IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 September 2023, accepted 4 October 2023, date of publication 16 October 2023, date of current version 27 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3324399

RESEARCH ARTICLE

Enhanced AI-Based CSI Prediction Solutions for Massive MIMO in 5G and 6G Systems

DAOUD BURGHAL[1], YANG LI[2], PRANAV MADADI[1], YEQING HU[2], JEONGHO JEON[1], (Senior Member, IEEE), JOONYOUNG CHO[2], ANDREAS F. MOLISCH[2], AND JIANZHONG ZHANG[2]

[1]Samsung Research America, Mountain View, CA 94043, USA
[2]Samsung Research America, Richardson, TX 75024, USA

Corresponding author: Daoud Burghal (daoud.burghal@outlook.com)

ABSTRACT Accurate Channel State Information (CSI) is critical for maximizing the throughput of massive Multi-Input Multi-Output (mMIMO) systems. Due to the environment dynamics and user mobility, CSI aging is a major challenge to achieving the large throughput of mMIMO promised by theory. CSI prediction can be used to overcome this without increasing the signaling overhead. Motivated by the anticipated native support for Artificial Intelligence (AI) in the fifth generation and beyond cellular standards, we propose deep learning CSI prediction solutions based on 3-Dimensional (3D) Complex Convolutional Neural Networks (CCNN). These solutions provide improved capabilities for capturing temporal and spatial correlations, enhancing CSI prediction performance. In particular, they utilize the angle delay decomposition of previously observed CSI to predict the future one. In one architecture, the network, dubbed CSI Prediction Network (CSI-PNet), uses small kernels with circular padding to efficiently capture the correlation between propagation paths in the angle domain. This architecture can be further improved by the use of an attention-like model to vary the weights and enhance prediction performance adaptively. We also propose methods to enhance robustness to noise and time and frequency offsets. We tested these solutions using 3GPP-compatible simulations and field measurements in a *commercial* network. Our solutions demonstrate stable performance and significantly outperform several benchmarks, especially at low and medium speeds. They strike a balance between performance and architecture complexity, indicating suitability for actual implementation.

INDEX TERMS Artificial intelligence, channel state information, CSI aging, MIMO, prediction, 6G.

I. INTRODUCTION

Massive Multi-Input Multi-Output (mMIMO) is a core technology to improve spectral efficiency and meet the throughput demands in fifth generation (5G) and sixth generation (6G) mobile networks [1]. The performance of mMIMO depends on the availability of accurate Channel State Information (CSI), e.g., for precoding [2], [3]. Typically, channel training using dedicated pilot transmission is needed for CSI estimation. Acquiring the CSI in mMIMO is challenging as it comes with additional overhead and energy costs, which usually scale with the number of User Equipments (UE) and/or the number of Base-Station (BS) antenna elements. The training should be repeated frequently when the wireless channel is time-varying, especially when the UEs are highly mobile. Otherwise, CSI becomes outdated due to the delay between the time of the estimate and the actual use. The problem is aggravated in Frequency Division Duplex (FDD) systems, where channel reciprocity does not hold due to the large separation between the uplink and the downlink frequencies. In such a case, the training process, along with signal feedback, might be needed for the uplink and the downlink, increasing the overhead and the delay. As a result, conventional methods to acquire the CSI for large MIMO systems result in lowered spectral efficiency and degradation of the overall system performance.

The associate editor coordinating the review of this manuscript and approving it for publication was Barbara Masini.

CSI prediction can be used to enhance performance and mitigate the limitation of conventional CSI acquisition techniques. For instance, the BS or the UE may utilize the previous channel observations to predict the future CSI, thus reducing the impact of processing and feedback delays. CSI prediction is one of the classical problems in wireless communication, and analytical techniques such as autoregression [4], [5], polynomial extrapolation [6], prediction based on high-resolution parameters estimation [7], [8], [9], and others have been suggested [10]. However, such *analytical* models inherently rely on channel models that may not reflect the complexity of the channel evolution.

Machine learning (ML) has recently emerged as a powerful data-driven framework to capture complex relations between observations and target quantities. Recent studies show that ML can be used to enhance different aspects of wireless communications systems [11], [12]. Thus it is envisioned that 5G-Advanced and 6G systems will provide native AI support [13], [14]. Due to its high impact, ML-based CSI prediction has been explored in the literature (see Sec. I-A) and is being actively discussed in the Third Generation Partnership Project (3GPP), the main standardization organization for cellular systems, as one possible application of AI-aided communication [13]. In this paper, we propose CSI prediction solutions using deep Neural Network (NN) architectures that efficiently utilize the physical characteristics of the propagation channels for improved CSI prediction.

In this paper, we introduce the CSI Prediction Network (CSI-PNet), which is based on a Complex-valued 3D Convolutional Neural Network (3D-CCNN) architecture that incorporates three key technologies. First, the network uses a Complex-valued Neural Network (CvNN), i.e., utilizes complex weights to *jointly* process the real and imaginary parts of the CSI signal. This approach offers a more natural way to handle complex input values than splitting/stacking the real and imaginary components. Second, the network employs the angle-delay channel representation of the CSI, which can (roughly) help in sparsifying the CSI representation and differentiating between the different propagation clusters and paths. Third, the network utilizes a 3D-CNN with circular padding, which facilitates a weight-sharing mechanism and reduces the number of trainable parameters in the network while capturing the impact of the *local* correlation between paths with small kernels. Numerous earlier studies (e.g., [3], [4], [15], [16], [17]) have confirmed that the correlation between the different clusters diminishes as the delay and/or angle separation increases, thus constraining the correlation to a small region in the angle-delay domain. The circular padding preserves the paths' relative relations throughout the network. As an enhancement, we introduce the Adaptive CSI-PNet (ACSI-PNet), which uses an additional 3D-CCNN to generate adaptive weights based on previous observations. This mechanism can be viewed as a temporal attention layer. Both solutions employ skip-connections to enhance learnability and accelerate convergence.

In addition, we propose structural and preprocessing enhancements to combat the impact of practical impairments, such as noise, and Timing- and Frequency-Offset (TFO). Specifically, the CSI-PNet architecture can be used as a denoiser that boosts the Signal to Noise Ratio (SNR). This simplifies the training process for the CSI prediction solutions. For the TFO, a processing step is used to reduce their impact. Finally, we study the performance of the proposed methods against several benchmarks, using datasets from 3GPP-based system-level simulations as well as field measurements. In the simulation dataset, we observed that the CSI-PNet provides considerable gains at low and medium speeds, where the path correlation can be significant. In the field measurement dataset, the proposed methods outperform all benchmarks. In this dataset, we observed that enhanced preprocessing is especially important.

A. PRIOR WORK

1) TEMPORAL CSI PREDICTION

Feedforward NN and Recurrent NN (RNN) have been investigated for channel prediction for a long time [18]. Early works, such as [19] and [20], consider narrow-band single antenna systems, where [19] uses RNN and [20] uses CvNN. A recent work, [21], compares the performance of a Fully Connected NN (FCN)-based AI solution to the Vector Kalman Filter (VKF) in a narrow-band system. The authors observe that the performance of the two is comparable.

Recently, there has been an increased interest in AI-based CSI prediction in wide-band MIMO systems. For instance, [22], [23], and [24] propose using RNNs based solutions. Reference [25] uses CNN to identify the channel variation patterns, then adopts an AutoregRession Model (ARM) or RNN to predict the CSI. In [26], the authors propose a hybrid network model that uses CNNs to extract a compact representation of the CSI, which can be then used as an input to RNN for CSI prediction. In [27], the authors use a deep 3D-CNN with residual connection network architecture to predict the CSI. However, these studies, among others, track the channel in the space-frequency domain, which does not take full advantage of the sparsity and, more importantly, do not utilize the reduced Doppler spread in angle-delay domain,[1] thus making the CSI prediction harder. As we discuss in Sec. III-A, the angle-delay domain representation is a natural representation of a multi-path environment, inherently sparser, and ensures that the correlation of the channel entries is reduced to the entries (paths) in the vicinity of each other.

In a recent work, [29] addresses the problem of multi-step CSI prediction using a transformer-based network. The considered system model utilizes a combiner matrix based on the strongest Discrete Fourier Transform (DFT) beams for a Uniform Linear antenna Array (ULA) at the BS. Thus the CSI prediction can be assumed to be in the angle domain

[1]Note that sparsity alone might not always be advantageous as several popular ML models may struggle with sparse data, e.g., [28].

of an Orthogonal Frequency Division Multiplexing (OFDM) MIMO system. The selection of the strongest beams is based on prior full beam sweeping and relative stationarity of the angle of arrival/departure. However, [29] considers a narrow-band system. In addition, transformers can be challenging to scale, train and store. In another work, [16] proposes a method based on the SNR to identify the significant propagation paths in the angle-delay domain, which are then fed to a CvNN for a *per-path* prediction. However, unlike our work, the authors use an FCN network structure to process the *pre-selected* paths. Although path selection has some advantages, such as filtering gain and restricting the learning process to strong paths, it has several potential drawbacks. First, with path pre-selection, the relative path information is lost at the input of the ML model. Second, the Diffuse Multipath Components (DMC), which carry a significant percentage of the overall signal energy, are typically difficult to capture with thresholding techniques [30]. Third, path identification is also sensitive to antenna array model mismatch and the angle-delay mapping process. Lastly, the method in [16] depends on the accuracy and the stability of the SNR estimates. Our proposal efficiently utilizes *all* the paths in the channel while maintaining their neighborhood information. However, it is worth mentioning that per-path prediction does show more gains in the simulation setup at high speeds at the possible cost of the above aspects.

Finally, different from the above works, except for [24], we validate the performance of the proposed solution using a field dataset. In contrast to [24], the data is collected in a commercial network, which is typically impacted by real world impairments, such as noise and TFO.

2) CSI PREDICTION WITH ONLINE TUNING

There are a number of recent AI solutions that focus on enabling online weight updates. For instance, [31] proposes a CvNN network for per-path CSI prediction and introduces regularization when updating of the CvNN weights. In [25], the authors propose a temporal frame structure for updating the network. In [32], a meta-learning solution is proposed that permits a fast fine-tuning process in new environments. Although updating the networks ''on-the-fly'' is interesting and might have a significant future impact, the anticipated signaling overhead and processing load make the real-time model update very challenging in the current and near-future systems.[2] Thus, we do not focus on this aspect, despite the fact that the CSI-PNet is designed in a modular fashion that can be well suited for such implementations.

3) CSI PREDICTION WITH SIDE INFORMATION

In another set of works, side information is used to enhance the CSI prediction. For instance, [33] uses location, temperature, weather, and CSI history to predict the CSI values with CNN and Long Short-Term Memory (LSTM) networks. In [34], the authors propose a multimodal learning framework to leverage data such as user location along with a partially observed CSI for channel prediction. While using side information can enhance the prediction, the availability of such information is not always guaranteed. Thus, this paper focuses on the ''classical'' temporal CSI prediction with observable CSI history.

4) VARIATIONS OF CSI PREDICTION

In the literature, we have observed that the ''CSI prediction'' can sometimes be used to refer to frequency extrapolation in FDD system [35], CSI compression for CSI feedback, or CSI spatial prediction for given antenna or locations [36]. We emphasize that the goal of this work is *temporal* CSI prediction. While one can draw some similarities between the temporal CSI prediction and other works, their goals, system setup, and possibly the evaluation metrics are different. Furthermore, even for temporal CSI prediction, different papers can have different goals [37]. For instance, some works focus on the CSI amplitude and/or statistics, e.g., [38], or only some large-scale parameters, such as the path-loss variations [39]. We emphasize that our goal is to predict the small-scale fading, more precisely, the instantaneous complex CSI.

B. CONTRIBUTION AND PAPER STRUCTURE

This work aims to address the temporal small-scale CSI prediction in a wide-band MIMO system. The CSI prediction is performed in the angle-delay domain. The contribution of this work can be summarized as follows:

- We propose NN structures that can exploit the temporal and spatial correlation between propagation paths with varying model complexities (number of trainable parameters and Floating Points Operations Per Second (FLOPS)).[3] The designs allow the networks to capture such relations, especially at low to medium speeds when the impact of correlation is prominent.
- The proposed networks use 3D-CCNN and operate on the whole CSI realization resulting in a relatively small number of parameters while being robust to some preprocessing or model mismatch.
- We suggest further enhancements, including using the proposed architectures as denoiser before CSI prediction. This can improve the robustness to noise and simplify the training process for CSI prediction tasks.
- The performance of the solutions against several benchmarks are studied in two datasets: a synthetic channel model (3GPP channel model) simulation and field measurements in a commercial network. To the best of the authors' knowledge, this is among the few works that validate the AI-based temporal CSI prediction with

[2]In addition, the fine-tuning process requires support and *maturity* in the standards, which could take time to be reached.

[3]Note that these are important. In practical wireless communication systems, model storage, loading, and transfer are expected to be performed frequently. Furthermore, upper limits on inference complexity can be set, especially for battery-powered devices and/or multi-user systems.

field measurements and the first to study and report the result based on field data in a commercial network.

The remainder of the paper is as follows. After introducing the notations, Sec. II summarizes the system model. Sec. III discusses the motivation and the process of angle-delay transformation. The section also reviews autoregression-based CSI prediction. Next, in Sec. IV, we introduce the proposed network structures, CSI-PNet, ACSI-PNet, and how to use the solutions as denoisers. In Sec. V, training aspects, including the normalization, impact of TFO, and the loss functions, are discussed. Sec. VI summarizes the considered benchmarks and evaluation metrics that we use in the later sections. In Sec. VII and Sec. VIII, the simulation and field datasets are respectively introduced along with the evaluation results. Finally, Sec. IX provides concluding remarks.

C. NOTATION

Let $\mathbf{X}$ be a matrix, $\mathbf{x}$ be a vector, $\mathcal{X}$ be a function, and $\mathcal{C}$ and $\mathcal{R}$ be the complex and real spaces, respectively. We use $\mathbb{X}$ to denote tensors with U dimensions. For $U = 3$, $\mathbb{X} \in \mathcal{C}^{n_1 \times n_2 \times n_3}$, let the i^{th} dimension be d_i, then $\mathbb{X}_{d_3,d_1,d_2}$ refers to reordering the dimensions such that d_3 is first, followed by d_1 and d_2, which we also write as $\mathbb{X}_{d_3}$. Furthermore, let $\mathbb{X}_{d_3:}$ transform the tensor to a matrix of size $n_3 \times (n_1 n_2)$ with d_3 being the first dimension and the reset dimensions are nested d_1, d_2. Furthermore, $\mathbb{X}_{:}$ is a vector of size $n_1 n_2 n_3 \times 1$. Finally, let $\mathbb{X}_{d_1,d_2,d_3}(d_1 = i) = \mathbb{X}_{d_1,d_2,d_3}(i)$ be a tensor of size $n_2 \times n_3$, where the values in dimension d_1 are fixed at the i^{th} "slice", $i \in \{1, \ldots, n_1\}$.

II. SYSTEM MODEL

We consider a single-cell wireless system where a BS is equipped with a dual-polarized Uniform Planner Array (UPA) with $N_h \times N_v$ antennas in the horizontal and the vertical domain, respectively. The BS communicates with a single antenna UE, i.e., Multi-Input-Single-Output (MISO), but generalization for MIMO is also possible.

We assume an OFDM system with K_s subcarriers that are grouped into K (contiguous) "Resource Blocks" (RBs), each of which, for simplicity, contains one pilot tone (Reference Signal (RS)).[4]

We consider a discretized normalized time, and the goal is to predict the channel at time $t+1$ based on the current and previous estimates, i.e., up to time t. The time normalization and resolution are design parameters that may depend on the RS periodicity and the application of the CSI prediction, e.g., scheduling vs. beamforming [41]. In this work, we focus on a single-step prediction; multi-step is left for future work. Furthermore, we do not restrict the discussion to whether the CSI prediction is done for a downlink or an uplink, or whether the system is a Time Division Duplex (TDD) or an FDD system. Therefore the goal is to design an ML solution for *temporal* CSI prediction that can conceptually be used in any of the above cases. For instance, for a downlink of an FDD system, where channel reciprocity does not hold, the BS needs to rely on the CSI feedback from UE to determine its downlink data transmission scheme. During the downlink training stage, the BS sends pilots to the UE, e.g., using CSI RS ("CSI-RS") in 5G, based on which the downlink channel is estimated. The UE will then feed back either (i) the estimated downlink CSI or (ii) the predicted downlink CSI to the BS. For (i), the BS does the prediction based on the fed-back past and present CSI, while for (ii) the UE does the prediction. For TDD, the channel prediction can be done either at the UE or the BS; in the former case, the UE feeds back the predicted CSI to the BS. The prediction can also be done at the BS side, by relying on channel reciprocity and uplink CSI as obtained from uplink pilots, such as the Sounding RS (SRS) in 5G.

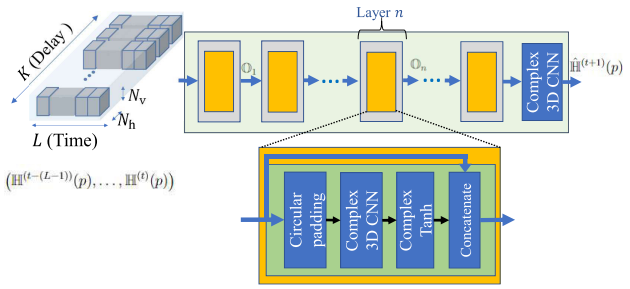


FIGURE 1. The structure of CSI-PNet (v1). In each layer (except the last) a 3D-CCNN followed by complex tanh and a concatenation process, the final layer has only 3D-CCNN layer for final path prediction and adaptation. The delay dimension is captured along the channel dimension of the CNN. Before each 3D-CCNN a circular padding can be performed. In each of the layers, the size of the used 3D kernel (n_t, n_v, n_h).

Let $\mathbb{G}^{(t)}_{f,p,v,h} \in \mathbb{C}^{K \times 2 \times N_v \times N_h}$ be the observed signal at time t at all frequencies, and all antennas for the two polarizations; we next, for brevity, drop the subscripts when they are clear from the context. A channel predictor takes in the past channel observations and outputs the future channel, that is

$$\widehat{\mathbb{G}}^{(t+1)} = \mathcal{P}_{\mathbf{\Theta}}\left(\mathbb{G}^{(t)}, \mathbb{G}^{(t-1)}, \ldots, \mathbb{G}^{(t-(L-1))}\right), \qquad (1)$$

where $\mathcal{P}_{\mathbf{\Theta}}$ is the channel predictor function parameterized by $\mathbf{\Theta}$, and L is the number of past channel observations used for prediction. Note that the true (unknown) mapping function $\mathcal{P}$ may rely on more or fewer previous channel observations to fully identify $\mathbb{G}^{(t+1)}$. The goal is to learn a mapping function $\mathcal{P}_\Theta$ for a given hyper-parameter L such that it closely resembles the true $\mathcal{P}$.

III. PRELIMINARIES

In this section, we first motivate and discuss the transformation into the angle-delay domain for CSI prediction. Next, we present a classical channel prediction technique based on the ARM.

[4] We emphasize that this is just one possible RS configuration, and the proposed solutions can be adapted to many others with minimal or no changes. Furthermore, when pilot sparsity is present, several techniques can be used for initial channel reconstruction, e.g., [40].

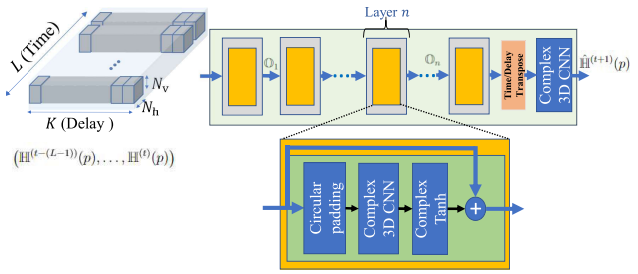


FIGURE 2. The structure of CSI-PNet (v2). The structure is similar to CSI-PNet (v1), except (i) the order of the dimensions of the input tensor is modified so that the time is placed at different channels, and (ii) the skip connection is used for a residual connection. In each of the layers, the size of the used 3D kernel (n_d, n_v, n_h), (iii) before the last layer the time and delay dimensions are swapped, such that the delay domain is placed at different channels (i.e., the last layer in (v2) matches the one in (v1)).

A. ANGLE DELAY DOMAIN REPRESENTATION

Angle delay CSI representation is a natural way to represent the wireless propagation channel, since each Multi-Path Component (MPC) is characterized by a distinct delay and angle. The MPCs are usually assumed to be independent, as per the generalization [15] of the well-known "WSSUS" assumption [42]. However, due to the limited system bandwidth and angular resolution, the MPCs from different clusters can fall within (and between) the adjacent angle delay bins, resulting in a residual correlation between the bins [3]. The correlation typically decreases with the increasing difference between the respective delays and/or angles of the MPCs. Furthermore, since the Doppler is related to the angular characteristics at the mobile link end, the evolution of each separate MPCs cluster may exhibit a reduced Doppler spread, which may simplify the CSI prediction. Thus, it is desirable to design a CSI prediction solution that can efficiently take full advantage of the separation between MPCs in the angle-delay domain while capturing the residual correlation between neighbor paths.

Given the received OFDM-MISO signal in the space (antenna) -frequency domain, we can transform it to the angle-delay domains through DFTs. For clarity of notation, we present that in two steps:

- Transforming $\mathbb{G}$ into the (discretized) delay domain, $\tilde{\mathbb{H}}$, using Inverse DFT (IDFT) (frequency to delay):

$$\tilde{\mathbb{H}}_{\tau:} = \mathbf{F}_K^H \mathbb{G}_{f:} \ , \tag{2}$$

  where τ denotes delay domain. $\mathbb{H}$ is the channel in the transform domain, $\mathbf{F}_X$ is $X \times X$ DFT matrix, and $(.)^H$ is the Hermitian transpose.
- To transform $\tilde{\mathbb{H}}$ from the spatial domain, sampled at the vertical $v \times$ horizontal h antennas, into the discretized elevation θ azimuth $\times\ \phi$ domains, we apply the 2D IDFT again for each polarization p and delay τ,

$$\mathbb{H}_{\tau,p,\theta,\phi}(\tau, p) = \mathbf{F}_{N_v}^H \tilde{\mathbb{H}}_{\tau,p,v,h}(\tau, p) \mathbf{F}_{N_h}^H. \tag{3}$$

In the following, we again drop the subscripts. With the angle-delay domain representation, the goal of the proposed ML solution is to learn $\mathcal{P}_{\boldsymbol{\Theta}}$ in the following:

$$\widehat{\mathbb{H}}^{(t+1)} = \mathcal{P}_{\boldsymbol{\Theta}} \left(\mathbb{H}^{(t)}, \mathbb{H}^{(t-1)}, \ldots, \mathbb{H}^{(t-(L-1))} \right). \tag{4}$$

B. ARM BASED CSI PREDICTION

ARM is one of the classical prediction solutions. An earlier study, [4], observed that using ARM in the angle-delay domain outperforms applying it in other domains. In this paper, we use ARM as one of the benchmark solutions (see Sec. VI). Furthermore, the ARM formulation can provide some insights into the dynamics of ML-based prediction solutions. Thus we summarize ARM-based CSI prediction below.

In ARM, for $p \in \{1, 2\}$, $\tau \in \{1, \ldots, K\}$, $\theta \in \{1, \ldots, N_v\}$, $\phi \in \{1, \ldots, N_h\}$, the predicted CSI at time $t+1$, based on P previous observations, is:

$$\widehat{\mathbb{H}}^{(t+1)}(\tau, p, \theta, \phi) = \sum_{l=1}^{P} \alpha^{(l)*}_{\tau,p,\theta,\phi} \widehat{\mathbb{H}}^{(t-l+1)}(\tau, p, \theta, \phi), \tag{5}$$

where $\alpha^{(l)*}_{p,\tau,\theta,\phi}$ is the regression parameter for given p, τ, θ, ϕ, and $(.)^*$ is the conjugate. Defining $\boldsymbol{\alpha} = [\alpha^{(1)}_{\tau,p,\theta,\phi}, \ldots, \alpha^{(P)}_{\tau,p,\theta,\phi}]^\top$, the parameters can be calculated using

$$\boldsymbol{\alpha} = \left(\mathbf{R} + \beta \mathbf{I} \right)^{-1} \mathbf{r}, \tag{6}$$

where $\mathbf{R}$ is the $P \times P$ correlation matrix between the P observations $\left\{ \widehat{\mathbb{H}}^{(t)}(\tau, p, \theta, \phi), \ldots, \widehat{\mathbb{H}}^{(t-P+1)}(\tau, p, \theta, \phi) \right\}$, $\mathbf{I}$ is the identity matrix, β is a coefficient that is typically set based on the noise variance, and $\mathbf{r}$ is the cross-correlation vector between the P observations and $\widehat{\mathbb{H}}^{(t+1)}(\tau, p, \theta, \phi)$. Note that the latter is not known; in Sec. VII, we use $P = L - 1$ and introduce a delay to calculate $\mathbf{R}$ and $\mathbf{r}$.

IV. PROPOSED SOLUTIONS

A. CSI-PNet

The data input, output, and structure of two variants of the proposed CSI-PNet are displayed in Fig. 1 and Fig. 2, we refer to them as CSI-PNet (v1) and CSI-PNet (v2), respectively. In the rest of the paper, when we refer to both designs, we use "CSI-PNet". In general, the operations and conclusions are valid to both variants unless otherwise stated.

The input tensors are first split into the two polarization, i.e., $\mathbb{H}_{\tau,p,\theta,\phi}(p = 1)$ and $\mathbb{H}_{\tau,p,\theta,\phi}(p = 2) \in \mathbb{C}^{K \times N_v \times N_h}$, then L tensors for a given polarization (after proper pre-processing, see Sec. V) are fed into the CSI-PNet. Before we discuss the details of the two variants, we start by explaining the core building block of the networks.

1) CCNN

The main building block of the proposed DL model is a 3D-CCNN layer. While CvNNs may be natural for treating wireless channels, they have received far less attention in the general ML literature than real NNs. The CCNN consists of pairs of real-valued CNNs; we demonstrate that relation with a simple single layer 2D CNN, with one input channel of size

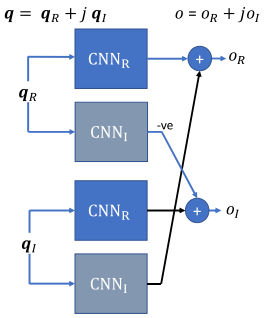


FIGURE 3. CCNN consists of two CNNs, one that represents the real weights CNN_R and the other represents the imaginary weights CNN_I.

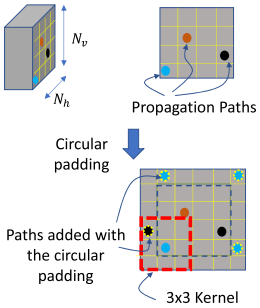


FIGURE 4. Example of the circular padding of size one for a 4 × 4 angle domain. After padding and convolution, the output value for each path (at the center of the kernel) considers neighboring paths.

$N_h \times N_v$ and using one kernel of size $n_h \times n_v$. Then a CNN convolution operation can be described by:

$$o^i = \mathbf{w}^\top \mathbf{q}^i,$$

where o^i is a scalar output at the i^{th} convolution step, $\mathbf{q}^i$ is a vector of size $n_h n_v \times 1$ with values corresponding to the input of the "image" that is covered by the kernel in the i^{th} convolution operation, and $\mathbf{w}$ is a flattened kernel weight of size $n_h n_v \times 1$. The CCNN generalizes the above such that $\mathbf{w} = \mathbf{w}_R + j\mathbf{w}_I$ and $\mathbf{q}^i = \mathbf{q}^i_R + j\mathbf{q}^i_I$, where subscripts R and I are, respectively, used to identify the real and imaginary components, and $j = \sqrt{-1}$. Then, we have

$$\begin{aligned} o^i &= o^i_R + jo^i_I = \mathbf{w}^H \mathbf{q}^i \\ &= \mathbf{w}^H_R \mathbf{q}^i_R + \mathbf{w}^H_I \mathbf{q}^i_I + j(\mathbf{w}^H_R \mathbf{q}^i_I - \mathbf{w}^H_I \mathbf{q}^i_R). \end{aligned}$$

It is straightforward to see that we can describe the CCNN as shown in Fig. 3.

2) CSI-PNet (v1)

In this structure, the delay domain is carried over different "channels" (in CNN terminology). The time domain (of size L) and the angle domains (of size N_v and N_h) form the 3D complex tensor. The output for the given polarization is the predicted $\hat{\mathbb{H}}^{(t+1)}_{\tau,p,\theta,\phi}(p)$. The other polarization is processed similarly, where we use weight sharing for the two polarizations, and thus the training and the inference use the same network weights; note that the inference can possibly be done in parallel with an identical network.

The structure of CSI-PNet is as follows. For the first five layers, each 3D-CCNN is followed by a tanh activation function, applied, separately, to the real and the imaginary components, which we refer to as Complex tanh (Ctanh). This is followed by a concatenation block. The input to the network is $\{\mathbb{H}^{(t-L+1)}(p), \ldots, \mathbb{H}^{(t)}(p)\}$. The first 3D-CCNN uses a 3D *kernel* of size (n_t, n_v, n_h), the $n_t = L$, i.e., matching the observed CSI in the buffer, while the n_v and n_h are hyper-parameters that can be selected based on the anticipated correlation of the paths in the angle domain, their impact to be discussed in Sec. VII-B.

Let the output of the n^{th} layer be $\mathbb{O}_n$. Throughout the network, the dimensions of the delay domain and the angle domain are maintained. This is achieved in the delay domain and the angle domain, respectively, by maintaining the number of channels and a proper choice of padding size. Given the circular nature of the angle domain (the DFT beam basis), we propose using circular padding. This is illustrated in Fig. 4: with circular padding, the per path prediction uses the neighbor paths within $\lfloor \frac{n_v}{2} \rfloor$ and $\lfloor \frac{n_h}{2} \rfloor$ in vertical and horizontal dimensions, respectively, where $\lfloor . \rfloor$ refer to flooring operation. Thus the correlation between adjacent bins is utilized. CSI-PNet parameters and input-output sizes are summarized in Tab. 2. In the table, B denotes the batch size. Note that due to the concatenation process, the size $\mathbb{O}_n$ grows in the time domain.

Finally, while not discussed in detail in the paper, we point out the modular property of the CSI-PNet. The skip connections that carry the input throughout the network help speed up the convergence during training and improve parallelism, where many corresponding values can be evaluated in parallel (applies mainly for (v1)). In addition, the last layer is designed such that it may be fine-tuned (in a transfer learning fashion) to adapt to different channel properties. For that, the layer is designed to be lightweight (based on CNN structure) and takes as input the extracted features from previous layers as well as the raw CSI. However, this is not studied further in this paper.

3) CSI-PNet (v2)

In general, the structure of the second variant of CSI-PNet is similar to the CSI-PNet (v1) except (i) the delay and the time at the input are swapped, where the time domain is carried over different channels, and the delay and the angle domains form 3D tensors. The 3D kernel in each 2D-CCNN has size (n_d, n_v, n_h), where n_d is the filter size in the delay domain. (ii) The concatenation steps are replaced with residual steps, where the output of the previous layer, $n-1$, is added to the output of layer n (after the activation). (iii) Before the last layer, a transpose operation is used to swap the delay and time domains such that the last layer matches the one in (v1). Fig. 2 shows an illustration of CSI-PNet (v2). This design provides a reduced complexity while preserving most of the gains, as we will explain later.

TABLE 1. Parameters of the CSI-PNet (v1).

Module	Parameter	Value
3D-CCNN (n^{th} Layer)	Dimension of input & Output	$(B, K, L+n-1, N_{\text{v}}, N_{\text{h}})$ & $(B, K, 1, N_{\text{v}}, N_h)$
	Kernel size & Padding & Stride	$(L+n-1, n_{\text{v}}, n_{\text{h}})$ & $(0, \lfloor \frac{n_{\text{v}}}{2} \rfloor, \lfloor \frac{n_{\text{h}}}{2} \rfloor)$ & $(1, 1, 1)$
Concatenation Block (n^{th} Layer)	Dimension of first input & Second input	$(B, K, 1, N_{\text{v}}, N_{\text{h}})$ & $(B, K, L+n-1, N_{\text{v}}, N_{\text{h}})$
	Dimension of the output	$(B, K, L+n, N_{\text{v}}, N_{\text{h}})$

TABLE 2. Parameters of the CSI-PNet (v2).

Module	Parameter	Value
3D-CCNN (n^{th} Layer)	Dimension of input & Output	$(B, L, K, N_{\text{v}}, N_{\text{h}})$ & $(B, L, K, N_{\text{v}}, N_h)$
	Kernel size & Padding & Stride	$(n_{\text{d}}, n_{\text{v}}, n_{\text{h}})$ & $(\lfloor \frac{n_{\text{d}}}{2} \rfloor, \lfloor \frac{n_{\text{v}}}{2} \rfloor, \lfloor \frac{n_{\text{h}}}{2} \rfloor)$ & $(1, 1, 1)$
3D-CCNN (Last Layer)	Dimension of input & Output	$(B, K, L, N_{\text{v}}, N_{\text{h}})$ & $(B, K, 1, N_{\text{v}}, N_h)$
	Kernel size & Padding & Stride	$(L, 1, 1)$ & $(0, 0, 0)$ & $(1, 1, 1)$

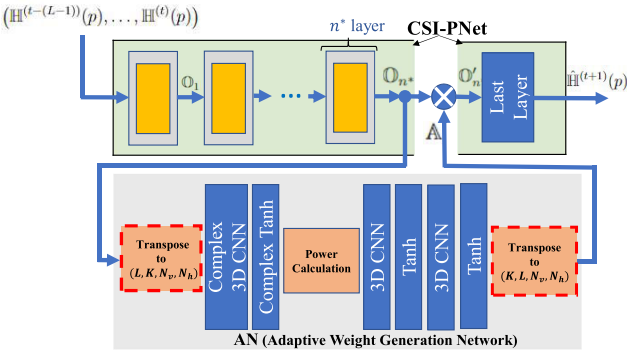


FIGURE 5. The ACSI-PNet architecture, consisting of an architecture similar to CSI-PNet plus an AN (Adaptive Weight Generation Network). The red dashed blocks are used for CSI-PNet (v1) only.

B. ACSI-PNet (ADAPTIVE CSI-PNet)

The CSI-PNet, as described above, relies on fixed weights similar to other AI-based CSI prediction solutions. However, similar to ARM, the coefficients can be designed to be adaptive to channel realizations.[5] Specifically, the relative importance of different paths in the L CSI realizations and their contribution to future CSI can vary widely. To address this challenge, we introduce an Adaptive weight generation Network (AN) into the CSI-PNet architecture. Using the ML terminology, the network provides the "attention" mechanism that allows the model to judiciously focus on and weigh the features and paths. However, different from the conventional attention weights that usually take values in $[0, 1]$, the adaptive weights here can take values in $[-1, 1]$, as we explain below.

[5]Note that in ARM the coefficients are function of the second order statistics, which vary over long time intervals (multiple stationarity times) [43].

The ACSI-PNet structure is shown in Fig. 5. Let the complex valued input tensor to the AN, $\mathbb{O}_{n^*}$, be the output of a CSI-PNet layer n^*, which consists of the transformed time domain features (along with the concatenated input in (v1)). Note that in CSI-PNet (v1), the size of the input depends on the position of the AN: While the output of each CSI-PNet layer is fixed, the size of $\mathbb{O}_n$ grows due to the concatenation process. In Sec. VII, we use six layers for CSI-PNet, and the AN is positioned after the fifth layer, i.e., $n^* = 5$.

AN is a multi-layered network. The first layer is a simple transpose operation if CSIP-Net (v1) is used, where the delay and the time are swapped. The number of channels is equal to the depth of the last layer. Then, a layer similar to the ones in CSI-PNet is used with a filter of size $(n_\tau, n'_v, n'_h) = (7, 5, 5)$ with a circular padding of size $(3, 2, 2)$. After Ctanh, to convert the output to real values, the power values are calculated, where the squares of the real and imaginary components are added. Next, the network uses two 3D-CNN layers with tanh activation, with kernels of size three and one, respectively. A transpose operation is then used to swap the delay and time dimensions (for CSI-PNet (v1)). The ACSI-PNet parameters in the evaluation sections are summarized in Tab. 3. Note that throughout the network, the output and input have equal sizes. However, the output is a *real-valued* tensor.

The output of the network is used for element wise product. Let $\mathbb{A}$, be the output of the AN and $\mathbb{O}'_{n^*}$ be the *input* to the $n^* + 1$ CSI-PNet layer, then

$$\mathcal{R}\{\mathbb{O}'_{n^*}\} = \mathcal{R}\{\mathbb{O}_{n^*}\} \odot \mathbb{A}, \text{ and } \mathcal{I}\{\mathbb{O}'_{n^*}\} = \mathcal{I}\{\mathbb{O}_{n^*}\} \odot \mathbb{A},$$

where $\odot$ is the element wise product, and the operators $\mathcal{R}\{.\}$ and $\mathcal{I}\{.\}$ extract the real and imaginary components of the tensor, respectively. Here the same weights are used for the real and imaginary components. Note that different from the conventional attention, due to the tanh activation, the output of the AN takes values in $[-1, 1]$, which we observed to give better performance than Softmax or non-negative

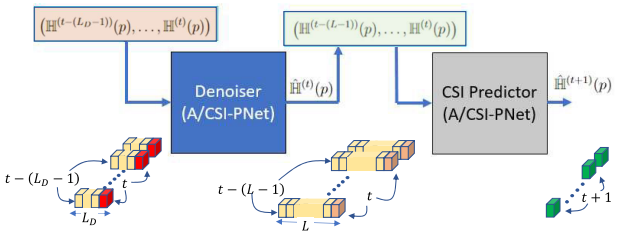


FIGURE 6. The overall flow with a denoiser. CSI-PNet or ACSI-PNet can be used as denoiser and/or for CSI prediction. With some abuse of notation, the output of the denoiser is added to the input buffer of the CSI predictor.

Sigmoid activation functions. In general, to develop this network, several other designs have been tested, e.g., the location of the input and output of AN in the CSI-PNet, the normalization of the power calculation, the type of activation functions, and whether the output of AN is different for the real and imaginary parts. However, when evaluated on the synthetic data, these variations showed marginal or no gains compared to the proposed version.

C. DENOISER

The above networks are proposed for CSI prediction, where the output CSI is at a future time instant. However, this prediction can be impacted severely by noise on the measured CSI. While it is possible to train the network for CSI prediction as well as handling the impact of noise, this could complicate and prolong the training process, and could deteriorate the overall performance. In this subsection, we propose using a ACSI-PNet architecture as a denoiser, where the output is $\mathbb{H}^{(t)}$. See Fig. 6 for the overall flow when a denoiser is added. Different from the actual predictor, we can set the buffer of the denoiser, L_D, to be smaller than L (i.e., $L_D < L$). This is motivated by the fact that one way to reduce the impact of noise can be with simple short-term smoothing. In Sec. VII, we use $L_D = 3$, so that the buffer takes the current CSI and two previously observed CSI values. Finally, we point out that other techniques to improve the channel estimation, e.g., [44] and [45] can be used as well. The comparison between these techniques is out of the scope of the paper.

V. PREPROCESSING, TRAINING AND LOSS FUNCTION

In this section, we discuss the training aspects of the models, including preprocessing and the loss function. As will be elaborated in Sec. VII, we use channel traces from several users drops. Let Q denote the number of channel realizations per UE (i.e., per trace). Then we construct $Q-L$ short channel sequence segments for the u^{th} UE, that is

$$\mathcal{D}_u = \left\{ \left(\mathbb{H}^{(s-(L-1))}, \ldots, \mathbb{H}^{(s-1)}, \mathbb{H}^{(s)}; \mathbb{H}^{(s+1)} \right)_{s=L}^{Q-1} \right\}, \quad (7)$$

where, for each "point" in $\mathcal{D}_u$, the first L channel tensors are the input of the network, and the last channel tensor is used as desired output, i.e., training target. Then, the dataset $\mathcal{D}$ is the collection from all UEs, i.e., $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots\}$. As typically done, to enhance the training and learning process, the dataset is normalized. In this work, we normalize the input sequence and the target output with the root mean square of the power of the input sequence V_s, where,

$$V_s = \sqrt{\frac{1}{LKN_vN_h} \sum_{l=0}^{L-1} \left\| \mathbb{H}^{(s-l)} \right\|^2}. \quad (8)$$

The $\| \cdot \|^2$ operator for a tensor $\mathbb{X}$ is used here to denote the l_2-norm square of $\mathbb{X}_{:}$. Note that calculating V_s requires only the knowledge of the input values.

A. TFO (TIME AND FREQUENCY OFFSET)

In commercial field operations, UEs will have TFO with respect to the BS due to imperfect synchronization. Thus, *field* data are typically impacted by *random* TFO, and thus in reality the *received* signal (at a given frequency, polarization, and vertical and horizontal antenna indices) can be written as follows[6]:

$$\tilde{\mathbb{G}}^{(t)}(f, p, v, h) = \mathbb{G}^{(t)}(f, p, v, h) e^{j(\psi_i - 2\pi f \tau_i)}, \quad (9)$$

where ψ_i is a random phase offset that is uniformly distributed in $[0, 2\pi)$, $\tau_i = cT_s$, with c being a random timing offset uniformly distributed in $[-c_{max}, c_{max}]$ and $T_s = \frac{1}{W}$, where c_{max} is a real number that depends on several factors including the clock, W is the bandwidth of the system. To combat that, we can employ one of the TFO correction techniques, e.g., [46] and [47]. Here, we use a simple preprocessing step that reduces the TFO impact.

The importance of such a preprocessing step is evident from the field data evaluation section (Sec. VIII). The solution is based on a grid search that maximizes the correlation between the received signals at $t-1$ and t.

$$\psi_i^*, \tau_i^* = \arg\max_{\psi_i, \tau_i} \| \tilde{\mathbb{G}}^{(t-1)} \odot \tilde{\tilde{\mathbb{G}}}^{(t)H} \|^2,$$

where $\tilde{\tilde{\mathbb{G}}}^{(t)}(f, p, v, h) = \tilde{\mathbb{G}}^{(t)}(f, p, v, h) e^{j(-\psi_i + 2\pi f \tau_i)}$. Once ψ_i^*, τ_i^* are identified, the TFO can be eliminated from t. The process is carried on for the sequences of interest. This procedure also resembles the correlation step between pilot symbols which is commonly used for TFO tracking [46]. Note that other techniques can be used as well for initial TFO estimate and tracking.

B. TRAINING AND THE LOSS FUNCTION

To learn the channel predictor, $\mathcal{P}_{\Theta}$, we pose our learning problem as a supervised regression problem. We choose the normalized mean square error (NMSE) as the training loss function. Note that although typically, the mean square error (MSE) is used in such a problem, we noticed NMSE shows better convergence in a number of cases while training. For

[6]This a slight deviation from the notation of the received signal in Sec. II.

TABLE 3. Parameters of the AN in ACSI-PNet.

Module	Parameter	Value
3D-CCNN (After the n^{th} CSI-PNet Layer)	Dimension of input & Output	$(B, L+n-1, K, N_{\text{v}}, N_{\text{h}})$ & $(B, L+n-1, K, N_{\text{v}}, N_{\text{h}})$
	Kernel size & Padding & Stride	$(7,5,5)$ & $(3,2,2)$ & $(1,1,1)$
3D CNN (1^{st} Layer)	Dimension of input & Output	$(B, L+n-1, K, N_{\text{v}}, N_{\text{h}})$ & $(B, L+n-1, K, N_{\text{v}}, N_{\text{h}})$
	Kernel size & Padding & Stride	$(3,3,3)$ & $(1,1,1)$ & $(1,1,1)$
3D CNN (2^{nd} Layer)	Dimension of input & Output	$(B, L+n-1, K, N_{\text{v}}, N_{\text{h}})$ & $(B, L+n-1, K, N_{\text{v}}, N_{\text{h}})$
	Kernel size & Padding & Stride	$(1,1,1)$ & $(0,0,0)$ & $(1,1,1)$

each predicted channel sample, the loss function is defined as

$$\mathcal{L}\left(\mathbb{H}^{(s+1)}, \widehat{\mathbb{H}}^{(s+1)}\right) = \frac{\left\|\mathbb{H}^{(s+1)} - \widehat{\mathbb{H}}^{(s+1)}\right\|^2}{\left\|\mathbb{H}^{(s+1)}\right\|^2}. \quad (10)$$

Thus the objective is to find a set of model parameters that minimize the loss averaged over the whole dataset, which can be expressed as $\min_{\boldsymbol{\Theta}} \frac{1}{N}\sum_{s=1}^{N} \mathcal{L}\left(\mathbb{H}^{(s)}, \widehat{\mathbb{H}}^{(s)}\right)$, where N is the size of the dataset, the details of which will be given in the next section, and $\boldsymbol{\Theta}$ is the vector of model parameters to be optimized. Finally, for simplicity, we use Adam optimizer, since the combination of CCNN and Ctanh can still be viewed as a particular combination of a real CNN as illustrated in Fig. 3. Other training parameters are listed in Tab. 4. We used windowed learning rate drop (see Tab. 4) and early stoppage based on the validation loss to combat overfitting and speed up the training process. The validation loss is calculated using the training cost function. During the training, we save the model with the smallest validation loss, which we later use when evaluating the performance of the models over the test data. To train the models, we used an Nvidia GPU, model Quadro RTX 6000.

VI. BENCHMARK SOLUTIONS AND EVALUATION METRIC

A. THE BENCHMARKS

To assess the relative performance of the proposed methods, in this paper, we consider the following benchmarks.

- The CSI prediction network in [27] (3D-CNN with residual connection), we refer to it as *3DCNN*. As per [27], the model is trained with MSE loss function.
- The CvNN with adaptive significant path selection [16], we refer to it as *CvNN*. As per [16], the model is trained with Mean Absolute Error (MAE) loss function.
- A model we here refer to as *LSTM+FCN*, consists of one LSTM layer that is applied separately to the two polarizations, followed by a FCN. The FCN is applied to the concatenation of the two polarizations. The input to the network is the transformed angle-delay domain signal and is trained using NMSE loss function. This is a variation of RNN-based solutions such as in [24] and other works discussed in Sec. I-A. We introduced such variation as we observed that LSTM plus FCN structure shows better performance than pure RNN or LSTM networks. However, this comes at the cost of increased number of network parameters.
- The ARM presented in Sec. III-B. As indicated, we use $P = L-1$ for fair causal implementation. The statistics in (6) are calculated based on running average.
- Sample and Hold (SH): Using $\widehat{\mathbb{H}}^{(s+1)} = \mathbb{H}^{(s)}$ is a simple solution that is used in practice for slowly varying channels.

In addition to the fact that these benchmarks include some of the state-of-the-art ML-based solutions, we chose them to capture three main aspects: (1) the impact of real-valued 3D-CNN network structures (with [27]), (2) the impact the complex path based prediction (with [16]), (3) the performance against RNNs, which are usually suitable for sequential data (with LSTM) and have been used extensively for CSI prediction (see Sec. I-A). Here we also use SH as a relative benchmark since it (i) reflects the speed of channel variation and (ii) is the default choice in practice.

Finally, in Tab. 5, we present the number of parameters and FLOPS used for every model. In the evaluation sections, for CSI-PNet we use $n_v = n_h = k \in \{1, 3\}$, i.e., we use equal angular filter values for the vertical and the horizontal array dimensions. For CSI-PNet (v2) we use $n_d = 3$. Note that our implementations of CvNN and 3DCNN are based on the reported structures in the papers, i.e., no further optimization was made. For "Denoiser + ACSI-PNet", we use CSI-PNet (v1), the Denoiser part uses $L_D = 3$, while for the other part (ACSI-PNet for the CSI prediction), we use $L = 8$. We here choose L_D and L such that the total number of used CSI samples at any time is equal to 10. The number of CSI-PNet layers for both parts (the denoiser and the predictor) is six, and this can be optimized further, especially for the denoiser, but that is omitted for brevity.

The CSI-PNet with any of the considered k values requires a relatively small number of parameters and FLOPS for both (v1) and (v2); (v2) uses a much smaller number of parameters and requires almost half of the number of FLOPS. It is worth noting that incorporating AN into the model, i.e., using ACSI-PNet, has a greater impact on the number of FLOPS than on the number of parameters. The addition of a denoiser with the above buffer size results in a slight increase in the number of parameters compared to ACSI-PNet alone. Nevertheless, when compared to the benchmarks,

TABLE 4. Hyper-parameters for model training.

Parameter	Value
B (Batch size)	512
Number of epochs	150
Initial learning rate	1×10^{-3}
Learning rate schedule	$0.1@[70, 110, 140]$

the proposed solutions have a smaller number of parameters and FLOPS. For instance, the CvNN uses 30 times as many parameters; the number of FLOPS depends on the number of significant paths, which in turn depends on the environment and the SNR; in our studies in Sec. VIII we have noticed that it can easily exceed 1000Million FLOPS. In general, these designs could indicate the possible utilization of the proposed models at either BS or UE side.

B. THE EVALUATION METRIC

Since two of the benchmarks are trained using different metrics (see Sec. VI), which are different from the metric we use (NMSE), and following the 3GPP recommendations, we consider an evaluation metric based on the cosine similarity. This quantity is useful for, e.g., beamforming, as a common phase rotation does not impact the correlation and the beamforming. In this paper, we use an equivalent metric that signifies the variations in the correlation values. In particular, let

$$\rho(\mathbb{H}^{(s)}, \widehat{\mathbb{H}}^{(s)}) = \frac{\|\mathbb{H}_{:}^{(s)^H} \widehat{\mathbb{H}}_{:}^{(s)}\|}{\|\mathbb{H}_{:}^{(s)}\| \|\widehat{\mathbb{H}}_{:}^{(s)}\|}, \tag{11}$$

be the correlation value. We define and use the "effective" SNR (eSNR) metric:

$$eSNR(\mathbb{H}^{(s)}, \widehat{\mathbb{H}}^{(s)}) = 10 \log_{10}(1/(1 - \rho(\mathbb{H}^{(s)}, \widehat{\mathbb{H}}^{(s)}))). \tag{12}$$

VII. EVALUATION WITH SIMULATION DATASET

A. CHANNEL MODEL AND DATASET

In this section, we describe the adopted channel model, the dataset, and the training parameters used in our simulations.

We consider a MISO-OFDM system where the BS is equipped with a dual-polarized UPA with $N_h = N_v = 4$. For the purpose of CSI prediction, we assume that $K = 48$ RBs over $W = 10$ MHz channel bandwidth are available, i.e., representing 180 kHz spacing between the utilized pilot subcarriers. The channel coefficients are generated by a system-level simulator that follows the 3GPP Urban Macro (UMa) channel model. The carrier frequency is 2.3 GHz. Only non-line-of-sight (NLOS) scenarios are considered. 260 UEs are dropped randomly in a cell with 1 km radius. About 60 UEs are used for testing. In addition, 10% of the UEs in the training dataset are dedicated to validation (for the early stoppage and model selection).

The UE speed, the total duration of a trace, the pilot periodicity, and the SNR vary depending on the scenario discussed. However, for most parts, the pilot periodicity is assumed to be 10 ms and 5 sec traces per UE, and each is generated with three different levels of UE mobility, low, medium, and high, respectively, denoting 3, 10, 30 kmph. Also, unless explicitly mentioned, in all the simulations, 20 dB SNR is maintained.

In this study, for brevity, we restrict the reported results to $L = 10$. For this data, absence of TFO was assumed, and thus only normalization is used as per Sec. V.

TABLE 5. Number of trainable parameters and FLOPS.

Solution	Trainable Parameters (Mill.)	FLOPS (Mill.)
CSI-PNet (v1/v2, k=1)	0.35/0.049	22/12
CSI-PNet (v1/v2, k=3)	3.1/0.073	199/86
ACSI-PNet (v1/v2, k=1)	0.43/0.087	274/124
Denoiser + ACSI-PNet (v1, k=1)	0.54 (0.19+0.35)	371 (208+163)
CvNN	17.9	36 (*per path*)
DCNN	5.98	7374
LSTM+FCN	28.3	387

B. SIMULATION RESULTS

We start with scenarios similar to [16]. Fig. 7 shows the Cumulative Distribution Function (CDF) of the eSNR gains to SH for 5 kmph and 20 ms CSI pilot periodicity and 30 kmph with 4 ms pilot periodicity. In these studies, we focus on CSI-PNet (v1) and (v2) with kernel sizes $k = 3$ and $k = 1$, and ACSI-PNet with $k = 1$ as well as the benchmarks.

In general, while all the networks show gains over SH, all three proposed networks, along with CvNN, outperform the other benchmarks. For both speeds, the proposed networks show comparable performance to CvNN, with slight advantages at the higher percentile at the lower speed, but CSI-PNet (v1) shows small loss at low percentiles for both kernel sizes of CSI-PNet.

1) VARYING SPEED WITH FIXED PERIODICITY

To understand and emphasize the capabilities of CSI-PNet to capture the residual correlation, we here vary the speed and CSI pilot periodicity.

Considering a CSI pilot periodicity at 10 ms, we use low speed in Fig. 8 and medium and high speeds in Fig. 9. In Fig. 8, the proposed networks outperform the CvNN. Furthermore, CSI-PNet (v1) and (v2) with $k = 3$ outperform the ones with $k = 1$. We conjecture that this can be attributed to the added capability to capture the residual correlation with $k = 3$. Using an adaptive weight in ACSI-PNet, provides a small but visible gain in this case. In comparison between (v1) and (v2), we notice that (v1) shows a small gain over (v2).

In general, we also see that not all AI solutions outperform a simple SH. For instance, while the 3DCNN results in good

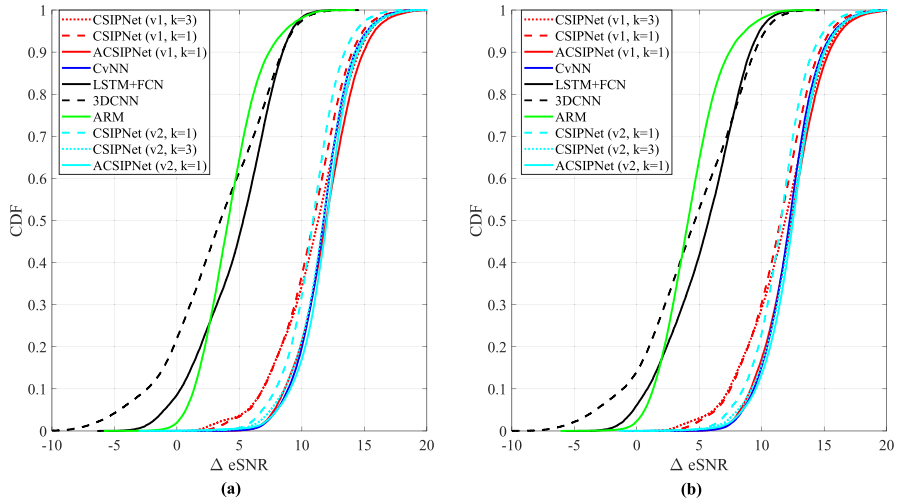


FIGURE 7. The CDF of the eSNR gain over SH for different speeds: (a) 5 kmph with 20 ms, (b) 30 kmph with 4 ms.

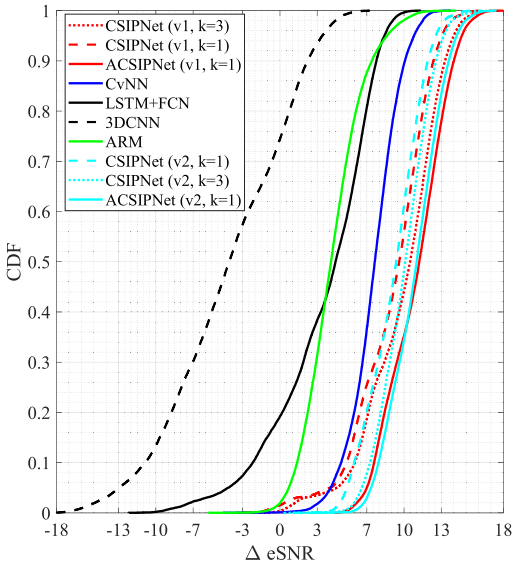


FIGURE 8. The CDF of the eSNR gain over SH for 3 kmph with 10 ms.

eSNR values (not shown), it still falls short of SH which, at this low speed, does well. The two other benchmarks show some gains compared to SH. The reason for this is that for AI solutions, there is a statistical learning aspect that depends on the training process, training set size, the capability of the model, etc., [48].

Next, in Fig. 9 we consider higher speeds for the same CSI pilot periodicity. For medium (10 kmph) speed, Fig. 9-(a), the performance is similar to the 5 kmph case, where we observe some gains for the proposed networks over CvNN, and that all AI solutions outperform the SH. However, as the speed increases, as shown in Fig. 9-(b), we observe increased losses versus SH for some AI techniques due to the inherent difficulty of predicting CSI accurately at higher speeds. Interestingly, we also note a visible advantage for CvNN over the other AI techniques and a reduction in the performance gap between the proposed architectures and the ARM approach. One possible explanation for the improved performance of CvNN lies in its simplified CSI prediction problem, which focuses on learning the prediction per path and prioritizes significant paths that usually suffer low noise. Overall, the gains of CvNN should be considered in light of the limitations discussed earlier in Sec. I-A. In comparison, between the (v1) and (v2) networks, networks based on (v2) show more resilience to speed than (v1). For (v2), $k = 3$ still provides some gain over $k = 1$, and all (v2) networks have about $1.6 \sim 1.9$ dB loss to CvNN, while (v1) networks have larger loss at the 50^{th} percentile ($1.9 \sim 3$ dB).

2) MIXED SPEEDS TRAINING AND SPEED MISMATCH

Next, considering the impact of high speed on learning, we explore training on mixed speeds for the same CSI pilot periodicity. In particular, we train the networks on UE speeds that are randomly generated in the range [2.5,50] kmph and then test on a medium speed (10 kmph). The results are presented in Fig. 10. Again, due to the presence of high speeds, we notice an enhanced gain for CvNN (compare to Fig. 9-(a)). Furthermore, the performance gains of ACSIPNet (both (v1) and (v2)) can be attributed to the design flexibility through the attention capabilities. The discussion above regarding $k = 1$ and $k = 3$ is also valid for this case.

In Tab. 6, we studied the impact of training on one high speed and testing on lower speeds for 10 ms pilot periodicity. The table also shows the case when the training and testing are equal. We observe that the proposed solutions show good performance when training with UE medium speed (10 kmph) and tested on low UE speed (3 kmph). However, they struggle when trained on higher UE speed and tested on UE lower speed. This supports the above observation.

Note that in this study, we focused on training on "high" speed and testing on a lower speed since (i) The mobility speed defines the *maximum* Doppler. Thus it is possible to

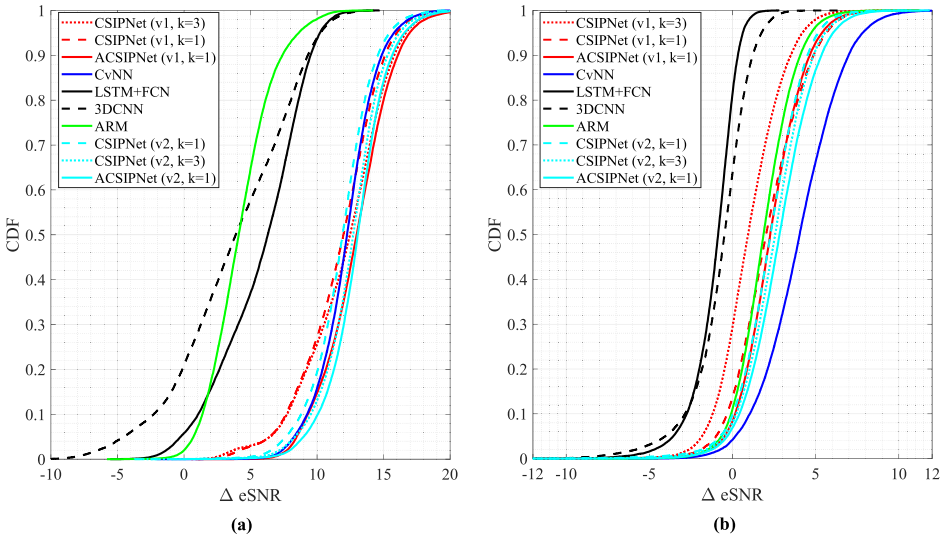


FIGURE 9. The CDF of the eSNR gain over SH for (a) 10 kmph (b) 30 kmph, both at 10 ms.

TABLE 6. CSI prediction performance, eSNR [dB] with speed mismatch.

Train Speed	Test Speed	SH	CSI-PNet (v1,k=1)	CSI-PNet (v1,k=3)	ACSI-PNet (v1,k=1)	ACSI-PNet (v2,k=1)	CvNN	LSTM+FCN	3DCNN	ARM
3	3	15.55	26.12	26.87	27.47	27.15	23.66	20.92	11.09	19.81
10	3	15.55	21.2	21.39	23.01	22.37	22.44	15.77	11.16	19.81
10	10	6.48	19.43	19.84	20.28	19.81	19.02	13.46	10.5	11.26
30	10	6.48	8.88	5.33	8.05	9.12	14.99	3.05	2.88	11.26
30	30	3.23	5.49	4.54	5.49	5.94	7.58	2.58	2.75	3.49

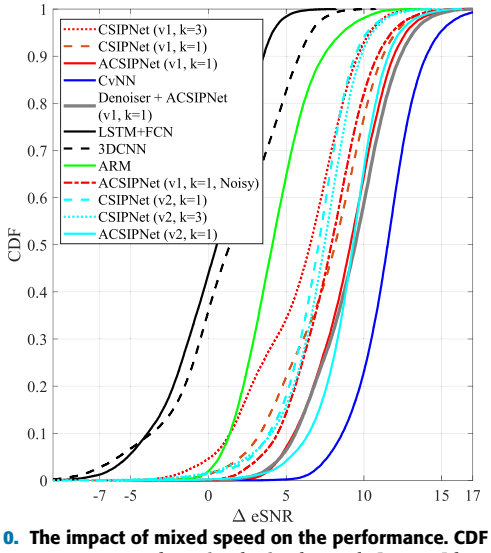


FIGURE 10. The impact of mixed speed on the performance. CDF of the eSNR gain over SH. Networks trained mixed speeds [2.5, 50] kmph and test on medium speed.

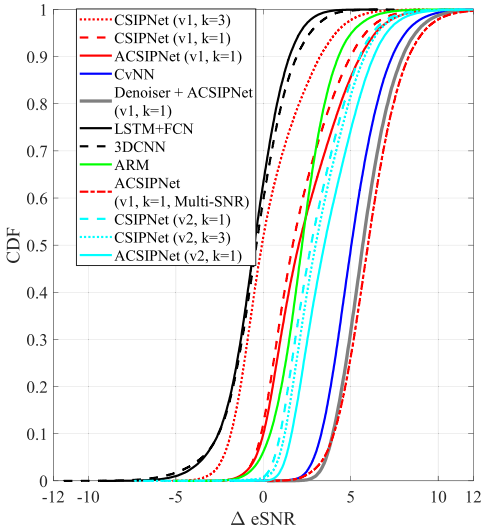


FIGURE 11. The impact of noise on scenario of Fig. 10. CDF of the eSNR gain over SH. Networks trained mixed speeds [2.5, 50] kmph and 20 dB SNR. The denoiser is trained on mixed speeds and mixed SNR values.

have paths for the same UE observing *up* to the maximum Doppler. (ii) Generalization to higher speed can possibly be achieved with some variations on the prediction setup, but that is out of the scope of the current paper.

3) IMPACT OF NOISE WITH MIXED SPEED TRAINING

To investigate the impact of noise on the performance of our networks, we use the scenario of the previous sub-section. However, for this case, we evaluate the networks in the low

TABLE 7. CSI prediction performance for CSI-PNet (v1, k=1), with noisy input *and* output. Training and testing speed = 10 kmph.

SNR_{in}/SNR_{out} [dB]	$20/\infty$	20/50	20/20	10/10
eSNR (σ_{eSNR})	19.4(2.4)	19.4(2.5)	19.5(2.7)	17.3(2.5)

SNR regime (SNR in [0,15] dB). The results are presented in Fig. 11. It should be noted that in this scenario, the SH itself is also affected by the presence of noise, making it difficult to draw a direct comparison between the results of Fig. 10 and Fig. 11. Nonetheless, analyzing the relative performance of the tested AI solutions can still provide meaningful insights. In Fig. 11, the gains of all the AI solutions are reduced, including our proposed solutions. Losses are observed for up to the 70^{th} percentile for LSTM and 3DCNN benchmarks. Regarding our proposed networks, CSI-PNet (v1) with $k = 3$ shows the worst performance among the variants, and other (v1) networks exhibit similar performance to ARM; however, (v2) shows again some resilience to noise, especially the ACSI-PNet that also shrinks the gap to CvNN. One factor that helps CvNN's ability to preserve some of its gains in this study can be attributed to the SNR filtering that is integrated into the path selection process.

Utilizing a denoiser architecture in combination with ACSI-PNet (v1) leads to further improvements in the network's performance compared to CvNN. Note that using the denoiser is introduced to simplify the training process. As a comparison, we also train the ACSI-PNet network over various noise realizations. Specifically, in this case, for "Denoiser + ACSI-PNet", we trained the denoiser on mixed speed and mixed SNR values, and the CSI prediction (with ACSI-PNet) was trained on fixed SNR value. In the other case, we trained the ACSI-PNet for CSI prediction and over different SNR values. Typically, ACSI-PNet has a larger number of parameters and requires a longer time to train. Adding the noise dimension can prolong the training process. For instance, a simple augmentation of the dataset with noise realization over different SNR values will extend the training time by the dataset augmentation order. Nonetheless, as seen in Fig. 11, both approaches are promising, indicating that they, depending on the available resources, can be viable options.

Finally, we point out that in all the presented results, the target output signal is assumed to be noiseless. We have tested several cases when the output itself is noisy (during training). A few examples are shown in Tab. 7. The training and testing are done at the same speed (10 kmph). We observed that the small noise does not significantly impact the result, except for a small increase in the standard deviation.

C. THE IMPACT OF THE LENGTH OF THE OBSERVATION WINDOW

In this study, we investigate the impact of the observation length L for the different models. The results are shown in Tab. 8. The proposed models, especially the (v1) based networks, show superior performance across the observation lengths with the capability of using all the observations. This is not the case for all AI models, where some have slight degradation with the increase in the observation window. This might be attributed to the training convergence issue, although we have tried to train these models using different random seeds.

FIGURE 12. The two locations for the LOS and NLOS measurements.

VIII. EVALUATION WITH FIELD MEASUREMENTS

For a realistic evaluation of the CSI prediction, we test the proposed method on field measurements. The measurement setup is presented in subsection VIII-A, and in subsection VIII-B we present the results. Note that unless otherwise stated, all parameters and preprocessing that were discussed in earlier sections apply.

A. MEASUREMENT SETUP

The data was collected in a commercial network in Suwon-si, South Korea in mixed outdoor channel conditions with both LOS and NLOS, see Fig. 12. The LOS data were collected near a soccer field where the BS is visible; however, since the area is surrounded by foliage, certain areas have partially obstructed LOS. The NLOS data were collected near a fire station in a typical urban environment. The BS antenna array, the channel bandwidth, and the number of used RBs are similar to Sec. VII-B. The CSI pilot periodicity is 10 ms. In total 22 traces were collected with two users. For each trace, the user was instructed to walk slowly or fast for about 10 s from a random starting location and orientation within the highlighted areas in Fig. 12. One user used a Samsung Galaxy J5, while the other used Samsung Galaxy Note9. We chose these two terminals since their clocks have different accuracies. In general, the estimated SNR in all the traces was above 18 dB.

In our study, we use 14 traces for training, four traces for validation, and four for testing, i.e., a split of (63.3%, 18.1%, 18.1%), respectively. The traces used for

TABLE 8. CSI prediction performance, eSNR [dB] with different observation lengths (*L*).

Number of Observations (L)	SH	CSI-PNet (v1,k=1)	CSI-PNet (v1,k=3)	ACSI-PNet (v1,k=1)	ACSI-PNet (v2,k=1)	CvNN	LSTM+FCN	3DCNN	ARM
3	6.48	16.57	17.19	17.27	15.58	15.86	11.58	8.14	10.21
5	6.48	18.5	19.07	19.14	17.46	17.5	13.07	8.91	10.97
10	6.48	19.43	19.84	20.28	19.83	19.02	13.46	10.5	11.26
15	6.48	19.64	20.07	20.64	20.43	18.52	13.87	10.45	11.25

TABLE 9. CSI prediction performance, eSNR [dB], based on the field measurement data.

TFO Correction	SH	CSI-PNet (v1,k=3)	CSI-PNet (v1,k=1)	ACSI-PNet (v1,k=1)	ACSI-PNet (v2,k=1)	CvNN	3DCNN	LSTM+FCN	ARM
Yes	13.06	15.64	15.94	15.87	15.92	10.53	13.69	13.39	13.37
No	13.5	*	4.16	3.89	*	6.47	3.37	*	6.87

testing are three from the J5 terminal with one low-speed LOS and one LOS and one NLOS with high walking speeds and one from the Note9 with NLOS high walking speeds.

B. RESULTS AND DISCUSSION

Table 9 shows the eSNR for several solutions. Results are given for two scenarios where the TFO is corrected or uncorrected. When no TFO correction is applied, we see very poor results for all CSI prediction solutions. Note that some solutions did not converge, marked by "*". Application of the TFO correction results in a significant improvement in the eSNR. We notice that the proposed networks outperform other benchmarks. For instance, the CvNN shows poor performance. This could be attributed to the path selection process and the per-path prediction structure, as we discussed in Sec. I-A, some of the challenges for this method may be attributed to the need for SNR estimation, the loss of the residual path correlation at the input of the prediction model, and the sensitivity to the angle domain transformation. In addition, the DMCs are usually difficult to capture.

Note that for this section, we used eSNR as a metric to present the results to highlight the impact of TFO on the SH. For SH, the eSNR did not change significantly with TFO. The reason comes from the fact the eSNR is based on the correlation metric, which is robust to phase rotation. However, that is not the case for the discussed CSI prediction solutions that use the previous observations to learn the mapping, and random phase offset can disturb the process.

IX. CONCLUSION

This paper proposes enhancements to AI-based CSI prediction using new deep-learning architectures. In particular, we propose CSI-PNet (with two different design variants) and ACSI-PNet. The solutions are based on 3D-CCNN (3D complex valued CNNs). The networks have the capabilities to capture the spatial and temporal correlation that show gains especially at low and medium speeds.

To evaluate the performance of the new methods, we consider two datasets. One is a synthetic dataset using an outdoor UMa 3GPP channel model. The other is based on measurements in a commercial network. In the latter, the correction of TFO (Time and Frequency Offset) plays a major role in the performance. Overall, the results show the advantage of the proposed solutions.

In future work, several enhancements to the network can be investigated, e.g., efficient methods to handle sparsity in the angle and the delay domains and further optimization of the network size, e.g., via compression network. The AN in ACSI-PNet can be implemented with an RNN that can adaptively track the possible long-term temporal correlation. Studying distributed implementation (in the UE and BS) or delayed feedback is another interesting research direction. The combination of temporal CSI prediction and feedback compression, or frequency domain extrapolation, are also interesting research directions.

REFERENCES

[1] Y.-H. Nam, B. L. Ng, K. Sayana, Y. Li, J. Zhang, Y. Kim, and J. Lee, "Full-dimension MIMO (FD-MIMO) for next generation cellular technology," *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 172–179, Jun. 2013.

[2] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, Nov. 2010.

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

[4] 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. 15183–15195, 2019.

[5] K. E. Baddour and N. C. Beaulieu, "Autoregressive modeling for fading channel simulation," *IEEE Trans. Wireless Commun.*, vol. 4, no. 4, pp. 1650–1662, Jul. 2005.

[6] Z. Shen, J. G. Andrews, and B. L. Evans, "Short range wireless channel prediction using local information," in *Proc. 37th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2003, pp. 1147–1151.

[7] J. Vanderpypen and L. Schumacher, "MIMO channel prediction using ESPRIT based techniques," in *Proc. IEEE 18th Int. Symp. Pers., Indoor Mobile Radio Commun.*, 2007, pp. 1–5.

[8] L. Dong, G. Xu, and H. Ling, "Prediction of fast fading mobile radio channels in wideband communication systems," in *Proc. GLOBECOM IEEE Global Telecommun. Conf.*, Nov. 2001, pp. 3287–3291.

[9] W. Li, H. Yin, Z. Qin, Y. Cao, and M. Debbah, "A multi-dimensional matrix pencil-based channel prediction method for massive MIMO with mobility," *IEEE Trans. Wireless Commun.*, vol. 22, no. 4, pp. 2215–2230, Apr. 2023.
[10] A. Duel-Hallen, "Fading channel prediction for mobile radio adaptive transmission systems," *Proc. IEEE*, vol. 95, no. 12, pp. 2299–2313, Dec. 2007.
[11] M. E. Morocho-Cayamcela, H. Lee, and W. Lim, "Machine learning for 5G/B5G mobile and wireless communications: Potential, limitations, and future directions," *IEEE Access*, vol. 7, pp. 137184–137206, 2019.
[12] G. Cerar, H. Yetgin, M. Mohorcic, and C. Fortuna, "Machine learning for wireless link quality estimation: A survey," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 696–728, 2nd Quart., 2021.
[13] 3GPP. *Release 18 Meeting Notes*. [Online]. Available: https://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_110
[14] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y. A. Zhang, "The roadmap to 6G: AI empowered wireless networks," *IEEE Commun. Mag.*, vol. 57, no. 8, pp. 84–90, Aug. 2019.
[15] B. H. Fleury, "First- and second-order characterization of direction dispersion and space selectivity in the radio channel," *IEEE Trans. Inf. Theory*, vol. 46, no. 6, pp. 2027–2044, Sep. 2000.
[16] 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 J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 1915–1930, Jul. 2021.
[17] L. Liu, C. Oestges, J. Poutanen, K. Haneda, P. Vainikainen, F. Quitin, F. Tufvesson, and P. D. Doncker, "The COST 2100 MIMO channel model," *IEEE Wireless Commun.*, vol. 19, no. 6, pp. 92–99, Dec. 2012.
[18] W. Jiang and H. D. Schotten, "Neural network-based fading channel prediction: A comprehensive overview," *IEEE Access*, vol. 7, pp. 118112–118124, 2019.
[19] W. Liu, L.-L. Yang, and L. Hanzo, "Recurrent neural network based narrowband channel prediction," in *Proc. IEEE 63rd Veh. Technol. Conf.*, May 2006, pp. 2173–2177.
[20] T. Ding and A. Hirose, "Fading channel prediction based on complex-valued neural networks in frequency domain," in *Proc. Int. Symp. Electromagn. Theory*, May 2013, pp. 640–643.
[21] H. Kim, S. Kim, H. Lee, C. Jang, Y. Choi, and J. Choi, "Massive MIMO channel prediction: Kalman filtering Vs. machine learning," *IEEE Trans. Commun.*, vol. 69, no. 1, pp. 518–528, Jan. 2021.
[22] W. Jiang and H. D. Schotten, "Recurrent neural network-based frequency-domain channel prediction for wideband communications," in *Proc. IEEE 89th Veh. Technol. Conf. (VTC-Spring)*, Apr. 2019, pp. 1–6.
[23] W. Jiang and H. D. Schotten, "Deep learning for fading channel prediction," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 320–332, 2020.
[24] M. K. Shehzad, L. Rose, S. Wesemann, and M. Assaad, "ML-based massive MIMO channel prediction: Does it work on real-world data?" *IEEE Wireless Commun. Lett.*, vol. 11, no. 4, pp. 811–815, Apr. 2022.
[25] J. Yuan, H. Q. Ngo, and M. Matthaiou, "Machine learning-based channel prediction in massive MIMO with channel aging," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 2960–2973, May 2020.
[26] C. Eom and C. Lee, "Hybrid neural network-based fading channel prediction for link adaptation," *IEEE Access*, vol. 9, pp. 117257–117266, 2021.
[27] Y. Zhang, A. Alkhateeb, P. Madadi, J. Jeon, J. Cho, and C. Zhang, "Predicting future CSI feedback for highly-mobile massive MIMO systems," 2022, *arXiv:2202.02492*.
[28] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant CNNs," in *Proc. Int. Conf. 3D Vis. (3DV)*, Oct. 2017, pp. 11–20.
[29] H. Jiang, M. Cui, D. W. K. Ng, and L. Dai, "Accurate channel prediction based on transformer: Making mobility negligible," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 9, pp. 2717–2732, Sep. 2022.
[30] S. Jiang, W. Wang, Y. Miao, W. Fan, and A. F. Molisch, "A survey of dense multipath and its impact on wireless systems," *IEEE Open J. Antennas Propag.*, vol. 3, pp. 435–460, 2022.
[31] T. Ding and A. Hirose, "Online regularization of complex-valued neural networks for structure optimization in wireless-communication channel prediction," *IEEE Access*, vol. 8, pp. 143706–143722, 2020.
[32] H. Kim, J. Choi, and D. J. Love, "Massive MIMO channel prediction via meta-learning and deep denoising: Is a small dataset enough?" 2022, *arXiv:2210.08770*.
[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 Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 227–236, Jan. 2020.
[34] Y. Yang, F. Gao, C. Xing, J. An, and A. Alkhateeb, "Deep multimodal learning: Merging sensory data for massive MIMO channel prediction," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 1885–1898, Jul. 2021.
[35] Y. Zhang, J. Wang, J. Sun, B. Adebisi, H. Gacanin, G. Gui, and F. Adachi, "CV-3DCNN: Complex-valued deep learning for CSI prediction in FDD massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 10, no. 2, pp. 266–270, Feb. 2021.
[36] M. Alrabeiah and A. Alkhateeb, "Deep learning for TDD and FDD massive MIMO: Mapping channels in space and frequency," in *Proc. 53rd Asilomar Conf. Signals, Syst., Comput.*, Nov. 2019, pp. 1465–1470.
[37] C. Huang, R. He, B. Ai, A. F. Molisch, B. K. Lau, K. Haneda, B. Liu, C.-X. Wang, M. Yang, C. Oestges, and Z. Zhong, "Artificial intelligence enabled radio propagation for communications—Part II: Scenario identification and channel modeling," *IEEE Trans. Antennas Propag.*, vol. 70, no. 6, pp. 3955–3969, Jun. 2022.
[38] T. Zhou, H. Zhang, B. Ai, C. Xue, and L. Liu, "Deep-learning-based spatial–temporal channel prediction for smart high-speed railway communication networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 7, pp. 5333–5345, Jul. 2022.
[39] M. Sasaki, N. Kuno, T. Nakahira, M. Inomata, W. Yamada, and T. Moriyama, "Deep learning based channel prediction at 2–26 GHz band using long short-term memory network," in *Proc. 15th Eur. Conf. Antennas Propag. (EuCAP)*, Mar. 2021, pp. 1–5.
[40] M. del Rosario and Z. Ding, "Learning-based MIMO channel estimation under practical pilot sparsity and feedback compression," *IEEE Trans. Wireless Commun.*, vol. 22, no. 2, pp. 1161–1174, Feb. 2023.
[41] D. Burghal, R. Wang, A. Alghafis, and A. F. Molisch, "Supervised ML solution for band assignment in dual-band systems with omnidirectional and directional antennas," *IEEE Trans. Wireless Commun.*, vol. 21, no. 9, pp. 7550–7565, Sep. 2022.
[42] P. Bello, "Characterization of randomly time-variant linear channels," *IEEE Trans. Commun.*, vol. COM-11, no. 4, pp. 360–393, Dec. 1963.
[43] F. Hlawatsch and G. Matz, *Wireless Communications Over Rapidly Time-Varying Channels*. New York, NY, USA: Academic, 2011.
[44] P. Dong, H. Zhang, G. Y. Li, I. S. Gaspar, and N. NaderiAlizadeh, "Deep CNN-based channel estimation for mmWave massive MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 5, pp. 989–1000, Sep. 2019.
[45] M. B. Mashhadi and D. Gunduz, "Pruning the pilots: Deep learning-based pilot design and channel estimation for MIMO-OFDM systems," 2020, *arXiv:2006.11796*.
[46] M. Morelli, C.-C.-J. Kuo, and M.-O. Pun, "Synchronization techniques for orthogonal frequency division multiple access (OFDMA): A tutorial review," *Proc. IEEE*, vol. 95, no. 7, pp. 1394–1427, Jul. 2007.
[47] H. Meyr, M. Moeneclaey, and S. A. Fechtel, *Digital Communication Receivers: Synchronization, Channel Estimation, and Signal Processing*, vol. 444. Hoboken, NJ, USA: Wiley, 1998.
[48] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.

DAOUD BURGHAL received the B.S. degree in electrical engineering from The University of Jordan, Amman, Jordan, in 2007, and the M.S. and Ph.D. degrees in electrical engineering and statistics from the University of Southern California, Los Angeles, CA, USA, in 2019. After the Ph.D. degree, he was a Postdoctoral Scholar with the WiDeS Laboratory. Later, he was a Wireless Research and Development System Engineer with Qualcomm. In 2022, he joined Samsung Research America as an AI Research Engineer. His research interests include wireless communications, AI-assisted communication, and joint communication and learning.

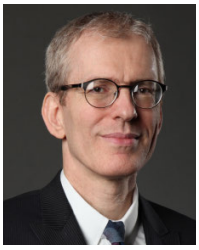
YANG LI received the B.S. degree from Shanghai Jiao Tong University, in 2005, and the Ph.D. degree from The University of Texas at Dallas, in 2012. From 2016 to 2019, he was with Apple, working on cellular, device-to-device, and satellite communications. He is currently the Director of the Standard and Mobility Innovation Laboratory, Samsung Research America, where he focuses on advanced signal processing and machine learning algorithms design in 5G cellular systems.

PRANAV MADADI received the bachelor's degree in electrical engineering from IIT Hyderabad, in 2014, and the Ph.D. degree in electrical and computer engineering from The University of Texas at Austin, in 2019. He is currently a Staff Research Engineer with Samsung Research America. His current research interests include applications of AI in wireless communications and 3GPP standards for 5G/6G technologies.

YEQING HU received the joint B.E. degree in electrical engineering from the Beijing Institute of Technology, Beijing, China, and the Australian National University, Canberra, Australia, in 2012, and the Ph.D. degree in wireless communication from Monash University, Melbourne, VIC, Australia, in 2017. Since 2017, she has been with Samsung Research America, Richardson, TX, USA. Her current research interests include MIMO systems, signal processing, and channel estimation.

JEONGHO JEON (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Maryland, College Park, MD, USA. He is currently the Director of the Standards and Mobility Innovation Laboratory, Samsung Research America, where he contributes to the next-generation cellular communications research and standardizations. Prior to joining Samsung, in 2018, he was with Intel Corporation, where he contributed to various advanced research as well as 3GPP 4G LTE and 5G NR standardizations. From 2021 to 2023, he was the Vice Chairman of the Technology Working Group, Next G Alliance, a 6G initiative by ATIS, and led the development of 6G candidate technologies. He was a recipient of the National Institute of Standards and Technology (NIST) Fellowship, from 2011 to 2013.

JOONYOUNG CHO received the B.S., M.S., and Ph.D. degrees in electrical engineering from the Pohang University of Science and Technology (POSTECH), South Korea, in 1993, 1995, and 2003, respectively. He engaged in CDMA modem algorithm development and design, while he was with SK Telecom, South Korea, from 1995 to 1998, and POSTECH as a Research Staff, until 1999. He was also involved in research and standardization for 3G(HSPA)/4G(LTE)/5G(NR) communications, while he was with Samsung Electronics, from 2003 to 2015, and Intel Corporation, from 2015 to 2018. He is currently the Senior Director of Standardization and Research for 5G/6G Communications with Samsung Research America (SRA).



ANDREAS F. MOLISCH received the Ph.D. and Habilitation degrees from TU Vienna, in 1994 and 1999, respectively. After ten years in the industry, he joined the University of Southern California, where he is currently the Solomon Golomb—Andrew and Erna Viterbi Chair Professor. He is the author of five books, including *Wireless Communications*, (Third edition, 2023), 22 book chapters, more than 300 journal articles, 400 conference papers, and 70 granted patents. His work has been cited more than 66,000 times and H-index of 108. His research interest includes wireless communications, with an emphasis on wireless propagation channels, multi-antenna systems, ultrawideband signaling and localization, novel modulation methods, caching for wireless content distribution, and edge computing. He is a fellow of the National Academy of Inventors, AAAS, and IET, and a member of the Austrian Academy of Sciences. He was a recipient of numerous awards.

JIANZHONG (CHARLIE) ZHANG received the Ph.D. degree from the University of Wisconsin–Madison. He was with the Nokia Research Center, from 2001 to 2006, and with Motorola, from 2006 to 2007. From 2009 to 2013, he was the Vice Chairman of the 3GPP RAN1 WG. He is currently the Senior Vice President and the Head of the Standards and Mobility Innovation Laboratory, Samsung Research America, where he leads research, prototyping, and standards for 5G cellular systems and future multimedia networks.