Spatio-Temporal Neural Network for Channel Prediction in Massive MIMO-OFDM Systems

Guanzhang Liu, Zhengyang Hu, Lei Wang, Jiang Xue, *Senior Member, IEEE*, Haifan Yin, *Member, IEEE*, and David Gesbert, *Fellow, IEEE*

***Abstract*—In massive multiple-input multiple-output orthogonal frequency division multiplexing (MIMO-OFDM) systems, a challenging problem is how to predict channel state information (CSI) (i.e., channel prediction) accurately in mobility scenarios. However, a practical obstacle is caused by CSI non-stationary and nonlinear dynamics in temporal domain. In this paper, we propose a spatio-temporal neural network (STNN) to achieve better performance by carefully taking into account the spatio-temporal characteristics of CSI. Specifically, STNN uses its encoder and decoder modules to capture the spatial correlation and temporal dependence of CSI. Further, the differencing-attention module is designed to deal with the non-stationary and nonlinear temporal dynamics and realize adaptive feature refinement for more accurate multi-step prediction. Additionally, an advanced training scheme is adopted to reduce the discrepancy between STNN training and testing. Evaluated on a realistic channel model with enhanced mobility and spherical waves, experimental results show that STNN can effectively improve the accuracy of prediction and perform well with respect to different signal to noise ratios (SNRs). Visualization and testing for unit root illustrate STNN is able to learn CSI time-varying patterns by alleviating series non-stationarity.**

***Index Terms*—Massive MIMO, channel prediction, spatial correlation, non-stationarity and non-linearity, deep learning.**

I. Introduction

MASSIVE multiple-input multiple-output (MIMO) has been widely regarded as a fundamental technology for the fifth generation (5G) and future wireless communications [1]. By equipping with a large number of antennas at the base station (BS), massive MIMO can achieve superior spectral efficiency [2] and energy efficiency [3]. In addition, orthogonal frequency division multiplexing (OFDM) is incorporated to cope with the effect of the inter-symbol interference (ISI) in the multipath fading channel [4]. In order to realize the potential of massive MIMO-OFDM systems, many emerging techniques have been introduced, such as transmit antenna selection [5], precoding [6] and beamforming [7]. It is worth mentioning that the promising benefits of all these techniques are based on accurate CSI acquisition at the BS. Nevertheless, when the user equipment (UE) is moving, CSI obtained at the BS would be outdated, leading to significant performance degradation. Considering higher mobility scenarios in the future, this problem will be more serious. To overcome the aforementioned problem, accurate and timely channel prediction is the key.

The conventional channel prediction methods can be mainly divided into two categories, the parametric radio channel (PRC) model based methods [8], [9] and the autoregressive (AR) model based methods [10], [11]. However, both of them are based on the theoretical channel models and strict assumptions, which are quasi-static and wide-sense stationary (WSS), respectively. In [11], inspired by employing the difference of the past CSI between adjacent times instead of the past CSI, two AR based channel prediction methods were proposed to suppress interference in MIMO systems. In [12], based on the first-order Taylor expansion (FIT), a FIT prediction method was designed to track CSI in time-varying massive MIMO systems. Both [11] and [12] assumed different transceiver antennas are mutually uncorrelated. Similarly, every sub-carrier in massive MIMO/MIMO-OFDM systems was treated as a SISO flat-fading channel in [13] and [14]. In the above methods, the temporal correlation was utilized to implement channel prediction, however, the array and frequency correlations were ignored.

In practice, CSI correlations exist not only in temporal domain, but also in array and frequency domains [15], [16], promoting the further development of channel prediction methods. In [17], two AR based channel predictors with data selection were derived to employ correlations existed in both array and temporal domains with a competitive complexity, which could effectively improve prediction accuracy. Instead of directly extracting correlations in array-frequency domain, CSI is transformed to angular-delay domain through inverse discrete Fourier transform (IDFT) and the corresponding channel prediction methods were investigated [18]–[20]. In [19], a Prony based angular-delay domain (PAD) predictor was presented to address the curse of mobility. In [20], a spatio-temporal autoregressive (ST-AR) predictor was designed to

The work of Jiang Xue was supported in part by the National Key R&D Program of China under Grant 2020YFA0713900, in part by the Fundamental Mathematical Theory of Network Communication, in part by the Major Key Project of PCL under Grant PCL2021A12, and in part by the Joint Project of Industries and Universities Shaanxi under Grant S2021-YF-GXZD 0076. The work of Haifan Yin was supported in part by the National Natural Science Foundation of China under Grant 62071191. The work of David Gesbert was supported by the 3IA Côte d'Azur Investments managed by the French National Research Agency (ANR) ANR-19-P3IA-0002.

Guanzhang Liu, Zhengyang Hu, and Lei Wang are with School of Mathematics and Statistics, Xi'an Jiaotong University, 710049 Xi'an, China (e-mail: lgzh97@stu.xjtu.edu.cn, hzyxjtu@stu.xjtu.edu.cn, wl_simple@stu.xjtu.edu.cn).

Jiang Xue is with the Pazhou Laboratory (Huangpu), Pengcheng Laboratory and School of Mathematics and Statistics, Xi'an Jiaotong University, 710049 Xi'an, China (e-mail: x.jiang@xjtu.edu.cn).

Haifan Yin is with School of Electronic information and Communications, Huazhong University of Science and Technology, 430074 Wuhan, China (e-mail: yin@hust.edu.cn).

David Gesbert is with the Department of Communication Systems, EURECOM, 06904 Sophia Antipolis, France (e-mail: gesbert@eurecom.fr).

The corresponding author is Jiang Xue.

employ residual correlations in angular-delay domain. The above predictors all exploited the characteristics of CSI structure in array/angular-frequency/delay domain. Nevertheless, in temporal domain, they still assumed that CSI time-varying patterns followed the linear and stationary models, which would be deviated from realistic wireless channels. Based on the above reason, the temporal dynamics of CSI also need to be carefully considered.

Due to the successful applications in computer vision (CV) and natural language processing (NLP), deep learning (DL) has recently been introduced into wireless communication [21]–[23]. Without any prior assumptions about the channel models, DL based methods can learn time-varying patterns of CSI in data-driven manners. Considering channel prediction as a time series problem, many powerful prediction methods have emerged [24]–[31]. In [24], motivated by the gap between the theoretical models and realistic wireless channels, a multi-layer perceptron (MLP) based method was proposed. In [25] and [26], recurrent neural network (RNN) and long short term memory (LSTM) based methods were investigated to implement channel prediction, respectively. In [29] and [30], inspired by the sequence-to-sequence (seq2seq) model in NLP, two LSTM based neural networks (NN) were designed to predict CSI. To better fit the characteristics of CSI, channel prediction has been extended to spatio-temporal series problem [32], [33]. In [32], by considering CSI as a two-dimension (2-D) image, convolutional neural network (CNN) and RNN were cascaded to extract the array and temporal correlations. In [33], an online prediction scheme based on CNN and RNN was proposed with an offline-online training mechanism. However, [32] and [33] simply stacked CNN and RNN, while few specific structures were designed to pertinently address difficulties in channel prediction. Moreover, RNN/LSTM adopted in the above methods does not take spatial correlation into consideration and may suffer from unaffordable complexity in massive MIMO-OFDM systems.

To fill the aforementioned research gaps, in this paper, we propose a novel spatio-temporal channel prediction method based on DL by exploiting the spatio-temporal characteristics of CSI in massive MIMO-OFDM systems, which will not only improve the prediction accuracy but also possess partial interpretability. More specifically, our predictor is designed based on the fact that CSI owns spatial correlation in array-frequency domain and reflects the non-stationary and nonlinear characteristics in temporal domain. Our ideas consist in jointly processing the spatio-temporal information, explicitly modeling the non-stationary and nonlinear dynamics and adaptively realizing feature refinement, in which "explicitly" means a deterministic differencing sub-module is designed and "adaptively" means that we use attention mechanism to decide "what","where" and "when" to focus for the proposed NN.

The major contributions of this paper are summarized as follows:

- After investigating the spatio-temporal characteristics of CSI from the aspects of correlation and time variation, we formulate the massive MIMO-OFDM channel prediction problem by taking into account the spatial and temporal information, and introduce a DL based approach to parallel exploit spatial and temporal information in high dimension space with tractable complexity.
- We propose a spatio-temporal neural network (STNN), which utilizes the differencing operation to improve the non-stationary modeling capability and three lightweight attention sub-modules for the enhancement of feature refinement power. Moreover, we improve the performance of STNN by combining it with an advanced training scheme, called scheduled sampling [34]. To the best of our knowledge, this is the first work to consider the differencing operation and attention mechanism in designing DL based channel prediction network structure.
- We evaluate STNN on a spherical waves and enhanced mobility based channel model with extensive experiments. Numerical results show that STNN can achieve excellent performance in terms of the prediction length, velocity of the UE and imperfect CSI. Moreover, we verify the robustness of STNN to noise, and the effect of the temporal dependence. Furthermore, we also design specific experiments to explore STNN itself, including the training scheme, hyper-parameter setting and model ablation.
- Through visualization of learned attention weights and the stationary analysis of series, we investigate the prediction mechanism of STNN in learning CSI time-varying patterns, which partially explains why STNN works and is the first to discuss the cause of the superior performance of DL based channel prediction methods.

The rest of this paper is organized as follows. In Section II, we introduce the massive MIMO-OFDM channel model [35] used in this paper. In Section III, we give the problem formulation of channel prediction and investigate the effectiveness of ConvLSTM for CSI spatio-temporal modeling. Section IV presents STNN and experimental results are shown in Section V. Finally, Section VI concludes this paper.

Notations: Throughout the paper, we use bold uppercase letter to denote matrix, bold lowercase letter to denote vector, and non-bold letter to denote scalar. $\|\cdot\|_2$ is the Euclidean norm. $(\cdot)^T$ represents the transpose of a vector or matrix. The complex number field is represented by $\mathbb{C}$ and the real number field is represented by $\mathbb{R}$. The symbol $\circ$ represents the Hadamard product, mod represents the module operation and $E\{\cdot\}$ represents the expectation operation.

II. Massive MIMO-OFDM Channel Model in Mobile Environment

In the general channel models, e.g., the 3GPP-3D model [36] and the 3GPP-NR model [37], the description of the Doppler shift is related to the arrival angles, while the modeling of arrival angles is time independent. In other words, the arrival angles are fixed as the UE moves, so are the delays. This kind of modeling approach is easy to implement, but there are non-negligible gaps compared to the real mobile environments. Furthermore, when the number of antennas is large, the plane waves assumption is not fulfilled [38]. To improve the effectiveness of channel prediction method in real deployments, we adopt a 3-D geometric based channel model

TABLE I: The definitions of the key parameters of the propagation environment.

Parameter	Definition
τ_l	the initial transmitting delay of the l-th path at time t_0
$\theta^a_{l,m}$	the initial elevation angle of arrival (AOA) of the m-th sub-path in the l-th path at time t_0
$\phi^a_{l,m}$	the initial azimuth AOA of the m-th sub-path in the l-th path at time t_0
$\theta^a_{l,m,t}$	the elevation AOA of the m-th sub-path in the l-th path at time t
$\phi^a_{l,m,t}$	the azimuth AOA of the m-th sub-path in the l-th path at time t
$\theta^d_{s,l,m}$	the elevation angle of departure (AOD) of the m-th sub-path in the l-th path for the s-th antenna element of the BS
$\phi^d_{s,l,m}$	the azimuth AOD of the m-th sub-path in the l-th path for the s-th antenna element of the BS
$\psi_{s,l,m,t}$	the phase between the s-th antenna element of the BS and the UE via the m-th sub-path in the l-th path at time t
$\tau_{s,l,t}$	the delay between the s-th antenna element of the BS and the UE via the l-th path at time t
$\mathbf{r}$	the distance vector pointing from the BS to the initial location of the UE at time t_0

[35], which supports enhanced mobility and spherical waves to provide a practical evaluation. The main idea is to calculate the location of the last-bounce scatterer[1] (LBS) based on the initial arrival angles.

Let us consider a single-cell massive MIMO-OFDM system where a mobile UE is communicating with a BS. The BS equips N_s antennas ($N_s \gg 1$), while the UE equips N_r antennas. Due to the BS has massive antennas, we thus mainly focus on exploiting the correlation among BS antennas, and the number of antennas of the UE, i.e., N_r, is assumed to 1. The number and spacing of sub-carriers are N_f and Δf. For clarity, the definitions of the key parameters of the propagation environment are listed in Table I. At first, the initial delay of the line of sight (LOS) path is assumed to be zero. Based on the initial path delay of the l-th path and the distance $\|\mathbf{r}\|_2$ between the BS and the initial location of the UE at time t_0, the initial length d_l of the l-th path follows

$$d_l = \|\mathbf{r}\|_2 + \tau_l \cdot c, \tag{1}$$

where c is the speed of light. Although different sub-paths are indistinguishable from the length (i.e., path delay), they have different arrival angles. Denote $\mathbf{p}_{l,m}$ as the arrival vector of the m-th sub-path in the l-th path pointing from the initial location of the UE at time t_0 to the LBS, its length can be obtained by

$$\|\mathbf{p}_{l,m}\|_2 = \frac{d_l^2 - \|\mathbf{r}\|_2^2}{2(d_l + \mathbf{r}^T \bar{\mathbf{p}}_{l,m})}, \tag{2}$$

where

$$\bar{\mathbf{p}}_{l,m} = \begin{bmatrix} \cos\phi^a_{l,m}\cos\theta^a_{l,m} \\ \sin\phi^a_{l,m}\cos\theta^a_{l,m} \\ \sin\theta^a_{l,m} \end{bmatrix}. \tag{3}$$

Once we get the arrival vector $\mathbf{p}_{l,m}$ and the initial location of the UE at time t_0, the corresponding location of the LBS becomes known. The geometric relationship between the BS, UE and LBS at time t_0 is shown in Fig. 1. Due to the movement of the UE, at time t, the arrival vector $\mathbf{p}_{l,m,t}$ and the corresponding elevation AOA $\theta^a_{l,m,t}$ and azimuth AOA $\phi^a_{l,m,t}$ can be updated by

$$\mathbf{p}_{l,m,t} = \mathbf{p}_{l,m} - \mathbf{e}_t, \tag{4}$$

[1]We illustrate here the single-bounce model in order to simplify the notations. In Section V, the proposed prediction method is evaluated on the multi-bounce model, which can be extended from the single-bounce model. More details about the multi-bounce model can be found in [35].

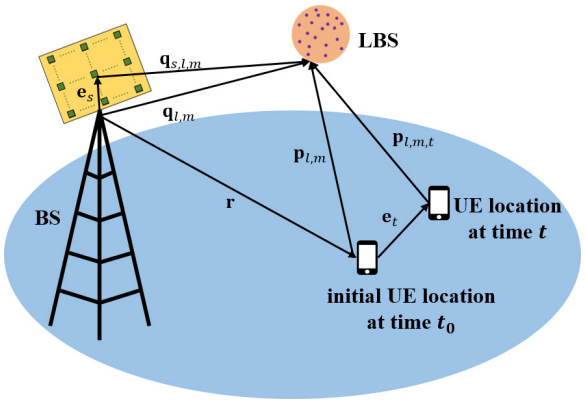


Fig. 1: The geometric relationship between the BS, UE, LBS and s-th antenna element of the BS at time t_0 and t.

$$\theta^a_{l,m,t} = \arcsin\frac{p_{l,m,t,z}}{\|\mathbf{p}_{l,m,t}\|_2}, \tag{5}$$

and

$$\phi^a_{l,m,t} = \arctan\frac{p_{l,m,t,y}}{p_{l,m,t,x}}, \tag{6}$$

where the vector $\mathbf{e}_t$ points from the initial location of the UE at time t_0 to the current location of the UE at time t, $p_{l,m,t,x}$, $p_{l,m,t,y}$ and $p_{l,m,t,z}$ are the Cartesian coordinate components of $\mathbf{p}_{l,m,t}$. To support spherical waves, the departure vector $\mathbf{q}_{s,l,m}$ for the s-th antenna element of the BS and the corresponding elevation AOD $\theta^d_{s,l,m}$ and azimuth AOD $\phi^d_{s,l,m}$ can be similarly obtained by

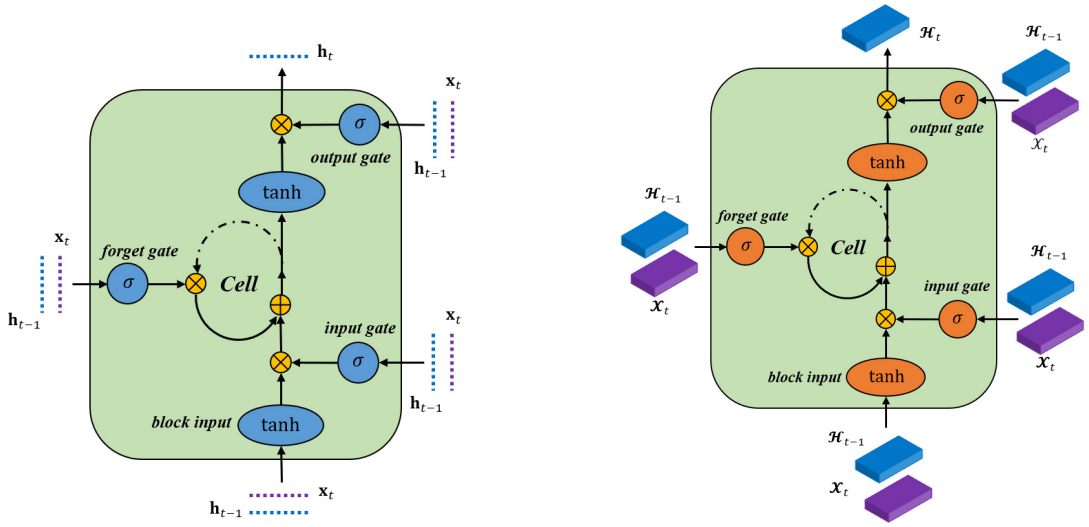
$$\mathbf{q}_{s,l,m} = \mathbf{q}_{l,m} - \mathbf{e}_s, \tag{7}$$

$$\theta^d_{s,l,m} = \arcsin\frac{q_{s,l,m,z}}{\|\mathbf{q}_{s,l,m}\|_2}, \tag{8}$$

and

$$\phi^d_{s,l,m} = \arctan\frac{q_{s,l,m,y}}{q_{s,l,m,x}}, \tag{9}$$

where the departure vector $\mathbf{q}_{l,m}$ of the m-th sub-path in the l-th path directs from the BS to the corresponding LBS, the direction of the vector $\mathbf{e}_s$ is from the BS to the s-th antenna element of the BS, $q_{s,l,m,x}$, $q_{s,l,m,y}$ and $q_{s,l,m,z}$



(a) LSTM (b) ConvLSTM

Fig. 2: Cell architecture of LSTM (a) and ConvLSTM (b). LSTM operates on 1-D vectors, while ConvLSTM oprates on 3-D tensors.

are the Cartesian coordinate components of $\mathbf{q}_{s,l,m}$. Now, the deterministic phase $\psi_{s,l,m,t}$ and delay $\tau_{s,l,t}$ can be derived as

$$\psi_{s,l,m,t} = \frac{2\pi}{\lambda_c} \cdot (d_{s,l,m,t} \bmod \lambda_c), \tag{10}$$

and

$$\tau_{s,l,t} = \frac{\sum_{m=1}^{M_l} d_{s,l,m,t}}{M_l \cdot c}, \tag{11}$$

where

$$d_{s,l,m,t} = \|\mathbf{q}_{s,l,m}\|_2 + \|\mathbf{p}_{l,m,t}\|_2, \tag{12}$$

M_l is the number of subpaths in the l-th path, λ_c represents the wavelength and mod represents the module operation.

Based on the above modeling of enhanced mobility and spherical waves, the channel between the s-th antenna element of the BS and the UE via the l-th path at time t can be described as [35]

$$g_{s,l,t} = \sum_{m=1}^{M_l} \beta_{s,l,m,t} e^{\left(-j\psi_{l,m}^{0} - j\psi_{s,l,m,t}\right)}, \tag{13}$$

where $j = \sqrt{-1}$ is the imaginary unit, $\psi_{l,m}^{0}$ is the random phase of the m-th sub-path in the l-th path, and $\beta_{s,l,m,t}$ is the attenuation coefficient between the s-th BS antenna and the UE via the m-th sub-path in the l-th path at time t. In equation (13), according to the modeling of the phase $\psi_{s,l,m,t}$, which is calculated by the BS location, the LBS location and the UE location at time t, mobility is explicitly integrated into the channel model. For the channel matrix $\bar{\mathbf{H}}_t \in \mathbb{C}^{N_s \times N_f}$ in array-frequency domain, its (s,k)-th element can be derived as

$$[\bar{\mathbf{H}}_t]_{s,k} = \sum_{l=1}^{L'} g_{s,l,t} e^{\left(-j2\pi \frac{k-1}{N_f} B \tau_{s,l,t}\right)}, \tag{14}$$

where B is the bandwidth, and L' is the number of path, $s = 1, \cdots, N_s$ and $k = 1, \cdots, N_f$. In practice, the channel matrix $\bar{\mathbf{H}}_t$ is obtained by reference signals.

III. Problem Formulation and Motivation

In this section, we provide the problem formulation of channel prediction in massive MIMO-OFDM systems and compare the ability of LSTM and ConvLSTM, two basic units of NN, in the spatio-temporal modeling.

A. Problem Formulation

Channel prediction is generally considered as a time series problem. As described in [19], the channel matrix has the specific array-frequency structure in massive MIMO-OFDM systems. Therefore, the prediction series contains both the spatial (i.e., array-frequency domain) and temporal characteristics, which is essentially a spatio-temporal series problem [39]. Thus, the specific formulation of channel prediction could be stated as

$$\hat{\bar{\mathbf{H}}}_{t+1}, \cdots, \hat{\bar{\mathbf{H}}}_{t+J} = f\left(\bar{\mathbf{H}}_{t-K+1}, \cdots, \bar{\mathbf{H}}_t\right), \tag{15}$$

where J denotes the length of the prediction series, K denotes the total length of the past series, $f(\cdot)$ denotes an arbitrary mapping function, and $\hat{\bar{\mathbf{H}}}_{t+i}$ denotes the $(t+i)$-th estimated CSI.

The main purpose of channel prediction is to determine the mapping function $f(\cdot)$. Over the last decades, a large number of linear predictors under the WSS assumption have been proposed. However, in real fast-varying environments, CSI may be non-stationary and nonlinear in temporal domain. Thus, the assumptions of the mapping function $f(\cdot)$ in equation (15) no longer require to satisfy the linear and stationary constraints, which is a major difference from the existing model. Besides, the mapping function $f(\cdot)$ needs to take the spatial correlation into consideration.

B. Why ConvLSTM?

By collecting huge amount of CSI, DL based methods are expert in discovering the inherent data characteristics.

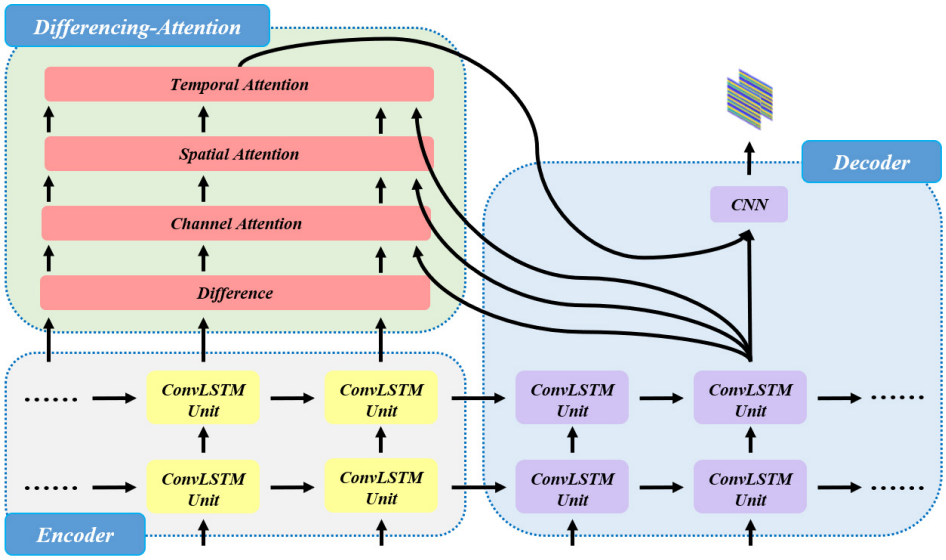


Fig. 3: The structure of STNN, including an encoder module, a differencing-attention module and a decoder module.

Thus, LSTM is widely applied to solve channel prediction problem [26], [28]–[30], [33]. The internal architecture of an LSTM cell unit is illustrated in Fig. 2a. For brevity, the principle of LSTM is not mentioned here, which can be found in [40]. Considering that the input, hidden states and cell states of LSTM are 1-D vectors, the channel matrix $\bar{\mathbf{H}}_t$ needs to be mapped into 1-D space. A straightforward way is to transform $\bar{\mathbf{H}}_t$ to its vectorized form. However, mapping from 2-D space to 1-D space could cause damage to CSI correlations and reduce the accuracy of channel prediction. One feasible solution is to jointly process $\bar{\mathbf{H}}_t$ in 2-D space to better exploit its spatial correlation.

Recently, ConvLSTM [39] is proposed to capture the spatio-temporal correlation and has been successfully applied to the high-speed railway and millimeter-wave communications [41], [42]. The major modifications of ConvLSTM are that all of the fully connected operators existed in LSTM are replaced by the convolution operators. As a result, the input, hidden states, cell states and gate signals are three-dimension (3-D) tensors, shown in Fig. 2b. From the perspective of physical propagation, antennas arranged in an array are pretty close to each other, which results in similar propagation paths of electromagnetic waves of different antennas [43]. Thus, the strong array correlation exists in the wireless channel. Meanwhile, different sub-carriers in nearby frequencies are also strongly correlated [16]. Therefore, the convolution operators are resorted to exploit the array and frequency correlations of the channel matrix $\bar{\mathbf{H}}_t$ with specific structural characteristics. As the convolution kernel slides over the input CSI and hidden states, the spatial information is encoded during the input-to-state and state-to-state transitions of ConvLSTM. The convolution operator for the feature extraction of CSI has also been verified in channel estimation [44] and feedback [45]. Due to the convolution operator is a local connection architecture, ConvLSTM has the advantages in terms of time complexity and space complexity. In addition, comparative experiments and visualization analysis are carried out to verify the ability of ConvLSTM for channel prediction, given in Appendix.

IV. Design of STNN and Training Scheme

By considering channel prediction as a spatio-temporal series problem, we here propose a DL based method, namely STNN, to implement multi-step CSI prediction. Fig. 3 shows the structure of STNN. Specifically, STNN is designed to deal with the nonlinear and non-stationary temporal dynamics and extract the spatial correlation of CSI. Besides, an advanced training scheme is adopted to fill in the gap between STNN training and testing.

A. The Encoder Module

The encoder module is designed to extract the spatial and temporal features from the input CSI, which is stacked with several ConvLSTM layers. Denote $\{\bar{\mathbf{H}}_t\}_{t=1}^{K}$ as the input series, $\bar{\mathbf{H}}_t$ at any time step can be considered as a $2\times N_s\times N_f$ grid of image $\mathbf{H}_t$, where its two channels[2] represent the real and imaginary parts of $\bar{\mathbf{H}}_t$, respectively. Hereafter, we use $\{\mathbf{H}_t\}_{t=1}^{K}$ to denote the input series. Considering the first layer of the encoder, at time step t, the input $\mathbf{H}_t$, hidden states $\boldsymbol{\mathcal{H}}_{t-1}^{\mathrm{e},1}$ and corresponding cell states $\boldsymbol{\mathcal{C}}_{t-1}^{\mathrm{e},1}$ are used to output the hidden states $\boldsymbol{\mathcal{H}}_{t}^{\mathrm{e},1}$ and cell states $\boldsymbol{\mathcal{C}}_{t}^{\mathrm{e},1}$ as

$$\boldsymbol{\mathcal{C}}_t^{\mathrm{e},1}, \boldsymbol{\mathcal{H}}_t^{\mathrm{e},1} = \mathrm{ConvLSTM}\left(\boldsymbol{\mathcal{C}}_{t-1}^{\mathrm{e},1}, \boldsymbol{\mathcal{H}}_{t-1}^{\mathrm{e},1}, \mathbf{H}_t\right). \tag{16}$$

Note that the parameters of ConvLSTM are shared along time in the same layer. Denote N as the total number of encoder layers, all the hidden states $\{\boldsymbol{\mathcal{H}}_0^{\mathrm{e},i}\}_{i=1}^{N}$ and cell states $\{\boldsymbol{\mathcal{C}}_0^{\mathrm{e},i}\}_{i=1}^{N}$ are initialized as pure-zero tensors.

[2]Here "channel" represents a dimension of feature map in CNN, not the wireless channel. In this paper, the specific meaning of "channel" can be inferred from the context.

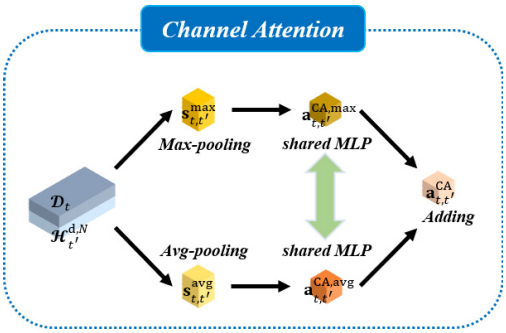


Fig. 4: Diagram for calculating the channel attention weights.

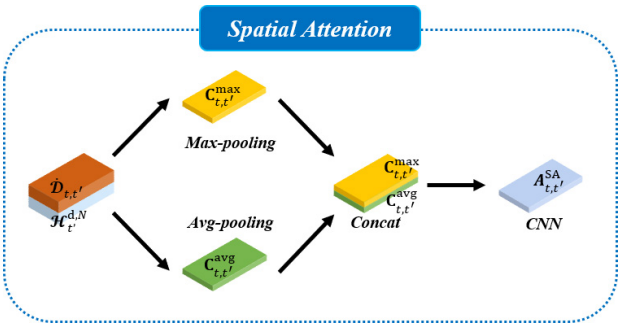


Fig. 5: Diagram for calculating the spatial attention weights.

B. The Differencing-Attention Module

For accurate multi-step prediction, the differencing-attention module is designed to deal with the non-stationary and nonlinear temporal dynamics of CSI and realize adaptive feature refinement, which is consist of four sub-modules, as the differencing, channel attention, spatial attention and temporal attention. Denote the prediction series by $\{\mathbf{H}_{t'}\}_{t'=K+1}^{K+J}$. At time step t', based on the hidden states $\{\boldsymbol{\mathcal{H}}_1^{\mathrm{e},N}, \cdots, \boldsymbol{\mathcal{H}}_K^{\mathrm{e},N}; \boldsymbol{\mathcal{H}}_{t'}^{\mathrm{d},N}\}$, a feature representation tensor $\mathbf{Z}_{t'}$ is obtained, where $\boldsymbol{\mathcal{H}}_{t'}^{\mathrm{d},N}$ denotes the hidden states of the last layer in the decoder at time step t'.

1) The Differencing Sub-Module: Differencing CSI between adjacent time steps is considered to perform channel prediction in [11], [28]. It can be explained from time series analysis that the non-stationary process can be gradually transformed to stationarity through differencing, improving the predictability of time series [46]. To capture the non-stationary temporal dynamic in STNN, the differencing sub-module is designed. The main idea is that the hidden states instead of the past CSI are considered, which could avoid damage to the intrinsic information of CSI structure in array-frequency domain and improve the non-stationary modeling capability of STNN. Specifically, based on the hidden states of the last layer in the encoder, differential states $\boldsymbol{\mathcal{D}}_t$ $(t = 1, \cdots, K-1)$ are obtained by

$$\boldsymbol{\mathcal{D}}_t = \boldsymbol{\mathcal{H}}_{t+1}^{\mathrm{e},N} - \boldsymbol{\mathcal{H}}_t^{\mathrm{e},N}. \tag{17}$$

Only the first-order differencing is executed for channel prediction in equation (17). If the stationarity of input series becomes worse, higher-order differencing can be considered with only minor modifications to STNN. In detail, the higher-order differencing is implemented by iteratively performing Q $(Q > 1)$ times first-order differencing. However, over-differencing may damage the intrinsic structure of series, causing negative impact for the accuracy of prediction. After obtaining the differential states $\{\boldsymbol{\mathcal{D}}_t\}_{t=1}^{K-1}$, attention mechanism is used to adaptively determine significant features from the approximately stationary process.

2) The Channel Attention Sub-Module: Each differential state $\boldsymbol{\mathcal{D}}_t$ is treated as a feature map of size $C \times H \times W$, where C denotes the number of channels and H and W denote the height and width of the feature map. The channel attention sub-module is designed to derive attention weights $\mathbf{a}_{t,t'}^{\mathrm{CA}}$ along the channel axis, as shown in Fig. 4. Specifically, the hidden states $\boldsymbol{\mathcal{H}}_{t'}^{\mathrm{d},N}$ and differential states $\boldsymbol{\mathcal{D}}_t$ are concatenated along the channel axis and both average-pooling and max-pooling operators are used to integrate the spatial information, generating two spatial feature vectors $\mathbf{s}_{t,t'}^{\mathrm{avg}} \in \mathbb{R}^{2C\times 1}$ and $\mathbf{s}_{t,t'}^{\mathrm{max}} \in \mathbb{R}^{2C\times 1}$. Then, $\mathbf{s}_{t,t'}^{\mathrm{avg}}$ and $\mathbf{s}_{t,t'}^{\mathrm{max}}$ are fed into a shared MLP with one hidden layer and two vectors $\mathbf{a}_{t,t'}^{\mathrm{CA,avg}} \in \mathbb{R}^{C\times 1}$ and $\mathbf{a}_{t,t'}^{\mathrm{CA,max}} \in \mathbb{R}^{C\times 1}$ are output. The hidden size is set to $\lfloor 2C/r_1 \rfloor$, where r_1 is a reduction ratio. The ReLU and linear activation functions are applied to the hidden and output layers, respectively. Finally, the channel attention weights $\mathbf{a}_{t,t'}^{\mathrm{CA}} \in \mathbb{R}^{C\times 1}$ are obtained by adding $\mathbf{a}_{t,t'}^{\mathrm{CA,avg}}$ and $\mathbf{a}_{t,t'}^{\mathrm{CA,max}}$ with element-wise. The overall processes can be summarized as

$$\begin{aligned}\mathbf{a}_{t,t'}^{\mathrm{CA}} =& \mathrm{MLP}(\mathrm{AvgPool}([\boldsymbol{\mathcal{H}}_{t'}^{\mathrm{d},N}; \boldsymbol{\mathcal{D}}_t])) \\ &+ \mathrm{MLP}(\mathrm{MaxPool}([\boldsymbol{\mathcal{H}}_{t'}^{\mathrm{d},N}; \boldsymbol{\mathcal{D}}_t])).\end{aligned} \tag{18}$$

The channel attention weights determine "what" to attend for a feature map [47]. Based on differential states $\boldsymbol{\mathcal{D}}_t$ and corresponding attention weights $\mathbf{a}_{t,t'}^{\mathrm{CA}}$, a new feature map $\dot{\boldsymbol{\mathcal{D}}}_{t,t'}$ is derived by

$$\dot{\boldsymbol{\mathcal{D}}}_{t,t'} = \boldsymbol{\mathcal{D}}_t \circ \mathbf{a}_{t,t'}^{\mathrm{CA}}, \tag{19}$$

where the channel attention weights $\mathbf{a}_{t,t'}^{\mathrm{CA}}$ are copied along the spatial axis in equation (19).

3) The Spatial Attention Sub-Module: As shown in Fig. 5, different from the channel attention sub-module, the spatial attention sub-module is designed to determine "where" to attend for a given feature map [47]. To calculate the spatial attention weights $\mathbf{A}_{t,t'}^{\mathrm{SA}} \in \mathbb{R}^{H\times W}$, we first concatenate the hidden states $\boldsymbol{\mathcal{H}}_{t'}^{\mathrm{d},N}$ and feature map $\dot{\boldsymbol{\mathcal{D}}}_{t,t'}$ along the channel axis and use average-pooling and max-pooling operators to integrate the channel information, obtaining two channel features matrices $\mathbf{C}_{t,t'}^{\mathrm{avg}} \in \mathbb{R}^{H\times W}$ and $\mathbf{C}_{t,t'}^{\mathrm{max}} \in \mathbb{R}^{H\times W}$. Next, we concatenate $\mathbf{C}_{t,t'}^{\mathrm{avg}}$ and $\mathbf{C}_{t,t'}^{\mathrm{max}}$ along the channel axis and utilize a convolution layer to output the spatial attention weights $\mathbf{A}_{t,t'}^{\mathrm{SA}}$. The overall process can be summarized as

$$\begin{aligned}\mathbf{A}_{t,t'}^{\mathrm{SA}} = \mathrm{CNN}([&\mathrm{AvgPool}([\boldsymbol{\mathcal{H}}_{t'}^{\mathrm{d},N}; \dot{\boldsymbol{\mathcal{D}}}_{t,t'}]); \\ &\mathrm{MaxPool}([\boldsymbol{\mathcal{H}}_{t'}^{\mathrm{d},N}; \dot{\boldsymbol{\mathcal{D}}}_{t,t'}])]),\end{aligned} \tag{20}$$

where the bias of the convolution layer is set to zero. Further, the spatial attention weights $\mathbf{A}_{t,t'}^{\mathrm{SA}}$ are applied to feature map $\dot{\boldsymbol{\mathcal{D}}}_{t,t'}$ and another new feature map $\ddot{\boldsymbol{\mathcal{D}}}_{t,t'}$ is derived by

$$\ddot{\boldsymbol{\mathcal{D}}}_{t,t'} = \dot{\boldsymbol{\mathcal{D}}}_{t,t'} \circ \mathbf{A}_{t,t'}^{\mathrm{SA}}. \tag{21}$$

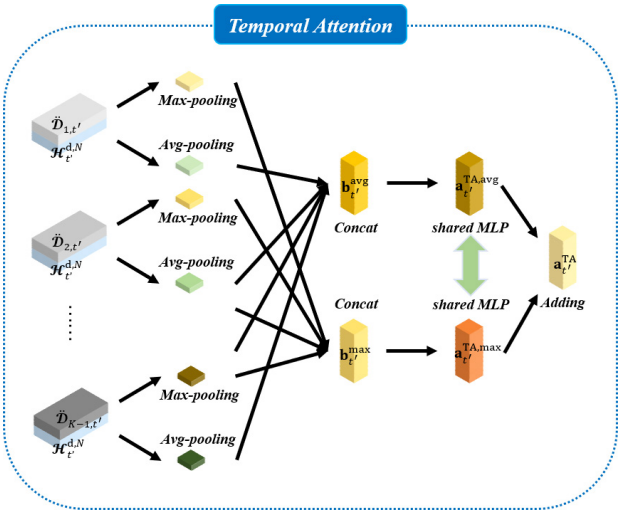


Fig. 6: Diagram for calculating the temporal attention weights.

Similarly, the spatial attention weights $\mathbf{A}^{\mathrm{SA}}_{t,t'}$ are copied along the channel axis in equation (21).

4) The Temporal Attention Sub-Module: After refining a specific feature map $\ddot{\boldsymbol{\mathcal{D}}}_{t,t'}$ for each differential state $\boldsymbol{\mathcal{D}}_t$ along the channel and spatial axes at time step t', the temporal attention sub-module is designed to combine all feature maps $\{\ddot{\boldsymbol{\mathcal{D}}}_{t,t'}\}_{t=1}^{K-1}$ along the temporal axis, as shown in Fig. 6. Specifically, each feature map $\ddot{\boldsymbol{\mathcal{D}}}_{t,t'}$ is concatenated with the hidden states $\boldsymbol{\mathcal{H}}^{\mathrm{d},N}_{t'}$ along the channel axis and both average-pooling and max-pooling operators are used to integrate all the spatial and channel information. Concatenating the corresponding output along the temporal axis, two channel-spatial feature vectors $\mathbf{b}^{\mathrm{avg}}_{t'} \in \mathbb{R}^{(K-1)\times 1}$ and $\mathbf{b}^{\mathrm{max}}_{t'} \in \mathbb{R}^{(K-1)\times 1}$ are obtained. Then, two new vectors $\mathbf{a}^{\mathrm{TA,avg}}_{t'} \in \mathbb{R}^{(K-1)\times 1}$ and $\mathbf{a}^{\mathrm{TA,max}}_{t'} \in \mathbb{R}^{(K-1)\times 1}$ are generated by inputting $\mathbf{b}^{\mathrm{avg}}_{t'}$ and $\mathbf{b}^{\mathrm{max}}_{t'}$ into a shared MLP with one hidden layer. The hidden size is set to $\lfloor (K-1)/r_2 \rfloor$, where r_2 is another reduction ratio. The hidden layer is deployed with the ReLU activation function and the output layer is activated by the linear function. By adding $\mathbf{a}^{\mathrm{TA,avg}}_{t'}$ and $\mathbf{a}^{\mathrm{TA,max}}_{t'}$, we can obtain the temporal attention weights $\mathbf{a}^{\mathrm{TA}}_{t'} \in \mathbb{R}^{(K-1)\times 1}$. The overall process can be summarized as

$$\begin{aligned}
\mathbf{b}^{\mathrm{avg}}_{t'} =& [\mathrm{AvgPool}([\boldsymbol{\mathcal{H}}^{\mathrm{d},N}_{t'}; \ddot{\boldsymbol{\mathcal{D}}}_{1,t'}]); \mathrm{AvgPool}([\boldsymbol{\mathcal{H}}^{\mathrm{d},N}_{t'}; \ddot{\boldsymbol{\mathcal{D}}}_{2,t'}]); \\
& \cdots; \mathrm{AvgPool}([\boldsymbol{\mathcal{H}}^{\mathrm{d},N}_{t'}; \ddot{\boldsymbol{\mathcal{D}}}_{K-1,t'}])] \\
\mathbf{b}^{\mathrm{max}}_{t'} =& [\mathrm{MaxPool}([\boldsymbol{\mathcal{H}}^{\mathrm{d},N}_{t'}; \ddot{\boldsymbol{\mathcal{D}}}_{1,t'}]); \mathrm{MaxPool}([\boldsymbol{\mathcal{H}}^{\mathrm{d},N}_{t'}; \ddot{\boldsymbol{\mathcal{D}}}_{2,t'}]); \\
& \cdots; \mathrm{MaxPool}([\boldsymbol{\mathcal{H}}^{\mathrm{d},N}_{t'}; \ddot{\boldsymbol{\mathcal{D}}}_{K-1,t'}])] \\
\mathbf{a}^{\mathrm{TA}}_{t'} =& \mathrm{MLP}(\mathbf{b}^{\mathrm{avg}}_{t'}) + \mathrm{MLP}(\mathbf{b}^{\mathrm{max}}_{t'}).
\end{aligned} \tag{22}$$

The temporal attention weights determine "when" to attend for the feature map series $\{\ddot{\boldsymbol{\mathcal{D}}}_{t,t'}\}_{t=1}^{K-1}$. Finally, feature representation tensor $\mathbf{Z}_{t'}$ is derived by

$$\mathbf{Z}_{t'} = \sum_{t=1}^{K-1} \ddot{\boldsymbol{\mathcal{D}}}_{t,t'} \circ a^{\mathrm{TA}}_{t',t}, \tag{23}$$

where $a^{\mathrm{TA}}_{t',t}$ denotes the t-th component of attention weights $\mathbf{a}^{\mathrm{TA}}_{t'}$. In equation (23), $a^{\mathrm{TA}}_{t',t} (t = 1, \cdots, K-1)$ are copied along the spatial and channel axes during multiplication.

C. The Decoder Module

The decoder module, consisting of several ConvLSTM layers and a convolution layer without bias, is designed to generate predictions of CSI step by step. The ConvLSTM layers of the decoder are symmetrical to that of the encoder and are initialized by the hidden states $\{\boldsymbol{\mathcal{H}}^{\mathrm{e},i}_{K}\}_{i=1}^{N}$ and cell states $\{\boldsymbol{\mathcal{C}}^{\mathrm{e},i}_{K}\}_{i=1}^{N}$. The convolution layer is used to recover CSI at each time step. Specifically, to predict $\mathbf{H}_{t'}$, the estimated $\hat{\mathbf{H}}_{t'-1}$, the hidden states of the first layer in the decoder $\boldsymbol{\mathcal{H}}^{\mathrm{d},1}_{t'-1}$ and the corresponding cell states $\boldsymbol{\mathcal{C}}^{\mathrm{d},1}_{t'-1}$ are input to the ConvLSTM to output the hidden states $\boldsymbol{\mathcal{H}}^{\mathrm{d},1}_{t'}$ and cell states $\boldsymbol{\mathcal{C}}^{\mathrm{d},1}_{t'}$ as

$$\boldsymbol{\mathcal{C}}^{\mathrm{d},1}_{t'}, \boldsymbol{\mathcal{H}}^{\mathrm{d},1}_{t'} = \mathrm{ConvLSTM}\left(\boldsymbol{\mathcal{C}}^{\mathrm{d},1}_{t'-1}, \boldsymbol{\mathcal{H}}^{\mathrm{d},1}_{t'-1}, \hat{\mathbf{H}}_{t'-1}\right). \tag{24}$$

Upward transmission layer by layer, the hidden state $\boldsymbol{\mathcal{H}}^{\mathrm{d},N}_{t'}$ in the last layer can be obtained at time step t'. After feature representation tensor $\mathbf{Z}_{t'}$ is calculated, it is concatenated with the hidden state $\boldsymbol{\mathcal{H}}^{\mathrm{d},N}_{t'}$ along the channel axis and convolved by the convolution layer to produce prediction $\hat{\mathbf{H}}_{t'}$, which can be expressed as

$$\hat{\mathbf{H}}_{t'} = \mathrm{CNN}\left([\boldsymbol{\mathcal{H}}^{\mathrm{d},N}_{t'}; \mathbf{Z}_{t'}]\right). \tag{25}$$

While for predicting $\hat{\mathbf{H}}_{t'+1}$, the estimated $\hat{\mathbf{H}}_{t'}$ and the corresponding hidden state $\boldsymbol{\mathcal{H}}^{\mathrm{d},1}_{t'}$ and cell state $\boldsymbol{\mathcal{C}}^{\mathrm{d},1}_{t'}$ are input at time step $t'+1$, and so on. Finally, we can obtain an estimated series $\{\hat{\mathbf{H}}_{t'}\}_{t'=K+1}^{K+J}$.

When multiple antennas are equipped on the UE, i.e., $N_r > 1$, a possible idea for STNN is to extend the number of channels to exploit the array correlation among user antennas. To be more specific, the input size of STNN at each time step needs to be extended to $2N_r \times N_s \times N_f$ and the same extension is needed for the output size. In this way, the future CSI corresponding to different antennas of the UE can be predicted. We prefer to leave this to future work.

D. The Advanced Training Scheme

At the training stage, the ground truth $\mathbf{H}_{t'-1}$ is available and generally used to replace the estimated $\hat{\mathbf{H}}_{t'-1}$ in equation (24). However, during the testing stage, the ground truth $\mathbf{H}_{t'-1}$ is unknown and the estimated $\hat{\mathbf{H}}_{t'-1}$ output by STNN itself is considered as the input to predict $\mathbf{H}_{t'}$. This gap among the input distributions between training and testing may cause a negative impact for channel prediction. So we stitch the gap by the scheduled sampling [34]. Specifically, during the i-th epoch of the training stage, a sampling probability η_i is set to decide the possibility of feeding the ground truth. With the increasing number of training epoch, the sampling probability η_i is gradually reduced to push the model to reach the testing stage.

The above three modules of STNN are trained jointly in an end to end manner and the mean squared error (MSE) is used as the loss function. For the estimated series $\hat{\mathbf{S}}_i = (\hat{\mathbf{H}}_{K+1}; \cdots; \hat{\mathbf{H}}_{K+t'}; \cdots; \hat{\mathbf{H}}_{K+J}) \in \mathbb{R}^{J\times 2\times N_s \times N_f}$ and

ground truth series $\mathbf{S}_i = (\mathbf{H}_{K+1}; \cdots; \mathbf{H}_{K+t'}; \cdots; \mathbf{H}_{K+J}) \in \mathbb{R}^{J\times 2\times N_s\times N_f}$, the specific loss function is defined as

$$L(\Theta) = \frac{1}{2\cdot JBN_fN_s}\sum_{i=1}^{B}\|\mathbf{S}_i - \hat{\mathbf{S}}_i\|_2^2, \tag{26}$$

where $\|\cdot\|_2$ denotes the Euclidean norm, B denotes the batch size and Θ denotes the parameter set of STNN. Besides, we utilize adaptive moment estimation (ADAM) optimizer [48] to train STNN.

V. Experimental Results and Analysis

In this section, we provide the details of our experiments and the prediction performance of the proposed STNN is evaluated on a realistic channel model with enhanced mobility and spherical waves. Moreover, we focus on analysis of the temporal attention sub-module to study the learning ability of STNN in temporal domain.

A. Experimental Setting

QuaDRiGa channel simulator [35] can not only support the baseline channel model of 3GPP New Ratio [37] but also provide some additional modeling components for more practical evaluation. In particularly, both enhanced mobility and spherical waves described in section II are implemented in the QuaDRiGa channel simulator. Based on QuaDRiGa, we consider the 3GPP rural macro (RMA) LOS scenario at frequency $f_c = 3.5$ GHz, to evaluate the performance of the proposed method. Note that our STNN is applicable to other communication scenarios. The number of clusters of multipath is 11 and the number of subpaths in each cluster is 20. The number of antennas equipped on BS, N_s, is set to 64, while the UE has only one antenna. The antenna spacing is half of the wavelength λ_c. The number of sub-carriers N_f is set to 64 and the sub-carrier spacing Δf is set to 30 kHz. The UE moves along a linear trajectory with the velocity $v = 60$ km/h, and the period T_s of the sounding reference signal is set to 0.5 ms. The normalized Doppler shift, which influences the time-varying patterns of channel matrix $\bar{\mathbf{H}}_t$ in equation (14), is defined as $f_n = T_s f_d$, where $f_d = \frac{v}{\lambda_c}$ denotes the maximum Doppler shift. We can obtain the normalized Doppler shift $f_n \approx 0.1$ and the maximum Doppler shift $f_d \approx 200$ Hz. To generate the dataset of STNN, we reconstruct the scattering environment 200 times, where the UE moves a distance of 5 meters every time, and the sliding window way is used to obtain each sample of the dataset. The size of window is set to the sum of the prediction length and the input length. The generated dataset is split into three parts, namely training, validation and testing datasets, containing 15000, 2000 and 3000 samples, respectively. The parameters of STNN are optimized based on the training dataset. It is worth mentioning that 3GPP TR 38.901 large scale calibration and full calibration have been performed in QuaDRiGa, which guarantee the reality and reliability of the dataset.

STNN is implemented in PyTorch. The total length of the past series (i.e., the input length) K is set to 20, and the length of the prediction series (i.e., the prediction length) J is set to 5. In other words, we predict CSI in the next 2.5 ms based on the past 10 ms. The reduction ratios r_1 and r_2 are both set to 2 in the channel attention and temporal attention sub-modules. The kernel size of convolution layer in the spatial attention sub-module is set to 5×5. The number of ConvLSTM layers both in the encoder and decoder modules is set to 2, and all the corresponding hidden and kernel sizes are set to 8 and 3×3. The final convolution layer in the decoder has 1×1 kernel size with 2 convolution channels to predict the real and imaginary parts of CSI. During training, we use ADAM optimizer with default setting ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-8$), and the learning rate, batch size and epoch are set to 0.0001, 100 and 1000. To realize the scheduled sampling, the sampling probability in the first epoch is set to 1, and it is then reduced by 0.002 in every epoch. Once the sampling probability decays to 0, it will be fixed. Normalized MSE (NMSE) is considered as the metric to evaluate the prediction performance on the testing dataset, which is defined as

$$\text{NMSE} = E\left\{\frac{\|\mathbf{S}-\hat{\mathbf{S}}\|_2^2}{\|\mathbf{S}\|_2^2}\right\}. \tag{27}$$

We compare the proposed STNN with four benchmarks, the outdated, linear regression, S times channel difference based forward AR (SDFAR) [11] and forward AR (FAR) [11]. The outdated method uses the last CSI for any time in the future. The linear regression is a statistical model, which predicts future CSI by fitting regression line to the past CSI. The FAR method can be seen as a special case of SDFAR when differencing is not performed. For the sake of fairness, the length of the available past series for calculating the above mentioned model parameters is also set to 20 and the prediction length is set to 5. The AR order of the FAR method is set to 4. For the SDFAR method, the AR order is set to 3 and the differencing order is set to 1. Unless otherwise specified, experimental setting mentioned above is applied throughout this paper.

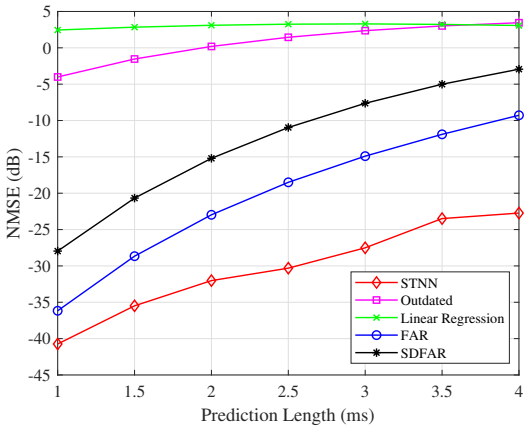


Fig. 7: NMSE (dB) performance comparison between STNN and four benchmarks under multi-step channel prediction. STNN shows the highest accuracy among all predictors.

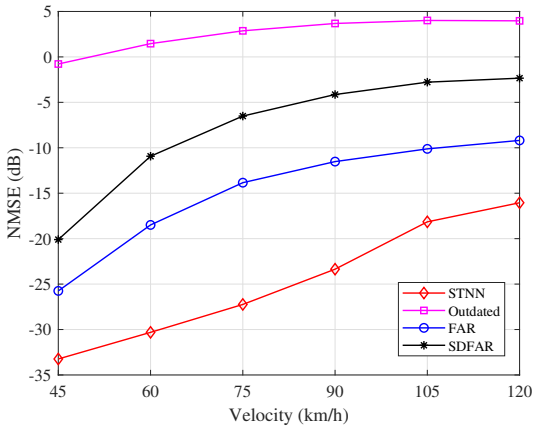


Fig. 8: Comparison of NMSE (dB) between STNN, FAR, SDFAR and linear regression with respect to the velocity v. The corresponding normalized Doppler shifts f_n are 0.075, 0.1, 0.125, 0.15, 0.175 and 0.2.

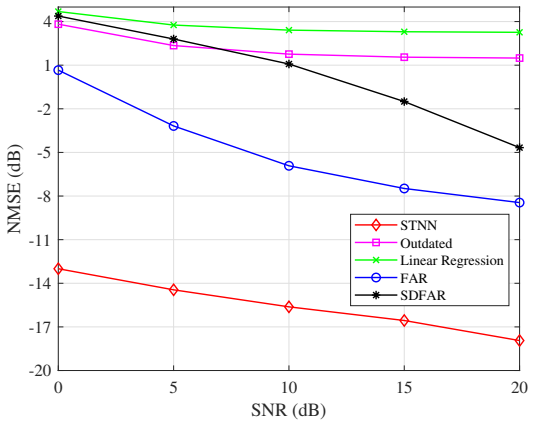


Fig. 9: Comparison of NMSE (dB) between STNN and other benchmarks in noisy CSI.

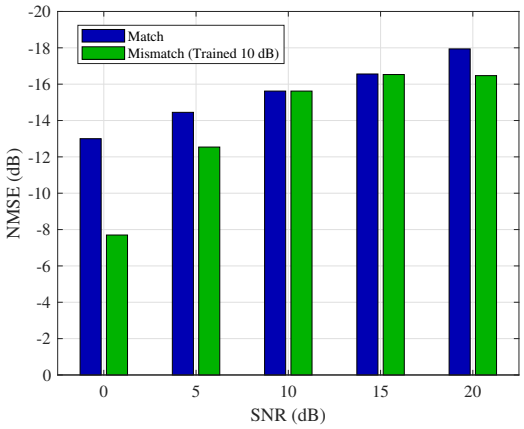


Fig. 10: Comparison of NMSE (dB) between the match and mismatch scheme in noisy CSI.

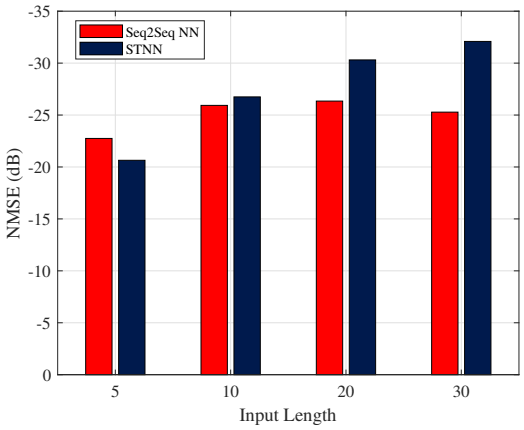


Fig. 11: Comparison of NMSE (dB) between STNN and Seq2Seq NN with respect to the input length K.

B. Performance of the Proposed STNN

1) Effect of the Prediction Length: Fig. 7 shows the NMSE of different methods under multi-step channel prediction. When the prediction length J increases from 2 to 8 (i.e., 1 ms to 4 ms), the performance of all predictors decline, except for linear regression. Compared with the FAR and SDFAR methods, the NMSE of STNN increases slowly as the prediction length increases. When the prediction length is 2 ms, STNN can improve the accuracy of the prediction by 39% compared to the FAR. This indicates that STNN has ability to provide more accurate predictions. In addition, the performance gap between the FAR and SDFAR methods implies that differencing CSI is not always useful for channel prediction. The reason is that differencing may cause a loss of intrinsic information of CSI in practical high-mobility scenarios.

2) Effect of the Normalized Doppler Shift: Fig. 8 shows the NMSE of different methods against the velocity of the UE. Compared with other methods, significant gains of performance can be observed from STNN at various velocities. The reason is the design of STNN carefully considers the characteristics of CSI in array-frequency-temporal domain. By comparing the performance of the FAR and SDFAR at $v = 45$ km/h and $v = 60$ km/h, the decreasing of performance implies that these two AR based methods encounter difficulties in high-mobility scenarios. Additionally, the poor performance of the outdated method reflects that CSI changes rapidly during the corresponding time period.

3) Effect of Imperfect CSI: Considering imperfect CSI case, we add the zero-mean white Gaussian noise with different variance levels to CSI. Fig. 9 shows the NMSE of different methods against signal to noise ratio (SNR). There is no doubt that STNN outperforms other predictors in all SNR regimes. At low SNR (e.g., 0 dB and 5dB), we can observe that SDFAR even performs worse than the outdated method.

4) Comparison Results under the RMSE and MAE Metrics: In addition to considering the NMSE as an evaluation metric, we also compare the performance of different methods under the root mean squared error (RMSE) and mean absolute error (MAE) metrics. Table II presents the RMSE and MAE of different methods related to the prediction length, Table III

TABLE II: RMSE and MAE performance comparison of different methods under multi-step channel prediction.

Method	Metric	Prediction Length						
		1.0 ms	1.5 ms	2.0 ms	2.5 ms	3.0 ms	3.5 ms	4.0 ms
Outdated	RMSE	0.0520	0.0691	0.0842	0.0972	0.1079	0.1160	0.1217
	MAE	0.0410	0.0534	0.0646	0.0745	0.0829	0.0898	0.0946
Linear Regression	RMSE	0.1084	0.1135	0.1172	0.1193	0.1197	0.1187	0.1165
	MAE	0.0897	0.0936	0.0964	0.0979	0.0980	0.0970	0.0947
FAR	RMSE	0.0010	0.0025	0.0049	0.0083	0.0127	0.0182	0.0247
	MAE	0.0006	0.0013	0.0025	0.0041	0.0062	0.0087	0.0115
SDFAR	RMSE	0.0028	0.0065	0.0124	0.0205	0.0304	0.0417	0.0536
	MAE	0.0018	0.0040	0.0072	0.0117	0.0172	0.0235	0.0302
STNN	RMSE	**0.0007**	**0.0013**	**0.0019**	**0.0024**	**0.0034**	**0.0051**	**0.0057**
	MAE	**0.0005**	**0.0009**	**0.0014**	**0.0017**	**0.0023**	**0.0035**	**0.0038**

TABLE III: RMSE and MAE performance comparison of different methods with respect to the velocity v.

Method	Metric	Velocity					
		45 km/h	60 km/h	75 km/h	90 km/h	105 km/h	120 km/h
Outdated	RMSE	0.0778	0.0982	0.1084	0.1112	0.1087	0.1026
	MAE	0.0590	0.0745	0.0841	0.0866	0.0839	0.0783
FAR	RMSE	0.0034	0.0083	0.0143	0.0180	0.0199	0.0210
	MAE	0.0015	0.0041	0.0074	0.0098	0.0112	0.0121
SDFAR	RMSE	0.0072	0.0205	0.0341	0.0431	0.0478	0.0475
	MAE	0.0039	0.0117	0.0201	0.0264	0.0298	0.0301
STNN	RMSE	**0.0017**	**0.0024**	**0.0034**	**0.0048**	**0.0073**	**0.0090**
	MAE	**0.0012**	**0.0017**	**0.0022**	**0.0031**	**0.0049**	**0.0057**

TABLE IV: RMSE and MAE performance comparison of different methods in noisy CSI.

Method	Metric	SNR				
		0 dB	5 dB	10 dB	15 dB	20 dB
Outdated	RMSE	0.1274	0.1077	0.1007	0.0983	0.0976
	MAE	0.0995	0.0832	0.0741	0.0756	0.0750
Linear Regression	RMSE	0.1410	0.1265	0.1216	0.1200	0.1195
	MAE	0.1138	0.1021	0.0993	0.0984	0.0981
FAR	RMSE	0.0905	0.0586	0.0426	0.0353	0.0314
	MAE	0.0677	0.0432	0.0307	0.0246	0.0214
SDFAR	RMSE	0.1364	0.1138	0.0932	0.0696	0.0488
	MAE	0.1052	0.0856	0.0684	0.0495	0.0336
STNN	RMSE	**0.0182**	**0.0155**	**0.0135**	**0.0123**	**0.0105**
	MAE	**0.0132**	**0.0111**	**0.0096**	**0.0085**	**0.0072**

lists the comparison results related to the velocity and Table IV shows the comparison results related to SNR. Similar to the results under the NMSE, STNN has the lowest prediction error among all methods in both the RMSE and MAE. More specifically, STNN shows better performance than FAR on MAE by decreasing 44% (at 2ms), 68% (at 90km/h) and 66% (at 20dB).

5) Robustness to imperfect CSI: We here study the robustness of STNN against SNR. As shown in Fig. 10, STNN was trained at SNR = 10 dB and then directly tested from 0 dB to 20 dB. The match scheme trains and tests STNN at the same SNR, which is chosen as comparison. When the testing SNR is lower than the training SNR, the NMSE of STNN drastically degrades. However, the prediction performance of STNN can almost reach the same result with the match scheme if the testing SNR is slightly higher than the training SNR. Therefore, when deploying STNN in the real communication systems, we should choose a relatively low SNR to train STNN.

6) Effect of the Input Length: For our STNN, another interesting problem is to study how many the past CSI are required to achieve a superior performance. Fig. 11 shows the NMSE of STNN and Seq2Seq NN[3] with respect to the input length K. When we increase the number of past CSI fed into STNN, the corresponding prediction performance has gradually improved. Meanwhile, we can observe that the performance gains become less and less when increase the input length K, because the correlation among CSI weakens. As the input length K increases from 10 to 30, the limited accuracy improvement and subsequent decline of Seq2Seq NN can be seen in Fig. 11. Therefore, we may conclude that STNN is able to cope with the long-term temporal dependence of CSI. It is also worth mentioning that the increase of the input length will bring higher time complexity for STNN. Considering the trade-off between prediction accuracy and complexity, hence we set the input length $K = 20$ in Section V-A.

7) Effect of the HP Setting: Table V compares the parameter numbers, floating point operations (FLOPs) and corresponding NMSE of STNN with different kernel sizes and depths. The HP setting $\{3\times3-8\}$ uses one ConvLSTM layer with hidden size 8 and kernel size 3×3 in both encoder and decoder modules, the setting $\{5\times5-8\}$ uses one ConvLSTM

[3]Seq2Seq NN stands for STNN without the differencing-attention module. The specific number of ConvLSTM layers, hidden size and kernel size of Seq2Seq NN are set to 2, 8 and 3×3. The other hyper-parameters are the same with STNN to provide a fair evaluation.

TABLE V: Comparison of the parameter numbers, Flops (G) and NMSE (dB) of STNN with different HP settings.

HP Setting	**Parameter Numbers**	**Flops** (G)	**NMSE** (dB)
{3×3-8}	6512	0.68	-25.66
{5×5-8}	16752	1.72	-27.76
{3×3-8-3×3-8}	15920	1.62	-30.31

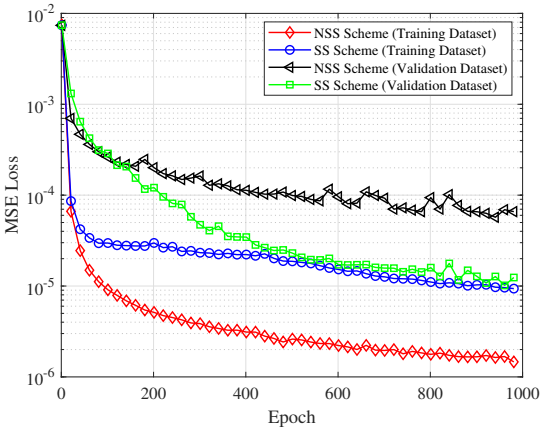


Fig. 12: Comparison of MSE loss on training dataset and validation dataset between the scheduled sampling (SS) and non-scheduled sampling (NSS) schemes with respect to training epoch.

layer with hidden size 8 and kernel size 5×5 and the setting $\{3\times3-8-3\times3-8\}$ uses two ConvLSTM layers, where each layer has the same kernel size 3×3 and hidden size 8. We can observe that STNN with kernel size 5×5 outperforms the one with kernel size 3×3. The reason is that more global information is obtained when STNN extracts CSI array-frequency correlations. Meanwhile, the performance improvement also brings the increase in parameter numbers and FLOPs. Due to the enhancement of nonlinear approximation ability, the third setting outperforms the second one with a little bit decrease in parameter numbers and FLOPs.

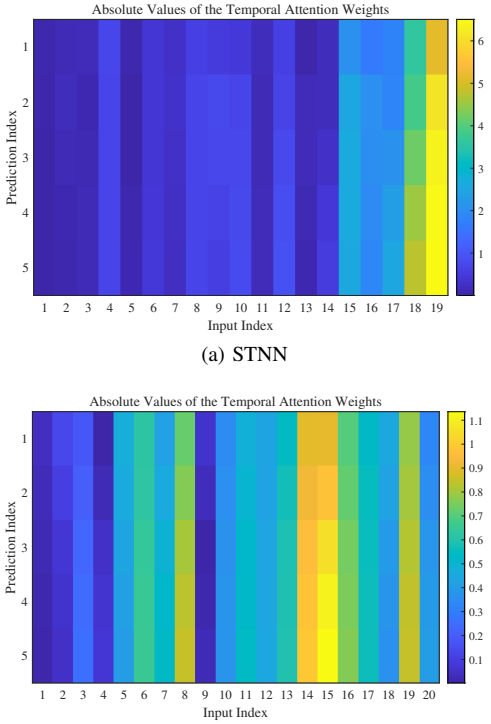
8) Effect of the Advanced Training Scheme: Fig. 12 illustrates the relationship between MSE loss and epoch of STNN under the SS and NSS[4] schemes. Due to avoiding iteration errors at the decoder module of STNN, the training loss of the NSS scheme is superior than that of the SS scheme. In contrast, the MSE loss of the SS scheme on validation dataset is lower than the NSS scheme. This is because the SS scheme fills in the discrepancy between STNN training and testing. It turns out that the SS scheme is more effective in the practical deployment of STNN.

9) Ablation Study: Model ablation is carried out to verify the necessity of the differencing sub-module for dealing with the non-stationary temporal dynamic. As illustrated in Table VI, the prediction performance is significantly reduced when we only remove the differencing sub-module from STNN. The comparison method has the same parameter setting, loss function and training scheme as STNN, only the first-order differencing for the hidden states $\boldsymbol{\mathcal{H}}_t^{\mathrm{e},N}$ is ignored in the structure of NN. Thus, it is important to consider CSI non-stationary temporal dynamic when implementing channel prediction.

[4]At time step t' of decoder during the training stage, the previous estimated $\hat{\mathbf{H}}_{t'-1}$ is ignored and the ground truth $\mathbf{H}_{t'-1}$ is fed into STNN.

TABLE VI: Ablation study with respect to the differencing sub-module.

Model	**NMSE** (dB)
STNN (Without the Differencing Sub-Module)	-23.74
STNN	-30.31



(a) STNN

(b) STNN (Without the Differencing Sub-module)

Fig. 13: Absolute values of the temporal attention weights at each decoder time step learned by STNN and STNN (without differencing sub-module) for a random sample in the testing dataset.

C. Analysis of the temporal attention sub-module

In this subsection, we focus on the analysis of the temporal attention sub-module to study the role of the differencing sub-module in STNN. We arbitrarily choose a sample from the testing dataset and visualize absolute values of the temporal attention weights to find "when" to attend for STNN. From Fig. 13a, we can observe that the differences in the last few time steps are assigned higher weights compared to the other time steps in STNN. The higher the weight, the greater the impact on the predictions will be imposed. In contrast, the temporal attention weights learned by STNN (Without the Differencing Sub-module) are in disorders, as shown in Fig. 13b.

Further, we use ADF test [49] to evaluate the stationary degree of feature map series $\{\ddot{\boldsymbol{\mathcal{D}}}_{t,t'}\}_{t=1}^{K-1}$ $(t' = 1, \cdots, J)$. The null hypothesis of the ADF test is that series has a unit root (i.e., the non-stationary process). Fig. 14 shows the

TABLE VII: Comparison of the NMSE (dB), RMSE, MAE, parameter numbers and Flops (G) between LSTM, GRU, the cascaded CNN and LSTM, and ConvLSTM.

Model	**NMSE** (dB)	**RMSE**	**MAE**	**Parameter Numbers**	**Flops** (G)
LSTM	-11.34	0.0210	0.0157	604 M	21.61
GRU	-11.29	0.0214	0.0160	470 M	16.24
CNN-LSTM	-12.45	0.0182	0.0137	604 M	21.63
ConvLSTM	-19.11	0.0087	0.0065	8112	1.31

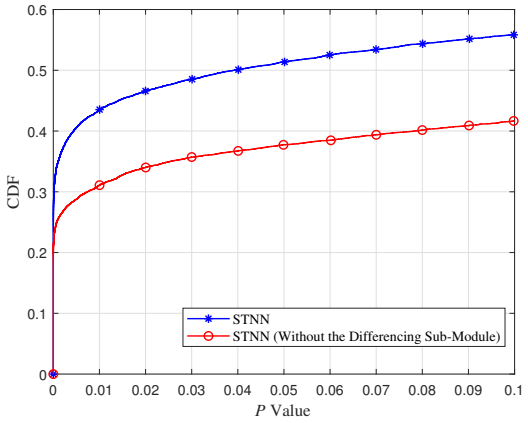
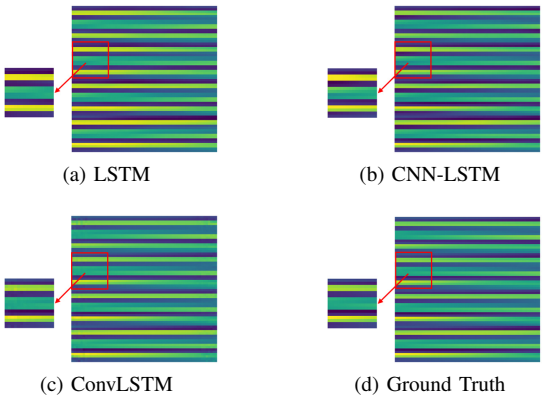


Fig. 14: The CDF of P value obtained by ADF test with the feature map series in the temporal attention sub-module of STNN and STNN (Without the Differencing Sub-module).



(a) LSTM (b) CNN-LSTM

(c) ConvLSTM (d) Ground Truth

Fig. 15: The output of different models for the real part of CSI and the corresponding ground truth CSI.

cumulative distribution function (CDF) of P value with STNN and STNN (Without the Differencing Sub-module). In the 1% (5%) significance level, we can conclude that the differencing sub-module brings significant improvement of feature map series stationarity in the temporal attention module of STNN, leading to an accurate pattern of the temporal attention wights $\mathbf{a}_{t'}^{\mathrm{TA}}$ $(t' = 1, \cdots, J)$.

VI. Conclusion

In this paper, we studied channel prediction problem in massive MIMO-OFDM systems and addressed several existing difficulties based on DL. After defining the specific formulation of channel prediction, we investigated the ability of CSI spatio-temporal modeling and complexity of ConvLSTM and compared it with LSTM. More importantly, we proposed a novel deep learning based channel prediction method, namely STNN. It can not only jointly extract CSI spatial-temporal information but also effectively deal with CSI non-stationary and non-linear temporal dynamics. When evaluated on a realistic channel model with enhanced mobility and spherical waves, experimental results showed that STNN could improve the prediction performance significantly and perform well with respect to different SNRs. Furthermore, we also demonstrated the superior ability of STNN in learning the time-varying patterns of CSI by the methods of visualization and the stationary analysis.

Appendix

Table VII compares the NMSE, RMSE, MAE, parameter numbers and Flops of four basic DL based channel prediction models. For the LSTM model, the input at each time step t is the vectorized form of CSI matrix $\mathbf{H}_t$ and the number of neurons (i.e., hidden size) is set to 8192. The experimental setting of the gate recurrent unit (GRU) model is same to the LSTM model, only the structure of NN is changed. For the CNN-LSTM model, CSI spatial correlation is first extracted by a convolution layer with kernel size 5×5 and convolution channel 2 and then the output feature map at each time step is vectorized and fed into LSTM with hidden size 8192. For the ConvLSTM model, we directly input CSI matrix $\mathbf{H}_t$ at each time step t and the hidden size and kernel size are set to 8 and 5×5. The number of LSTM and ConvLSTM layers are set to 1. We here predict a single CSI matrix $\mathbf{H}_{K+J}$ instead of series $\{\mathbf{H}_{t'}\}_{t'=K+1}^{K+J}$ for simplicity.

As listed in Table VII, GRU and LSTM have similar prediction accuracy, but GRU enjoys lower parameter numbers and Flops. After adding an extra convolution layer, obvious performance gain can be observed from CNN-LSTM. This indicates that the extraction of the spatial correlation is beneficial. Compared to CNN-LSTM, convolutional features of ConvLSTM are incorporated into the transmission of the recurrent states over time step, and we can observe that ConvLSTM outperforms the other three basic DL models. The improvement of performance verifies the superiority of the "deeper" integration between the convolution operators and LSTM cell unit, which is similar to the results in [50]. Meanwhile, the complexity of ConvLSTM is the lowest. Furthermore, the visualization analysis in the same way as [51] is used to reveal the reason behind the performance gain of ConvLSTM. An arbitrary sample in the testing dataset is selected, and the output of different basic DL models and the ground truth are visualized. By comparing the output of different DL models and the corresponding ground truth in Fig. 15, we can observe

that LSTM loses the most information in array-frequency domain, followed by CNN-LSTM, and ConvLSTM predicts the characteristics of CSI best. The results of visualization analysis illustrate the superiority of ConvLSTM from another perspective.

REFERENCES

[1] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 186–195, 2014.

[2] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, "An overview of massive MIMO: Benefits and challenges," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 742–758, 2014.

[3] E. Björnson, J. Hoydis, M. Kountouris, and M. Debbah, "Massive MIMO systems with non-ideal hardware: Energy efficiency, estimation, and capacity limits," *IEEE Transactions on Information Theory*, vol. 60, no. 11, pp. 7112–7139, 2014.

[4] T. Schmidl and D. Cox, "Robust frequency and timing synchronization for OFDM," *IEEE Transactions on Communications*, vol. 45, no. 12, pp. 1613–1621, 1997.

[5] B. Lin, F. Gao, S. Zhang, T. Zhou, and A. Alkhateeb, "Deep learning-based antenna selection and CSI extrapolation in massive MIMO systems," *IEEE Transactions on Wireless Communications*, vol. 20, no. 11, pp. 7669–7681, 2021.

[6] J. Du, W. Xu, H. Shen, X. Dong, and C. Zhao, "Hybrid precoding architecture for massive multiuser MIMO with dissipation: Sub-connected or fully connected structures?" *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5465–5479, 2018.

[7] S. Lakshminarayana, M. Assaad, and M. Debbah, "Coordinated multicell beamforming for massive MIMO: A random matrix approach," *IEEE Transactions on Information Theory*, vol. 61, no. 6, pp. 3387–3412, 2015.

[8] J.-K. Hwang and J. Winters, "Sinusoidal modeling and prediction of fast fading processes," in *IEEE GLOBECOM 1998 (Cat. NO. 98CH36250)*, Sydney, Australia, Nov. 1998, pp. 892–897.

[9] J. Vanderpypen and L. Schumacher, "MIMO channel prediction using ESPRIT based techniques," in *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, Athens, Greece, Dec. 2007, pp. 1–5.

[10] A. Duel-Hallen, "Fading channel prediction for mobile radio adaptive transmission systems," *Proceedings of the IEEE*, vol. 95, no. 12, pp. 2299–2313, 2007.

[11] M. Ozawa, T. Ohtsuki, F. H. Panahi, W. Jiang, Y. Takatori, and T. Nakagawa, "A low-complexity high-accuracy AR based channel prediction method for interference alignment," in *2018 IEEE Global Communications Conference (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–7.

[12] W. Peng, M. Zou, and T. Jiang, "Channel prediction in time-varying massive MIMO environments," *IEEE Access*, vol. 5, pp. 23 938–23 946, 2017.

[13] C. Min, N. Chang, J. Cha, and J. Kang, "MIMO-OFDM downlink channel prediction for IEEE802.16e systems using Kalman filter," in *2007 IEEE Wireless Communications and Networking Conference*, Hong Kong, China, Mar. 2007, pp. 942–946.

[14] L. Zhang, Z. Jin, W. Chen, and X. Zhang, "An improved adaptive channel prediction for MIMO-OFDM systems," in *2008 Third International Conference on Communications and Networking in China*, Hangzhou, China, Aug. 2008, pp. 1008–1012.

[15] T. Svantesson and A. Swindlehurst, "A performance bound for prediction of MIMO channels," *IEEE Transactions on Signal Processing*, vol. 54, no. 2, pp. 520–529, 2006.

[16] M. Khani, M. Alizadeh, J. Hoydis, and P. Fleming, "Adaptive neural signal detection for massive MIMO," *IEEE Transactions on Wireless Communications*, vol. 19, no. 8, pp. 5635–5648, 2020.

[17] L. Liu, H. Feng, T. Yang, and B. Hu, "MIMO-OFDM wireless channel prediction by exploiting spatial-temporal correlation," *IEEE Transactions on Wireless Communications*, vol. 13, no. 1, pp. 310–319, 2014.

[18] C. Lv, J.-C. Lin, and Z. Yang, "Channel prediction for millimeter wave MIMO-OFDM communications in rapidly time-varying frequency-selective fading channels," *IEEE Access*, vol. 7, pp. 15 183–15 195, 2019.

[19] H. Yin, H. Wang, Y. Liu, and D. Gesbert, "Addressing the curse of mobility in massive MIMO with Prony-based angular-delay domain channel predictions," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 12, pp. 2903–2917, 2020.

[20] C. Wu, X. Yi, Y. Zhu, W. Wang, L. You, and X. Gao, "Channel prediction in high-mobility massive MIMO: From spatio-temporal autoregression to deep learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 1915–1930, 2021.

[21] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[22] H. He, S. Jin, C.-K. Wen, F. Gao, G. Y. Li, and Z. Xu, "Model-driven deep learning for physical layer communications," *IEEE Wireless Communications*, vol. 26, no. 5, pp. 77–83, 2019.

[23] H. Huang, S. Guo, G. Gui, Z. Yang, J. Zhang, H. Sari, and F. Adachi, "Deep learning for physical-layer 5G wireless techniques: Opportunities, challenges and solutions," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 214–222, 2020.

[24] H. Kim, S. Kim, H. Lee, C. Jang, Y. Choi, and J. Choi, "Massive MIMO channel prediction: Kalman filtering vs. machine learning," *IEEE Transactions on Communications*, vol. 69, no. 1, pp. 518–528, 2021.

[25] W. Jiang and H. D. Schotten, "Neural network-based fading channel prediction: A comprehensive overview," *IEEE Access*, vol. 7, pp. 118 112–118 124, 2019.

[26] T. Peng, R. Zhang, X. Cheng, and L. Yang, "LSTM-based channel prediction for secure massive MIMO communications under imperfect CSI," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, Dublin, Ireland, Jun. 2020, pp. 1–6.

[27] Z. Tao and S. Wang, "Improved downlink rates for FDD massive MIMO systems through Bayesian neural networks-based channel prediction," *IEEE Transactions on Wireless Communications*, vol. 21, no. 3, pp. 2122–2134, 2022.

[28] Y. Zhu, X. Dong, and T. Lu, "An adaptive and parameter-free recurrent neural structure for wireless channel prediction," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 8086–8096, 2019.

[29] Y. Huangfu, J. Wang, R. Li, C. Xu, X. Wang, H. Zhang, and J. Wang, "Predicting the mumble of wireless channel with sequence-to-sequence models," in *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Istanbul, Turkey, Sep. 2019, pp. 1–7.

[30] A. Kulkarni, A. Seetharam, A. Ramesh, and J. D. Herath, "DeepChannel: Wireless channel quality prediction using deep learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 443–456, 2020.

[31] J. Ma, S. Zhang, H. Li, F. Gao, and Z. Han, "Time-varying downlink channel tracking for quantized massive MIMO networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6721–6736, 2020.

[32] J. Yuan, H. Q. Ngo, and M. Matthaiou, "Machine learning-based channel prediction in massive MIMO with channel aging," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 2960–2973, 2020.

[33] C. Luo, J. Ji, Q. Wang, X. Chen, and P. Li, "Channel state information prediction for 5G wireless communications: A deep learning approach," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 227–236, 2020.

[34] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," *arXiv:1506.03099*, 2015.

[35] S. Jaeckel, L. Raschkowski, K. Börner, and L. Thiele, "QuaDRiGa: A 3-D multi-cell channel model with time evolution for enabling virtual field trials," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 6, pp. 3242–3256, 2014.

[36] 3GPP, "Study on 3D channel model for LTE," Tech. Rep. 36.873, 2017, v12.5.0.

[37] 3GPP, "Study on channel model for frequencies from 0.5 to 100GHz," Tech. Rep. 38.901, 2017, v14.1.0.

[38] J. Bian, C.-X. Wang, X. Gao, X. You, and M. Zhang, "A general 3D non-stationary wireless channel model for 5G and beyond," *IEEE Transactions on Wireless Communications*, vol. 20, no. 5, pp. 3211–3224, 2021.

[39] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, Cambridge, MA, USA, Dec. 2015, pp. 802–810.

[40] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[41] C. Xue, T. Zhou, H. Zhang, L. Liu, and C. Tao, "Deep learning based channel prediction for massive MIMO systems in high-speed railway scenarios," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, Helsinki, Finland, Apr. 2021, pp. 1–5.

[42] T. Nishio, H. Okamoto, K. Nakashima, Y. Koda, K. Yamamoto, M. Morikura, Y. Asai, and R. Miyatake, "Proactive received power

prediction using machine learning and depth images for mmWave networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2413–2427, 2019.

[43] Z. Xiao, Z. Zhang, C. Huang, X. Chen, C. Zhong, and M. Debbah, "C-GRBFnet: A physics-inspired generative deep neural network for channel representation and prediction," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 8, pp. 2282–2299, 2022.

[44] P. Jiang, C.-K. Wen, S. Jin, and G. Y. Li, "Dual CNN-based channel estimation for MIMO-OFDM systems," *IEEE Transactions on Communications*, vol. 69, no. 9, pp. 5859–5872, 2021.

[45] W. Utschick, V. Rizzello, M. Joham, Z. Ma, and L. Piazzi, "Learning the CSI recovery in FDD systems," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2022.

[46] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.

[47] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proceedings of the European conference on computer vision*, Cham, German, Oct. 2018, pp. 3–19.

[48] D. P. Kingma and J. Ba, "ADAM: A method for stochastic optimization," *arXiv: 1412.6980*, 2014.

[49] J. G. MacKinnon, "Approximate asymptotic distribution functions for unit-root and cointegration tests," *Journal of Business & Economic Statistics*, vol. 12, no. 2, pp. 167–176, 1994.

[50] Y. Wang, L. Jiang, M.-H. Yang, L.-J. Li, M. Long, and L. Fei-Fei, "Eidetic 3D LSTM: A model for video prediction and beyond," in *International Conference on Learning Representations*, Vancouver, BC, Canada, May 2019, pp. 1–14.

[51] Z. Hu, J. Guo, G. Liu, H. Zheng, and J. Xue, "MRFNet: A deep learning-based CSI feedback approach of massive MIMO systems," *IEEE Communications Letters*, vol. 25, no. 10, pp. 3310–3314, 2021.