The following publication C. Ding and I. W. -H. Ho, "Digital-Twin-Enabled Federated Learning and CNN-Based Channel Estimation for Urban Vehicular Channels," in IEEE Internet of Things Journal, vol. 12, no. 17, pp. 36034-36044, 1 Sept.1, 2025 is available at https://doi.org/10.1109/JIOT.2025.3580360.

Digital-Twin-Enabled Federated Learning and CNN-based Channel Estimation for Urban Vehicular Channels

Cao Ding, Ivan Wang-Hei Ho, *Senior Member, IEEE*

***Abstract*—Urban vehicular channel estimation (UVCE) has long been a difficult task due to the inter-carrier interference (ICI) and path loss caused by high-speed vehicle motion and urban geographical features (e.g., buildings, vehicles, and trees). Conventional estimators based on pilot symbols and channel statistics generally assume static signal propagation models, such as Free-space and Rayleigh fading. These models are inadequate for addressing path losses caused by geographical features, leading to limited performance. In contrast, centralized learning (CL)-based estimators can provide higher estimation performance by collecting channel data from a specific geographical area for training. However, when the UVCE is scaled to city size, CL estimators cannot precisely recognize the channel characteristics of each location in the city, resulting in decreased estimation accuracy. To further improve the scalability and accuracy of UVCE, this paper proposes a federated learning (FL) and convolutional neural network (CNN)-based channel estimator, referred to as FL-CNN. FL is used to aggregate multiple local channel models (LCM), which are clustered by the K-Dimension (KD)-tree technique. For each LCM, we employ the ray-tracing model to calculate the path loss caused by geographical features and use CNN to estimate the channel. Our results show that at the low signal-to-noise-ratio (SNR) regime (e.g., 10 dB), the estimation and data recovery performance of the FL-CNN estimator are respectively 61% and 65% higher than those of benchmark estimators on average.**

***Index Terms*—Digital Twin, Federated Learning, Channel Estimation, Ray-Tracing, Vehicular Channels.**

I. Introduction

The evolution toward 6G networks promises to revolutionize urban mobility through ultra-reliable, low-latency connectivity, and the Internet of everything (IoE). However, the unique characteristics of urban vehicular environments, coupled with the ambitious goals of 6G, introduce multifaceted challenges for urban vehicular channel estimation (UVCE), a pain point for reliable communication. Due to the inter-carrier interference (ICI) [1] and path loss caused by high-speed vehicle motion and urban geographical features (e.g., buildings, vehicles, and trees), the performance of conventional estimators based on pilot symbols and channel statistics is very limited. Therefore, researchers have turned to machine learning (ML)-based estimators, such as estimators based on deep neural networks (DNN) [2], convolutional neural networks (CNN) [3], and federated learning (FL) [4]. DNN- and CNN-based estimators adopt a centralized learning (CL) approach, where channel data or matrices are collected to a central server (CS) for neural network training. CL estimators has shown high channel estimation performance in a specific urban geographic area. However, when UVCE is scaled to a city-wide level, CL estimators cannot precisely recognize the channel characteristics of each location in the city, resulting in degraded estimation accuracy. To solve the scalability problem of UVCE, the FL estimator adopts a decentralized learning (DL) approach, where multiple clients (e.g. edge devices) collaboratively train a global model under the orchestration of a CS. Meanwhile, the training data from local models remains decentralized and private. Such a FL paradigm introduces two major challenges: i) data across clients are highly non-independent and identically distributed (non-IID); and ii) the presence of communication bottlenecks and delays [5].

The non-IID challenge refers to the issue that the local data of an individual client cannot reflect the overall data distribution of the environment, potentially hindering the convergence of the global model. Moreover, FL communication can be divided into two types: uplink and downlink. Uplink refers to the client uploading the locally trained model, while downlink refers to the client downloading the global model from the CS. Due to the limited bandwidth of clients and the far distance between them and the CS, there are communication delays in both the uplink and downlink. To combat the non-IID and communication bottleneck challenges, some client clustering and weighted averaging aggregation methods have been proposed. For instance, [6] utilized a K-dimensional (KD)-tree client clustering method, which regarded sub-domains divided by root nodes as clusters and selected clients from different clusters to participate in the training. In addition, Wang *et al*. proposed a trimmed mean aggregation (TMA) method [7], which aggregated the global model according to the contribution and weight of each local model. Results show that its privacy preservation and convergence performance are better than those of baselines. Although the above methods can balance data distribution and reduce the communication overheads, the CS still undergoes delayed model updates due to the far transmission distance.

Moreover, most of existing channel estimators generally assume static signal propagation models, such as Free-space and Rayleigh fading [8], which determine the path loss based on the propagation distance and homogeneous environment.

C. Ding and I. W.-H. Ho (corresponding author) are with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong (e-mail: caosnow.ding@connect.polyu.hk; ivanwh.ho@polyu.edu.hk). This work was supported in part by the Research Institute for Artificial Internet of Things (RIAIoT) (Project 1-CDJ5) and the Otto Poon Charitable Foundation Smart Cities Research Institute (Project Q-CDAS) at The Hong Kong Polytechnic University.

However, in realistic vehicular channels, signal propagation experiences path losses caused by geographical features (e.g., buildings, vehicles, and trees), which is known as the urban canyon effect [9]. To accurately model the vehicular channels and path losses caused by obstacles, the ray-tracing (RT) technique [10] is a promising solution. In the RT model, each line-of-sight (LoS) and non-line-of-sight (NLoS) path is considered as an independent ray and its path loss after reflection and diffraction from obstacles is calculated.

Furthermore, the digital twin (DT) paradigm can provide a virtual testbed for ML-based RT channel estimators, which has been applied to the city-model-aware DNN (CMA-DNN) estimator in our prior work [11]. The prior work focused on geographical features-based channel estimation in a centralized DT structure, which is constrained to the channel characteristics of specific locations in the city model. To address the scalability and generalization limitations, this work proposes a FL-driven decentralized DT structure. To further improve the UVCE performance in terms of channel estimation and data recovery, this work also proposes a FL and CNN-based (FL-CNN) estimator. In summary, the main contributions of this work are as follows.

i) We propose a FL-CNN estimator that uses a KD-tree for client clustering in an urban geographic model. For each client, the RT model generates actual channel response matrices (CRM) as the ground truth, and the CNN uses CRM as labels to train the local channel model. Finally, the parameters of the local models are uploaded to the CS for TMA. The KD-tree clustering together with TMA combat the non-IID and communication bottlenecks challenges in FL. They combine with RT to fill the gap in geographical features-based UVCE.
ii) We propose a FL-driven decentralized DT that combines the above techniques (e.g., RT, KD-tree, CNN, and TMA) and implement them in a real-time joint traffic and communication simulation. Compared with the centralized DT in the prior work, FL enhances the DT by enabling geographically distributed model training across clients, avoiding the centralization of raw channel data. FL is uniquely effective in urban DT due to spatially heterogeneous channel conditions. Local models trained by FL capture region-specific propagation effects (e.g., urban highways vs. intersections), while global aggregation ensures generalization. This feature greatly enhances the scalability of UVCE.

The rest of this paper is as follows. Section II presents related work and summarizes the shortcomings of existing channel estimators. Section III introduces the system model for UVCE. Section IV proposes the FL-CNN estimator. Section V compares the performance of the proposed FL-CNN with other benchmark estimators. Finally, Section VI draws conclusions and discusses future research directions.

II. Related Work

Current vehicular channel estimators can be roughly divided into two categories according to the technologies they apply. The first category is conventional estimators based on pilot symbols and channel statistics, such as least square (LS) [12] and linear minimum mean squared error (LMMSE) [13]. LS uses the pre-defined pilot symbols to calculate the first partial derivatives of the estimated channel matrix, which has low complexity but ignores noise. LMMSE adds a weighting matrix based on LS to further optimize the impact of noise and uses linear expected value to reduce complexity. Spectral temporal averaging (STA) is another conventional estimator [14] that uses the time and frequency correlation ratio (TFCR) to continuously estimate the channel to improve the accuracy at low SNR. However, STA requires prior knowledge of the channel to determine the TFCR, which is not feasible in practice.

The first category of conventional estimators has obvious shortcomings. Most of them assume static channels for estimation and cannot combat carrier frequency offsets (CFOs) and Doppler shifts caused by high-speed movement of vehicle nodes, so the estimation performance is very limited. In addition, they highly rely on pilot symbols and prior knowledge of channel statistics, which results in high computational complexity and impracticality.

To further improve the performance of the vehicular channel estimation, researchers have turned to the second category of ML-based estimators. These estimators generally apply DNN or CNN for the estimation process. Specifically, Abdul *et al*. proposed a hybrid estimator STA-DNN [15], which first uses STA to roughly estimate the channel and then uses DNN to learn STA channel statistics. The results show that STA-DNN can effectively track the changes of channels over time, thereby improving the estimation accuracy significantly and reducing the complexity by at least 55%. Furthermore, CNN is capable to recognize and learn channel characteristics in channel response matrices (CRM), which enables CNN to accurately predict channels. For example, Dong et al. proposed a spatial-frequency CNN (SF-CNN) estimator [3], which divides the three-dimensional CRM into two-dimensional frames according to adjacent subcarriers and inputs the CRM into CNN training frame by frame. The results show that SF-CNN can provide higher estimation accuracy than DNN estimators, but at the cost of extremely high complexity. To further improve the estimation performance in terms of the accuracy and complexity, FL-based estimators have been explored by researchers. One of the state-of-art is the FL estimator in massive multiple-input multiple-output (MIMO) systems [4], which uses local datasets to input CNN to train local models and sends the parameters of each local model to the base station (BS) for the global model aggregation, thereby reducing the huge communication overhead of global data collection.

The second category of estimators can precisely capture the changes of dynamic channels over time through ML, thus can perform more accurate estimations. However, they still have shortcomings, such as high computational complexity, difficulty in data collection, and they all ignore the impact of the geographical features on vehicular channels. To the best of our knowledge, there are very few literature that explore the impact of geographical features on UVCE. In our prior work [11], a CMA-DNN estimator is proposed, which first obtains the channel data after experiencing the penetration and

reflection losses of the buildings, and then uses the channel dataset to train the DNN estimator. However, CMA-DNN was trained on channel data collected centrally from a small set of specific street corners and cannot accurately predict the channels at various locations in the city. The FL-CNN approach proposed in this paper overcomes this limitation by leveraging FL to train a global model based on decentralized channel data from distributed clients across diverse urban locations. In this way, we can aggregate channel characteristics from varying geographical scenarios (e.g., intersections, straight roads, highways, etc.) and environmental dynamics (e.g., traffic density), enabling the model to learn robust and environmentally adaptive features.

In general, there is a research gap between the two categories of estimators in terms of the trade-off between estimation accuracy and complexity. To compare the performance of the proposed FL-CNN, the LS, LMMSE, STA, STA-DNN, CMA-DNN and SF-CNN estimators in the above literature are reproduced as benchmarks under the same simulation settings.

III. System Models

In this section, a digital twin paradigm for vehicular channel estimation is proposed, as shown in Fig. 1. In the physical layer, real-time traffic and vehicular communication are constructed. It is assumed that vehicle nodes are equipped with a global navigation satellite system (GNSS) and send location data to the BS. In the cyber layer, the FL server uses GNSS locations matrix and channel matrices to train the CNN local model, and then aggregates the global model for channel estimation.

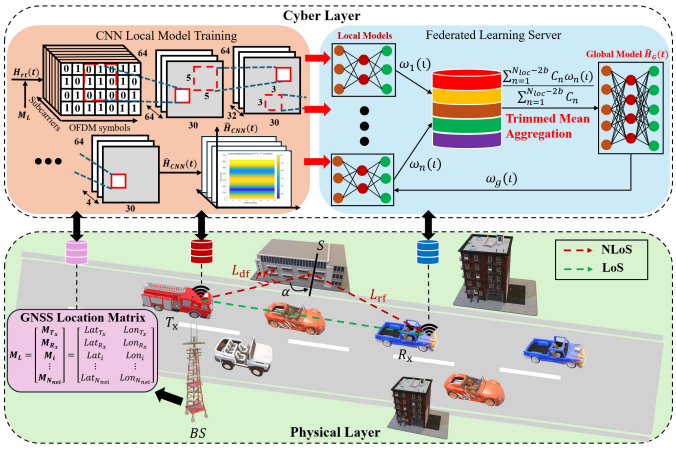


Fig. 1. The FL-driven decentralized DT structure for UVCE.

A. Vehicular Channels

Fig. 1 depicts the multi-path propagation of 5G vehicular channels, that is, the signal sent by the transmitter (T_x) will be transmitted to the receiver (R_x) through multiple paths (i.e., LoS and NLoS), and each path will experience varying degrees of delay spread and path loss. Assuming that the 5G new radio (NR) channels model [16] is adopted in this system,

TABLE I
Permittivity and Conductivity of Common Materials

Material	Frequency (GHz)	Real Part of Relative Permittivity		Conductivity (S/m)	
		σ_1	σ_2	ξ_1	ξ_2
Vacuum (air)	[0.001, 100]	1	0	0	0
Concrete	[1, 100]	5.31	0	0.0326	0.8095
Glass	[0.1, 100]	6.27	0	0.0043	1.1925
Wood	[0.001, 100]	1.99	0	0.0047	1.0718
Dry ground	[1, 10]	3	0	0.00015	2.52

the complex channel gain of p-th propagation path can be mathematically expressed as

$$G_p(t) = \sqrt{\frac{E_p(t)}{N_r}} \sum_{n_r=1}^{N_r} e^{j\phi_{m,n_r}} e^{-jv\cos(\alpha(t)-\theta(t))t}, \quad (1)$$

where $E_p(t)$ denotes the time-varying electromagnetic power, N_r denotes the number of ray components, and $\phi_{m,n_r} \in [0, 2\pi]$ represents the random phase of the n_r-th radio ray. $e^{j\phi_{m,n_r}}$ represents the complex amplitude (i.e., magnitude and phase) of the received signal pulse. v is the absolute value of the lateral relative speed between T_x and R_x, $\alpha(t)$ is the angle of arrival (AoA), and $\theta(t)$ is the angle between the driving direction and the main path. Furthermore, the channel impulse response based on multi-path propagation can be expressed as

$$h(t) = \sum_{p=1}^{P} G_p(t)\delta(t - \tau_p(t)), \quad (2)$$

where P denotes the total number of paths, $\tau_p(t)$ denotes the delay spread of the path, and $\delta(\cdot)$ represents the Dirac delta function. Based on the multi-path channel model in (2), this paper further considers the propagation loss caused by the urban canyon effect.

B. Ray-Tracing Propagation Model

To measure the path loss caused by obstacles, the RT technology is adopted to model the dynamic propagation. The RT model considers each signal transmission path as an independent ray and follows the principles of reflection and diffraction of light propagation. In the case shown in Fig. 1, the LoS path between T_x and R_x is blocked by an obstacle vehicle, while the NLoS path can be achieved after experiencing the reflection loss (L_{rf}) and diffraction loss (L_{df}) caused by the building surface. The propagation matrix of the i-th reflection/diffraction point of an NLoS ray can be expressed as

$$\boldsymbol{R_i} = \begin{bmatrix} s & p & k \end{bmatrix}_i \begin{bmatrix} F_V(\alpha) & 0 & 0 \\ 0 & F_H(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}_i \begin{bmatrix} s' & p' & k' \end{bmatrix}_i^{-1}, \quad (3)$$

where s, p denote the horizontal and vertical polarization directions, k denotes the incident ray direction. s', p', and k' represent the same properties of the exiting ray. α denotes the incident ray angle, $F_V(\alpha)$ and $F_H(\alpha)$ are Fresnel reflection coefficients for horizontal and vertical polarization, which can be expressed as

$$F_V(\alpha) = \frac{\cos(\alpha) - \sqrt{\epsilon - \sin^2(\alpha)}}{\cos(\alpha) + \sqrt{\epsilon - \sin^2(\alpha)}}, \tag{4}$$

$$F_H(\alpha) = \frac{\cos(\alpha) - \sqrt{(\epsilon - \sin^2(\alpha))/\epsilon^2}}{\cos(\alpha) + \sqrt{(\epsilon - \sin^2(\alpha))/\epsilon^2}}, \tag{5}$$

where ϵ is the complex relative permittivity of the obstacle material based on the ITU-R P.2040-1 standard [17]. Table I gives the permittivity (σ_1, σ_2) and conductivity (ξ_1, ξ_2) of common materials. ϵ can be calculated as

$$\epsilon = \sigma_1 f^{\sigma_2} + j(\frac{\xi_1 f^{\xi_2}}{2\pi\epsilon_0 f}), \tag{6}$$

where $\sigma_1 f^{\sigma_2}$ is the real part and $\frac{\xi_1 f^{\xi_2}}{2\pi\epsilon_0 f}$ is the imaginary part. f is the frequency of the ray, ϵ_0 is the permittivity of free space. The propagation matrix of an NLoS ray is $\boldsymbol{R} = \prod \boldsymbol{R_i}$, and the polarization matrix between T_x and R_x can be expressed as

$$\boldsymbol{P_m} = \begin{bmatrix} \boldsymbol{R}(\boldsymbol{V_{T_x}} \times \boldsymbol{k_{T_x}}) \cdot \boldsymbol{H_{R_x}} & \boldsymbol{RV_{T_x}} \cdot \boldsymbol{H_{R_x}} \\ \boldsymbol{R}(\boldsymbol{V_{T_x}} \times \boldsymbol{k_{T_x}}) \cdot \boldsymbol{V_{R_x}} & \boldsymbol{RV_{T_x}} \cdot \boldsymbol{V_{R_x}} \end{bmatrix}, \tag{7}$$

where $\boldsymbol{V_{T_x}}$ denotes the direction of the nominal vertical polarization for the ray leaving T_x, $\boldsymbol{k_{T_x}}$ denotes the direction of the ray leaving Tx. $\boldsymbol{H_{R_x}}$ and $\boldsymbol{V_{R_x}}$ represent the directions of the horizontal and vertical polarizations for R_x. The reflection loss can be calculated as

$$L_{rf} = -20log_{10}|\boldsymbol{J_{R_x}}^{-1} \boldsymbol{P_m} \boldsymbol{J_{T_x}}|, \tag{8}$$

where $\boldsymbol{J_{T_x}}$, $\boldsymbol{J_{R_x}}$ represent two normalized Jones polarization vectors for T_x and R_x. When T_x and R_x are unpolarized, both $\boldsymbol{J_{T_x}}$ and $\boldsymbol{J_{R_x}}$ vectors are $[\frac{\sqrt{2}}{2} \ \frac{\sqrt{2}}{2}]^{-1}$. For the first order signal diffraction, the diffraction loss can be calculated as

$$L_{df} = \boldsymbol{J_{R_x}}^{-1} \boldsymbol{P'_m} \boldsymbol{J_{T_x}}, \tag{9}$$

where $\boldsymbol{P'_m}$ is the polarization matrix containing the diffraction coefficients in the uniform theory of diffraction (UTD) [18]. In general, the RT model analyzes the reflection and diffraction losses in signal propagation, which further improves the accuracy of 5G vehicular channel modeling.

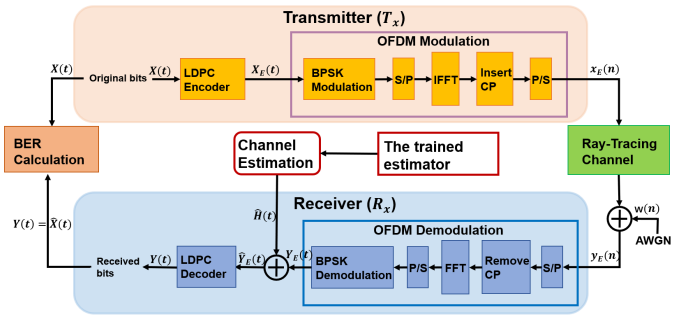


Fig. 2. OFDM link in ray-tracing vehicular channels.

C. OFDM Link

In addition to the RT propagation, T_x and R_x use the orthogonal frequency-division multiplexing (OFDM) link for signal transmission, as shown in Fig. 2. At the transmitter stage, T_x first generates the original data bits $\boldsymbol{X(t)}$ and performs low-density parity-check code (LDPC) encoding [19]. Note that LDPC is a linear error correction code that can recover the correct data even with some erroneous bytes, thereby greatly reducing the bit-error-rate (BER). The encoded data $\boldsymbol{X_E(t)}$ and the predefined pilot signal are converted from the time domain into frequency domain symbols $\boldsymbol{x_E(n)}$ through OFDM modulation. Afterwards, $\boldsymbol{x_E(n)}$ symbols go through path losses and additive white Gaussian noises (AWGN) before receiving by R_x. The received symbols can be expressed as

$$\boldsymbol{y_E(n)} = \boldsymbol{h_{rt}(n)} \times \boldsymbol{x_E(n)} + \boldsymbol{\omega(n)}, \tag{10}$$

where $\boldsymbol{h_{rt}(n)}$ is the ray-tracing channel in the frequency domain, $\boldsymbol{\omega(n)}$ is the AWGN under a certain signal-to-noise ratio (SNR) level. At the receiver stage, R_x first performs OFDM demodulation on $\boldsymbol{y_E(n)}$ to obtain time domain data $\boldsymbol{Y_E(t)}$. Then, R_x estimates the channel matrices $\boldsymbol{\hat{H}(t)}$ using the trained estimator and recovers the received data as

$$\boldsymbol{\hat{Y}_E(t)} = \boldsymbol{\hat{H}(t)} \boldsymbol{Y_E(t)} + \boldsymbol{\Omega(t)}, \tag{11}$$

where $\boldsymbol{\Omega(t)}$ is the AWGN in time domain. Finally, R_x performs LDPC decoding on $\boldsymbol{\hat{Y}_E(t)}$ to obtain the recovered data $\boldsymbol{Y(t)}$. To evaluate the data recovery performance, the recovered data $\boldsymbol{Y(t)}$ is compared with the original data (i.e., $\boldsymbol{X(t)}$) for bit-error-rate (BER) calculation.

IV. Federated Learning and CNN-based Channel Estimation

In this section, we present the proposed FL-CNN estimator step by step.

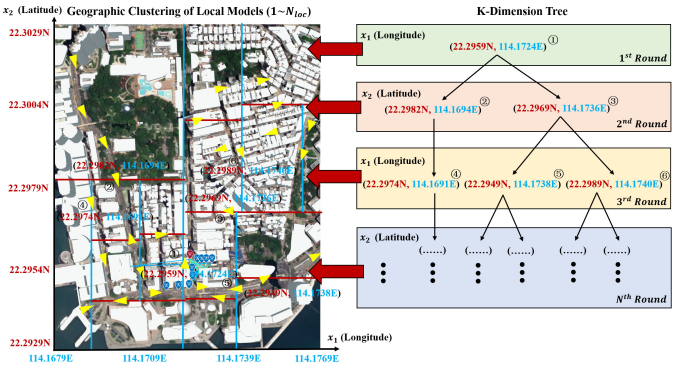


Fig. 3. KD-tree geographical client clustering for the TST model.

A. KD-Tree Geographical Clustering of Clients

Vehicular channels are highly dependent on geographical features, such as the layout of buildings, the location of obstacle vehicles, and even trees and foliage. This means that knowledge of geographical features can help improve the accuracy of local channel estimation. Therefore, we apply the

KD-tree to cluster the urban geographic model into multiple clients, as shown in Fig. 3.

Taking the geographical model of Tsim Sha Tsui (TST) in Hong Kong, China as a case study, we first obtained the 3D spatial data from Lands Department of the HKSAR, which includes the coordinates of the buildings, building heights, number of floors, coordinates of trees and overpasses, etc. Secondly, we randomly generated traffic flows and the sampled receivers as R_x. Then, a KD-tree (K=2) is constructed based on the longitude (i.e., x_1) and latitude (i.e., x_2) coordinates of the receivers. It selects the median of x_1 or x_2 coordinate as the root node each round and divides the area into two subsets. Note that when the building density in the area is lower than the minimum building density threshold, the clustering stops. For instance, the park area in the upper left corner of Fig. 3 is not further divided into sub-areas. After N^{th} rounds, the TST model is divided into $N_{loc} = 16$ clients according to the longitude range (i.e., [22.29290, 22.3092]°N) and the latitude range (i.e., [114.1679, 114.1769]°E). When a pair of T_x and R_x are located in two clients (e.g., N_1 and N_2), their channel data will be saved in both training sets. Therefore, the training set for each client only contains geographic features within that region. Finally, the contribution of the n-th client to the whole training datasets can be expressed as follows.

$$\zeta_n = \frac{TS(n)}{\sum_{i=1}^{N_{loc}} TS(i)} \times 100\%, \tag{12}$$

where $TS(n)$ denotes the training set size of the n-th client. The KD-tree clusters clients based on data similarity metrics (e.g., geographic features), which combats the Non-IID challenge because clients in the same cluster share statistically similar channel data, making their local distribution closer to IID. Through the KD-tree client clustering, channel data from clients are collected into training and testing sets.

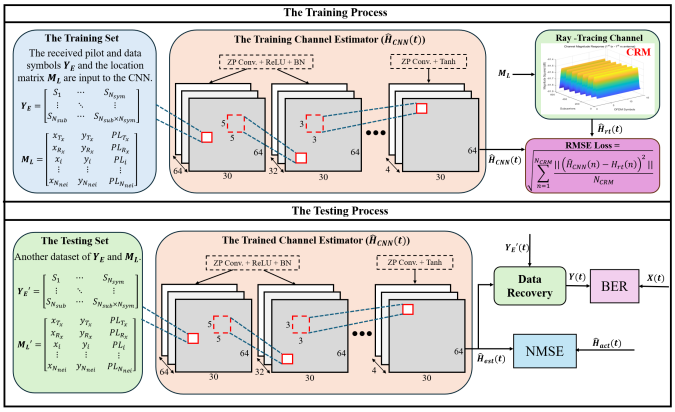


Fig. 4. The training and testing processes of the CNN estimator.

B. CNN Channel Estimator for Local Models

Convolutional neural network (CNN) can precisely model and predict vehicular channels by using kernels or filters to learn features in the channel matrices. Therefore, we propose a CNN structure to train local channel models (LCM) for clients, as shown in Fig. 4. Before training, the channel data of each client will be collected into training and testing sets, which include the received pilot and data symbols $\boldsymbol{Y_E} \in \mathbb{C}^{N_{sub} \times N_{sym}}$ corresponding to N_{sub} subcarriers and N_{sym} OFDM symbols, the location matrix $\boldsymbol{M_L}$ of T_x, R_x, and neighbor vehicles. In particular, the coordinates of T_x, R_x, and neighbor vehicles (e.g., x_{R_x}, y_{R_x}) are input into the RT model to generate channel response matrices (CRM) $\boldsymbol{H_{rt}}$ as groundtruth labels. The path loss at each location (e.g., PL_{R_x}) is calculated according to the geographical features and added to the location matrix $\boldsymbol{M_L}$ as follows.

$$\boldsymbol{M_L} = \begin{bmatrix} x_{T_x} & y_{T_x} & PL_{T_x} \\ x_{R_x} & y_{R_x} & PL_{R_x} \\ x_i & y_i & PL_i \\ \vdots & \vdots & \vdots \\ x_{N_{nei}} & y_{N_{nei}} & PL_{N_{nei}} \end{bmatrix}, \tag{13}$$

During the training, a set of $\boldsymbol{Y_E}$ and $\boldsymbol{M_L}$ data are input to the CNN, and the CNN predicts the channel state information (CSI) and outputs the estimated CRM $\hat{\boldsymbol{H}}_{\boldsymbol{CNN}}(\boldsymbol{t})$. The structure of CNN includes 7 zero padding (ZP) convolutional layers. For the first ZP convolutional layer, it uses 64 kernels of size 5×5 in the filter, then passes through the rectified linear unit (ReLU) activation function to generate an output of 64×30×64, and is finally processed by the batch normalization (BN) layer. For the next 5 ZP convolutional layers, each layer uses 32 kernels of size 3×3 in the filter and outputs 64×30×32 real-valued matrices with the ReLU function, and then processed by a BN layer. The last ZP convolution layer uses 4 kernels of size 3×3 in the filter to estimate the real and imaginary parts of the channel matrices ($\hat{\boldsymbol{H}}_{\boldsymbol{CNN}}(\boldsymbol{t})$), and employs the hyperbolic tangent activation function to map the output into [-1,1]. Specifically, the output of the l-th convolutional layer can be mathematically expressed as follows.

$$\mu(l) = \sum_{c=1}^{C} \sum_{i=1}^{I_k} \sum_{j=1}^{J_k} \omega_{c,i,j}(l) \cdot \mu(l-1) + b_c(l), \tag{14}$$

where $c = 1, ..., C$ represents the input channel matrice indices, $i = 1, ..., I_k$ and $j = 1, ..., J_k$ represent the height and width of the kernel, respectively. $\omega_{c,i,j}(l)$ denotes the weight at (i, j) kernel of the c-th channel on the l-th layer, $\mu(l-1)$ denotes the output of the $(l-1)$-th convolutional layer, and $b_c(l)$ denotes the deviation of the c-th channel on the l-th layer. Afterwards, the estimated CRM ($\hat{\boldsymbol{H}}_{\boldsymbol{CNN}}(\boldsymbol{t})$) and the groundtruth CRM ($\boldsymbol{H_{rt}}$) are used to calculate the root-mean-squared-error (RMSE) loss function, which can be expressed as follows.

$$RMSE = \sqrt{\sum_{n=1}^{N_{CRM}} \frac{||(\hat{\boldsymbol{H}}_{\boldsymbol{CNN}}(\boldsymbol{n}) - \boldsymbol{H_{rt}}(\boldsymbol{n}))^2||}{N_{CRM}}}, \tag{15}$$

where N_{CRM} is the total number of CRM in the training set. After training, the trained channel estimator ($\hat{\boldsymbol{H}}_{\boldsymbol{CNN}}$) is saved as the N_{loc}-th LCM ($\hat{\boldsymbol{H}}_{\boldsymbol{N_{loc}}}$).

For testing, another set of $\boldsymbol{Y'_E}$ and $\boldsymbol{M'_L}$ data that are completely different from the training set is input to the trained estimator ($\hat{\boldsymbol{H}}_{\boldsymbol{CNN}}$). Then, the trained estimator $\hat{\boldsymbol{H}}_{\boldsymbol{CNN}}$ predicts the CSI and outputs the estimated CRM ($\hat{\boldsymbol{H}}_{\boldsymbol{est}}$). Finally, R_x

uses the estimated CRM ($\hat{\boldsymbol{H}}_{\boldsymbol{est}}$) to recover the received data symbols. Note that the CNN estimator not only can provide training for LCM, but also can be used as an independent channel estimation method. The stand-alone CNN estimator serves as a benchmark in Section V.

C. Trimmed Mean Aggregation (TMA) for The Global Model

In this work, the TMA is adopted to aggregate the global model (i.e., $\hat{\boldsymbol{H}}_{\boldsymbol{G}}(\boldsymbol{t})$) due to the geographical characteristics of channels in different LCM. As illustrated in Fig. 1, the FL central server (CS) first distributes the initialized global model parameters (i.e., $\omega_g(\iota)$) to each LCM (i.e., $\hat{\boldsymbol{H}}_{\boldsymbol{CNN}}(\boldsymbol{t})$). Each LCM is trained based on the same features and using local channel data, and the trained parameters (i.e., $\omega_n(\iota)$) are uploaded to the CS. In each FL iteration, the LCM experiences K steps of gradient descent, which can be expressed as follows

$$\omega_n(\iota) = \omega_g(\iota) - \eta_k \nabla \ell(\omega_n(\iota - 1)), \tag{16}$$

where ι represents the number of FL iteration, $n = 1, ..., N_{loc}$ denotes the number of LCM, and η_k denotes the learning rate of the k-th step of gradient descent. $\nabla \ell$ is the loss function of the local gradient descent, which is the MSE of the CNN in this case. The CS aggregates and updates the global model (i.e., $\omega_g(\iota)$) with the TMA of parameters from different local models, which can be expressed as

$$\omega_g(\iota) = \frac{\sum_{n=1}^{N_{loc}-2b} \zeta_n \omega_n(\iota)}{\sum_{n=1}^{N_{loc}-2b} \zeta_n}, \tag{17}$$

where $b < \frac{N_{loc}}{2}$ is a constant parameter, ζ_n is the contribution of each client calculated by the KD-tree clustering. Specifically, the CS eliminates the largest b parameters and the smallest b parameters (i.e., outliers), and then average the remaining parameters to obtain the global model parameters. By removing outliers, TMA minimizes the need for multiple rounds of communication to address noisy updates, effectively tackling the challenge of communication bottlenecks. After each iteration, the FL server again distributes the updated global model parameters to LCM for the next round of training. For the actual UVCE, R_x downloads the global model for estimation instead of LCM.

V. Numerical Results and Performance Evaluation

In this section, we conducted multiple simulations to evaluate the performance of the proposed FL-CNN channel estimation. The simulation framework of the digital twin is illustrated in Fig. 5. In the physical layer, the simulation of urban mobility (SUMO) is used to generate traffic flows of stochastic trajectories on TST roads and obtain the geographic locations of multiple pairs of T_x and R_x. Simultaneously, in the cyber layer, MATLAB retrieves location data through a TCP link over the traffic control interface (TraCI) to perform RT propagation and generate channel matrices for training and test sets. Finally, the collected training sets are used to train the FL-CNN estimator using Keras in Python.

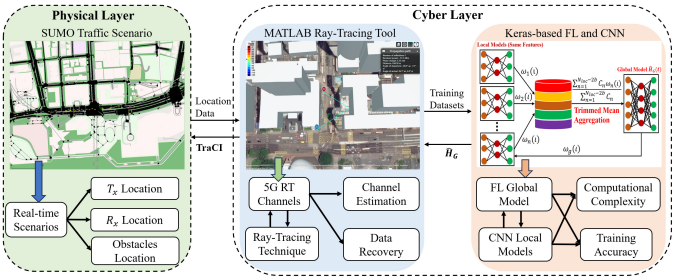


Fig. 5. The simulation framework of the digital twin.

TABLE II
Parameters In Simulation

Parameter	Value	Unit
Simulation Duration	1800.00	s
Geographical Model	Tsim Sha Tsui (TST)	N/A
Mobility model	Stochastic Trajectory	N/A
Traffic Flow	1000 - 3000	veh/h
Speed Limit	50	km/h
Vehicular Channel	5G New Radio (NR)	N/A
Channel Coding, Signal Interference	LDPC, AWGN	N/A
Propagation Model	Ray Tracing	N/A
Signal-to-Noise Ratio (SNR)	0 - 35	dB
Physical layer, Modulation	IEEE 802.11p, BPSK	N/A
Number of Pilot, Data, and Total Subcarrier	4, 48, 64	N/A
Subcarrier Spacing	312.5	KHz
Length of Cyclic Prefix	16	bits
Symbol, Data, and Guard Interval	4, 3.2, 0.8	μs
Packet Rate	10	packets/s
Number of Base Station (BS)	5	meter
Communication Radius for Vehicle	200	meter
Communication Radius for BS	2000	meter
Number of Convolutional Layers	7	layer
Padding Method	Zero Padding (ZP)	N/A
Total Training Sets Size	16,000	matrices
Epoch, Learning Rate	10,000, 0.01	N/A
Number of FL server	1	N/A
Number of Clients (Local Models) N_{loc}	16	N/A
FL Communication Delays	10-50	ms
Bandwidth Constraints for Uplink and Downlink	10-50, 50-100	Mbps
Positive-to-Negative Samples Ratio	4:1	N/A
Number of FL Iteration	100	round
Constant b in TMA	3	N/A

A. Simulation Parameters and Datasets

The simulation parameters are given in Table II. Firstly, we consider half an hour of vehicular communication data from the TST geographic model, which is sufficient to train the FL-CNN estimator. Secondly, the traffic volume is set at 3,000 vehicles per hour, which is a moderate vehicular density, and the speed limit on urban roads is 50 kilometers per hour. For the vehicular communication, we adopt 5G NR channel, LDPC channel coding, and the RT propagation model. In addition to the propagation loss, we also consider AWGN signal interference and set SNR regimes from 0 to 35 dB. Each OFDM block consists of 64 subcarriers in total, including 4 pilot subcarriers and 48 data subcarriers, with a subcarrier spacing of 312.5 KHz. We deploy the Keras, TensorFlow platform for FL-CNN programming. For each client, we collect at least 1,000 channel matrices of OFDM communications and input them into a CNN network

consisting of 7 convolutional, ZP and BN layers. The training epoch is set to 10,000, the learning rate is 0.01, and total training sets size is over 16,000 matrices. For FL settings, we deploy a centralized FL server and 16 clients (local models) geographically distributed across the TST area. The number of FL iterations is set to 100 and the constant b in TMA is set to 3. Moreover, we model the communication delays and bandwidth constraints between the server and clients based on empirical measurements in 5G vehicular networks. Delays range from 10 to 50 ms, uplink bandwidth ranges from 10 to 50 Mbps, and downlink bandwidth ranges from 50 to 100 Mbps, depending on the distance between each client and the server. The training datasets include positive and negative samples, where positive samples are the channel data ($\boldsymbol{Y_E}$, $\boldsymbol{M_L}$) selected from successful LoS and NLoS signal transmissions (i.e., received power is higher than -80 dBm), and negative samples are selected from failed signal transmissions with NLoS (i.e., received power is lower than -80 dBm). We use synthetic minority oversampling (SMOTE) to enforce a 4:1 positive-to-negative samples ratio to prevent bias. The rest of the parameters are shown in Table II.

To fairly compare the performance, we reproduce two types of state-of-the-art estimators as benchmarks in the same parameters setting. The first type is conventional estimators based on pilot symbols, including LS, LMMSE and STA. The second type is learning-based estimators, including STA-DNN, CMA-DNN, and SF-CNN. We use CMA-DNN to replace CNN to train local models and aggregate the global model as the FL-DNN estimator.

B. Channel Estimation Performance

The estimation performance is defined as the normalized mean square error (NMSE) between the estimated channel matrices (i.e., $\hat{\boldsymbol{H}}_{\boldsymbol{est}}$) and the actual channel matrices (i.e., $\boldsymbol{H}_{\boldsymbol{act}}$), which can be expressed mathematically as

$$NMSE = \frac{\sum_{j=1}^{J}(\hat{\boldsymbol{H}}_{\boldsymbol{est}}(j) - \boldsymbol{H}_{\boldsymbol{act}}(j))^2}{\sum_{j=1}^{J}(\boldsymbol{H}_{\boldsymbol{act}}(j))^2}, \tag{18}$$

where $j = 1, \ldots, J$ denotes the j-th channel matrix of $\hat{\boldsymbol{H}}_{\boldsymbol{est}}$ or $\boldsymbol{H}_{\boldsymbol{act}}$ in the testing dataset. It can be seen from Fig. 6 that the NMSE of the learning-based estimators is generally lower than that of the conventional estimators, especially FL-CNN and FL-DNN. Specifically, at the SNR regime of 35 dB, the NMSE of FL-CNN is 27.2%, 53.8%, 45.9%, 64.9%, 75.5%, 94.3%, 39.61%, and 96.2% lower than that of FL-DNN, SF-CNN, CNN, CMA-DNN, STA-DNN, STA, LMMSE, and LS, respectively. Note that LMMSE assumes that the channel correlation is known and all channel information is obtained, so the estimation accuracy is high. However, at the SNR regime of 10 dB, the NMSE between FL-CNN and other estimators reaches the maximum performance gap. The NMSE of FL-CNN is 42.6%, 45.9%, 40.2%, 83.8%, 94.8%, 97.2%, 96.6%, and 98.5% lower than that of FL-DNN, SF-CNN, CNN, CMA-DNN, STA-DNN, STA, LMMSE, and LS, respectively. This means that the NMSE of benchmark estimators increases significantly at low SNR, while FL-CNN still maintains the best performance. In addition, the NMSE of FL-CNN and FL-DNN is lower than that of CNN and CMA-DNN, which proves that the global model aggregated by TMA more comprehensively captures the characteristics of urban vehicular channels.

C. Data Recovery Performance

Fig. 7 depicts the bit-error-rate (BER) performance of each estimator during data recovery, and it can be seen that FL-CNN has the lowest BER at all SNR regimes. Since LDPC is deployed in the OFDM process, the BER performance of estimators can be degraded to an extremely low level (i.e., 10^7 to 10^6). We can see from Fig. 7 that the BER of each estimator drops significantly when the SNR increases. Specifically, at the SNR regime of 35 dB, the BER between FL-CNN and other estimators reaches the maximum performance gap. The BER of FL-CNN is 32.1%, 65.6%, 76.4%, 90.5%, 94.7%, 95.8%, 98.3%, and 96.1% lower than that of FL-DNN, SF-CNN, CNN, CMA-DNN, STA-DNN, STA, LMMSE, and LS, respectively. This result illustrates that FL-CNN is more adaptable to different levels of SNR than other estimators.

D. Impact of Geographic Features

In addition, this paper also investigates the impact of geographic features on channel estimation and data recovery performance. Since the main factor of geographical features is building layout, we consider building density as a quantitative indicator of geographical features, which is defined as $\rho_B(n) = \frac{S_B(n)}{S(n)} \times 100\%$, where $S_B(n)$ is the building area in the n-th LCM and $S(n)$ is the total area of the n-th LCM. According to the statistics of the 16 LCM within TST, the smallest building density is 35.2% and the largest building density is 94.8%. Therefore, we divide the local models into six intervals from 35% to 95% and summarize their performance at the SNR regime of 25 dB, as shown in Fig. 8 and Fig. 9.

Fig. 8 shows that the NMSE of each estimator increases with the increase of building density. Specifically, when the building density increases from $[35, 45)$ to $[85, 95]$, the NMSE of FL-CNN, FL-DNN, CNN, CMA-DNN, SF-CNN, STA-DNN and LMMSE increases by 16.3%, 20.4%, 23.9%, 29.1%, 42.8%, 58.4% and 48.6%, respectively. This result demonstrates that estimators trained using the RT model (i.e., FL-CNN, FL-DNN, CNN, CMA-DNN) are better adapted to high building densities and thus provide accurate estimates of NLoS channels. Fig. 9 gives the BER performance of channel estimators under different building densities. Compared with other benchmark estimators, the BER of FL-based estimators (i.e., FL-CNN, FL-DNN) is lower and more stable. Specifically, when the building density increases from $[35, 45)$ to $[85, 95]$, the BER of FL-CNN, FL-DNN, CNN, and CMA-DNN increases by 11.7%, 21.6%, 25.1% and 28.4%, respectively. Since the FL better learns the geographic features of LCM, which cannot be achieved by training with the entire TST dataset.

Furthermore, the FL-CNN is designed to address cross-scenario generalization via FL aggregation. While our experiments focused on the TST area, the FL-CNN inherently supports adaptation to new environments. When deploying to

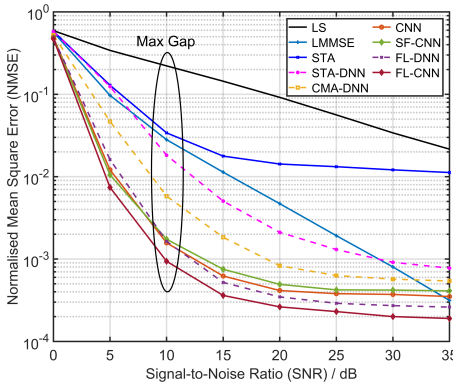


Fig. 6. NMSE performance versus SNR.

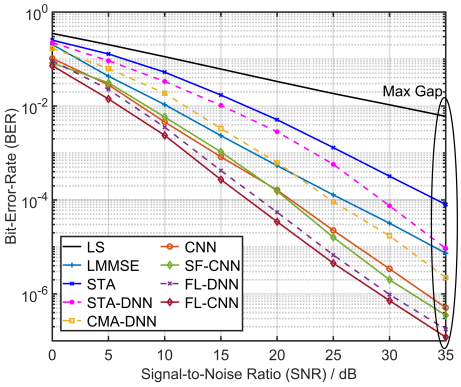


Fig. 7. BER performance versus SNR.

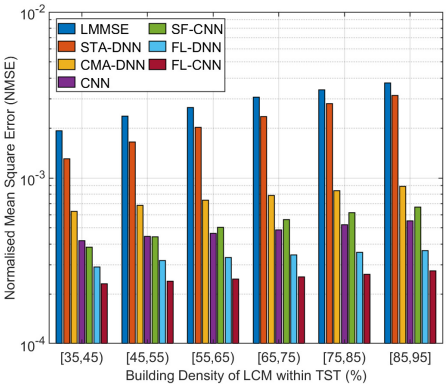


Fig. 8. NMSE performance versus building density.

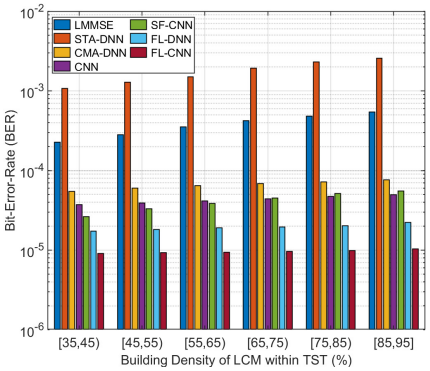


Fig. 9. BER performance versus building density.

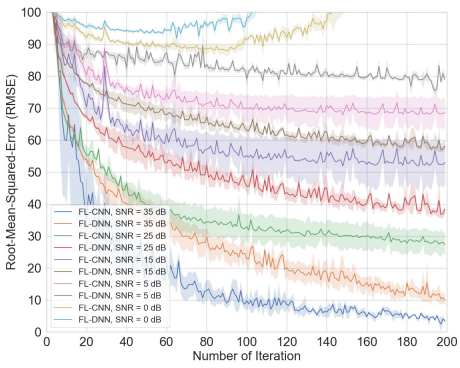


Fig. 10. RMSE versus the number of iterations.

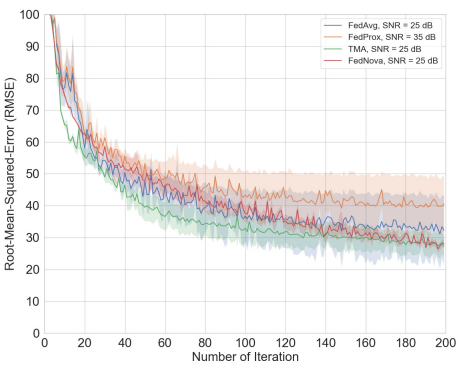


Fig. 11. RMSE of different aggregation methods.

a new city (e.g., grid-based vs. irregular layouts), the FL-CNN global model with TST pre-trained weights can be fine-tuned using local data from the target environment. This avoids full retraining by leveraging prior knowledge (e.g., geographic-channel relationships learned from TST) while adapting to new layouts incrementally. In addition, the KD-tree strategy is not static, but dynamically adapts to new environments by recalculating KD-tree clusters based on the new geographical features. The clustering is data-driven and does not depend on pre-defined TST regions or local models.

E. Impact of Federated Learning

This paper also explores the impact of federated learning on training performance. Fig. 10 shows the RMSE of the FL-CNN and FL-DNN estimators at different number of iterations. We collected the RMSE of three rounds of training for fitting, where the solid line is the average RMSE. It can be seen that the RMSE of FL-CNN and FL-DNN gradually decreases and converges with the increase of iterations, except for the case where SNR is 0 dB. Specifically, at the SNR regime of 35 dB, the RMSE of FL-CNN quickly reaches the minimum value and converges, while the RMSE of FL-DNN is higher and converges more slowly. This result illustrates that FL-CNN captures vehicular channel characteristics more accurately than FL-DNN. As an extreme case that is unlikely to occur in reality, when the SNR regime is 0 dB, the RMSE of FL-CNN and FL-DNN first decreases gradually and then rises significantly to an ultra-high level. This is because the vehicular channel is almost full of noise and interference, so the estimation process is extremely difficult.

To compare the performance of the TMA global model aggregation method used in FL-CNN, we reproduced the following methods to replace TMA in FL-CNN. Federated averaging (FedAvg) [20] is the foundational method, which works by weighted averaging model updates from multiple local models to form a global model. Federated proximal (FedProx) [21] extends FedAvg by introducing a proximal term to handle heterogeneity among local models. Federated normalized averaging (FedNova) [22] performs local updates and computes a normalized update based on the number of local steps taken. Fig. 11 shows that the RMSE of the TMA aggregation method is generally lower than that of FedAvg and FedProx. After the number of iterations reaches 160, the RMSE performance of FedNova is very close to that of TMA. However, the RMSE of TMA converges faster than that of FedNova, which shows that the TMA method outperforms other benchmarks in aggregating models efficiently and accurately.

F. Component-Wise Analysis

To quantify the contribution from individual components, we conduct an ablation study for FL-CNN components (i.e., FL framework, CNN architecture, KD-tree clustering, and RT model), as shown in Table III. It can be seen from the table that at 10 dB SNR, the NMSE performance of the full framework (FL+CNN+KD+RT) is improved by 61%, as the baseline. When we replace the RT model with a statistical channel

TABLE III
COMPONENT-WISE ANALYSIS.

Scenario (at 10 dB SNR)	NMSE Improvement	Key Insight
Full Framework (FL+CNN+KD+RT)	61%	Baseline
Centralized Training (No FL)	35%	FL contributes 26% of the gain by aggregating diverse LCM
DNN (No CNN)	42%	CNN's geographical path loss extraction adds 19% gain
No KD-Tree Clustering	50%	Client clustering improves the performance by 11%
Statistical Channel Model	30%	Ray-tracing adds 31% gain by capturing real-world geometry

model (Rayleigh), the improvement of NMSE is only 30%, which means that the RT model adds 31% gain by capturing real-world geometry, and contributes to the most performance gain. This result reveals that without the RT model, the FL-CNN estimator lacks the geometric context and cannot capture the channel characteristics properly under different geographical features. Moreover, FL contributes 26% of the gain by aggregating diverse LCM compared with centralized learning (CL), while the CNN architecture improves 19% gain by extracting path losses based on geographical features. In contrast, KD-tree improves the NMSE performance the least, only 11%. This is because KD-tree clustering ensures clients with similar channel conditions being trained together, reducing the model's divergence. For example, LOS-dominant clients converge faster than NLOS-heavy ones. Finally, TMA-based FL aggregation specifically enhance generalization compared to CL and FedAvg, as CL suffers from geographical bias and data imbalance when scaling to city-wide models, while FedAvg averages all LCM equally, allowing outliers (e.g., a model trained based on obstructed data) to degrade the global model. On the contrary, TMA discards most extreme LCM, and hence reduce the RMSE by 15% compared to FedAvg as shown in Fig. 11.

TABLE IV
COMPUTATIONAL COMPLEXITY OF CHANNEL ESTIMATORS.

Channel Estimator	Computational Complexity (Times of Multiplication)	Time Complexity
LS	$2N_{sub}$	$O(2N_{sub})$
STA	$22N_p + 2N_d$	$O(22N_p)$
CMA-DNN	$L_1L_2 + L_2L_3 + L_3L_4 + L_1L_4$	$O(L_1L_2)$
STA-DNN	$22N_p + 2N_d + L_1L_2 + L_2L_3 + L_3L_4 + L_1L_4$	$O(L_1L_2)$
FL-DNN	$N_{loc}(L_1L_2 + L_2L_3 + L_3L_4 + L_1L_4)$	$O(N_{loc}L_1L_2)$
CNN, SF-CNN	$\sum_{l=1}^{N_{con}}(D_l^x D_l^y W_l^x W_l^y F_l^i F_l^o)$	$O(D_1^x D_1^y W_1^x W_1^y F_1^i F_1^o)$
FL-CNN	$N_{loc}\sum_{l=1}^{N_{con}}(D_l^x D_l^y W_l^x W_l^y \cdot F_l^i F_l^o)$	$O(N_{loc}D_1^x D_1^y W_1^x W_1^y \cdot F_1^i F_1^o)$
LMMSE	$(N_{sub})^3$	$O((N_{sub})^3)$

G. Computational Complexity Performance

Finally, this paper compares the computational complexity of different channel estimators, as shown in Table IV. We define the computational complexity of an estimator as the number of multiplication operations and the time complexity as the worst-case complexity. The order of the estimators from lowest to highest computational complexity is LS, STA, DNN, STA-DNN, FL-DNN, CNN/SF-CNN, FL-CNN and LMMSE. LS divides the received symbols by the predefined pilot symbols. Since we use BPSK modulation, the total number of divisions is $2N_{sub}$. Note that N_{sub}, N_p, and N_d represent the number of total subcarriers, data subcarriers, and pilot subcarriers, respectively. For STA, it needs $2N_d$ or $4N_p$ multiplications to complete one frequency domain averaging or time domain averaging, respectively. Thus, the total number of multiplications is $22N_p + 2N_d$. CMA-DNN, STA-DNN uses a 5-layer fully connected structure, and all neurons perform multiplications except the output layer. L_1, L_2, L_3, and L_4 represent the number of neurons within the input layer, the first, second, and third hidden layers, respectively. STA-DNN performs an additional number of STA multiplications. FL-DNN performs N_{loc} times of CMA-DNN to aggregate the global model. The computational complexity of CNN and SF-CNN is the number of multiplications of N_{con} convolutional layers. D_l^x, D_l^y denote the 2-dimension feature map size of the l-th layer, W_l^x, W_l^y represent the column and row sizes of each filter, and F_l^i, F_l^o are the number of input and output feature maps. FL-CNN performs CNN convolutions on N_{loc} local models. LMMSE has the highest complexity since it performs low-rank estimation by computing the SVD of the channel autocorrelation matrix. The SVD requires $(N_{sub})^3$ times of multiplication. In general, the complexity of FL-CNN is higher than that of other estimators except LMMSE. The trade-off between estimation accuracy and complexity should be considered in practical applications. At low SNR regimes, FL-CNN can provide high-precision estimation performance. At high SNR regimes, FL-DNN and CMA-DNN estimators can provide estimation performance close to FL-CNN, but with greatly reduced computational complexity.

VI. CONCLUSION

In this paper, we propose a FL-CNN channel estimator that can provide high-precision estimation performance in urban vehicular environments. It combines RT, CNN, and FL techniques to pave the way for UVCE. Our results indicate that at 10 dB SNR, the proposed FL-CNN estimator outperforms the average of other benchmark estimators by 61% and 65% in channel estimation and data recovery performance, respectively. Although the proposed FL-CNN estimator can achieve impressive performance in the DT paradigm, it still faces challenges and limitations in practical implementations. To align this work with the requirements and challenges of 6G, potential future research directions are discussed as follows.

i) Ultra-reliable low-latency communication (URLLC): Ultra reliability and sub-millisecond latency are paramount for safety-critical applications in 6G IoE. To ensure URLLC for 6G, the integration of FL-CNN with beamforming and reconfigurable intelligent surfaces (RIS) techniques are promising solutions. For instance, by leveraging FL-CNN to optimize beamforming vectors in real time, ultra-reliable connections can be ensured even

in dense networks. In addition, the FL-CNN can enhance signal reliability with minimal latency by collaboratively optimizing RIS configurations across devices.

ii) AI-native architectures: 6G mandates embedded ML for proactive resource management (PRM). To achieve AI-native architecture, optimization of neural networks is required, which can be roughly divided into two aspects. The first aspect is the state-of-art architectures of CNN, such as VGG-16, ResNet-50, DenseNet-40 and MobileNet. The other aspect is the fine-tuning of parameters, such as feature map size, filter size, learning rate and other parameters in CNN, number of iterations in FL, etc.

iii) Sustainability: Energy-efficient channel estimation is significant to align with the green networking goal of 6G. In practical implementation, the training of the FL-CNN estimator needs to reduce complexity and thus reduce energy consumption. Some channel modeling and knowledge distillation techniques can be exploited. For example, the basis expansion model (BEM) can approximate a large number of channels based on existing channel data, reducing the number of RT simulations and speeding up the DT operations.

Overall, the FL-CNN and DT paradigm proposed in this paper lays the groundwork for future 6G networks and various smart city applications.

REFERENCES

[1] G. Acosta-Marum and M. A. Ingram, "Six time- and frequency-selective empirical channel models for vehicular wireless lans," in *2007 IEEE 66th Vehicular Technology Conference*, 2007, pp. 2134–2138.

[2] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in ofdm systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2018.

[3] P. Dong, H. Zhang, G. Y. Li, I. S. Gaspar, and N. NaderiAlizadeh, "Deep cnn-based channel estimation for mmwave massive mimo systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 5, pp. 989–1000, 2019.

[4] A. M. Elbir and S. Coleri, "Federated learning for channel estimation in conventional and ris-assisted massive mimo," *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 4255–4268, 2022.

[5] S. Arisdakessian, O. A. Wahab, A. Mourad, and H. Otrok, "Coalitional federated learning: Improving communication and training on non-iid data with selfish clients," *IEEE Transactions on Services Computing*, vol. 16, no. 4, pp. 2462–2476, 2023.

[6] G. Wang, "Interpret federated learning with shapley values," *arXiv preprint arXiv:1905.04519*, 2019.

[7] T. Wang, Z. Zheng, and F. Lin, "Federated learning framework based on trimmed mean aggregation rules," *Expert Systems with Applications*, p. 126354, 2025.

[8] N. Seshadri, V. Tarokh, and A. Calderbank, "Space-time codes for wireless communication: code construction," in *1997 IEEE 47th Vehicular Technology Conference. Technology in Motion*, vol. 2, 1997, pp. 637–641 vol.2.

[9] W. J. Farrell, L. Deville Cavellin, S. Weichenthal, M. Goldberg, and M. Hatzopoulou, "Capturing the urban canyon effect on particle number concentrations across a large road network using spatial analysis tools," *Building and Environment*, vol. 92, pp. 328–334, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360132315002061

[10] Z. Yun and M. F. Iskander, "Ray tracing for radio propagation modeling: Principles and applications," *IEEE access*, vol. 3, pp. 1089–1100, 2015.

[11] C. Ding and I. W.-H. Ho, "Digital-twin-enabled city-model-aware deep learning for dynamic channel estimation in urban vehicular environments," *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 3, pp. 1604–1612, 2022.

[12] S. Coleri, M. Ergen, A. Puri, and A. Bahai, "Channel estimation techniques based on pilot arrangement in ofdm systems," *IEEE Transactions on Broadcasting*, vol. 48, no. 3, pp. 223–229, 2002.

[13] O. Edfors, M. Sandell, J.-J. van de Beek, S. Wilson, and P. Borjesson, "Ofdm channel estimation by singular value decomposition," *IEEE Transactions on Communications*, vol. 46, no. 7, pp. 931–939, 1998.

[14] J. A. Fernandez, K. Borries, L. Cheng, B. V. K. Vijaya Kumar, D. D. Stancil, and F. Bai, "Performance of the 802.11p physical layer in vehicle-to-vehicle environments," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 3–14, 2012.

[15] A. K. Gizzini, M. Chafii, A. Nimr, and G. Fettweis, "Deep learning based channel estimation schemes for ieee 802.11p standard," *IEEE Access*, vol. 8, pp. 113 751–113 765, 2020.

[16] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, and F. Gutierrez, "Millimeter wave mobile communications for 5g cellular: It will work!" *IEEE Access*, vol. 1, pp. 335–349, 2013.

[17] P. Series, "Effects of building materials and structures on radiowave propagation above about 100 mhz," *recommendation itu-r*, pp. 2040–1, 2015.

[18] R. Paknys, *Uniform Theory of Diffraction*, 2016, pp. 268–316.

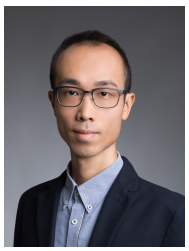
[19] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 599–618, 2001.

[20] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[21] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[22] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *Advances in neural information processing systems*, vol. 33, pp. 7611–7623, 2020.

Cao Ding received the B.Eng. degree in automation from Beijing University of Chemical Technology, Beijing, China, and the M.Sc. degree in electronic and information engineering from The Hong Kong Polytechnic University, Hong Kong. He is now a Ph.D. student of The Hong Kong Polytechnic University. His research interests include vehicular networks and intelligent transport systems (ITS), specifically in vehicular ad hoc network (VANET). His research focuses most on the digital twin of vehicular networks and intelligent transport systems.



Ivan Wang-Hei Ho (M'10–SM'18) received the B.Eng. and M.Phil. degrees in information engineering from The Chinese University of Hong Kong, Hong Kong, in 2004 and 2006, respectively, and the Ph.D. degree in electrical and electronic engineering from the Imperial College London, London, U.K., in 2010. He was a Research Intern with the IBM Thomas J. Watson Research Center, Hawthorne, NY, USA, and a Postdoctoral Research Associate with the System Engineering Initiative, Imperial College London. In 2010, he cofounded P2 Mobile Technologies Ltd., where he was the Chief Research and Development Engineer. He is currently an Associate Professor with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong. His research interests include wireless communications and networking, specifically in vehicular networks, intelligent transportation systems (ITS), and Internet of things (IoT). He primarily invented the MeshRanger series wireless mesh embedded system, which received the Silver Award in Best Ubiquitous Networking at the Hong Kong ICT Awards 2012. His work on indoor positioning and IoT also received the Gold Medal at the International Trade Fair Ideas and Inventions New Products (iENA) in Germany in 2019, and the Gold Medal with the Organizer's Choice Award in the International Invention Innovation Competition in Canada (iCAN) in 2020. He is currently an Associate Editor for the IEEE Access and IEEE Transactions on Circuit and Systems II, and was the TPC Co-Chair for the PERSIST-IoT Workshop in conjunction with ACM MobiHoc 2019 and IEEE INFOCOM 2020.