LLM4CP: Adapting Large Language Models for Channel Prediction

Boxun Liu, Xuanyu Liu, Shijian Gao, Xiang Cheng, Liuqing Yang

Abstract—Channel prediction is an effective approach for reducing the feedback or estimation overhead in massive multi-input multi-output (m-MIMO) systems. However, existing channel prediction methods lack precision due to model mismatch errors or network generalization issues. Large language models (LLMs) have demonstrated powerful modeling and generalization abilities, and have been successfully applied to cross-modal tasks, including the time series analysis. Leveraging the expressive power of LLMs, we propose a pre-trained LLM-empowered channel prediction (LLM4CP) method to predict the future downlink channel state information (CSI) sequence based on the historical uplink CSI sequence. We fine-tune the network while freezing most of the parameters of the pre-trained LLM for better cross-modality knowledge transfer. To bridge the gap between the channel data and the feature space of the LLM, preprocessor, embedding, and output modules are specifically tailored by taking into account unique channel characteristics. Simulations validate that the proposed method achieves state-of-the-art (SOTA) prediction performance on full-sample, few-shot, and generalization tests with low training and inference costs.

Keywords—channel prediction, massive multi-input multi-output (m-MIMO), large language models (LLMs), fine-tuning, time-series

B. X. Liu, X. Y. Liu, X. Cheng. State Key Laboratory of Advanced Optical Communication Systems and Networks, School of Electronics, Peking University, Beijing 100871, China (e-mail: boxunliu@stu.pku.edu.cn; xvanyvliu@gmail.com; xiangcheng@pku.edu.cn).

S. J. Gao. Internet of Things Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511400, China (e-mail: shijiangao@hkust-gz.edu.cn).

L. Q. Yang. Internet of Things Thrust and Intelligent Transportation Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511400, China, and Department of Electronic and Computer Engineering and Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Hong Kong, China (e-mail: lqyang@ust.hk).

I. INTRODUCTION

Massive multi-input multi-output (MIMO) technology is regarded as a core technology in the fifth-generation (5G) and beyond 5G mobile communication systems[1] for enhancing the spectral efficiency (SE). Accurate channel state information (CSI) plays a fundamental role in facilitating m-MIMO related design, such as transceiver optimization[2], adaptive modulation[3], resource allocation[4], and so on. Typically, CSI is acquired through channel estimation[5-6] whose updating frequency is dictated by the channel coherence time. For mobility scenarios involving high-velocity users, the shortened channel coherence time significantly increases the overhead of channel estimation, thereby leading to an appreciable reduction in the system SE. Additionally, in frequency-division duplex (FDD) systems where channel reciprocity does not hold, the base station (BS) can only obtain downlink CSI through user feedback, resulting in increased overhead and delay.

Channel prediction[7-9] is a promising technology to reduce the CSI acquisition overhead, which predicts future CSI based on historical CSI data. The historical CSI and the predicted CSI can either be in the same or different frequency bands, corresponding to time-division duplex (TDD) and FDD modes, respectively. For instance, in FDD systems, the downlink CSI can be inferred from previous uplink CSI, thereby avoiding the need for channel estimation and feedback. Existing studies on channel prediction can be categorized into three types, i.e., model-based methods, deep learning-based methods, and hybrid (physics-informed deep learning-based) methods.

For model-based methods, several parametric models have been investigated for temporal channel prediction, including the autoregressive (AR) model[10], the sum-of-sinusoids model[11], and the polynomial extrapolation model[12]. In Ref. [13], a prony-based angular-delay domain (PAD) channel prediction algorithm was proposed by exploiting the high

resolution of multipath angle and delay in massive MIMO orthogonal frequency division multiplexing (OFDM) systems. Additionally, a joint angle-delay-Doppler (JADD) CSI acquisition framework was designed[14] for FDD systems, utilizing the partial reciprocity between uplink and downlink channels. Nevertheless, the effectiveness of model-based approaches is heavily dependent on the accuracy of the theoretical model, which can be challenging to fit into the complex multipath characteristics of the practical channel.

Deep learning has demonstrated its powerful capabilities in automatically adapting to data distribution without prior assumptions. Recently, several classical neural networks[15-20] have been applied for channel prediction tasks. In Ref. [15], a multi-layer perceptron (MLP)-based channel prediction method demonstrates comparable performance to vector Kalman filter (VKF)-based channel predictor. To better learn temporal variations, recurrent neural network (RNN)[16] and long short-term memory (LSTM)[17] are applied to channel prediction. In addition, a transformer-based parallel channel prediction scheme[18] was proposed to avoid error propagation in the sequential CSI prediction process. In Refs. [19-20], convolutional neural networks (CNN) and generative adversarial networks (GANs) are utilized for downlink CSI prediction by treating the prediction process as image processing. However, due to the lack of consideration for the unique structure of the channel, the above methods struggle to handle complex channel prediction tasks and exhibit high complexity.

Therefore, a few physics-informed deep learning-based[21-22] works have considered the unique characteristics of CSI, referred to as hybrid methods[23-24]. For instance, a 3-dimensional (3D) complex CNN-based predictor[23] is utilized to capture temporal and spatial correlations based on angle-delay domain representation. In Ref. [24], a ConvLSTM-based spatio-temporal neural network (STNN) is proposed to parallel process high-dimensional spatial-temporal CSI. Nevertheless, the scalability of hybrid methods is relatively poor, requiring sufficient understanding of the channel structure.

Despite significant progress achieved by deep learning-empowered methods, there are still some shortcomings limiting their application in practical scenarios. Firstly, the predictive capability is constrained by the size of networks. Existing methods struggle to accurately model complex spatial, temporal, and frequency relationships, especially for high-velocity scenarios and FDD systems. Secondly, in contrast to model-based approaches, deep learning-based methods exhibit poor generalization ability, requiring retraining when the CSI distribution changes. Although a few studies aim to improve generalization ability by meta-learning[25] or hypernetwork[26], the additional adaptation stage or the hypernetwork branch increases the operational complexity. In summary, existing deep learning-empowered prediction models struggle to meet the requirements for high generalization performance and accurate prediction capabilities.

Large language models (LLMs) have achieved tremendous success in the field of natural language processing (NLP) and have led to a new paradigm, namely, fine-tuning models pre-trained on large-scale datasets for downstream tasks with few or zero labels. This provides a promising solution to address the shortcomings of existing channel prediction schemes. However, previous downstream tasks were limited to the field of NLP. Recently, several studies[27-31] have provided initial evidence of the powerful cross-modal transfer capabilities of pre-trained LLMs. For instance, Ref. [28] fine-tunes frozen pre-trained LLM on time series datasets and achieves state-of-the-art (SOTA) performance on main time series analysis tasks. In Ref. [30], a traffic prediction framework based on pre-trained LLM (TPLLM) is proposed with the low-rank adaptation (LoRA) fine-tuning approach. Still, existing cross-modal fine-tuning works mainly focused on temporal or spatio-temporal series prediction rather than channel prediction tasks. Unlike time series forecasting tasks, there are certain difficulties in adapting LLM for channel prediction[32]. First, CSI is the high-dimensional structural data with the multi-path channel model rather than simple one-dimensional data, which increases the complexity of processing. Moreover, there is a huge domain gap between CSI and the natural language. In addition, for FDD channel prediction tasks, extrapolation is achieved in both the time and frequency domain, further increasing the difficulty.

Unlike the existing approaches that designed entire networks specifically for channel prediction tasks, in this paper, we attempt to adapt LLM for multi-input single-output (MISO) OFDM channel prediction to achieve improved predictive capability and generalization ability. Specifically, we build a channel prediction neural network based on pre-trained GPT-2 and fine-tune it to predict the future downlink CSI sequence based on the historical uplink CSI sequence. Unlike existing studies where LLMs are used for time series prediction, we fully consider the specific characteristics of the channel, and design preprocessor, embedding, and output modules to bridge the gap between CSI data and the LLM. In detail, considering multi-path effects, we process CSI from both the frequency and delay domain to extract the underlying physical propagation feature. To fully preserve the general knowledge in the pre-trained LLM, most of its parameters are frozen during training. Simulations evaluate the proposed method for both the TDD and FDD channel prediction tasks and demonstrate its superiority among existing baselines. The main contributions of our work are summarized as follows.

• We propose a novel LLM-empowered channel prediction (LLM4CP) method for MISO-OFDM systems, i.e., fine-tuning pre-trained GPT-2 on channel prediction datasets. To the best of our knowledge, it is the first attempt to adapt pre-

trained LLM for channel prediction.

- In light of the unique channel characteristics, we design dedicated modules and processing to bridge the gap between channel data and the feature space of the LLM, thereby facilitating cross-modality knowledge transfer.

- Preliminary results validate the SOTA performance of the proposed method on TDD/FDD channel prediction tasks. In addition, it demonstrates superior few-shot and generalization prediction performance, as well as low training and inference costs.

Notation: $(\cdot)^{\mathrm{T}}$, $(\cdot)^{\mathrm{H}}$, $(\cdot)^{\dagger}$, $\|\cdot\|$, and $\|\cdot\|_{\mathrm{F}}$ denote the transpose, the conjugate transpose, pseudo-inverse, l_2 norm, and Frobenius norm, respectively. $\otimes$ represents the Kronecker product operation. $\boldsymbol{a}[i]$ is the ith element of a vector $\boldsymbol{a}$, and $\boldsymbol{A}[i,j]$ denotes the element of a matrix or tensor $\boldsymbol{A}$ at the ith row and the jth column. $\boldsymbol{A}[:,i]$ denotes the ith column of $\boldsymbol{A}$.

II. SYSTEM MODEL

We focus on a single-cell MISO-OFDM system comprising a BS and a mobile user. The BS is equipped with a dual-polarized uniform planar array (UPA) with $N_{\mathrm{t}} = N_{\mathrm{h}} \times N_{\mathrm{v}}$ antennas, where N_{h} and N_{v} represent the number of antennas in the horizontal and vertical directions, respectively. The mobile user is equipped with an omnidirectional antenna, and it can be readily applied for cases with multiple antennas through parallel processing. The system is capable of operating in both TDD and FDD modes.

A. Channel Model

For a given polarization, adopting the classical cluster-based multi-path channel model[33], the downlink CSI between BS and the user at time t and frequency f is

$$\boldsymbol{h}(t,f) = \sum_{n=1}^{N} \sum_{m=1}^{M_n} \beta_{n,m} e^{\mathrm{j}[2\pi(\upsilon_{n,m}t - f\tau_{n,m}) + \Phi_{n,m}]} \boldsymbol{a}(\theta_{n,m}, \varphi_{n,m}), \quad (1)$$

where N and M_n are the number of clusters and paths in each cluster, respectively. $\beta_{n,m}$, $\upsilon_{n,m}$, $\tau_{n,m}$, and $\Phi_{n,m}$ represent the complex path gain, Doppler frequency shift, delay, and random phase of the mth path of the nth cluster, respectively. Assuming the instantaneous velocity of the mobile user at time t is v and the angle between the direction of the velocity vector and the path is $\phi_{m,n}$. The Doppler frequency shift is derived as $\upsilon_{n,m} = \frac{v f \cos \phi_{m,n}}{2\pi c}$, where c represents the velocity of light. It is worth noting that the Doppler frequency shift caused by the user's movement is the main factor contributing to the time variation of the channel. $\boldsymbol{a}(\theta_{n,m}, \varphi_{n,m})$ represents the steering vector of the corresponding path, where $\theta_{n,m}$ and $\varphi_{n,m}$ denote the azimuth and elevation angles. Considering the structural characteristics of UPA as shown in Fig. 1, $\boldsymbol{a}(\theta_{n,m}, \varphi_{n,m})$ is de-
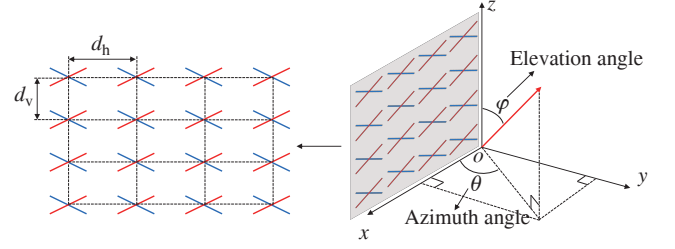

Fig. 1 An illustration of the architecture of a dual-polarized UPA[13]

rived as

$$\boldsymbol{a}(\theta_{n,m}, \varphi_{n,m}) = \boldsymbol{a}_{\mathrm{h}}(\theta_{n,m}, \varphi_{n,m}) \otimes \boldsymbol{a}_{\mathrm{v}}(\theta_{n,m}), \quad (2)$$

where $\boldsymbol{a}_{\mathrm{h}}(\theta, \varphi)$ and $\boldsymbol{a}_{\mathrm{v}}(\theta)$ are the horizontal and vertical steering vector, respectively. $\boldsymbol{a}_{\mathrm{h}}(\theta, \varphi)[i] = e^{\frac{\mathrm{j}2\pi i f d_{\mathrm{h}} \sin(\varphi) \cos(\theta)}{c}}$ and $\boldsymbol{a}_{\mathrm{v}}(\theta)[i] = e^{\frac{\mathrm{j}2\pi i f d_{\mathrm{v}} \sin(\theta)}{c}}$, where d_{h} with d_{v} being the antenna spacing along the horizontal and vertical directions, respectively.

B. Signal Model

We consider a downlink MISO-OFDM signal transmission process, where K_{s} subcarriers are activated, with the kth subcarrier denoted as f_k. According to Eq. (1), the downlink CSI at time t and the kth subcarrier is $\boldsymbol{h}_k = \boldsymbol{h}(t, f_k)$, which can be obtained through channel estimation or prediction. Considering transmit precoding at the BS side, the received downlink signal of the kth subcarrier at the user side is derived as

$$y_k = \boldsymbol{h}_k^{\mathrm{H}} \boldsymbol{w}_k x_k + n_k, \quad (3)$$

where n_k is the additive white Gaussian noise (AWGN) with noise power σ_n^2, and $\boldsymbol{w}_k$ is the transmit precoder. The achievable SE[34] of the downlink transmission process is derived as

$$R = \sum_{k=1}^{K_{\mathrm{s}}} \mathrm{lb}\left(1 + \frac{|\boldsymbol{h}_k^{\mathrm{H}} \boldsymbol{w}_k|^2}{\sigma_n^2}\right). \quad (4)$$

To maximize the downlink transmission rate, the matched-filtering based precoding is applied as

$$\boldsymbol{w}_k = \frac{\boldsymbol{h}_k}{\|\boldsymbol{h}_k\|}. \quad (5)$$

It is worth noting that inaccurate $\boldsymbol{h}_k$ will lead to mismatched $\boldsymbol{w}_k$, thereby impairing the SE.

III. PROBLEM FORMULATION FOR CHANNEL PREDICTION

In this section, we introduce a channel prediction-based transmission scheme and formulate a problem of inferring future downlink CSI based on historical uplink CSI.
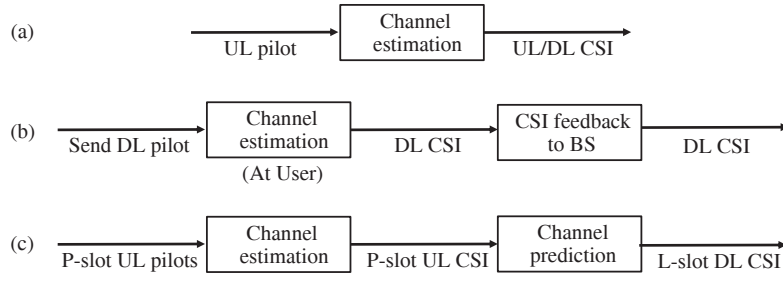
Fig. 2 An illustration of three schemes for downlink CSI acquisition: (a) Traditional downlink CSI acquisition scheme for TDD systems; (b) Traditional downlink CSI acquisition scheme for FDD systems; (c) Channel prediction-based downlink CSI acquisition for TDD/FDD systems
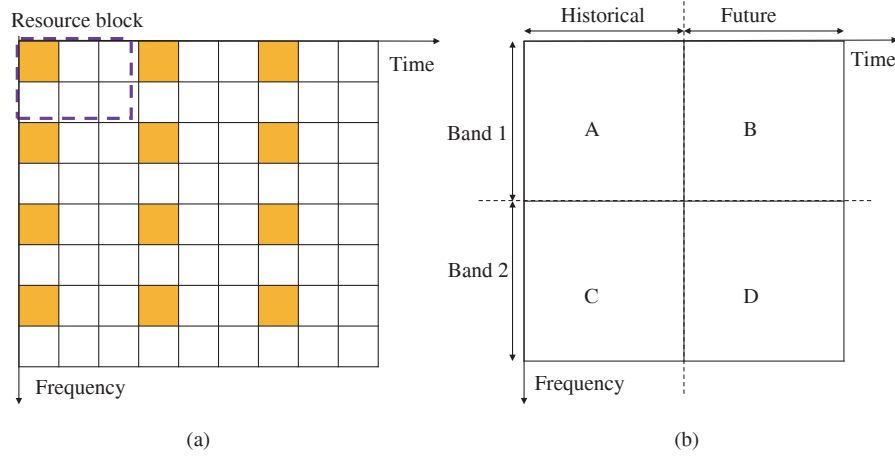

Fig. 3 An illustration of the resource block and channel prediction scheme in the time-frequency domain: (a) Resource block (RB); (b) Channel prediction in the time-frequency domain

A. Channel Prediction-Based Transmission

Traditional downlink CSI acquisition scheme for TDD and FDD systems are illustrated in Fig. 2 (a) and (b), respectively. In TDD systems, due to channel reciprocity, the downlink CSI can be obtained at the BS side by channel estimation on uplink pilots. In FDD systems where the frequency of the uplink and downlink channels differs, downlink CSI can only be estimated at the user side and then fed back to the BS. However, there are some shortcomings in existing downlink CSI acquisition methods. First, the CSI estimation and feedback process incur additional computational and transmission time overhead, causing channel aging[18] in high dynamic scenarios. In addition, extra downlink pilots occupy some of the time-frequency resources, reducing the SE of FDD systems.

Channel prediction-based transmission scheme provides a promising solution to address the above two drawbacks, as shown in Fig. 2(c). Specifically, it predicts future downlink CSI sequences based on historical uplink CSI sequences, avoiding the overhead of downlink pilots and feedback delay. For further clarification, the time and frequency relationship between uplink and downlink CSI of the channel prediction-based scheme can be illustrated in Fig. 3(b). Region A represents the uplink CSI, while regions B and D correspond to the predicted downlink CSI under TDD and FDD modes, respectively. Each time-frequency region consists of multiple time-frequency resource blocks (RBs), and each RB contains a pilot, as shown in Fig. 3(a). In the following channel prediction process, we only consider the CSI associated with the pilots' positions, while CSI between pilots can be obtained through interpolation methods. We assume that the uplink and downlink links have the same bandwidth and each covers K RBs in the frequency domain. In the time domain, future L RBs are predicted based on historical P RBs. For simplicity, we denote the uplink and downlink CSI of each RB as $h_{\mathrm{u}}^{k,s}$ and $h_{\mathrm{d}}^{k,s}$, where k and s represent the indices of RBs in the frequency domain and time domain, respectively.

B. Problem Formulation

We aim to accurately predict future downlink CSI of $K \times L$ RBs based on historical CSI of $K \times P$ RBs. The uplink CSI of K subcarriers at time i is represented in matrix form as

$$\boldsymbol{H}_{\mathrm{u}}^{i} = [\boldsymbol{h}_{\mathrm{u}}^{1,i}, \cdots, \boldsymbol{h}_{\mathrm{u}}^{K,i}]^{\mathrm{H}}. \tag{6}$$

The downlink CSI can be obtained similarly as $\boldsymbol{H}_{\mathrm{d}}^{i} = [\boldsymbol{h}_{\mathrm{d}}^{1,i}, \cdots, \boldsymbol{h}_{\mathrm{d}}^{K,i}]^{\mathrm{H}}$. The normalized mean square error (NMSE)[18] between predicted and actual downlink CSI is
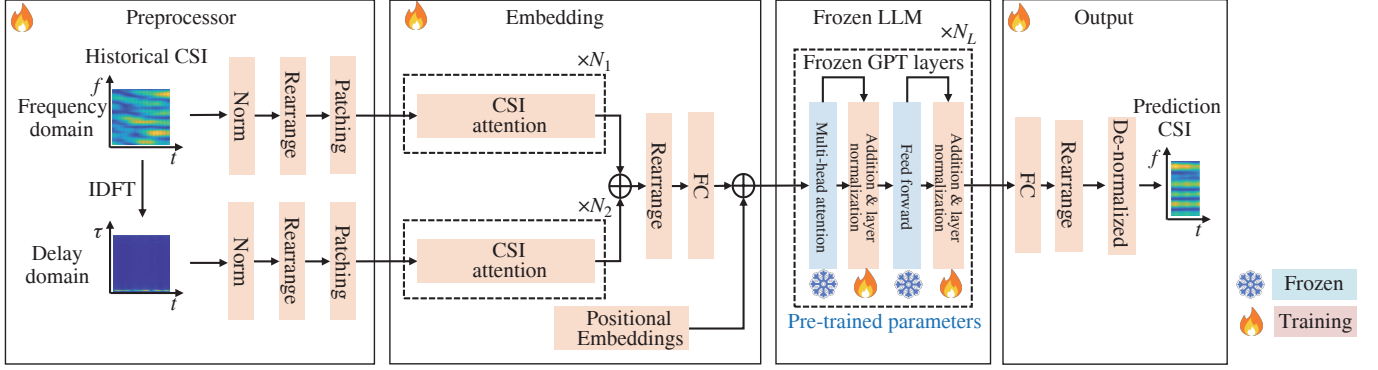
Fig. 4 An illustration of the network architecture of LLM4CP

used to evaluate the prediction accuracy. Then the entire problem can be described as follows.

$$\min_{\Omega} \quad \text{NMSE} = \mathbb{E}\left\{ \frac{\sum_{i=1}^{L} \|\hat{\boldsymbol{H}}_{\text{d}}^{s+i} - \boldsymbol{H}_{\text{d}}^{s+i}\|_{\text{F}}^2}{\sum_{i=1}^{L} \|\boldsymbol{H}_{\text{d}}^{s+i}\|_{\text{F}}^2} \right\} \quad (7a)$$

$$\text{s.t.} \quad (\hat{\boldsymbol{H}}_{\text{d}}^{s+1}, \cdots, \hat{\boldsymbol{H}}_{\text{d}}^{s+L}) = f_{\Omega}(\boldsymbol{H}_{\text{u}}^{s}, \cdots, \boldsymbol{H}_{\text{u}}^{s-P+1}), \quad (7b)$$

where $\boldsymbol{H}_{\text{u}}^{i}$, $\hat{\boldsymbol{H}}_{\text{d}}^{i}$, and $\boldsymbol{H}_{\text{d}}^{i}$ represent the estimated uplink CSI, predicted downlink CSI, and actual downlink CSI of RBs at time i, respectively. f_{Ω} is the constructed mapping function and Ω is its variable parameters. In previous work, f_{Ω} represents either a parameterized model[10-12] or a deep learning network[18-20]. In this paper, we consider a novel pre-trained LLM-based channel prediction method to achieve higher prediction accuracy and generalization capability. Specifically, f_{Ω} is the proposed LLM-based neural network, and Ω represents the trainable parameters. It is worth noting that the trained network is designed to handle the CSI prediction under two polarizations.

IV. LLM FOR CHANNEL PREDICTION

In this section, we propose an LLM-empowered MISO-OFDM channel prediction method, named LLM4CP, to predict the future downlink CSI sequence based on the historical uplink CSI sequence. In order to adapt text-based pre-trained LLM to the complex matrix format of CSI data, specific modules are designed for format conversion and feature extraction, including preprocessor, embedding, backbone, and output. The network components are shown in Fig. 4 and the training process is illustrated in detail below.

A. Preprocessor Module

Given the uplink CSI at time i as $\boldsymbol{H}_{\text{u}}^{i} \in \mathbb{C}^{K \times N_t}$, directly applying it to the network will bring significant complexity and training time, especially for a large number of antennas and subcarriers. Therefore, we parallelize the processing of antennas, i.e., predicting the CSI for each pair of transmitter

and receiver antennas separately. For the jth transmit antenna, the input sample of the network $\boldsymbol{H}_f \in \mathbb{C}^{K \times P}$ is derived as

$$\boldsymbol{H}_f = [\boldsymbol{H}_{\text{u}}^1[:,j], \cdots, \boldsymbol{H}_{\text{u}}^P[:,j]], \quad (8)$$

where P is the temporal length of historical CSI.

Given that the delay domain, as referenced in Ref. [13], is the dual domain of the frequency domain, it serves to characterize the delay information of each multipath component, providing a more comprehensive perspective on frequency domain information[23]. Consequently, we simultaneously offer a representation in the delay domain through inverse discrete Fourier transform (IDFT) as well.

$$\boldsymbol{H}_{\tau} = \boldsymbol{F}_K^{\text{H}} \boldsymbol{H}_f, \quad (9)$$

where $\boldsymbol{F}_K$ represents the K dimensional DFT matrix. Since neural networks generally deal with real numbers, we convert $\boldsymbol{H}_f$ and $\boldsymbol{H}_{\tau}$ into real tensors $\boldsymbol{X}_f \in \mathbb{R}^{2 \times K \times P}$ and $\boldsymbol{X}_{\tau} \in \mathbb{R}^{2 \times K \times P}$, respectively. To facilitate network training and convergence, we first normalize the input data as $\bar{\boldsymbol{X}}_f = \frac{\boldsymbol{X}_f - \mu_f}{\sigma_f}$ and $\bar{\boldsymbol{X}}_{\tau} = \frac{\boldsymbol{X}_{\tau} - \mu_{\tau}}{\sigma_{\tau}}$, where (μ_f, μ_{τ}) and $(\sigma_f, \sigma_{\tau})$ represent the mean value and standard deviation of a batch of corresponding domain input data. Then the tensors are rearranged to merge feature dimensions, i.e., $\tilde{\boldsymbol{X}}_f \in \mathbb{R}^{2K \times P}$ and $\tilde{\boldsymbol{X}}_{\tau} \in \mathbb{R}^{2K \times P}$.

To capture the local temporal features and reduce computational complexity, patching[35] operation along the temporal dimension is adopted, as shown in Fig. 5. Specifically, $\tilde{\boldsymbol{X}}_f$ and $\tilde{\boldsymbol{X}}_{\tau}$ are grouped into non-overlapping patches of size N as the $\boldsymbol{X}_f^p \in \mathbb{R}^{2K \times N \times P'}$ and $\boldsymbol{X}_{\tau}^p \in \mathbb{R}^{2K \times N \times P'}$, where $P' = \lceil \frac{P}{N} \rceil$ is the number of patches. Zero-padding is applied to the last patch that is not fully filled.

B. Embedding Module

The embedding module is designed for preliminary feature extraction before LLM, including CSI attention and position embedding. First, the patched tensors are processed by corresponding CSI attention modules, as detailed below.
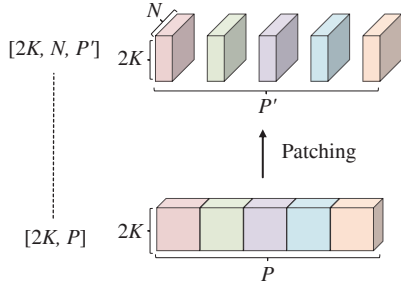
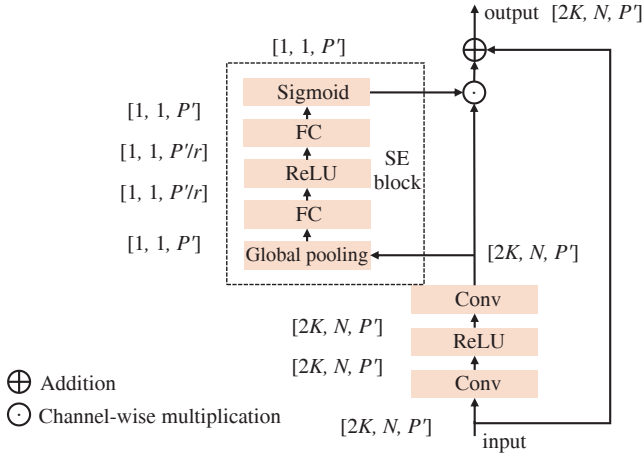**Fig. 5** An illustration of the patching operation

Fig. 6 An illustration of the network structure and dimension changes of the CSI attention module. The network within the dashed box is the squeeze-and-excitation (SE) module

Inspired by Ref. [36], the CSI attention module is designed for feature analysis, as shown in Fig. 6. The dimensions of each tensor during processing are described. For simplicity, we use $\boldsymbol{X}^{\mathrm{i}} \in \mathbb{R}^{2K \times N \times P'}$ and $\boldsymbol{X}^{\mathrm{o}} \in \mathbb{R}^{2K \times N \times P'}$ to represent the input and output tensors of the proposed CSI attention module, respectively. First, the feature map is obtained as

$$\boldsymbol{X}^{\mathrm{fm}} = \mathrm{Conv}(\mathrm{ReLU}(\mathrm{Conv}(\boldsymbol{X}^{\mathrm{i}}))) \in \mathbb{R}^{2K \times N \times P'}, \quad (10)$$

where $\mathrm{Conv}(\cdot)$ represents the 2D convolution operator, and $\mathrm{ReLU}(\cdot)$ represents the ReLU[37] activation function. The convolution layers extract temporal and frequency features within each patch and integrate features across different patches. Then $\boldsymbol{X}^{\mathrm{fm}}$ is passed through the SE block, which compromises squeeze-and-excitation parts, to obtain weights for different patches, i.e.,

$$\boldsymbol{X}^{\mathrm{SE}} = \mathrm{SE}(\boldsymbol{X}^{\mathrm{fm}}) \in \mathbb{R}^{1 \times 1 \times P'}. \quad (11)$$

For the squeeze part, the global average pooling is first applied to generate channel-wise statistics $\boldsymbol{X}^{\mathrm{GAP}} \in \mathbb{R}^{1 \times 1 \times P'}$ and

$$\boldsymbol{X}^{\mathrm{GAP}}[i] = \frac{1}{2KP'} \sum_{j=1}^{2K} \sum_{k=1}^{P'} \boldsymbol{X}^{\mathrm{fm}}[j, k, i]. \quad (12)$$

For the excitation part, two fully connected (FC) layers are adopted to model the correlation between different patches. To reduce computational complexity, the first FC layer shrinks the patch dimension size from P' to P'/r with $r \geqslant 1$, followed by a ReLU operation. Then, the second FC layer expands the patch dimension to P'. Following this, the sigmoid function is utilized to generate the attention weight tensor $\boldsymbol{X}^{\mathrm{SE}}$, with each element falling within the range $[0, 1]$. The scale operation weights each patch with $\boldsymbol{X}^{\mathrm{SE}}$ and obtains scaled features $\boldsymbol{X}^{\mathrm{Sca}} \in \mathbb{R}^{2K \times N \times P'}$, where

$$\boldsymbol{X}^{\mathrm{Sca}}[:,:,i] = \boldsymbol{X}^{\mathrm{SE}}[i] \times \boldsymbol{X}^{\mathrm{fm}}[:,:,i]. \quad (13)$$

Then the residual connection is established as

$$\boldsymbol{X}^{o} = \boldsymbol{X}^{\mathrm{Sca}} + \boldsymbol{X}^{i}. \quad (14)$$

It is worth noting that the CSI attention module can be concatenated multiple times to enhance feature extraction effectiveness. The processed $\boldsymbol{X}_f^p$ and $\boldsymbol{X}_\tau^p$ are added as

$$\boldsymbol{X}^{\mathrm{CA}} = \mathrm{CA}^{(N_1)}(\boldsymbol{X}_f^p) + \mathrm{CA}^{(N_2)}(\boldsymbol{X}_\tau^p), \quad (15)$$

where CA^N represents the CSI attention module cascaded N times. To adapt to the input format of LLM, $\boldsymbol{X}^{\mathrm{CA}}$ is first rearranged to $\tilde{\boldsymbol{X}}^{\mathrm{CA}} \in \mathbb{R}^{2KN \times P'}$ and then mapped to $\bar{\boldsymbol{X}}^{\mathrm{CA}} \in \mathbb{R}^{F \times P'}$ with a single FC layer, where F is the feature dimension of the pre-trained LLM. To incorporate positional information, non-learnable positional encoding[38] $\boldsymbol{X}^{\mathrm{PE}} \in \mathbb{R}^{F \times P'}$ is introduced to enable the network to distinguish between patches at different positions. Its structure is shown as

$$\boldsymbol{X}^{\mathrm{PE}}(2i, j) = \sin\left(\frac{j}{10\,000^{\frac{2i}{F}}}\right), \quad (16)$$

and

$$\boldsymbol{X}^{\mathrm{PE}}(2i+1, j) = \cos\left(\frac{j}{10\,000^{\frac{2i}{F}}}\right), \quad (17)$$

finally, the embedding $\boldsymbol{X}^{\mathrm{EB}} \in \mathbb{R}^{F \times P'}$ is obtained as

$$\boldsymbol{X}^{\mathrm{EB}} = \bar{\boldsymbol{X}}^{\mathrm{CA}} + \boldsymbol{X}^{\mathrm{PE}}. \quad (18)$$

C. Backbone Network

Recent studies[28,30] have revealed that pre-trained LLMs can be fine-tuned for cross-modal downstream tasks, including time series analysis. In other words, during the pre-training process on extensive textual data, LLM has acquired general knowledge[28]. Inspired by this, we attempt to leverage the universal modeling capability of LLMs for channel prediction tasks. However, the pre-trained LLM cannot directly process non-linguistic data due to the significant disparity between textual and CSI information. Therefore, sim-

ilar to how text is preprocessed into tokens by the tokenizer, we obtain the embeddings of CSI data using the proposed pre-processor and embedding module. The preprocessed CSI "tokens" are then fed into the backbone networks of the LLM, i.e.,

$$\boldsymbol{X}^{\mathrm{LLM}} = \mathrm{LLM}(\boldsymbol{X}^{\mathrm{EB}}) \in \mathbb{R}^{F \times P'}, \tag{19}$$

where $\mathrm{LLM}(\cdot)$ denotes backbone networks of the LLM.

Without loss of generality, GPT-2[39] is chosen as the LLM backbone in this work. The backbone of GPT-2 is composed of a learnable positional embedding layer and stacked transformer decoders[38], where the number of stacks and feature dimensions can be flexibly adjusted according to the requirements. Each layer consists of multi-head attention layers, feed forward layers, addition, and layer normalization, as shown in Fig. 4. During the training process, multi-head attention and feed forward layers are frozen to retain universal knowledge, while addition, layer normalization, and positional embedding are fine-tuned for adapting the LLM to the channel prediction task. It is worth noting that in the proposed method, the GPT-2 backbone can be flexibly replaced with other LLMs, such as Llama[40]. The selection of the type and size of the LLM needs to consider the trade-off between training costs and performance.

D. Output Module

The output is designed to convert the output features of the LLM into the final prediction results. First, two FC layers are adopted to transform the dimension of $\boldsymbol{X}^{\mathrm{LLM}}$.

$$\hat{\boldsymbol{X}} = \mathrm{FC}(\mathrm{FC}(\boldsymbol{X}^{\mathrm{LLM}})) \in \mathbb{R}^{2K \times L}, \tag{20}$$

where $\mathrm{FC}(\cdot)$ represents the FC layer and L is the prediction length. Then, $\hat{\boldsymbol{X}}$ is rearranged to $\boldsymbol{X}^{\mathrm{re}} \in \mathbb{R}^{2 \times K \times L}$, where the first and the second dimension respectively correspond to the real part and the imaginary part. Then $\boldsymbol{X}^{\mathrm{re}}$ is de-normalized to generate the final output of the network, i.e.,

$$\boldsymbol{X}^{\mathrm{de}} = \sigma_f \boldsymbol{X}^{\mathrm{re}} + \mu_f, \tag{21}$$

and the final prediction result $\hat{\boldsymbol{H}}_f \in \mathbb{C}^{K \times L}$ is obtained as

$$\hat{\boldsymbol{H}}_f = \boldsymbol{X}^{\mathrm{de}}[1,:,:] + \mathrm{j}\boldsymbol{X}^{\mathrm{de}}[2,:,:], \tag{22}$$

where $\mathrm{j} = \sqrt{-1}$ represents the imaginary unit.

E. Training Configuration

The proposed neural network is initially trained on channel prediction datasets and then applied for testing. In the training phase, the ground truth of the predicted CSI $\hat{\boldsymbol{H}}_f$ is available. Then we obtain the ground truth of the network output as $\boldsymbol{X}^{\mathrm{gt}} \in \mathbb{R}^{2 \times K \times L}$. The NMSE[23] is adopted as the loss function

to minimize the prediction error, i.e.,

$$\mathrm{Loss} = \frac{\|\boldsymbol{X}^{\mathrm{de}} - \boldsymbol{X}^{\mathrm{gt}}\|_{\mathrm{F}}^2}{\|\boldsymbol{X}^{\mathrm{gt}}\|_{\mathrm{F}}^2}. \tag{23}$$

In addition, the validation loss also adopts the same loss function. It is worth noting that the multi-head attention and feed forward layers of the pre-trained GPT-2 are frozen, while other parameters of the network are trainable, as shown in Fig. 4. Since the former includes the main parameters of the network, the number of trainable parameters is relatively small, which will be explained in detail in section V.B.6). The model with the smallest validation loss is saved for the testing phase.

V. EXPERIMENTS

In this section, we first illustrate the simulation settings and then evaluate the prediction performance of the proposed LLM4CP method.[1]

A. Simulation Setup

1) Dataset: We adopt the widely used channel generator QuaDRiGa[41] to simulate time-varying CSI datasets compliant with 3GPP standards[33]. We consider a MISO-OFDM system, where a BS is equipped with a dual-polarized UPA with $N_{\mathrm{h}} = N_{\mathrm{v}} = 4$, and a user is equipped with a single omni-directional antenna. The antenna spacing is half of the wavelength at the center frequency. We suppose the bandwidth of both the uplink and the downlink channels is 8.64 MHz and covers $K = 48$ RBs, i.e., the frequency interval of pilots is 180 kHz. For both the TDD and FDD modes, we set the center frequency of the uplink channel as 2.4 GHz. For FDD modes, the uplink and downlink channels are adjacent. We predict future $L = 4$ RBs based on historical $P = 16$ RBs and set the time interval of pilots as 0.5 ms. We consider the 3GPP urban macro (UMa) channel model[33] and non-line-of-sight (NLOS) scenarios. The number of clusters is 21 and the number of paths per cluster is 20. The initial position of the user is randomized and the motion trajectory is set as linear type. The training dataset and validation dataset respectively contain 8 000 and 1 000 samples, with user velocities uniformly distributed between 10 km/h and 100 km/h. The testing dataset contains 10 velocities ranging from 10 km/h to 100 km/h, with 1 000 samples for each velocity.

2) Baselines: To validate the superiority of the proposed method, several model-based and deep learning-based channel prediction methods are implemented as baselines.

- **PAD**[13]: PAD is an advanced model-based channel prediction method to overcome the curse of mobility for TDD

---

[1]The code is publicly available at Github

systems. In this approach, the order of the predictor is set as $N = 8$, and historical $2N - 1 = 15$ CSI samples are utilized for parameter calculation[13].

- **RNN**[16]: RNN is a classical neural network for series processing and has been adopted for channel prediction tasks. In experiments, the number of RNN layers is set as 4.

- **LSTM**[17]: LSTM is designed with memory cells and multiplicative gates to deal with long-term dependency. An LSTM-based predictor with 4 LSTM layers is applied.

- **GRU**[42]: Gate recurrent unit (GRU)[42] is a variant of LSTM to address the vanishing gradient problem. Similarly, the number of GRU layers is set as 4 for channel prediction.

- **CNN**[19]: In Ref. [19], a CNN-based predictor is proposed for FDD systems, treating the prediction process of time-frequency CSI data as a 2D image processing task. This CNN model consists of 10 convolutional layers, with a convolution kernel size of 3×3.

- **Transformer**[18]: In Ref. [18], a transformer-based parallel channel predictor is proposed for TDD systems to avoid error propagation problems. Additionally, it is used as a basis for comparison.

- **No prediction**: For the case of no prediction, we directly take the value of the future downlink L CSI as the value of the latest uplink CSI to illustrate the severity of channel aging.

To ensure fairness, the above deep learning-based methods process antenna dimensions in parallel and adopt NMSE as the loss function for training. Since deep learning-based approaches do not rely on prior assumptions, all schemes except for PAD are applied to both TDD and FDD modes.

3) Network and Training Parameters: In the simulation, we adopt $N_1 = N_2 = 4$ CSI attention module for frequency and delay domain data processing. For patching, the size of patches is set as $N = 4$, and historical CSI is grouped into $P' = P/N = 4$ non-overlapping patches. For the SE block, the value of the reduction ratio is set as 2 and the convolution kernel size is set as 3×3 for all convolution layers. The smallest version[38] of GPT-2 with $F = 768$ feature dimension is adopted, the first $N_L = 6$ layers of which are deployed. Several hyper-parameters for model training are illustrated in Tab. 1.

4) Performance Metric: In order to comprehensively evaluate the performance of the proposed scheme, we adopt three performance metrics, namely NMSE, SE, and bit error rate (BER).

- **NMSE**: As shown in Eq. (7a), NMSE is a widely used[23] performance metric to directly characterize the accuracy of channel prediction. Therefore, NMSE is adopted as a major metric in our experiments.

- **SE**: SE is an important metric that reveals the achievable rate of the system, reflecting the effectiveness of communication. It is calculated by Eq. (4), where h_k is the actual CSI,

Tab. 1 Hyper-parameters for network training

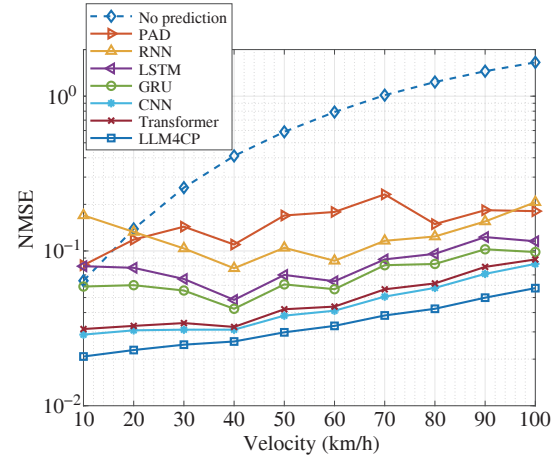| Parameter	Value
Batch size	512
Epochs	500
Optimizer	Adam (betas = $(0.9, 0.999)$)
Starting learning rate	0.001
Learning rate decay rate	0.1 per 150 epochs

Fig. 7 The NMSE performance of LLM4CP and other baselines versus different user velocities for TDD systems

and w_k is obtained as Eq. (5) with predicted h_k. The communication signal-to-noise ratio (SNR) is defined as $1/\sigma_n^2$ and set as 10 dB.

- **BER**: BER describes the communication reliability at a certain transmission rate. During the simulation, 4-QAM (quadrature amplitude modulation) is adopted and the communication SNR is set as 10 dB.

B. Performance Evaluation

1) Performance Under Varying Velocities: For TDD systems, we compare the NMSE performance of the proposed LLM4CP with baselines among different user velocities, as shown in Fig. 7. It can be observed that as the user velocity increases, the NMSE of all baselines gradually increases. This is due to that as the user velocity increases, the CSI changes rapidly with a shorter channel coherence time, resulting in increased prediction difficulty. Directly adopting the latest CSI as the predicted CSI will bring huge prediction errors, especially for highly dynamic scenes. The proposed method consistently outperforms other baselines among testing velocities and demonstrates its high prediction accuracy.
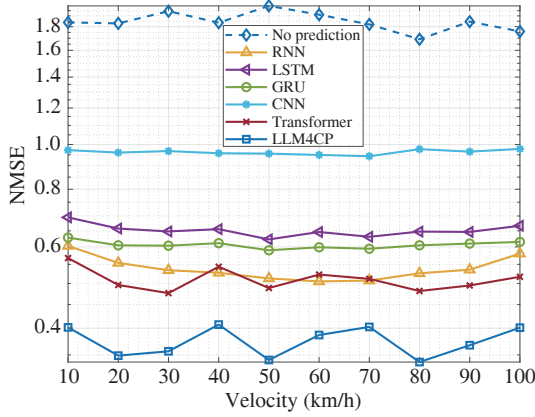
For FDD systems, the NMSE performance among different user velocities is given in Fig. 8. Due to the frequency gap between the uplink and downlink channels, the no-prediction scheme is unacceptable for the FDD mode. LLM4CP shows obvious advantages under all user velocities. It is worth noting that compared with TDD systems, the superiority of LLM4CP

Tab. 2 The SE and BER performance of LLM4CP and other baselines for TDD systems (Maximum achievable SE: 7.33 bit $\cdot$ (s $\cdot$ Hz)$^{-1}$)

Metric	No prediction	PAD	RNN	LSTM	GRU	CNN	Transformer	LLM4CP
SE (bit $\cdot$ (s $\cdot$ Hz)$^{-1}$)	6.238	7.007	6.692	6.816	6.772	6.992	6.963	**7.036**
BER	0.125 9	0.005 0	0.020 8	0.014 8	0.018 9	0.006 5	0.008 1	**0.003 9**

Tab. 3 The SE and BER performance of LLM4CP and other baselines for FDD systems (Maximum achievable SE: 7.33 bit $\cdot$ (s $\cdot$ Hz)$^{-1}$)
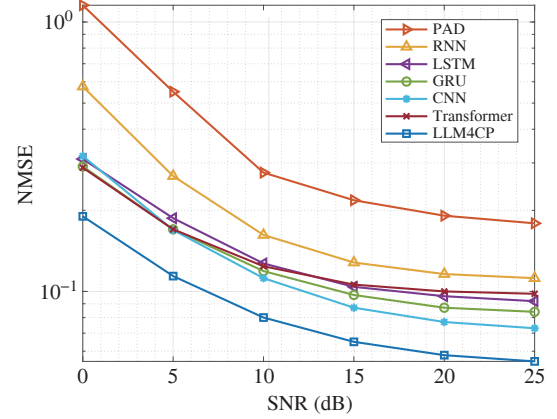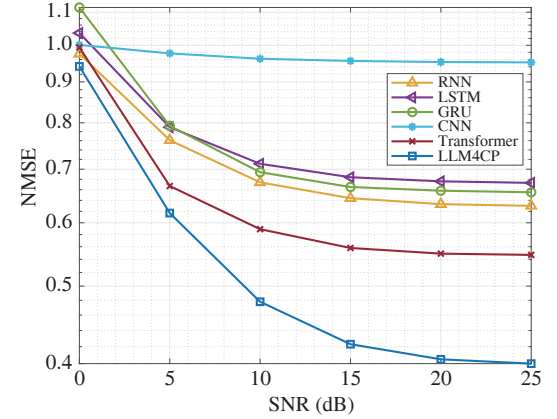
Metric	No prediction	RNN	LSTM	GRU	CNN	Transformer	LLM4CP
SE (bit $\cdot$ (s $\cdot$ Hz)$^{-1}$)	4.992	5.336	5.306	5.292	4.119	5.754	**6.303**
BER	0.318 9	0.131 3	0.130 3	0.131 5	0.283 2	0.080 7	**0.034 7**

Fig. 8 The NMSE performance of LLM4CP and other baselines versus different user velocities for FDD systems

Fig. 9 The NMSE performance versus SNR of nosiy historical CSI for TDD systems

Fig. 10 The NMSE performance versus SNR of nosiy historical CSI for FDD systems

is greater in FDD systems, thanks to its powerful ability to model complex time-frequency relationships.

In addition, to evaluate the communication effectiveness and reliability of these methods, both the SE and BER performance are presented for TDD and FDD systems, as shown in Tabs. 2 and 3, respectively. The maximum achievable SE is obtained with perfect CSI. Both the SE and BER have been averaged over all test velocities. Compared with TDD mode, the performance of all baselines in FDD mode shows an overall decrease due to the greater difficulty of prediction. Nevertheless, the proposed method achieves SOTA SE and BER performance for both the TDD and FDD modes.

2) Robustness Against Noise: Due to the inaccuracy of channel estimation, the predictor's robustness against noise is crucial, that is, predicting future CSI based on noisy historical CSI. In Figs. 9 and 10, the NMSE of these methods with noisy historical CSI is provided for both the TDD and FDD systems. Specifically, in the testing phase, the historical CSI data is added by Gaussian white noise with variance σ_n^2 and SNR is defined as $1/\sigma_n^2$. To enhance the robustness, during the training phase, noise with SNR uniformly distributed between 0 and 25 dB is added to all baselines. It can be observed that for all schemes, lower SNR results in higher prediction NMSE. The proposed method exhibits the lowest

NMSE at most SNRs, indicating its high robustness performance against CSI noise.

3) Few-Shot Prediction: To reduce the cost of CSI data collection and network training, few-shot prediction is crucial for the rapid deployment of deep learning-based models. In this part, we evaluate the few-shot learning ability of the proposed methods, where only 10% of the dataset is utilized for network training.

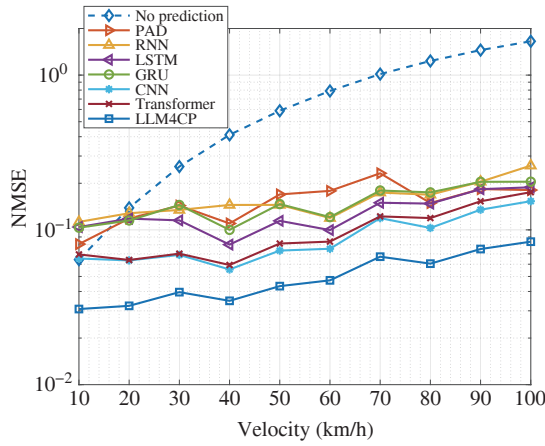In Fig. 11, the few-shot performance of the proposed

Fig. 11 The NMSE performance of LLM4CP and other baselines versus different user velocities for TDD systems (Few-shot: 10% training dataset)
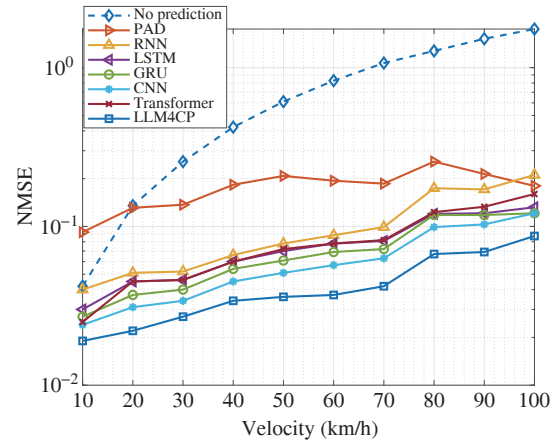

Fig. 13 The zero-shot generalization performance for the TDD systems in UMi scenario
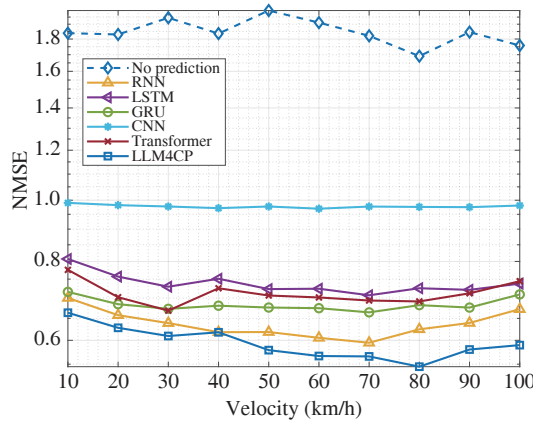

Fig. 12 The NMSE performance of LLM4CP and other baselines versus different user velocities for FDD systems (Few-shot: 10% training dataset)
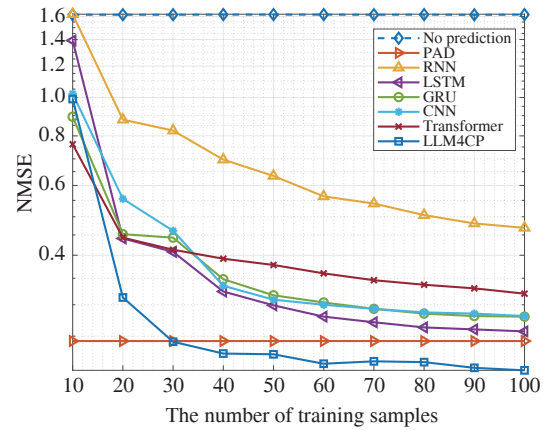

Fig. 14 The cross-frequency generalization performance versus different training samples for the TDD systems

LLM4CP and other baselines for TDD systems is illustrated. The powerful few-shot learning capability of LLM enables the proposed method to perform well even with limited training samples. When compared to full-sample training, as demonstrated in Fig. 7, the advantages of LLM4CP over other baselines are more evident in few-shot prediction scenarios. In Fig. 12, the few-shot performance for FDD systems is also given. Even though FDD channel prediction is more difficult, the proposed method maintains its advantages at most velocities.

4) Generalization Experiments: Generalization ability is crucial for model deployment in real-world scenarios, where well-trained models are applied to new scenarios with few-shot or even zero-shot training. In Fig. 13, considering TDD systems, we directly apply the model trained in the UMa scenario to the 3GPP urban micro (UMi) scenario[33] without additional training process, while keeping the other settings unchanged. The proposed model surpasses other baselines in terms of NMSE metric, demonstrating its strong generalization capability across different channel distributions. Further-

more, we consider the model's cross-frequency generalization capability, which is important for practical multi-band communication systems. Specifically, we apply the model trained at 2.4 GHz TDD systems to 4.9 GHz TDD systems with few-shot and zero-shot training. In Fig. 14, all deep learning-based baselines show poor zero-shot learning capabilities due to the channel's striking frequency selectivity. Nevertheless, as the number of training samples increases, the prediction NMSE of the proposed scheme drops significantly and surpasses other schemes. The proposed LLM4CP method achieves great NMSE performance at the new frequency with only a few training samples, demonstrating its strong frequency generalization ability.

5) Ablation Experiments: To validate the effectiveness of several specific modules, the ablation experiment is conducted by removing relevant modules, as shown in Tab. 4. The NMSE, SE, and BER performance for TDD systems are given and averaged over all testing velocities. For LLM4CP without delay domain processing, the subsequent network of the delay domain transformation is removed. For LLM4CP without

Tab. 4 Results of ablation experiments

Metric	LLM4CP	w/o delay domain	w/o CSI attention	w/o patching	w/o LLM
NMSE	**0.043**	0.045	0.053	**0.043**	0.049
SE $(\text{bit} \cdot (\text{s} \cdot \text{Hz})^{-1})$	**7.073**	7.051	7.036	7.066	7.036
BER	**0.003 1**	0.003 6	0.004 9	0.003 6	0.004 8

Tab. 5 Network parameters (training parameters/total parameters) and the training/interference cost per batch

Metric	PAD	RNN	LSTM	GRU	CNN	Transformer	LLM4CP
Network parameters (1×10^6)	0/0	0.30/0.30	1.13/1.13	0.86/0.86	3.14/3.14	1.76/1.76	1.73/82.87
Training time (ms)	0	4.59	7.40	5.88	2.03	21.84	6.84
Interference time (ms)	31.96	3.55	5.19	4.02	0.52	18.35	5.84

Tab. 6 The NMSE performance, network parameters, and interference time of LLM4CP with different numbers of GPT-2 layers (LLM4CP (n) represents the proposed method with n GPT-2 layers)

LLM4CP	NMSE	Network parameters (1×10^6)	Interference time (ms)
LLM4CP (2)	0.0626	1.72/54.49	3.67
LLM4CP (4)	0.0568	1.73/68.66	5.02
LLM4CP (6)	0.0509	1.73/82.87	5.81
LLM4CP (8)	0.0610	1.74/97.01	6.77

CSI attention modules, the patched tensors of both the frequency and the delay domain are directly added for the following processing. For LLM4CP without patching, the output of the preprocessor is directly input into CSI attention modules. For LLM4CP without LLM, the frozen LLM is removed while other parts remain unchanged. Observably, the removal of any of these four modules results in a loss of performance, indicating the necessity of these modules for high predictive accuracy.

6) Training and Inference Cost: We compare the model training and inference cost of the proposed method with other baselines to assess the difficulty of deploying the model in practical scenarios, as shown in Tab. 5. All experiments are conducted on the same machine with 4 Intel Xeon Platinum 8358P CPUs, 4 NVIDIA GeForce RTX4090 GPUs, and 188 GB of RAM. Since PAD is a model-based method, the number of its model parameters is relatively small to neglect and the training process is not required. However, it takes the longest inference time due to the high processing complexity. Although LLM4CP has large total parameters, the actual trainable parameters are similar to those of other deep learning-based models since most parameters in the LLM are frozen. It is worth noting that LLM's inference time is much shorter than that of the Transformer thanks to inference acceleration specific to the GPT model. Therefore, the proposed LLM4CP has the potential to serve real-time channel prediction.

In addition, we comprehensively evaluated the impact of selecting different numbers of GPT-2 layers on channel prediction performance, parameters cost, and inference time, as shown in Tab. 6. The relevant experimental results are obtained under the TDD few-shot prediction settings with 10% training dataset, where NMSE is the average across different velocities. The network parameters and inference time are both increased with the number of GPT-2 layers. It is worth noting that the proposed model with 6 GPT-2 layers performs the best within the testing range, indicating that having more layers does not necessarily favor prediction. In practical deployment, the selection of the type and size of the LLM backbone needs to consider both the requirements for prediction accuracy and the constraints of device storage and computational resources.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed an LLM-empowered channel prediction method, which fine-tunes pre-trained GPT-2 for MISO-OFDM channel prediction tasks. It predicts the future downlink CSI sequence based on the historical uplink CSI sequence and can be applied for both the TDD and FDD systems. To account for channel characteristics, we have tailored preprocessor, embedding, and output modules to bridge the gap between CSI data and LLM, with the aim of fully leveraging the transfer knowledge across models from the pre-trained LLM. Preliminary simulations validate its superiority over existing model-based and deep learning-based channel prediction methods in full-sample, few-shot, and generalization tests with acceptable training and inference costs.

In the future, we plan to explore more comprehensive experimental setups and validate the proposed method using a more realistic and challenging CSI dataset. Additionally, we will incorporate link-level simulation with channel coding and evaluate the frame error rate (FER) of the system.

Conflict of interest statement. None declared.

References

[1] CHENG X, ZHANG H, ZHANG J, et al. Intelligent multi-modal sensing-communication integration: synesthesia of machines[J]. IEEE Communications Surveys and Tutorials, 2024, 26(1): 258-301.

[2] ZHANG H, GAO S, CHENG X, et al. Integrated sensing and communications towards proactive beamforming in mmWave V2I via multi-modal feature fusion (MMFF)[J]. IEEE Transactions on Wireless Communications, early access, 2024: doi.org/10.1109/TWC.2024.3401686.

[3] CHUNG S T, GOLDSMITH A J. Degrees of freedom in adaptive modulation: a unified view[J]. IEEE Transactions on Communications, 2001, 49(9): 1561-1571.

[4] SADR S, ANPALAGAN A, RAAHEMIFAR K. Radio resource allocation algorithms for the downlink of multiuser OFDM communication systems[J]. IEEE Communications Surveys and Tutorials, 2009, 11(3): 92-106.

[5] GAO S, CHENG X, YANG L. Estimating doubly-selective channels for hybrid mmWave massive MIMO systems: a doubly-sparse approach[J]. IEEE Transactions on Wireless Communications, 2020, 19(9): 5703-5715.

[6] MA X, GIANNAKIS G B, OHNO S. Optimal training for block transmissions over doubly selective wireless fading channels[J]. IEEE Transactions on Signal Processing, 2003, 51(5): 1351-1366.

[7] ROTTENBERG F, CHOI T, LUO P, et al. Performance analysis of channel extrapolation in FDD massive MIMO systems[J]. IEEE Transactions on Wireless Communications, 2020, 19(4): 2728-2741.

[8] CHOI T, ROTTENBERG F, GOMEZ-PONCE J, et al. Experimental investigation of frequency domain channel extrapolation in massive MIMO systems for zero-feedback FDD[J]. IEEE Transactions on Wireless Communications, 2020, 20(1): 710-725.

[9] ARNOLD M, DÖRNER S, CAMMERER S, et al. Towards practical FDD massive MIMO: CSI extrapolation driven by deep learning and actual channel measurements[C]//Proceedings of 2019 53rd Asilomar Conference on Signals, Systems, and Computers. Piscataway: IEEE Press, 2019: 1972-1976.

[10] TRUONG K T, HEATH R W. Effects of channel aging in massive MIMO systems[J]. Journal of Communications and Networks, 2013, 15(4): 338-351.

[11] WONG I C, EVANS B L. Joint channel estimation and prediction for OFDM systems[C]//Proceedings of IEEE Global Telecommunications Conference. Piscataway: IEEE Press, 2005: 2255-2259.

[12] SHEN Z, ANDREWS J G, EVANS B L. Short range wireless channel prediction using local information[C]//Proceedings of the Thrity-Seventh Asilomar Conference on Signals, Systems and Computers. Piscataway: IEEE Press, 2003, 1: 1147-1151.

[13] YIN H, WANG H, LIU Y, et al. Addressing the curse of mobility in massive MIMO with prony-based angular-delay domain channel predictions[J]. IEEE Journal on Selected Areas in Communications, 2020, 38(12): 2903-2917.

[14] QIN Z, YIN H, CAO Y, et al. A partial reciprocity-based channel prediction framework for FDD massive MIMO with high mobility[J]. IEEE Transactions on Wireless Communications, 2022, 21(11): 9638-9652.

[15] KIM H, KIM S, LEE H, et al. Massive MIMO channel prediction: Kalman filtering vs. machine learning[J]. IEEE Transactions on Communications, 2020, 69(1): 518-528.

[16] JIANG W, SCHOTTEN H D. Neural network-based fading channel prediction: a comprehensive overview[J]. IEEE Access, 2019, 7: 118112-118124.

[17] JIANG W, SCHOTTEN H D. Deep learning for fading channel prediction[J]. IEEE Open Journal of the Communications Society, 2020, 1: 320-332.

[18] JIANG H, CUI M, NG D W K, et al. Accurate channel prediction based on transformer: making mobility negligible[J]. IEEE Journal on Selected Areas in Communications, 2022, 40(9): 2717-2732.

[19] SAFARI M S, POURAHMADI V, SODAGARI S. Deep UL2DL: data-driven channel knowledge transfer from uplink to downlink[J]. IEEE Open Journal of Vehicular Technology, 2019, 1: 29-44.

[20] ZHANG Z, ZHANG Y, ZHANG J, et al. Adversarial training-aided time-varying channel prediction for TDD/FDD systems[J]. China Communications, 2023, 20(6): 100-115.

[21] FAN S, XU W, XIE R, et al. Deep CSI compression for dual-polarized massive MIMO channels with disentangled representation learning[J]. IEEE Transactions on Communications, 2024: doi.org/10.1109/TCOMM.2024.3384256.

[22] GAO S, CHENG X, FANG L, et al. Model enhanced learning based detectors (Me-LeaD) for wideband multi-user 1-bit mmWave communications[J]. IEEE Transactions on Wireless Communications, 2021, 20(7): 4646-4656.

[23] BURGHAL D, LI Y, MADADI P, et al. Enhanced AI based CSI prediction solutions for massive MIMO in 5G and 6G systems[J]. IEEE Access, 2023, 11: 117810-117825.

[24] LIU G, HU Z, WANG L, et al. Spatio-temporal neural network for channel prediction in massive MIMO-OFDM systems[J]. IEEE Transactions on Communications, 2022, 70(12): 8003-8016.

[25] KIM H, CHOI J, LOVE D J. Massive MIMO channel prediction via meta-learning and deep denoising: is a small dataset enough?[J]. IEEE Transactions on Wireless Communications, 2023, 22(12): 9278-9290.

[26] LIU G, HU Z, WANG L, et al. A hypernetwork based framework for non-stationary channel prediction[J]. IEEE Transactions on Vehicular Technology, 2024: doi.org/10.1109/TVT.2024.3355557.

[27] SU J, JIANG C, JIN X, et al. Large language models for forecasting and anomaly detection: a systematic literature review[J]. arXiv preprint arXiv:2402.10350, 2024.

[28] ZHOU T, NIU P, SUN L, et al. One fits all: power general time series analysis by pretrained lm[J]. Advances in Neural Information Processing Systems, 2023, 36: 43322-43355.

[29] JIN M, WANG S, MA L, et al. Time-LLM: time series forecasting by reprogramming large language models[J]. arXiv preprint arXiv:2310.01728, 2023.

[30] REN Y, CHEN Y, LIU S, et al. TPLLM: a traffic prediction framework based on pretrained large language models[J]. arXiv preprint arXiv:2403.02221, 2024.

[31] LIANG Y, WEN H, NIE Y, et al. Foundation models for time series analysis: a tutorial and survey[J]. arXiv preprint arXiv:2403.14735, 2024.

[32] ZHOU H, HU C, YUAN Y, et al. Large language model (LLM) for telecommunications: a comprehensive survey on principles, key techniques, and opportunities[J]. arXiv preprint arXiv:2405.10825, 2024.

[33] 3GPP Radio Access Network Working Group. Study on channel model for frequencies from 0.5 to 100 GHz (Release 15)[R]. 3GPP TR 38.901, 2018.

[34] MARZETTA T L, LARSSON E G, YANG H. Fundamentals of massive MIMO[M]. Cambridge: Cambridge University Press, 2016.

[35] NIE Y, NGUYEN N H, SINTHONG P, et al. A time series is worth 64 words: long-term forecasting with transformers[J]. arXiv preprint arXiv:2211.14730, 2022.

[36] HU J, SHEN L, SUN G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2018: 7132-7141.

[37] NAIR V, HINTON G E. Rectified linear units improve restricted boltzmann machines[C]//Proceedings of the 27th International Conference on Machine Learning (ICML-10). Piscataway: IEEE Press, 2010: 807-

814.

[38] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems. [S.l.]: NeurIPS, 2017: 5998-6008.

[39] RADFORD A, WU J, CHILD R, et al. Language models are unsupervised multitask learners[J]. OpenAI Blog, 2019, 1(8): 9.

[40] TOUVRON H, MARTIN L, STONE K, et al. Llama 2: open foundation and fine-tuned chat models[J]. arXiv preprint arXiv:2307.09288, 2023.

[41] JAECKEL S, RASCHKOWSKI L, BÖRNER K, et al. QuaDRiGa: a 3-D multi-cell channel model with time evolution for enabling virtual field trials[J]. IEEE Transactions on Antennas and Propagation, 2014, 62(6): 3242-3256.

[42] CHO K, VAN MERRIËNBOER B, GULCEHRE C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[J]. arXiv preprint arXiv:1406.1078, 2014.

About the Authors

Boxun Liu received the B.E. degree from University of Electronic Science and Technology of China, Chengdu, China, in 2023. He is currently pursuing the Ph.D. degree with the School of Electronics, Peking University, Beijing, China. His current research interests include integrated sensing and communication (ISAC), transceiver design, channel prediction, and machine learning.

Xuanyu Liu is currently pursuing the Ph.D. degree with the School of Electronics, Peking University, Beijing, China. His current research interests include wireless communications, channel prediction, and large language model(LLM).

Shijian Gao received the Ph.D. degree from University of Minnesota, Minneapolis, USA, in 2022. After graduation, he served as a Senior RF Engineer at Samsung SoC Lab, San Diego, USA. In February 2024, he joined the Internet of Things Thrust at The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China, as an Assistant Professor. His research lies in the broad areas of statistical signal processing, wireless communication, RF systems, and machine learning. He was the co-recipient of 2021 MICCAI Young Scientist Paper Award and serves as an Associate Editor for IET Communications.

Xiang Cheng [corresponding author] received the Ph.D. degree jointly from Heriot-Watt University and The University of Edinburgh, Edinburgh, UK, in 2009. He is currently a Boya Distinguished Professor with Peking University. His general research interests are in areas of channel modeling, wireless communications, and data analytics, subject on which he has published more than 280 journal and conference papers, nine books, and holds 26 patents. He is a recipient of the IEEE Asia-Pacific Outstanding Young Researcher Award in 2015 and Xplorer Prize 2023. He was a co-recipient of the 2016 IEEE JSAC Best Paper Award: Leonard G. Abraham Prize, and the 2021 IET Communications Best Paper Award: Premium Award. He has also received the Best Paper Awards at IEEE ITST'12, ICCC'13, ITSC'14, ICC'16, ICNC'17, GLOBECOM'18, ICCS'18, and ICC'19. He has been a Highly Cited Chinese Researcher since 2020. In 2021 and 2023, he was selected into two world scientist lists, including World's Top 2% Scientists released by Stanford University and top computer science scientists released by Guide2Research. He has served as the Symposium Lead Chair, the Co-Chair, and a member of the Technical Program Committee for several international conferences. He led the establishment of three Chinese standards (including one industry standard and two group standards) and participated in the formulation of ten 3GPP international standards and two Chinese industry standards. He is currently a Subject Editor of IET Communications and an Associate Editor of the IEEE Transactions on Wireless Communications, IEEE Transactions on Intelligent Transportation Systems, IEEE Wireless Communications Letters, and the Journal of Communications and Information Networks, and is an IEEE Distinguished Lecturer. He was a Distinguished Lecturer of IEEE Vehicular Technology Society.

Liuqing Yang received the Ph.D. degree in Electrical and Computer Engineering from University of Minnesota, Minneapolis, USA, in 2004. She is currently a Professor with The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China. Before joining The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China, she has been a Faculty Member with the Department of Electrical and Computer Engineering, University of Florida (2004-2010), Colorado State University (2010-2020), and University of Minnesota (2020-2021), USA. Her research interests include communications and networking subjects on which she has published more than 370 journals and conference papers, four book chapters, and five books. She was a recipient of the ONR Young Investigator Program (YIP) Award in 2007, the NSF Faculty Early Career Development (CAREER) Award in 2009, and the Best Paper Award at IEEE ICUWB 2006, ICCC 2013, ITSC 2014, GLOBECOM 2014, ICC 2016, WCSP 2016, GLOBECOM 2018, ICCS 2018, and ICC 2019. She is the Editor-in-Chief of IET Communications, an Executive Editorial Committee (EEC) Member of the IEEE Transactions on Wireless Communications, and a Senior Editor of the IEEE Transactions on Signal Processing. She has also served as an Editor for the IEEE Transactions on Communications, the IEEE Transactions on Intelligent Transportation Systems, IEEE Intelligent Systems, and PHYCOM: Physical Communication, and as a Program Chair, a Track/Symposium, or a TPC Chair for many conferences.