connections* between the encoder and the decoder parts. Skip connections allow the decoder to access feature maps from the encoder, which helps to propagate context information to higher-resolution layers.

*3) Atrous Convolution: Receptive field* of a convolutional layer is the region of the input feature map that contributes to the output feature map at a given location. The size of the receptive field is determined by the resolution of the input feature map and the field-of-view (FoV) of the filter. There is a logarithmic relationship between the localization accuracy of a model and the size of its receptive field. This means the receptive field size should be sufficient if the given dataset and task are observed with wide FoV. A standard convolutional filter detects a particular feature by sliding over the input feature map, resulting in the output feature map seeing only the adjacent part of the input feature map. In terms of computational complexity, having a wide receptive field with the standard convolutional filter is expensive. Thus, broadly speaking, the receptive field of the standard convolution filter is somewhat narrow, seeing only little context.

Atrous convolution, also known as dilated convolution, is a technique that addresses this limitation [39]. It allows capturing a larger context with a wider FoV by modifying the standard convolution operation. For the two-dimensional case, atrous convolution is applied over the input feature map $f$ to produce the output feature map $g$ at location $\{i, j\}$ using the convolution filter $w$. This operation can be expressed as

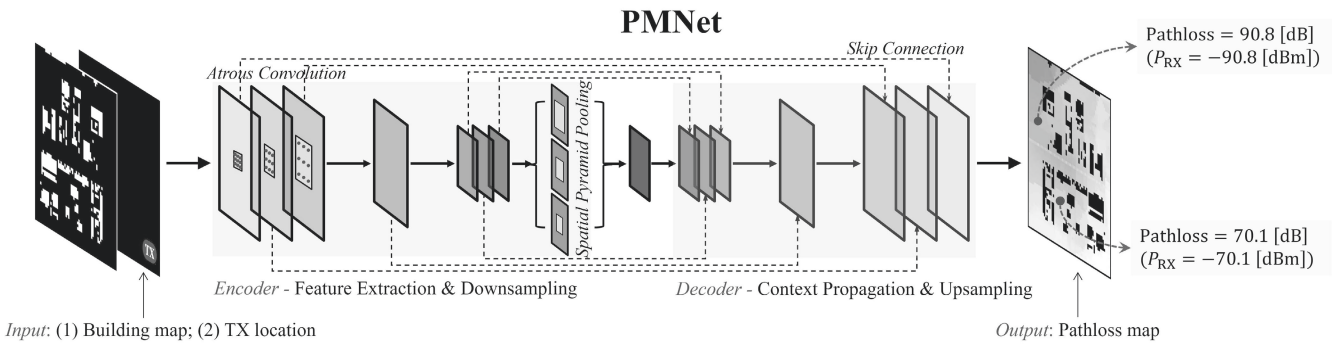



Fig. 3. Overview of the PMP task and the PMNet architecture.

follows:

$$g_{\{i,j\}} = \sum_{m=1}^{k} \sum_{n=1}^{k} f_{\{i+rm,j+rn\}} w_{\{m,n\}}. \tag{11}$$

Here, $k$ represents the kernel size, and $r$ is the atrous rate, which determines the stride level. Notably, the atrous rate $r$ allows to adaptively control the FoV of the filter. For example, an atrous rate of $r = 2$ doubles the FoV of the filter, while an atrous rate of $r = 3$ triples it. The standard convolution can be seen as a special case of (11) where $r = 1$.

In the context of the PMP task, the encoder-decoder architecture of PMNet facilitates efficient context propagation from the encoder to the decoder, while atrous convolution enables it to handle scale variations and capture broader context in map data, setting it apart from other UNet-based networks [15], [16], [17], [18]. The combination of these two features enables PMNet to efficiently and accurately predict pathloss maps, while also accounting for complex wireless propagation physics.

*4) Design Parameters:* PMNet architectures are composed of a stack of *ResLayers*, each containing multiple residual blocks [40]. These ResLayers can be configured with varying numbers of blocks, atrous rates, multi-grids, and output strides. These elements are summarized as follows:

- *Number of blocks*: The number of residual blocks in a ResLayer controls the complexity and depth of the network. Increasing the number of blocks may improve the accuracy of the model, but it also increases the computational cost.
- *Atrous rates*: Atrous rates control the spacing between the convolutions in a ResLayer. Larger atrous rates allow the network to capture more larger spatial contexts in the PMP task.
- *Multi-grids*: Multi-grids allow the network to capture multi-scale information from different levels of the CNN architecture.
- *Output stride*: The output stride of a ResLayer controls the ratio between the resolution of the input image and the output image's resolution. A higher output stride results in a lower-resolution output image. This can be useful to strike a balance between accuracy and speed.

Parameters are optimized through thorough simulations. Note that the impact of output stride in the PMP task is shown in Table IV in Sec. IV-E (*e.g.*, the case of $\frac{H}{8} \times \frac{W}{8}$), demonstrating the most substantial effect compared to other parameters. With these design choices and parameters, PMNet effectively predicts pathloss maps even for different channel datasets (*e.g.*, *RadioMapSeer* [21]). For an architectural overview, please refer to Fig. 3 and Table II. For more details, please see our source code repository.

### C. Training

Table III lists the hyper-parameters that are used for the training of PMNet. We implement the PMNet using PyTorch and use an NVIDIA GeForce RTX 3080 Ti GPU. For more stable training, we normalize the input values into $[0, 1]$ via scaling. During the training, we evaluate the PMNet by mean squared error (MSE) on the validation set at the end of every epoch. For testing, we use the parameters of PMNet with the best MSE score on the validation set. Consequently, the pathloss map for a given map can be generated within a few milliseconds after training.

### D. Evaluation

*1) Root Mean Square Error (RMSE):* RMSE is a widely used loss function in regression analysis and is used as the primary evaluation metric for this task. It measures the overall difference between the prediction $\hat{\boldsymbol{y}}$ and ground-truth $\boldsymbol{y}$ and quantifies the overall accuracy of the model. The formula for RMSE is:

$$\mathrm{RMSE}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \sqrt{\frac{1}{N} \sum_{n=1}^{N} (\hat{y}_n - y_n)^2}, \tag{12}$$

where $\hat{y}_n \in \hat{\boldsymbol{y}}$ and $y_n \in \boldsymbol{y}$ denote predicted and ground-truth gray value (corresponding $P_{\mathrm{RX}}$) at the $n$-th pixel, respectively, and $N$ is the number of pixels in a pathloss map, *i.e.*, $256 \times 256$. The RMSE averaged over all samples is the primary evaluation metric for the PMP task.

*2) RoI Segmentation Error:* The RoI segmentation error, calculated using the intersection over union (IoU) metric, quantifies the accuracy of RoI and non-RoI area segmentation for all pixels in the ground-truth ($\{i, j\}$) and prediction ($\{\hat{i}, \hat{j}\}$) - that is calculated as follows:

$$\text{RoI Segmentation Err.} = \frac{\sum_i \sum_j \mathrm{Err}^{\mathrm{B}}{}_{\{i,j\}}}{\sum_i \sum_j \mathrm{Bld}_{\{i,j\}}}. \tag{13}$$

TABLE II
PMNET ARCHITECTURES AND PARAMETERS. ↓ AND ↑ REPRESENT THE DOWNSAMPLING AND UPSAMPLING LAYERS, RESPECTIVELY

| PMNet | | | | | |
|---|---|---|---|---|---|
| Encoder | | | Decoder | | |
| *#* | *Type* | *Output Size* | *#* | *Type* | *Output Size* |
| Input | Image | $2 \times 256 \times 256$ | Output | Image | $1 \times 256 \times 256$ |
| 1(↓) | Conv2d, MaxPool2d | $64 \times 65 \times 65$ | 1(↑) | Conv2d | $(128+2) \times 256 \times 256$ |
| 2 | ResLayer | $256 \times 65 \times 65$ | 2 | Conv2d | $(256+64) \times 65 \times 65$ |
| 3(↓) | ResLayer | $512 \times 33 \times 33$ | 3 | Conv2d | $(256+256) \times 65 \times 65$ |
| 4(↓) | ResLayer | $512 \times 17 \times 17$ | 4(↑) | ConvTranspose2d | $(256+256) \times 65 \times 65$ |
| 5 | ResLayer | $1024 \times 17 \times 17$ | 5(↑) | ConvTranspose2d | $(512+512) \times 33 \times 33$ |
| 6 | Conv2d, AdaptiveAvgPool2d | $512 \times 17 \times 17$ | 6 | Conv2d | $(512+512) \times 17 \times 17$ |

TABLE III
TRAINING CONFIGURATION AND HYPER-PARAMETERS FOR PMNET TRAINING

| ***Model*** | PMNet |
|---|---|
| ***Dataset*** (USC) | |
| Map | USC campus |
| Split for training (test) set | 90% (10%) of dataset |
| ***Hyper-parameter*** | |
| Learning rate (LR) | $10^{-3} \sim 5 \times 10^{-4}$ |
| LR gamma, step size | 0.5, 10 |
| Batch size | $16 \sim 32$ |
| Optimizer | Adam |
| # of of epochs | 50 |

Here, $\mathrm{Err}^{\mathrm{B}}{}_{\{i,j\}}$ and $\mathrm{Bld}_{\{i,j\}}$ are defined as:

$$\mathrm{Err}^{\mathrm{B}}{}_{\{i,j\}} = \begin{cases} 1, & \{i,j\} \in \mathcal{B} \text{ and } \{\hat{i},\hat{j}\} \in \mathcal{A}^{\star} \\ 1, & \{i,j\} \in \mathcal{A}^{\star} \text{ and } \{\hat{i},\hat{j}\} \in \mathcal{B} \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

$$\mathrm{Bld}_{\{i,j\}} = \begin{cases} 1, & \{i,j\} \in \mathcal{B} \\ 0. & \text{otherwise} \end{cases} \tag{15}$$

Within a given map, the non-RoI area, denoted as black (gray value 0), is represented by $\mathcal{B}$, while the RoI area, denoted as non-black (grayscale $1-255$), is represented by $\mathcal{A}^{\star}$. $\mathcal{B}$ and $\mathcal{A}^{\star}$ are complementary set within $\mathcal{A}$. $\mathcal{B}$ can include buildings, foliage, and/or small objects.

*3) Channel Prediction Error:* Channel prediction error directly evaluates path gain accuracy for pixels within the RoI area, evaluating power in [dBm] (or path gain in [dB]) unlike RMSE, which quantifies differences based on gray values.

To calculate channel prediction error, gray values within the RoI area of both the predicted and ground-truth pathloss maps are converted into corresponding received power values. The RMSE formula is then applied to these power values:

$$\mathrm{RMSE}(\hat{\boldsymbol{p}}, \boldsymbol{p}) = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\hat{p}_n - p_n)^2}, \tag{16}$$

where $\hat{p}_n \in \hat{\boldsymbol{p}}$ and $p_n \in \boldsymbol{p}$ represent the predicted and ground-truth $P_{\mathrm{RX}}$ at the $n$-th pixel, respectively. Channel Prediction Error is then computed by averaging $\mathrm{RMSE}(\hat{\boldsymbol{p}}, \boldsymbol{p})$ across all given samples.

### *E. Simulation Result*

*1) Training Optimization:* Table IV presents an ablation study to identify the factors that significantly contribute to PMNet's performance in the PMP task, such as *data augmentation* and *feature map size*.[9]

*2) Impact of Data Augmentation:* For the data augmentation, we do horizontal, vertical and diagonal flips. In other words, including the original images, we use the $\times 4$ number of images for training. Note that data augmentation has several advantages in general: first, it enhances the diversity of the training data by generating additional examples that capture various variations of the original data. Second, it reduces overfitting by exposing the model to a wider range of input patterns. Finally, data augmentation helps to make the model more robust to noise and variability in the input data. As shown in Table IV, it improves the performance of PMNet by $15.7\%$ in terms of RMSE.

*3) Impact of Feature Map Size:* We analyze the performances of PMNet according to the size of the feature map, which is the output of the encoder. Table IV compares the results with the feature map sizes $\frac{H}{8} \times \frac{W}{8}$ and $\frac{H}{16} \times \frac{W}{16}$, where $H$ and $W$ are the height and width of an input image, respectively. To adjust the feature map size, we modify the strides of the convolution layers in the encoder. We employ the feature map size of $\frac{H}{8} \times \frac{W}{8}$ as the default option, because PMNet yields better performance with the feature map size of $\frac{H}{8} \times \frac{W}{8}$ than that of $\frac{H}{16} \times \frac{W}{16}$.

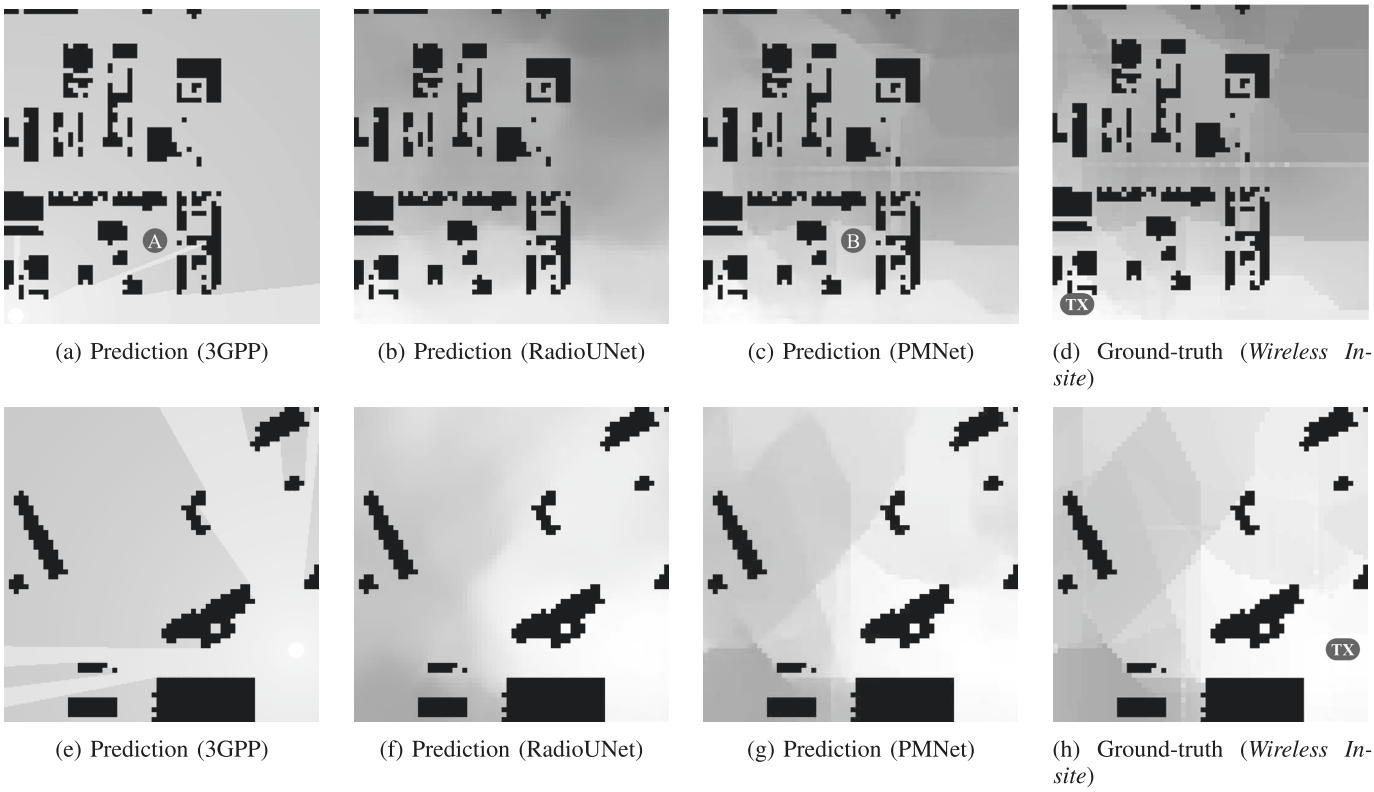
*4) Accuracy:* We compare the ML-based PMP with our proposed PMNet model to two other methods for the PMP task: a model-based approach, *3GPP*, and an ML-based approach, *RadioUNet*. All three methods produce a single-channel $256 \times 256$ image of the pathloss map as the output, given the input of a two-channel $256 \times 256$ image containing the geographical map and the TX location. Here are the details of these baseline methods:

1) **3GPP (with map info.)** As discussed in Sec. III-A.2, the 3GPP model determines the pathloss at a particular location based on the Euclidean distance and whether the link between the TX and RX is in LoS or NLoS. To ensure a fair comparison with other baselines, we utilize map information to determine the LoS or NLoS condition of

[9]Our extensive experiments tested other factors, such as different sampling methods, training loss functions, and additional input features (*e.g.*, TX distance heatmap), but these factors did not show a meaningful improvement to justify the additional complexity.

TABLE IV
ABLATION STUDY FOR PMNET TRAINING OPTIMIZATION. LOWER VALUES INDICATE BETTER PERFORMANCE

| *Case* | *Data Aug. ($\times 4$)* | *Feature Size* | *RMSE*$\downarrow$ | *RoI Segmentation Err.*$\downarrow$ | *Channel Prediction Err.*$\downarrow$ |
|---|---|---|---|---|---|
| w/o Data-Aug. | ✗ | $\frac{H}{16} \times \frac{W}{16}$ | 0.01637 | 0.00263 | 0.01860 |
| w/ Data-Aug. | ✓ | $\frac{H}{16} \times \frac{W}{16}$ | 0.01259 | 0.00025 | 0.01403 |
| $\frac{H}{8} \times \frac{W}{8}$ | ✓ | $\frac{H}{8} \times \frac{W}{8}$ | 0.01057 | 0.00096 | 0.01175 |




(a) Prediction (3GPP) (b) Prediction (RadioUNet) (c) Prediction (PMNet) (d) Ground-truth (*Wireless Insite*)

(e) Prediction (3GPP) (f) Prediction (RadioUNet) (g) Prediction (PMNet) (h) Ground-truth (*Wireless Insite*)

Fig. 4. Comparison of the predicted pathloss map of 3GPP, RadioUNet, and PMNet. TX in ground-truth represents the TX location. The scenes are randomly selected, not cherry-picked.

specific pixels to the TX.[10] Note that it does not require any NN training as it is a model-based approach.

2) **RadioUNet** [15] is an ML-based PMP method that extends the UNet architecture by employing two UNets. Each UNet comprises 8 encoder layers with convolution, ReLU, and Maxpool layers, followed by 8 decoder layers with transposed convolution and ReLU layers. The encoders and decoders are concatenated, as in the original UNet architecture. Here, RadioUNet employs curriculum training to enhance training: in the first stage, the first UNet is trained for a specific number of epochs, with the second UNet frozen. In the second stage, the second UNet is trained using the two-channel input features and the output of the first UNet, effectively making it a three-channel input network.

3) **PMNet (Proposed)** is our proposed ML-based PMP method. This network employs several parallel atrous convolutions with different rates and the encoder-decoder network. The encoder consists of 6 ResNet-based layers. Each ResNet layer comprises several bottleneck layers consisting of convolution, batch normalization, max pooling, and ReLU. The decoder consists of 6 layers consisting of convolution, adaptive average pooling, ReLU, transposed convolution, and ReLU. Skip connections are used between encoders and decoders.

[10]The original 3GPP pathloss model uses a probabilistic model to determine LoS/NLoS condition at a particular distance. However, to ensure a fair comparison, we use here the deterministic LoS/NLoS condition determined from the map information in calculating the pathloss gain.

*5) Qualitative Analysis:* Fig. 4 shows the prediction results of the baselines. Recall that each pixel in the RoI corresponds to the predicted received power $P_{\mathrm{RX}}$ (or the path gain $\mathrm{PG}$). Note that some pixel values in the ground-truth data appear noisy due to interpolation during the gray conversion process after RT simulation.

3GPP exhibits a substantial deviation from ground-truth obtained through RT simulation, highlighting the differences between how RT simulation and 3GPP model calculate a pathloss. Specifically, for RX locations with LoS conditions close to the TX, the results obtained using the 3GPP model approximately match the ground-truth data obtained from *Wireless Insite*. However, for RX locations farther from the TX

TABLE V
COMPARISON STUDY FOR PMP SCHEMES (3GPP, RADIOUNET, AND PMNET). LOWER VALUES INDICATE BETTER PERFORMANCE, AND THE LOWEST ERRORS ARE HIGHLIGHTED

| *Scheme* | *ML-based* | *RMSE*↓ | *RoI Segmentation Err.*↓ | *Channel Prediction Err.*↓ |
|---|---|---|---|---|
| 3GPP (with map info.) [20] | ✗ | 15.9451 | - | 17.5973 |
| RadioUNet [15] | ✓ | 0.02634 | 0.00840 | 0.01249 |
| PMNet | ✓ | **0.01057** | **0.00096** | **0.01175** |

TABLE VI
NUMERICAL RESULTS OF PMNET ON AN UNSEEN NETWORK SCENARIO. PMNET WAS TRAINED ON THE USC DATASET AND EVALUATED ON THE UCLA AND BOSTON DATASET

| *Case* | *Model* | *Train Data* | *Eval. Data* | *RMSE*↓ | *RoI Segmentation Err.*↓ | *Channel Prediction Err.*↓ |
|---|---|---|---|---|---|---|
| Vanilla | PMNet | USC | USC | 0.01057 | 0.00096 | 0.01175 |
| Cross-scenario (UCLA) | PMNet | USC | UCLA | 0.19146 | 0.03925 | 0.21700 |
| Cross-scenario (Boston) | PMNet | USC | Boston | 0.25842 | 0.04602 | 0.32436 |

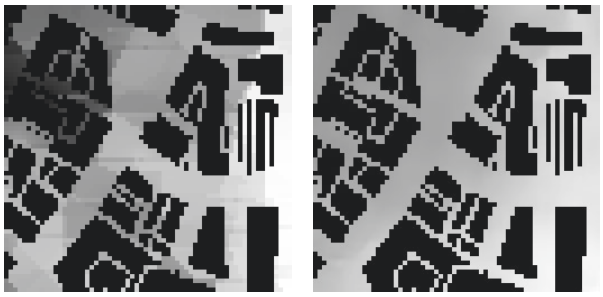
or under NLoS conditions, the 3GPP model exhibits significant discrepancy from the ground-truth data. It is worth noting that the 3GPP pathloss model does not provide results for near-field within a link distance of 10 meters; so, we arbitrarily set the power in the near-field area to gray value 255, which does not introduce significant errors. The 3GPP pathloss model is a simplified model that does not account for the complex wireless propagation physics of reflection, diffraction, and scattering (highlighted in Ⓐ). Instead, it relies solely on two models for LoS and NLoS locations, respectively, and only considers link distance and carrier frequency. This simplified approach inevitably leads to significant inaccuracies in the pathloss prediction.

RadioUNet demonstrates impressive RoI segmentation results, while its channel prediction outputs appear somewhat blurry. It is worth noting that RadioUNet conducts curriculum-based training with 50 epochs each in the first and second stages, utilizing the same training/validation set as PMNet, which is trained with a total of 50 epochs.

PMNet, on the other hand, achieves notable results for both RoI segmentation and channel prediction. As highlighted in Ⓑ, PMNet effectively captures the intricate wireless propagation physics of reflection, diffraction, and scattering. This can be attributed to PMNet's ability to incorporate a broader contextual understanding of the environment, enabling it to capture the representation of wireless propagation physics in the surrounding environment.

*6) Quantitative Analysis:* Table V compares our proposed PMNet model to the model-based 3GPP method and the ML-based RadioUNet method in terms of three accuracy metrics for the PMP task: RMSE, RoI segmentation error, and channel prediction error. Note that the ground-truth dataset is made by RT simulation; therefore, the error shows the difference between a scheme and the RT simulation.

The model-based 3GPP method has inferior results compared to ML-based methods, which can be explained by the oversimplifications inherent in this model, as discussed above. While our proposed PMNet model achieves the best score on all three metrics, another ML-based PMP method, RadioUNet, also achieves high accuracy ($RMSE \leq 0.03$). This result highlights the capability of ML-based PMP approaches to learn a representation of the wireless propagation physics implicit in the ground-truth RT channel data.



(a) Ground-truth (*Wireless Insite*) (b) Prediction (PMNet)

Fig. 5. Prediction results of PMNet on an unseen network scenario (*i.e.*, cross-scenario evaluation). The model is trained on the USC dataset and evaluated on the Boston dataset.

## V. TRANSFERABLE PATHLOSS MAP PREDICTION

### A. Challenge: PMP for Unseen Network Scenario

As demonstrated in the previous section, PMNet exhibits high accuracy of the PMP task for a given dataset. However, minimizing re-training efforts for new network scenarios remains a challenge. To evaluate PMNet's generalizability across different scenarios, we conducted a cross-scenario evaluation, testing PMNet trained on USC data on the Boston dataset.

As shown in Fig. 5 and Table VI, the PMNet achieves the RoI segmentation error on the order of $10^{-2}$ and the channel prediction error on the order of $10^{-1}$ in a new scenario. Such deterioration is due to differences in network configuration and environmental characteristics between the two scenarios (*e.g.*, different map scales and geographical features). This highlights the need for further development to improve PMNet's performance across different network scenarios, a task we refer to as *cross-scenario PMP*.

### B. Task (2): Cross-Scenario PMP

To enable better performance, we now allow cross-scenario PMP to *improve* the model trained on a different network scenario through training with a *reduced-size* training in the new scenario. This will allow the network to adapt to the new scenario with less time and resource effort, while maintaining high accuracy. To address this challenge, we leverage *transfer learning* (TL).

*1) Approach: Transfer Learning:* TL is an ML technique that allows knowledge transfer from one task or dataset to another, reducing the amount of data and training time required for new scenarios. In the context of cross-scenario PMP, we can transfer the knowledge from the source scenario, which learns a predictive function $f_S(\cdot)$ from a source dataset $\mathcal{D}_S$ (*e.g.*, USC), to the target scenario, which learns a predictive function $f_T(\cdot)$ from a target dataset $\mathcal{D}_T$ (*e.g.*, UCLA and Boston).

There are two main ways to use TL for the cross-scenario PMP.

- *Feature extraction*: We can train a feature extractor on a source scenario and then use that feature extractor to extract features from data from a target scenario. Once we have extracted the features, we can train a simple model (*e.g.*, a linear regressor) to predict the pathloss map for the target scenario.
- *Fine-tuning*: We can fine-tune a pre-trained model on the target scenario. This can be done by unfreezing some or all of the layers of the pre-trained model and training the model on data from the target scenario.

The choice between those two methods depends on a number of factors, including the size and complexity of the pre-trained model, the availability of training data for the target dataset, and the computational resources available.

In this work, we focus on the fine-tuning TL approach with all of the layers of the pre-trained model unfrozen.[11] This approach is simple yet effective, achieving higher accuracy on various cross-scenario PMP tasks with less training data and shorter training time, as elaborated in the following subsection.

We prepare and use the following pre-trained models in our experiments:

(i) **VGG16$_{\mathbf{ImgNet}}$** is the pre-trained CNN model trained on the ImageNet dataset, which contains 140k images belonging to 22k categories. It is a powerful image classification model that has been used to achieve state-of-the-art results on a variety of image classification benchmarks.

(ii) **PMNet$_{\mathbf{3gpp}}$** is the pre-trained PMNet model trained on the 3GPP pathloss map dataset. The 3GPP pathloss map dataset is prepared with the 3GPP pathloss model in [20] (see 3GPP in Sec. IV-E.4, Fig. 4, and Table V).

(iii) **PMNet$_{\mathbf{usc}}$** is the pre-trained PMNet model trained on the USC RT dataset. It is similar to PMNet$_{\mathbf{3gpp}}$ but is trained on a different dataset. This is our main pre-trained model.

Each pre-trained model is available on our GitHub page.

[11] While we have performed sample experiments with unfreezing certain layers, such as the encoder-frozen and decoder-unfrozen, performance did not improve significantly. A more comprehensive investigation of this topic is, however, beyond the scope of this paper.

TABLE VII
TRAINING CONFIGURATION AND HYPER-PARAMETERS IN CROSS-SCENARIO PMP

| | |
|---|---|
| ***Model*** | |
| Backbone | PMNet, VGG16 |
| Pre-trained model | PMNet$_{\mathrm{usc}}$, PMNet$_{\mathrm{3gpp}}$, VGG16$_{\mathrm{ImgNet}}$ |
| ***Dataset*** (UCLA, Boston) | |
| Map | UCLA campus, Boston |
| Split for training (test) set | $10\% \sim 90\%$ ($10\%$) of dataset |
| ***Hyper-parameter*** | |
| LR | $10^{-3} \sim 5 \times 10^{-4}$ |
| LR gamma, step size | 0.5, 10 |
| Batch size | 16 |
| Optimizer | Adam |
| # of of epochs | 50 |

TABLE VIII
IMPACT OF TL ON TRAINING SPEED ($= \frac{1}{\text{steps}}$). PMNET MODELS WITH OR WITHOUT PMNET$_{\mathrm{usc}}$ PRE-TRAINED MODEL ARE TRAINED AND EVALUATED ON THE BOSTON DATASET

| *Case* | *# of Required Step (Training Speed)* | |
|---|---|---|
| | ***RMSE*** $\leq 0.1$ | ***RMSE*** $\approx 0.03$ |
| Vanilla (90% Data) | 5841 ($\times\mathbf{1.0}$) | 6195 ($\times\mathbf{1.0}$) |
| PMNet$_{\mathrm{usc}}$ (20% Data) | 1040 ($\times\mathbf{5.6}$) | 1520 ($\times\mathbf{4.1}$) |

### C. Simulation Results

As demonstrated in the cross-scenario evaluation results (in Fig. 5 and Table VI), there is a need for further development to make PMNet adapt to different network scenarios. To this end, our approach is fine-tuning a pre-trained model with down-sized data for the new scenario. Here, the main questions in performing cross-scenario PMP are: (1) How quickly and with how minimal data PMNet can effectively adapt to new scenarios; and (2) Which pre-trained model should be utilized for optimal performance in cross-scenario PMP.

*1) Efficiency:* For cross-scenario PMP, rapidly adapting PMNet models to new network scenarios using limited data is essential due to the time-consuming and expensive nature of channel measurement using RT simulation or channel sounder. This is particularly critical for applications like beam management and localization using ML-based PMP, which demand quick adjustments for new scenarios.

*2) Impact of TL:* TL can significantly improve the training speed of PMNet models for cross-scenario PMP. As shown in Fig. 6 and Table VIII, the TL case with the PMNet$_{\mathrm{usc}}$ pre-trained model achieves a given level of accuracy much faster even with much less amount of training data. In particular, PMNet$_{\mathrm{usc}}$ achieves the same level of accuracy (*RMSE* $\leq 0.1$ and *RMSE* $\approx 0.03$) $\times 5.6$ and $\times 4.1$ faster, respectively, as the Vanilla case (highlighted in Ⓒ), where we define as "Vanilla" the training from scratch in a particular environment.

Furthermore, the TL can also significantly save the required amount of data for cross-scenario PMP. As shown in Fig. 7, the TL (PMNet$_{\mathrm{usc}}$) trained with about $20\%$ of the Boston dataset achieves equivalent results to the Vanilla case trained with about $90\%$ of the dataset.

It is worth noting that limited training data can easily induce overfitting, as observed in the Vanilla case with $20\%$ Data (highlighted in Ⓓ). For the same amount of new scenario

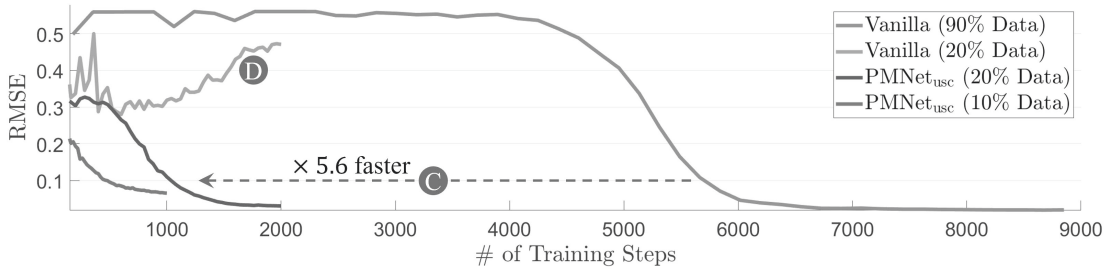



Fig. 6. Comparison of the training efficiency of PMNet models with and without TL. PMNet models are trained for 50 epochs and evaluated on the Boston dataset.

TABLE IX

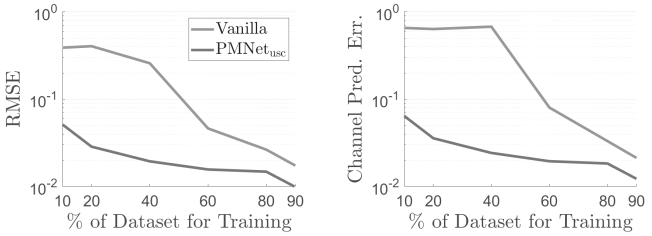
COMPARISON OF PRE-TRAINED MODELS (VGG16$_{\text{ImgNet}}$, PMNET$_{\text{3gpp}}$, AND PMNET$_{\text{usc}}$) IN TERMS OF ACCURACY. MODELS ARE EVALUATED ON THE UCLA AND BOSTON DATASETS, USING 90% OF THE DATA FOR TRAINING AND 10% OF THE DATA FOR VALIDATION. 50 EPOCHS ARE USED FOR TRAINING. LOWER VALUES INDICATE BETTER PERFORMANCE, AND THE LOWEST ERRORS ARE HIGHLIGHTED

| *Case* | *Pre-training* | *Model* | *RMSE*↓ | *RoI Segmentation Err.*↓ | *Channel Prediction Err.*↓ |
|---|---|---|---|---|---|
| Vanilla | ✗ | PMNet | 0.03415 | 0.02935 | 0.03844 |
| TL (ImageNet) | ✓ (ImageNet) | VGG16 | 0.04528 | 0.01814 | 0.05108 E |
| TL (3GPP) | ✓ (3GPP) | PMNet | 0.02809 | **0.00655** | 0.03238 |
| TL (USC) | ✓ (USC) | PMNet | **0.02792** | 0.01666 | **0.03145** |

(a) UCLA

| *Case* | *Pre-training* | *Model* | *RMSE*↓ | *RoI Segmentation Err.*↓ | *Channel Prediction Err.*↓ |
|---|---|---|---|---|---|
| Vanilla | ✗ | PMNet | 0.01736 | 0.02417 | 0.02125 |
| TL (ImageNet) | ✓ (ImageNet) | VGG16 | 0.01999 | **0.02040** | 0.02512 |
| TL (3GPP) | ✓ (3GPP) | PMNet | 0.01762 | 0.04030 | 0.02187 |
| TL (USC) | ✓ (USC) | PMNet | **0.00987** | 0.03530 | **0.01225** |

(b) Boston




(a) RMSE (b) Channel prediction error

Fig. 7. Impact of TL on training data requirements. PMNet models with or without PMNet$_{\text{usc}}$ pre-trained model are trained with 50 epochs and evaluated on the Boston dataset.

data, the TL case (PMNet$_{\text{usc}}$ (20%)) does not experience the overfitting issue. This suggests that TL also enhances training stability (less overfitting issue with limited data) in cross-scenario PMP.

Our findings demonstrate that the pre-trained PMNet$_{\text{usc}}$ model efficiently accelerates the training process by leveraging its knowledge of PMP tasks, including the physics of wireless channel propagation and RoI segmentation, and this model can be readily adapted to new scenarios with minimal data and training steps.

Consequently, we confirm that fine-tuning with a *suitable* pre-trained model is an effective cross-PMP task method. Another key question is which pre-trained model is suitable and which is not, which is discussed further in the following.

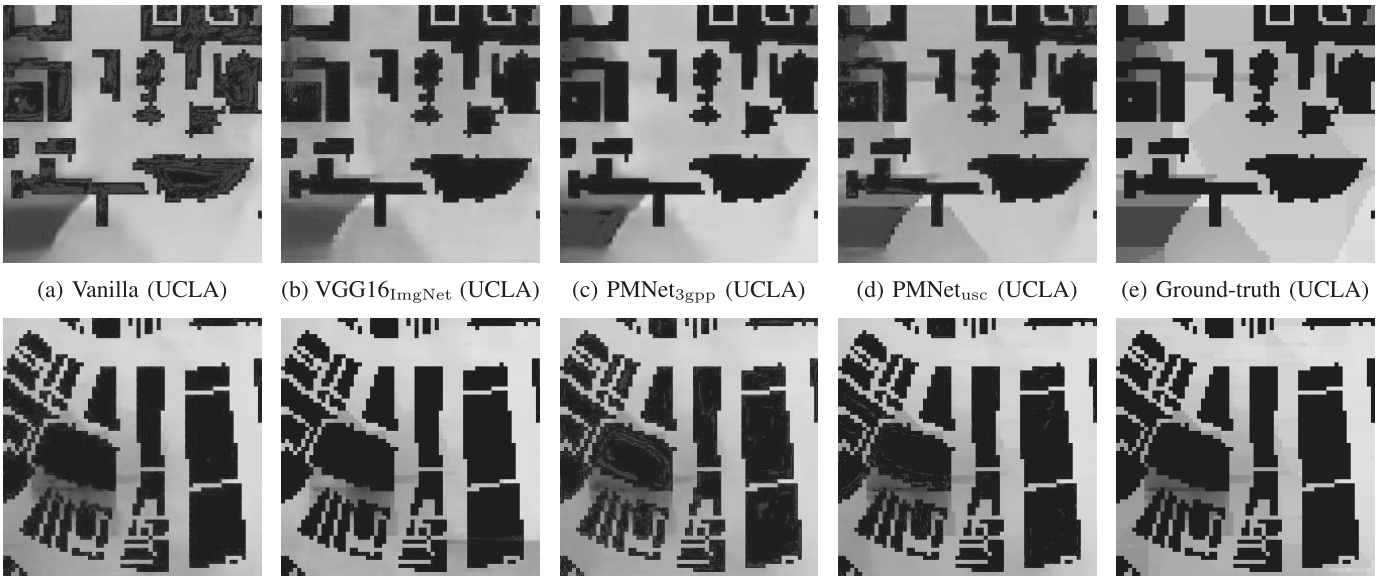
*3) Accuracy:* As discussed in Sec. V, the source and target scenario (task or domain) should be sufficiently similar for effective TL to occur. For instance, to successfully apply TL to the target task of predicting wireless communication channels, the NN should extract relevant features of wireless propagation physics from the source task.

*4) "Suitable" Pre-Trained Model:* Table IX compares the performance of the PMNet model with and without TL. The baseline model, referred to as Vanilla, is trained without any TL (without any pre-trained model). Additionally, we compare the performance of TL using a pre-trained model trained on an unrelated source scenario (*i.e.*, VGG16 trained on ImageNet) with TL using a pre-trained model trained on a related source scenario (*i.e.*, PMNet trained on USC or 3GPP datasets).

As shown in Table IX, both PMNet models trained on PMNet$_{\text{usc}}$ and PMNet$_{\text{3gpp}}$ outperform the Vanilla case on all performance metrics, suggesting that using a pre-trained model trained on a related source task can significantly improve accuracy.

Interestingly, while the VGG16 model trained on ImageNet (VGG16$_{\text{ImgNet}}$) outperforms the Vanilla for RoI segmentation, it fails to do so for channel prediction (highlighted in E). This discrepancy stems from the VGG16 pre-trained model, which has an inherent understanding of segmentation and image representation from its source task; however, does not have any knowledge of the physics of wireless propagation.

Fig. 8 visually confirms the findings from Table IX. All models achieve high accuracy for RoI segmentation, while



(a) Vanilla (UCLA) (b) $\text{VGG16}_{\text{ImgNet}}$ (UCLA) (c) $\text{PMNet}_{\text{3gpp}}$ (UCLA) (d) $\text{PMNet}_{\text{usc}}$ (UCLA) (e) Ground-truth (UCLA)

(f) Vanilla (Boston) (g) $\text{VGG16}_{\text{ImgNet}}$ (Boston) (h) $\text{PMNet}_{\text{3gpp}}$ (Boston) (i) $\text{PMNet}_{\text{usc}}$ (Boston) (j) Ground-truth (Boston)

Fig. 8. Comparison of the prediction results of pre-trained models ($\text{VGG16}_{\text{ImgNet}}$, $\text{PMNet}_{\text{3gpp}}$, and $\text{PMNet}_{\text{usc}}$). 50 epochs are used for training. Brighter colors indicate higher PG. Note that pixels with non-zero gray value are converted to color to highlight differences between results.

only the TL case using a pre-trained model trained on a related source scenario (*e.g.*, $\text{PMNet}_{\text{3gpp}}$ and $\text{PMNet}_{\text{usc}}$) achieves high accuracy for channel prediction, capturing subtle details of the wireless propagation physics. This suggests that our PMNet pre-trained model is generalizable to different scenarios due to its inherent knowledge of channel propagation representation, and that TL can further improve accuracy.

These results empirically demonstrate that pre-trained model's source dataset (task or domain) should be similar to the target dataset (task or domain) to transfer useful information during TL. Specifically, for cross-scenario PMP, it is important to use a pre-trained model that has been trained extensively on data related to wireless propagation physics.

Therefore, we conclude that the suggested TL approach, fine-tuning with a stable and closely related pre-trained model (such as $\text{PMNet}_{\text{usc}}$), is a simple yet effective way to address the cross-scenario PMP task, which is important for practical applications.

## VI. Conclusion

This work introduces an ML-based large-scale channel prediction framework, PMNet, which can create highly accurate pathloss predictions for a given map in a few milliseconds. Utilizing an RT channel dataset of real-world scenarios (*e.g.*, USC, UCLA, and Boston area), PMNet is verified for its accuracy and training efficiency. In particular, TL with our PMNet pre-trained model, which has generalization capability for different network scenarios, enables the PMNet to adapt itself quickly and efficiently to a new network scenario, while achieving an RMSE of $10^{-2}$ level.

The high accuracy and low runtime of the PMNet framework make it suitable for deployment planning in dense networks as well as online optimization of network parameters.

Still, it remains an open question whether the knowledge of wireless propagation physics in our PMNet pre-trained model can be transferred to other downstream tasks beyond the PMP task; this question will be the topic of our future research.

## Acknowledgment

The help of Dr. Zheda Li in the creation of the USC data set is gratefully acknowledged.

## References

[1] J.-H. Lee, J. Lee, S.-H. Lee, and A. F. Molisch, "PMNet: Large-scale channel prediction system for ICASSP 2023 first pathloss radio map prediction challenge," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–2.

[2] J.-H. Lee, O. G. Serbetci, D. P. Selvam, and A. F. Molisch, "PMNet: Robust pathloss map prediction via supervised learning," in *Proc. IEEE Global Commun. Conf.*, Dec. 2023, pp. 1–20.

[3] X. Lin, L. Kundu, C. Dick, E. Obiodu, T. Mostak, and M. Flaxman, "6G digital twin networks: From theory to practice," *IEEE Commun. Mag.*, vol. 1, no. 1, pp. 1–7, Aug. 2023.

[4] A. Alkhateeb, S. Jiang, and G. Charan, "Real-time digital twins: Vision and research directions for 6G and beyond," *IEEE Commun. Mag.*, vol. 61, no. 11, pp. 128–134, Nov. 2023.

[5] R. Valenzuela, "A ray tracing approach to predicting indoor wireless transmission," in *Proc. IEEE 43rd Veh. Technol. Conf.*, Apr. 1993, pp. 214–218.

[6] S.-C. Kim et al., "Radio propagation measurements and prediction using three-dimensional ray tracing in urban environments at 908 MHz and 1.9 GHz," *IEEE Trans. Veh. Technol.*, vol. 48, no. 3, pp. 931–946, May 1999.

[7] *Wireless Insite*. Accessed: Sep. 2024. [Online]. Available: https://www.remcom.com/wirelessinsite

[8] (2024). *Path Loss Comparison in Rosslyn*. [Online]. Available: https://www.remcom.com/examples/path-loss-comparison-in-rosslyn-virginia.html

[9] V. Degli-Esposti, F. Fuschini, E. M. Vitucci, and G. Falciasecca, "Measurement and modelling of scattering from buildings," *IEEE Trans. Antennas Propag.*, vol. 55, no. 1, pp. 143–153, Jan. 2007.

[10] R. Wahl, G. Wolfle, P. Wildbolz, and F. Landstorfer, "Dominant path prediction model for urban scenarios," in *Proc. 14th IST Mobile Wireless Commun. Summit*, 2005, pp. 1–26.

[11] M. Lee and D. Han, "Voronoi tessellation based interpolation method for Wi-Fi radio map construction," *IEEE Commun. Lett.*, vol. 16, no. 3, pp. 404–407, Mar. 2012.
[12] D. Romero and S.-J. Kim, "Radio map estimation: A data-driven approach to spectrum cartography," 2022, *arXiv:2202.03269*.
[13] T. Orekondy, P. Kumar, S. Kadambi, H. Ye, J. Soriaga, and A. Behboodi, "WiNeRT: Towards neural ray tracing for wireless channel modelling and differentiable simulations," in *Proc. Int. Conf. Learn. Represent.*, 2023, pp. 1–20.
[14] X. Zhao, Z. An, Q. Pan, and L. Yang, "NeRF2: Neural radio-frequency radiance fields," in *Proc. 29th Annu. Int. Conf. Mobile Comput. Netw.*, vol. 33, Oct. 2023, pp. 1–15.
[15] R. Levie, Ç. Yapar, G. Kutyniok, and G. Caire, "RadioUNet: Fast radio map estimation with convolutional neural networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 4001–4015, Jun. 2021.
[16] V. V. Ratnam et al., "FadeNet: Deep learning-based mm-Wave large-scale channel fading prediction and its applications," *IEEE Access*, vol. 9, pp. 3278–3290, 2021.
[17] E. Krijestorac, H. Sallouha, S. Sarkar, and D. Cabric, "Agile radio map prediction using deep learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–2.
[18] K. Qiu, S. Bakirtzis, H. Song, I. Wassell, and J. Zhang, "Deep learning-based path loss prediction for outdoor wireless communication systems," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–2.
[19] Ç. Yapar, F. Jaensch, R. Levie, G. Kutyniok, and G. Caire, "The first pathloss radio map prediction challenge," 2023, *arXiv:2310.07658*.
[20] *Study on Channel Model for Frequencies From 0.5 to 100 GHz*, Standard TR 38.901, 2018.
[21] Ç. Yapar, R. Levie, G. Kutyniok, and G. Caire, "Dataset of pathloss and ToA radio maps with localization application," 2022, *arXiv:2212.11777*.
[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[23] A. F. Molisch, *Wireless Communications: From Fundamentals to Beyond 5G*, 3rd ed., Hoboken, NJ, USA: Wiley, 2023.
[24] F. Fuschini, H. El-Sallabi, V. Degli-Esposti, L. Vuokko, D. Guiducci, and P. Vainikainen, "Analysis of multipath propagation in urban environment through multidimensional measurements and advanced ray tracing simulation," *IEEE Trans. Antenna Propag.*, vol. 56, no. 3, pp. 848–857, Mar. 2008.
[25] T. Rautiainen, G. Wolfle, and R. Hoppe, "Verifying path loss and delay spread predictions of a 3D ray tracing propagation model in urban environment," in *Proc. IEEE 56th Veh. Technol. Conf.*, vol. 4, Jun. 2002, pp. 2470–2474.
[26] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.
[27] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–20.
[28] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" 2014, *arXiv:1411.1792*.
[29] K. James et al., "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.
[30] S. Kornblith, J. Shlens, and Q. V. Le, "Do better ImageNet models transfer better?" 2018, *arXiv:1805.08974*.
[31] M. Huh, P. Agrawal, and A. A. Efros, "What makes ImageNet good for transfer learning?" 2016, *arXiv:1608.08614*.
[32] M. Steinbauer, A. F. Molisch, and E. Bonek, "The double-directional radio channel," *IEEE Antennas Propag. Mag.*, vol. 43, no. 4, pp. 51–63, Aug. 2001, doi: 10.1109/74.951559.
[33] R. Araki, T. Onishi, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "MT-DSSD: Deconvolutional single shot detector using multi task learning for object detection, segmentation, and grasping detection," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 10487–10493.
[34] T. Xu and W. Takano, "Graph stacked hourglass networks for 3D human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1–23.
[35] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with atrous separable convolution for semantic image segmentation," in *Proc. ECCV*, 2018, pp. 1–27.
[36] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5168–5177.
[37] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters—Improve semantic segmentation by global convolutional network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1743–1751.
[38] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*. Cham, Switzerland: Springer, 2015, pp. 234–241.
[39] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. ICLR*, 2016, pp. 1–29.
[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

**Ju-Hyung Lee** (Member, IEEE) received the Ph.D. degree from Korea University, Seoul, South Korea. He was a Post-Doctoral Researcher in electrical and computer engineering with the University of Southern California (USC), Los Angeles, CA, USA, where he worked under the supervision of Prof. Andreas F. Molisch. Before that, he was a Research Professor with Korea University. He is currently a Principal Researcher with Nokia, Sunnyvale, CA, USA. His research interests include generative AI solutions for wireless communication systems and on-device AI/LLM. He has been recognized with several prestigious awards, including first rank in a machine learning (ML) competition, best paper awards at key IEEE conferences, and the Grand Prize for Research Excellence from Korea University. For more information visit the link (https://juhyung-lee.com/) and (https://juhyung-lee.com/).

**Andreas F. Molisch** (Fellow, IEEE) received the Dipl. (Ing.), Ph.D., and Habilitation degrees from Technical University Vienna, Austria, in 1990, 1994, and 1999, respectively.

He spent the next ten years in industry, FTW, AT&T (Bell) Laboratories, and Mitsubishi Electric Research Laboratories (where he rose to the Chief Wireless Standards Architect). In 2009, he joined the University of Southern California (USC), Los Angeles, CA, USA, as a Professor and founded the Wireless Devices and Systems (WiDeS) Group. In 2017, he was appointed to the Solomon Golomb—Andrew and Erna Viterbi Chair. His research interests revolve around wireless propagation channels, wireless systems design and their interaction, wireless channel measurement and modeling for 5G and 6G systems, joint communication-caching-computation, hybrid beamforming, UWB/TOA-based localization, and novel modulation/multiple access methods. Overall, he has published five books [among them the textbook *Wireless Communications* (third edition in 2023)], 22 book chapters, more than 300 journal articles, and more than 400 conference papers. He is also the inventor of 80 patents, and the co-author of some 70 standards contributions. His work has been cited more than 73,000 times, his H-index is 113, and he is a Clarivate Highly Cited Researcher.

Dr. Molisch is a fellow of the National Academy of Inventors, AAAS, and IET; an IEEE Distinguished Lecturer, and a member of Austrian Academy of Sciences. He has received numerous awards, among them the IET Achievement Medal, the Technical Achievement Awards of IEEE Vehicular Technology Society (Evans Avant-Garde Award) and the IEEE Communications Society (Edwin Howard Armstrong Award), and the Technical Field Award of the IEEE for Communications (Eric Sumner Award). He has been an editor of a number of journals and special issues, the general chair, the technical program committee chair, or the symposium chair of multiple international conferences; and the Chairperson of various international standardization groups.