

CSI-4CAST: A Hybrid Deep Learning Model for CSI Prediction with Comprehensive Robustness and Generalization Testing

Sikai Cheng*, Reza Zandehshahvar*, Haoruo Zhao*, Daniel A. Garcia-Ulloa[†], Alejandro Villena-Rodriguez[†], Carles Navarro Manchón[†], Pascal Van Hentenryck*

Abstract—Channel state information (CSI) prediction is a promising strategy for ensuring reliable and efficient operation of massive multiple-input multiple-output (mMIMO) systems by providing timely downlink (DL) CSI. While deep learning-based methods have advanced beyond conventional model-driven and statistical approaches, they remain limited in robustness to practical non-Gaussian noise, generalization across diverse channel conditions, and computational efficiency. This paper introduces *CSI-4CAST*, a hybrid deep learning architecture that integrates 4 key components, i.e., Convolutional neural network residuals, Addaptive correction layers, ShuffleNet blocks, and Transformers, to efficiently capture both local and long-range dependencies in CSI prediction. To enable rigorous evaluation, this work further presents a comprehensive benchmark, *CSI-RRG* for Regular, Robustness and Generalization testing, which includes more than 300,000 samples across 3,060 realistic scenarios for both TDD and FDD systems. The dataset spans multiple channel models, a wide range of delay spreads and user velocities, and diverse noise types and intensity degrees. Experimental results show that *CSI-4CAST* achieves superior prediction accuracy with substantially lower computational cost, outperforming baselines in 88.9% of TDD scenarios and 43.8% of FDD scenarios—the best performance among all evaluated models—while reducing FLOPs by 5× and 3× compared to LLM4CP, the strongest baseline. In addition, evaluation over *CSI-RRG* provides valuable insights into how different channel factors affect the performance and generalization capability of deep learning models. Both the dataset (Hugging Face)¹ and evaluation protocols (GitHub)² are publicly released to establish a standardized benchmark and to encourage further research on robust and efficient CSI prediction.

Index Terms—CSI Prediction, mMIMO System, Time Series Forecasting, Deep Learning for Wireless Communications, Computational Efficiency, Robustness, Generalization

I. INTRODUCTION

MASSIVE multiple-input multiple-output (mMIMO) system has been widely adopted in fifth-generation (5G) wireless communication networks [1]. By deploying large antenna arrays at base stations (BS) and employing advanced antenna configuration techniques such as directional beamforming, efficient precoding, and adaptive power allocation [2, 3], mMIMO significantly enhances spectral and energy efficiency,

improves coverage, and reduces multiuser interference [4]–[6]. These advantages enable high user density and diverse services such as calling, video streaming, and internet browsing simultaneously, even within the same frequency band. However, the benefits of mMIMO rely on accurate real-time downlink (DL) channel state information (CSI) acquisition at BS [7, 8]. In practice, CSI acquisition remains challenging due to the well-known aging effect [9, 10], which arises from inevitable delays in wireless systems, including transmission, estimation [11], and feedback delays, particularly in frequency division duplexing (FDD) mode [12]. Rapid channel variations caused by user mobility, multipath propagation, and channel noise further exacerbate this problem. Consequently, the acquired CSI often diverges from the true channel conditions, degrading BS operations such as precoding and power allocation.

To address this issue, CSI prediction has emerged as a promising strategy to mitigate the aging effect and provide timely DL CSI. In CSI prediction, the BS attempts to predict the DL CSI at a future time instant based on (noisy) observations of the uplink (UL) CSI at a previous instant. By forecasting future CSI from past observations, CSI prediction alleviates the impact of aging and reduces CSI acquisition overhead. In time division duplexing (TDD) systems, where UL and DL transmissions occur sequentially over the same frequency band, prediction constitutes an intra-band task. In contrast, in FDD systems, UL and DL transmissions occupy separate frequency bands, making prediction an inter-band task.

Conventional CSI prediction methods can be broadly classified into model-based and statistical approaches. This includes autoregressive (AR) models [10, 13], low-complexity polynomial approximation predictors [14], first-order Taylor expansion-based channel models [15], and Kalman filter frameworks [16, 17]. In addition, Prony-based predictors have been introduced to exploit the angular-delay domain structure inherent in mMIMO channels [18]. While AR and polynomial predictors are computationally efficient, they accumulate errors quickly and fail to generalize under large channel variations. More advanced approaches, such as Kalman filters, Prony-based, and Taylor expansion models, incur high parameter estimation overhead. In general, all model-based methods often remain sensitive to model mismatch and non-Gaussian noise, tend to scale poorly to large antenna arrays, and to degrade rapidly beyond short prediction horizons [17, 19]–[21].

Deep learning models have recently emerged as powerful tools for high-dimensional time-series forecasting and have been increasingly applied to CSI prediction [22, 23]. Once trained, deep learning architectures offer efficient inference

*S. Cheng, R. Zandehshahvar, H. Zhao, and P. Van Hentenryck are with the H. Milton Stewart School of Industrial and Systems Engineering, NSF AI Institute for Advances in Optimization (AI4OPT), Georgia Institute of Technology, Atlanta, GA, USA. Corresponding author: Sikai Cheng (email: sikaicheng@gatech.edu).

[†]Daniel A. Garcia-Ulloa, Alejandro Villena-Rodriguez, and Carles Navarro Manchón are with Keysight Technologies in Atlanta, GA; Málaga, Spain; and Barcelona, Spain, respectively.

¹Dataset: <https://huggingface.co/CSI-4CAST>

²Code: <https://github.com/AI4OPT/CSI-4CAST>

and can capture complex temporal and spatial dependencies beyond the reach of conventional statistical and model-based approaches. Motivated by these advantages, researchers have developed a diverse set of architectures for CSI prediction. The combination of LSTM and GRU was proposed in [24] to address the vanishing gradient problem and improve training stability. A CNN-based architecture that models the CSI matrix as a complex-valued image to preserve phase information was introduced in [25]. A generative adversarial network (GAN) model is used in [26] for CSI prediction, reconstructing the full CSI matrix, including the future part, from corrupted inputs containing only historical data. A transformer-based model with an attention mechanism was proposed in [27] to support parallel multi-step prediction, aiming to reduce error propagation and mitigate data loss in sequential forecasting. Spectral-temporal graph neural network (STEMGNN) was recently employed in [28] to jointly capture frequency-domain spatial correlations and time-domain dynamics of CSI. Recently, large language model (LLM)-based methods have also been explored by framing CSI prediction as a time-series forecasting task analogous to next-token prediction in language modeling. For example, pre-trained GPT-2 layers were used in a CSI predictor in [29], showing superior generalization compared to other deep learning approaches. Similarly, a BERT-based predictor was proposed in [30], where the masked token training strategy effectively addressed missing data in CSI prediction. Despite notable advancements, deep learning-based CSI prediction methods still encounter three core challenges:

- **Robustness to noise:** In practice, real-world wireless channels are affected by various noise sources that deviate from the standard additive white Gaussian noise (AWGN) assumption. These include phase noise stemming from imperfect local oscillators [31, 32], burst noise marked by short, high-amplitude spikes due to electromagnetic interference [33, 34], and packet drop noise resulting from system-level issues such as scheduling delays or network congestion [35]. Despite their practical significance, the robustness of CSI prediction methods under such noise conditions had not been properly explored.
- **Limited generalization:** Deep learning models often struggle to generalize to unseen scenarios or maintain performance under distribution shifts. Most models perform well only when training and evaluation data distributions are closely aligned. However, some previous works consider narrow in-distribution testing and adopt impractical assumptions, such as training separate models for different user velocities (e.g., [28]). In contrast, real-world wireless systems feature continuously varying channel conditions and user mobility, demanding models that can generalize reliably across diverse environments.
- **High computational cost:** Leading approaches like LLM4CP [29] and CSI-BERT [30] impose high computational costs due to their reliance on large pre-trained language models. These models demand substantial hardware resources, often exceeding what is feasible for deployment at each BS, particularly due to their high memory and throughput requirements. Moreover, recent

findings suggest that such complex architectures may not be necessary for time-series forecasting, as simpler designs can deliver comparable or even superior performance [36].

This paper addresses the challenges mentioned above in CSI prediction through the following core contributions:

- **CSI-4CAST:** This paper proposes a novel deep learning architecture that significantly enhances the accuracy–efficiency trade-off in CSI prediction. The *CSI-4CAST* integrates 4 key components: Convolutional neural network-based residuals, Addaptive correction layer modules for temporal and subcarrier intrinsic dependencies extraction, ShuffleNet blocks for compact feature learning, and Transformer encoders for long-range modeling. These elements together enhance noise robustness and computational efficiency. Experimental results demonstrate that *CSI-4CAST* achieves the lowest NMSE in 88.9% of test scenarios under TDD, while reducing computational cost by approximately 5× in FLOPs compared to LLM4CP, the second-best model. In the more challenging FDD setting, *CSI-4CAST* leads in 43.8% of test scenarios—ranking highest among all models—and achieves over a 3× reduction in FLOPs relative to LLM4CP.
- **Comprehensive evaluation suite:** This paper presents a large-scale and realistic benchmark designed for the training and evaluation of CSI prediction models. The dataset includes over 300,000 samples across 3,060 distinct scenarios for both TDD and FDD, encompassing multiple standardized channel models, a range of delay spreads, varying user mobility speeds, and a wide spectrum of SNR conditions. Additionally, it incorporates several types of non-Gaussian noise—such as burst, phase, and packet-drop noise—at multiple intensity levels, reflecting perturbations commonly encountered in practice yet often overlooked in previous studies. Collectively, these elements constitute the most comprehensive framework to date for rigorously stress-testing CSI prediction methods under diverse and realistic conditions.
- **Reproducibility and impact:** The *CSI-RRG*, short for *Regular, Robustness, and Generalization*, together with its evaluation protocols, is publicly released to provide a standardized benchmark for CSI prediction. It covers TDD/FDD operation, noise stress testing, and cross-scenario generalization. The *CSI-RRG* benchmark is designed to lower the barrier to rigorous comparison and foster progress toward more robust and efficient CSI prediction.

Throughout this paper, the following notational conventions are adopted. The bold capital notation $\mathbf{H}$ represents the CSI matrix or tensor, while the bold lowercase notation $\mathbf{h}$ denotes the CSI vector (e.g., along the antenna or subcarrier dimension), and the non-bold lowercase notation h indicates the element-wise CSI. For any matrix, vector, or element of CSI, $(\cdot)^t$ and $(\cdot)^T$ denote the CSI at a specific time index t and over a time sequence $\mathcal{T}$, respectively. The notation $(\hat{\cdot})$ represents the noisy observation of the CSI, while $(\hat{\cdot})^\dagger$ denotes its predicted value. For a general complex matrix, $(\cdot)^\dagger$ denotes the Hermitian transpose, and $\|\cdot\|_F$ represents the Frobenius norm.

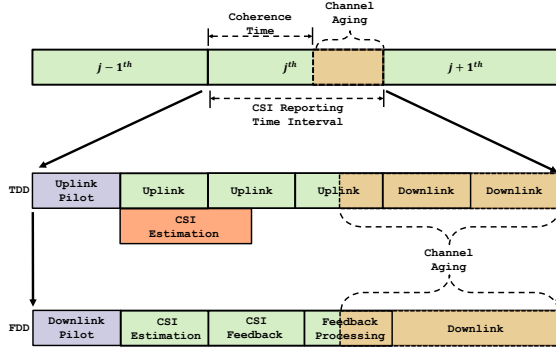


Fig. 1. The channel aging problem.

II. PROBLEM DEFINITION

This study investigates the task of DL CSI prediction in a MIMO system, employing Orthogonal Frequency Division Multiplexing (OFDM) for signal transmission. In this system, the BS is equipped with a dual-polarized uniform planar array (UPA) composed of M_h rows and M_v columns of antenna elements. The User Equipment (UE) has a single omnidirectional receive antenna. OFDM splits the transmission bandwidth into N_{sc} orthogonal subcarriers, allowing for efficient frequency-domain processing.

According to established models [37]–[39], the CSI at time t , denoted as $\mathbf{H}^t$, is represented as a complex-valued tensor:

$$\mathbf{H}^t \in \mathbb{C}^{N_{tr} \times N_{re} \times N_{sc}}. \quad (1)$$

It captures the channel coefficients between each transmit–receive antenna pair (spatial dimension) and across all subcarriers (frequency dimension) at a specific time instant t . Each matrix element is complex-valued: its magnitude reflects path gain (attenuation) and its phase reflects the propagation-induced phase shift, including hardware/oscillator offsets. The transmitter employs $N_{tr} = 2 \times M_h \times M_v$ antennas, while the receiver is equipped with $N_{re} = 1$. A single snapshot $\mathbf{H}^t$ implicitly encodes delay and angular structure across subcarriers and antennas, and a sequence of CSI further captures the channel’s temporal evolution (e.g., Doppler and path dynamics).

Accurate CSI conveys essential information for BS functions such as precoding, scheduling, and power control. However, channel aging often prevents the BS from accessing the up-to-date DL CSI. Fig. 1 illustrates the detailed transmission sequence in TDD and FDD. In the top part of the figure, the time axis is divided into successive CSI reporting intervals $j-1, j, j+1$. Ideally, for each interval, a pilot signal—known reference signals shared between the BS and UE—is transmitted at the beginning. CSI is then estimated by comparing the received pilot with the shared reference, and the resulting CSI is assumed stationary, guiding the BS’s operations. However, in modern communication systems characterized by mobility and rapidly changing multipath environments, the coherence time—i.e., the time interval over which channel conditions can be considered constant—shortens. This, combined with inevitable estimation, processing, scheduling delays, and feedback delays (in FDD systems), leads to the channel aging

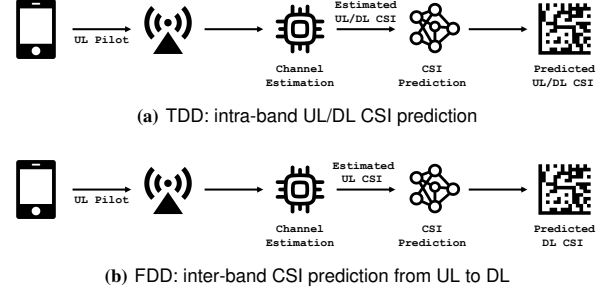


Fig. 2. An illustration of the DL CSI acquisition schemas.

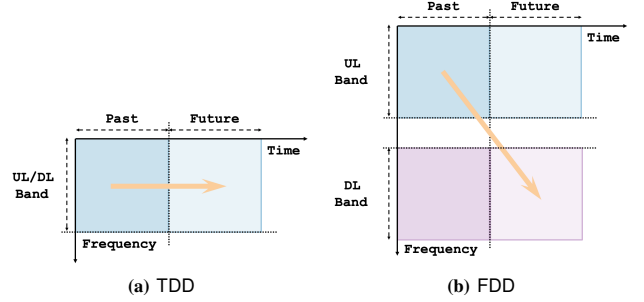


Fig. 3. An illustration of the CSI Prediction in the time-frequency domain.

problem: by the time DL transmission occurs, the reported CSI is already outdated and no longer reflects current DL conditions. The bottom part of the figure illustrates detailed time-interval breakdowns for TDD and FDD scenarios, highlighting the occurrence of channel aging. According to 3GPP procedures [40], in TDD systems, where UL and DL share the same frequency band and are separated in time, the BS can—after reciprocity calibration—use UL pilot-based estimates for precoding in the next DL slot. However, a scheduled UL transmission block typically follows the UL pilot transmission, and the delay before the next scheduled DL transmission may render these estimates stale. In FDD systems, where UL and DL operate on different frequencies and reciprocity does not apply, DL CSI is estimated at the UE based on the DL pilots. The BS transmits DL pilots; the UE estimates the DL channel, compresses or quantizes it, and sends it back on the UL band. The BS decodes this feedback and applies it at the subsequent DL scheduling boundary. Despite the concurrent operation of UL and DL in FDD—eliminating the need to wait for UL transmissions to complete—latency from feedback and processing may still exceed the coherence time.

To mitigate channel aging, CSI prediction forecasts near-future DL CSI based on recent pilot-based observations, thereby reducing the pilot-to-use delay and improving the BS’s precoding accuracy. The prediction framework aligns with the CSI acquisition pipeline: as illustrated in Fig. 2a, in TDD systems, where UL and DL share the same frequency band, intra-band prediction leverages historical UL/DL CSI to predict future UL/DL CSI (Fig. 3a) [21]. In contrast, in FDD systems, CSI prediction is utilized to further eliminate the feedback and associated processing overhead by shifting the prediction entirely to the BS side. In this case, the BS uses historical UL CSI (estimated from UL pilots) to directly predict future

DL CSI, as shown in Fig. 2b. Since this approach involves prediction across both time and frequency bands, it is referred to as inter-band prediction (Fig. 3b) [41, 42].

Consistent with the intra-band and inter-band schemas described above, both settings are unified under a single formulation by treating band-specific details as part of the input–target definitions. Denote the historical CSI sequence as follows:

$$\mathbf{H}^{\mathcal{T}} = \{\mathbf{H}^{t-|\mathcal{T}|+1}, \mathbf{H}^{t-|\mathcal{T}|+2}, \dots, \mathbf{H}^t\} \quad (2)$$

where $\mathcal{T}$ represents the historical window $\mathcal{T} = \{t - |\mathcal{T}| + 1, \dots, t - 1, t\}$. CSI prediction can be considered as learning the following mapping:

$$\hat{\mathbf{H}}^{\mathcal{P}} = f_{\Omega}(\mathbf{H}^{\mathcal{T}}), \quad (3)$$

where $\mathcal{P} = \{t + 1, \dots, t + |\mathcal{P}|\}$ is the prediction horizon, and $\hat{\mathbf{H}}^{\mathcal{P}} \in \mathbb{C}^{|\mathcal{P}| \times N_{\text{tr}} \times N_{\text{sc}}}$ represent the predicted future DL CSI at time slot t . The parameters $|\mathcal{T}|$ and $|\mathcal{P}|$ denote the length of the input (past) CSI sequence and the prediction horizon, respectively. The prediction function $f_{\Omega}(\cdot)$ is parameterized by Ω .

Under realistic conditions, clean and accurate CSI is generally unattainable due to the presence of transmission noise and inevitable estimation errors. Consequently, the model utilizes noisy CSI as input instead of ideal CSI. The noisy past CSI at time t , denoted by $\tilde{\mathbf{H}}^t$, is defined as:

$$\tilde{\mathbf{H}}^t = \mathbf{H}^t + \mathbf{E}^t, \quad (4)$$

where $\mathbf{E}^t$ represents the additive noise at time t . Consequently, the predicted CSI under the noisy channel is obtained as:

$$\hat{\mathbf{H}}^{\mathcal{P}} = f_{\Omega}(\tilde{\mathbf{H}}^{\mathcal{T}}). \quad (5)$$

This paper introduces a novel deep learning model to learn the mapping f_{Ω} for CSI prediction. The proposed model is designed to address efficiency, robustness, and generalization and is evaluated against various deep learning models using a comprehensive dataset.

III. METHODOLOGY

This section outlines the proposed method for CSI prediction and describes the proposed deep learning architecture, denoted as *CSI-4CAST*.

Let $\mathcal{D}_{\text{train}} = \{(\tilde{\mathbf{H}}^{\mathcal{T}_i}, \mathbf{H}^{\mathcal{P}_i})\}_{i=1}^N$ be the training dataset, consisting $N = |\mathcal{D}_{\text{train}}|$ pairs of historical and future CSI sequences. The learning process is then formulated as the following optimization problem:

$$\begin{aligned} \min_{\Omega} \quad & \mathbb{E}_{(\tilde{\mathbf{H}}^{\mathcal{T}_i}, \mathbf{H}^{\mathcal{P}_i}) \sim \mathcal{D}_{\text{train}}} [\mathcal{L}(\hat{\mathbf{H}}^{\mathcal{P}_i}, \mathbf{H}^{\mathcal{P}_i})] \\ \text{s.t.} \quad & \hat{\mathbf{H}}^{\mathcal{P}_i} = f_{\Omega}(\tilde{\mathbf{H}}^{\mathcal{T}_i}), \end{aligned} \quad (6)$$

where the loss function $\mathcal{L}(\cdot)$ is considered as Normalized Mean Squared Error (NMSE) between the actual and predicted CSI, and is defined as follows:

$$\text{NMSE}(\hat{\mathbf{H}}^{\mathcal{P}}, \mathbf{H}^{\mathcal{P}}) = \frac{\sum_{t \in \mathcal{P}} \|\hat{\mathbf{H}}^t - \mathbf{H}^t\|_F^2}{\sum_{t \in \mathcal{P}} \|\mathbf{H}^t\|_F^2} \quad (7)$$

where $\|\cdot\|_F$ is the Frobenius norm.

Fig. 4 illustrates the schematic of *CSI-4CAST*. To handle the complexity and high dimensionality of CSI sequences, *CSI-4CAST* integrates several specialized components—Convolutional neural networks (CNN), Adaptive Correction Layers (ACLs), ShuffleNet Blocks, and Transformer encoders. This design enables efficient feature extraction and robustness in CSI prediction.

a) Per-Antenna Modeling for Scalability and Efficiency:

Following standard practice in the literature [29], each transmitter–receiver pair is modeled independently for CSI prediction. This allows *CSI-4CAST* to operate separately on each pair, improving scalability and efficiency. The input of the model is defined as:

$$\mathbf{X} = \tilde{\mathbf{H}}_m^{\mathcal{T}} \in \mathbb{C}^{|\mathcal{T}| \times N_{\text{sc}}}, \quad (8)$$

representing the CSI sequence for a single transmitter–receiver pair m . To comply with the common requirement that neural networks operate on real-valued tensors, the real and imaginary parts of $\mathbf{X}$ are stacked and presented as:

$$\mathbf{X}_r = [\text{Re}(\mathbf{X}), \text{Im}(\mathbf{X})] \in \mathbb{R}^{2 \times |\mathcal{T}| \times N_{\text{sc}}}. \quad (9)$$

b) *CNN-based Residual Representation*: Given the input CSI $\mathbf{X}_r$, a CNN-based module, inspired by [43, 44], is employed to extract structured features. By capturing local correlations across time and frequency, the CNN learns residual representations that refine noisy CSI observations, while its inherent smoothing property mitigates measurement inaccuracies, jointly enhancing robustness. The resulting representation is then denoted as $\mathbf{X}_{\text{CNN}} \in \mathbb{R}^{2 \times |\mathcal{T}| \times N_{\text{sc}}}$.

The module consists of stacked 2D convolutional layers, combined with batch normalization and nonlinear activation functions. The channel configuration evolves across the network depth as $[2, 4, \dots, 2^{\nu}, \dots, 4, 2]$, where the initial value of 2 reflects the real and imaginary components of the input, and ν represents the depth of the network. By virtue of carefully selected kernel sizes, padding schemes, and a symmetric channel structure, the output of the CNN module is guaranteed to preserve the dimensionality of the input.

c) *Delay-Domain Representation for Multi-Path Pattern Utilization*: To complement the frequency-domain representation, the CSI is also transformed into the delay domain, where the signal is described in terms of propagation delays rather than subcarrier frequencies. In this view, each tap corresponds to the signal arriving through a distinct propagation path—for example, direct transmission or reflections from surrounding objects. [45, 46] The resulting path-level representation aggregates the per-subcarrier responses into a small number of significant taps that capture the dominant paths and overall delay spread. This form is typically sparser and more stable, offering a more physically interpretable structure that can improve learning performance [47, 48]. Let $\mathbf{X}_f \in \mathbb{R}^{|\mathcal{T}| \times 2N_{\text{sc}}}$, obtained from the $\mathbf{X}_{\text{CNN}}$ by concatenating the real and imaginary dimensions, denote the real-valued representation of the CSI sequence in the frequency domain. The transformation from frequency to delay domain is accomplished via the inverse discrete Fourier transform (IDFT), expressed as

$$\mathbf{X}_d^{\text{C}} = \mathbf{X}_f^{\text{C}} \mathbf{F}_d^{\dagger} \in \mathbb{C}^{|\mathcal{T}| \times N_{\text{sc}}}. \quad (10)$$

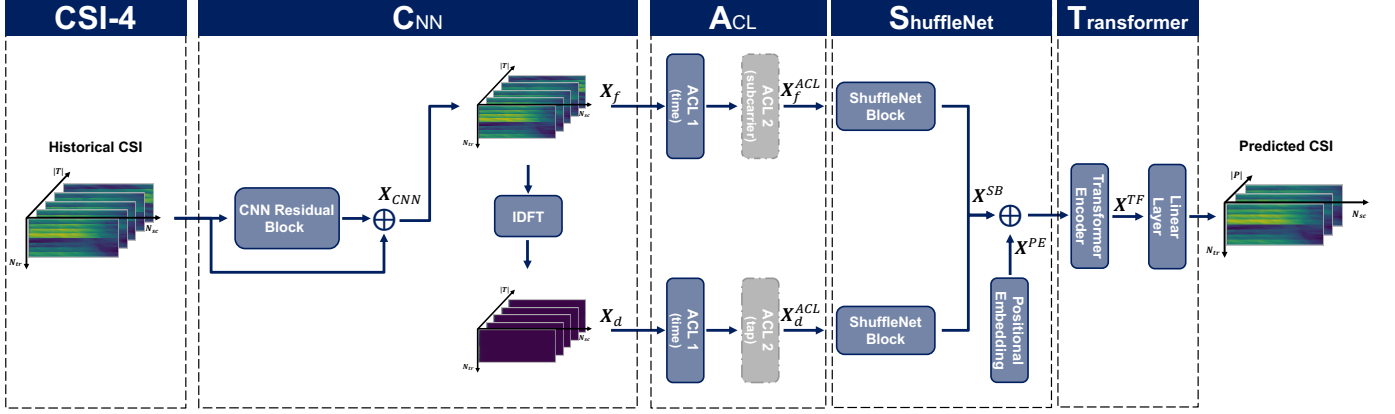


Fig. 4. **The proposed CSI-4CAST.** Historical CSI is first processed by a CNN residual block, followed by an inverse DFT (IDFT) to obtain the delay-domain representation. Both frequency- and delay-domain features are then refined by ACL layers and passed through a ShuffleNet block. Finally, the Transformer block maps the embedded features to predict future CSI. The ACL2 layer (in gray) applies only to the FDD.

Here, $\mathbf{X}_f^C \in \mathbb{C}^{|\mathcal{T}| \times N_{sc}}$ denotes the complex-valued tensor converted from $\mathbf{X}_f$ for the IDFT operation, and $\mathbf{F}_d \in \mathbb{C}^{N_{sc} \times N_{sc}}$ is the unitary DFT matrix. Subsequently, the tensors are converted back into real-valued form: $\mathbf{X}_d^C \in \mathbb{R}^{|\mathcal{T}| \times 2N_{sc}}$ is obtained from $\mathbf{X}_d^C$ by concatenating its real and imaginary components along the last dimension, representing the real-valued CSI sequence in the delay domain.

d) Adaptive Correction Layer (ACL) for Underlying Structures: To capture intrinsic dependencies within $\mathbf{X}_f$ and $\mathbf{X}_d$, ACLs are introduced, motivated by [30,49]. The ACL serves as an adaptive calibration mechanism that dynamically corrects learned representations along the temporal and subcarrier/delay dimensions. Owing to the correlated yet non-stationary nature of wireless channels, a fixed extractor may overlook subtle structural variations; ACLs address this limitation through learnable residual mappings that flexibly modulate inter-time and inter-subcarrier dependencies via additive or multiplicative operations. In TDD systems, ACLs are applied only along the temporal dimension, as prediction is restricted to time (i.e., intra-band). In FDD systems, ACLs are applied along both temporal and subcarrier/delay dimensions, reflecting the joint prediction task across time and frequency (i.e., inter-band).

For a general case $\mathbf{X} \in \mathbb{R}^{|\mathcal{T}| \times 2N_{sc}}$, the ACL is formulated as

$$\begin{aligned} \mathbf{X}^{\text{ACL1}}[:, k] &= \text{MLP}_1(\mathbf{X}[:, k]) \oplus \mathbf{X}[:, k], \forall k \in \{1, 2, \dots, 2N_{sc}\} \\ \mathbf{X}^{\text{ACL2}}[t, :] &= \text{MLP}_2(\mathbf{X}^{\text{ACL1}}[t, :]) \oplus \mathbf{X}^{\text{ACL1}}[t, :], \forall t \in \mathcal{T} \\ \text{MLP}_1: \mathbb{R}^{|\mathcal{T}|} &\rightarrow \mathbb{R}^{|\mathcal{T}|} \\ \text{MLP}_2: \mathbb{R}^{2N_{sc}} &\rightarrow \mathbb{R}^{2N_{sc}} \end{aligned} \quad (11)$$

where MLP_1 and MLP_2 are multilayer perceptrons, with the number of layers, hidden dimensions, and activation functions treated as tunable hyperparameters. The corrected representations are obtained by combining the MLP outputs with the original inputs using an element-wise operation $\oplus$ (addition or multiplication), which is treated as a tunable hyperparameter. This mechanism enables the model to emphasize essential dependencies along the temporal or spectral dimensions. For TDD, the final corrected representation is $\mathbf{X}_f^{\text{ACL}} = \mathbf{X}_f^{\text{ACL1}}$, while for FDD, it is $\mathbf{X}_f^{\text{ACL}} = \mathbf{X}_f^{\text{ACL2}}$. Similarly, $\mathbf{X}_d^{\text{ACL}}$ is

derived in an analogous manner. The ACLs will not change the dimensionality of the input, i.e., $\mathbf{X}_f^{\text{ACL}}, \mathbf{X}_d^{\text{ACL}} \in \mathbb{R}^{|\mathcal{T}| \times 2N_{sc}}$.

e) ShuffleNet Block for Feature Extraction: Inspired by [50,51], the ShuffleNet Block is employed to perform efficient and expressive feature extraction on each input tensor $\mathbf{X}$ obtained from the ACL. In CSI prediction, high-dimensional features across time and subcarriers require a balance between representational richness and computational efficiency. The ShuffleNet Block architecture fulfills this need by combining lightweight convolutions with channel-mixing operations. The input is first reshaped into $\mathbb{R}^{2 \times |\mathcal{T}| \times N_{sc}}$ and projected into ρ feature maps using Conv1d, resulting in $\mathbf{X}_\rho \in \mathbb{R}^{\rho \times |\mathcal{T}| \times N_{sc}}$. A sequence of ShuffleNet Blocks is then applied to extract higher-level representations.

Each ShuffleNet Block integrates several lightweight yet synergistic operations: *Point-wise convolution (PW)*: A Conv1d grouped convolution with group size η is first applied, enabling intra-group cross-channel interaction while remaining computationally efficient. *Channel Shuffle (CS)*: A permutation step redistributes channels across groups, allowing later grouped convolutions to incorporate information from diverse groups, thereby enriching feature representation. *Depth-wise convolution (DW)*: After shuffling, a depth-wise convolution with kernel size $\mu \times \mu$ is applied, where each channel is convolved independently to capture spatial structure within channels while preserving their count. This operation offers spatial expressiveness with low computational cost. *Second point-wise convolution*: A final Conv1d grouped convolution further refines intra-group features post depth-wise processing. The overall feature extraction procedure can be expressed as Eq. (12a), resulting in $\mathbf{X}^{\text{FE}} \in \mathbb{R}^{\rho \times |\mathcal{T}| \times N_{sc}}$. To adaptively emphasize informative channels, a *Squeeze-and-Excitation (SE)* module is applied to $\mathbf{X}^{\text{FE}}$, producing channel attention weights $\mathbf{X}^{\text{SE}} \in \mathbb{R}^{\rho \times 1 \times 1}$ as in Eq. (12b). Finally, the channel-wise Hadamard product ($\odot$) combines $\mathbf{X}^{\text{SE}}$ and $\mathbf{X}^{\text{FE}}$ to yield the refined feature map $\mathbf{X}^{\text{SB}} \in \mathbb{R}^{\rho \times |\mathcal{T}| \times N_{sc}}$ as shown in Eq. (12c).

$$\mathbf{X}^{\text{FE}} = \text{PW}(\text{DW}(\text{CS}(\text{PW}(\mathbf{X}_p)))) \quad (12a)$$

$$\mathbf{X}^{\text{SE}} = \text{SE}(\mathbf{X}^{\text{FE}}) \quad (12b)$$

$$\mathbf{X}^{\text{SB}} = \mathbf{X}^{\text{SE}} \odot \mathbf{X}^{\text{FE}} \quad (12c)$$

After that, the resulting $\mathbf{X}_f^{\text{SB}}, \mathbf{X}_d^{\text{SB}} \in \mathbb{R}^{|\mathcal{T}| \times 2N_{\text{sc}}}$ are obtained by projecting the outputs of the ShuffleNet Blocks back to the original space using Conv1d convolutions and then concatenating the real and imaginary parts. The delay and frequency representation are then added to feed into the subsequent modules, $\mathbf{X}^{\text{SB}} = \mathbf{X}_f^{\text{SB}} + \mathbf{X}_d^{\text{SB}} \in \mathbb{R}^{|\mathcal{T}| \times 2N_{\text{sc}}}$.

f) *Position Embedding & Transformer Encoder*: A Transformer encoder is employed to model long-range and non-local temporal dependencies in the CSI sequence. Through multi-head self-attention, it adaptively aggregates information across all historical time steps. Due to the permutation-invariant nature of the Transformer architecture, it lacks an inherent sense of sequence order. Position embedding (PE) is therefore crucial for incorporating relative positional information. Let γ denote the latent dimension of the Transformer, u the index along the latent dimension, and v the position index of the input sequence. The position embedding $\mathbf{X}^{\text{PE}} \in \mathbb{R}^{|\mathcal{T}| \times \gamma}$ is defined as

$$\mathbf{X}^{\text{PE}}(u, v) = \begin{cases} \sin\left(\frac{v}{|\mathcal{T}|^{u/\gamma}}\right), & u = 0, 2, \dots, 2\lfloor \gamma/2 \rfloor \\ \cos\left(\frac{v}{|\mathcal{T}|^{(u-1)/\gamma}}\right), & u = 1, 3, \dots, 2\lfloor \gamma/2 \rfloor - 1 \end{cases} \quad (13)$$

Simultaneously, token embeddings (TE) are obtained by projecting $\mathbf{X}^{\text{SB}}$ into the latent space of dimension γ using 1×1 convolutions, as shown in (14a). Following the standard Transformer approach [52], the position embeddings are added to the token embeddings to form the input to the Transformer encoder. As illustrated in (14b), the resulting embedded sequence is passed through a stack of Transformer encoder layers, yielding the final representation of the input CSI sequence.

$$\mathbf{X}^{\text{TE}} = \text{Conv1d}(\mathbf{X}^{\text{SB}}) \in \mathbb{R}^{|\mathcal{T}| \times \gamma} \quad (14a)$$

$$\mathbf{X}^{\text{TF}} = \text{Transformer}(\mathbf{X}^{\text{TE}} + \mathbf{X}^{\text{PE}}) \quad (14b)$$

g) *Prediction Module*: The final prediction module comprises two MLPs that transform the learned CSI embeddings into the predicted CSI sequence. The first MLP maps the Transformer embeddings from the latent dimension back to the original subcarrier dimension, $\mathbb{R}^{\gamma} \rightarrow \mathbb{R}^{2N_{\text{sc}}}$. The second MLP projects the historical time dimension into the predicted time dimension, $\mathbb{R}^{|\mathcal{T}|} \rightarrow \mathbb{R}^{|\mathcal{P}|}$. The final predicted CSI sequence is reconstructed by converting the real-valued tensor back into a complex-valued tensor through stacking the real and imaginary parts.

IV. EXPERIMENTS

This section outlines the experimental setup for CSI prediction, covering data generation for training and testing, baseline models, and evaluation metrics.

TABLE I
SYSTEM CONFIGURATION

Parameter	Value
BS antenna	dual-polarized [4,4] UPA
UE antenna	Single omnidirectional antenna
Carrier frequency	2.4 GHz
Subcarriers (UL / DL)	300 for UL/DL
Subcarrier spacing	30 kHz

TABLE II
TRAINING DATASET CONFIGURATION

Parameter	Value
Channel models	CDL-A, CDL-C, CDL-D
Delay spreads	30, 100, 300 ns
User velocities	1, 10, 30 m/s
Noise Type	AWGN
Noise SNR	uniformly distributed in [0, 25] dB

A. Dataset

a) *Data generation and system configurations*: This study uses the Sionna library [53] to synthesize time-varying CSI. The system configurations are summarized in Table I. Specifically, the system configured an OFDM link with a dual-polarized [4,4] UPA at the BS ($N_{\text{tr}} = 4 \times 4 \times 2$) and a single omnidirectional receive antenna at the UE ($N_{\text{re}} = 1$) with the carrier frequency 2.4 GHz. Both TDD and FDD duplexing modes are considered. The OFDM grid consists of 750 subcarriers with 30 kHz subcarrier spacing—300 assigned to the UL and 300 to the DL—corresponding to 9 MHz bandwidth for each link. A guard band of 150 subcarriers (4.5 MHz) separates the UL and DL bands. CSI reports are spaced by 5 slots, corresponding to 2.5 ms under the considered system parameters. The historical CSI window and prediction horizon lengths are considered as $|\mathcal{T}| = 16$, and $|\mathcal{P}| = 4$. As a result, each CSI snapshot is represented by a $32 \times 1 \times 750$ complex tensor, corresponding to the N_{tr} , receive antennas N_{re} , and total number of subcarriers ($2N_{\text{sc}} + 150$). After separating the UL and DL subcarriers, the model input and prediction target are shaped as $32 \times 1 \times 16 \times 300$ and $32 \times 1 \times 4 \times 300$, respectively. These system configurations are carefully aligned with the 3GPP specifications [54], ensuring consistency with standardized practices and relevance to real-world deployment settings.

A large set of scenarios are considered for training and evaluation of the models under various channel conditions and noise types. The set of generated scenarios is denoted as:

$$\mathbf{S} = \left\{ [v_{\text{ue}}, \sigma_{\tau}, \text{CM}, \text{NT}, \text{ND}] \mid v_{\text{ue}} \in \mathcal{V}, \sigma_{\tau} \in \Sigma, \right. \\ \left. \text{CM} \in \mathfrak{M}, \text{NT} \in \mathfrak{N}, \text{ND} \in \mathfrak{D} \right\} \quad (15)$$

which enumerates the combinations of user speed (v_{ue}), delay spread (σ_{τ}), and channel model (CM), along with the noise type (NT) and noise degree (ND) that control the characteristics of additional synthesized noise. The noise component is introduced to emulate real-world channel conditions. Specifically, ND denotes the packet drop probability for packet drop noise,

TABLE III
TESTING DATASET CONFIGURATIONS

Test type	Parameter	Value
Regular		
	Channel models	CDL-A/C/D
	Delay spreads	30, 100, 300 ns
	User velocities	1, 10, 30 m/s
	Noise Type	AWGN
	Noise SNR	[0, 5, 10, 15, 20, 25] dB
Robustness		
	Channel models	CDL-A/C/D
	Delay spreads	30, 100, 300 ns
	User velocities	1, 10, 30 m/s
	Noise Type	phase, burst, packet-drop
	Noise SNR	[10, 15, 20, 25] dB for both phase and burst
	Packet drop probability	[0.01, 0.02, ..., 0.10]
Generalization		
	Channel models	CDL-A/B/C/D/E
	Delay spreads	30, 50, 100, 200, 300, 400 ns
	User velocities	3, 6, ..., 45 m/s; 1, 10 m/s
	Noise Type	AWGN
	Noise SNR	[0, 5, 10, 15, 20, 25] dB

and the SNR for other noise types. These parameters govern the statistical properties of the synthesized CSI sequence (1).

b) Training set: Table II summarizes the training configurations. The 3GPP TR 38.901 [55] CDL-A, CDL-C, and CDL-D models with delay spreads [30, 100, 300] ns and user velocities [1, 10, 30] m/s are adopted. For each scenario 1,000 samples are generated, yielding $3 \times 3 \times 3 = 27$ configurations and 27,000 samples in total. To model noisy CSI observations, AWGN (Appendix D-A) is added to the historical inputs with SNR uniformly distributed in [0, 25] dB. Note that the noise is only added to the historical sequence, and the clean future CSI is used for training and evaluation. The training set is further split into a training set and a validation set with a ratio of 9:1.

c) Testing suite: The testing suite comprises three scenarios (Table III).

- **Regular** retains the same 27 configurations and AWGN setting as the training data. For each configuration, AWGN with SNRs [0, 5, 10, 15, 20, 25] dB is evaluated, resulting in a total of $27 \times 6 = 162$ scenarios. For each scenario, 100 samples are generated (this sample size is consistently used across all subsequent testing scenarios and will not be restated), resulting in a total of 16,200 pairs of historical and future CSI instances. This setup provides an exact in-distribution evaluation of model performance.
- **Robustness** keeps the same 27 configurations as training, but replaces the AWGN assumption with three realistic noise types. This setting evaluates the robustness of the model against realistic noises. The total number of scenarios in *Robustness* is $27 \times (4 + 4 + 10) = 486$, detailed as follows:
 - **Phase noise (Appendix D-B1):** simulates irregular fluctuations in the channel phase. Evaluated at [10, 15, 20, 25] dB.

- **Burst noise (Appendix D-B2):** models short, high-amplitude, pulse-like disturbances in the channel. Evaluated at [10, 15, 20, 25] dB.
- **Packet drop noise (Appendix D-B3):** represents random erasures of CSI matrices within the sequence. Evaluated at drop rates [0.01, 0.02, ..., 0.10].
- **Generalization** retains the AWGN setting used in training but extends the channel model, delay spread, and user velocity configurations to evaluate the model’s generalization capability. The final dataset includes 5 channel models, 6 delay spreads, and 17 user velocities. For each of the $5 \times 6 \times 17$ configurations, AWGN is evaluated at SNRs [0, 5, 10, 15, 20, 25] dB. Accordingly, the total number of scenarios in *Generalization* is $5 \times 6 \times 17 \times 6 = 3,060$.
- **Channel model:** According to [53,55], CDL-A/B/C correspond to Non-Line-of-Sight (NLOS) channels, where signals reach the receiver through reflection, scattering, and diffraction, whereas CDL-D/E represent Line-of-Sight (LOS) channels characterized by a dominant direct path between the BS and UE. In the experimental setup, CDL-A/C/D are included during training while CDL-B/E are reserved for evaluation, enabling the model to learn from both channel types and to demonstrate its generalization across NLOS and LOS conditions.
- **Delay spread:** The training dataset includes delay spreads of 30, 100, and 300 ns, while additional values of 50, 200, and 400 ns are used for evaluation. This allows assessment with both within-range values (50 ns and 200 ns) and an outside-range value (400 ns).
- **User velocity:** In addition to the training velocities of 1, 10, and 30 m/s, the dataset includes velocities 3, 6, 9, ..., 45 m/s. This setup enables evaluation both within the training range (velocities below 30 m/s) and outside the training range (velocities above 30 m/s), similar to the delay-spread configuration.

B. Baseline Models

To evaluate the proposed model, this study compares against the following baselines:

- **LLM4CP [29]:** Together with CSI-BERT2 [30], LLM4CP is a representative example of recent CSI predictors that leverage (pre-trained) large language model (LLM) layers. LLM4CP is included due to reported strong generalization and publicly available code.
- **STEMGNN [28]:** An advanced deep learning architecture that incorporates a graph neural network (GNN) component to capture spatiotemporal dependencies. The authors of [28] applied STEMGNN to CSI prediction and demonstrated strong performance. It should be noted that [28] integrated the encoder-decoder structure from STNet [56], allowing STEMGNN to operate in the latent space between the encoder and decoder. For a fair comparison, this study applies the STEMGNN predictor directly, without incorporating the additional encoder-decoder structure.
- **RNN [57]:** A recurrent neural network baseline that models temporal dependencies; among the earliest deep

learning approaches introduced for CSI prediction, showing promising results.

- **CNN [26]:** A convolutional baseline that leverages structural similarity between CSI tensors and images, particularly for FDD scenarios.
- **No Prediction (NP):** A naive baseline that repeats the last observed CSI across the four-step prediction horizon (i.e., persistent model). This provides a direct measure of channel aging and offers an intuitive understanding of task difficulty across scenarios.

Model-based algorithms are not included as baselines for two reasons. First, extensive prior work has already compared model-based and deep learning approaches, with a consistent conclusion that deep learning models are superior for high-dimensional, complex CSI prediction [29,57,58]. Second, initial experiments with PAD [18] indicate prohibitive computational cost: single-sample inference (batch size = 1) requires 800-1300 ms (approximately 100× slower than deep learning baselines), making it impractical for comparison over *CSI-RRG*.

C. Training Configuration

Hyperparameter tuning is performed with Optuna [59] for *CSI-4CAST* and all baselines in both TDD and FDD modes. For each (model, duplexing) pair, an Optuna study explores a predefined search space for up to 30 hours. Depending on the model's complexity, between 1 and 3 NVIDIA H200 GPUs are allocated in parallel to accelerate the search. Each trial is budgeted for 10–20 training epochs, with the length chosen empirically according to the typical convergence speed of the model. Trials are evaluated on the validation set, recording accuracy (validation NMSE) together with efficiency in terms of FLOPs. The detailed hyperparameter ranges and training configurations are provided in Appendix F.

After tuning, all trial outcomes are mapped to the accuracy-efficiency plane, and the set of *non-dominated* configurations is extracted as the Pareto frontier. Every configuration on this frontier is then *retrained from scratch* under a full schedule: at most 50 epochs with early stopping. In practice, models typically converge and stop between 15 and 30 epochs, so the actual training duration does not deviate substantially from the trial budgets. For each (model, duplexing) setting, the final checkpoint is the one achieving the lowest validation NMSE among these full trainings. The Pareto frontier approach is employed to explicitly capture the trade-off between computational complexity and predictive performance.

D. Evaluation Metrics

In order to evaluate the performance of the proposed model on *CSI-RRG*, the following metrics are employed:

- **NMSE** As shown in (7), NMSE serves as a straightforward numerical indicator of prediction performance. Moreover, the increasing or decreasing trend of NMSE under changes in the channel conditions offers intuitive insights into the mechanisms by which different factors influence prediction performance.
- **Spectral Efficiency (SE)** In addition to NMSE, SE (Appendix E) quantifies the practical performance of the

predicted CSI in terms of achievable data rate. It measures how prediction accuracy translates into overall system efficiency.

- **Rank Score and the Percentage of Models with Rank 1** The evaluation dataset contains thousands of diverse scenarios that vary substantially in difficulty. As a result, NMSE values are not directly comparable across scenarios—for instance, the NMSE on CDL-A may be an order of magnitude larger than on CDL-D under identical settings (Table V). Consequently, averaging NMSE across scenarios may introduce bias toward the harder cases and complicate fair comparisons (see Sections V-B and V-C). To mitigate this issue, the scenario-wise rank distribution is introduced to better reflect model performance within specific subsets of scenarios, and is defined as:

$$\text{rank}(\pi, \mathbf{s}) \in \{1, \dots, |\Pi|\}, \quad (16)$$

where π denotes the model, $\mathbf{s}$ the scenario, and Π the set of all models. Based on this definition, for a selected subset of scenarios $\mathbf{S}' \subset \mathbf{S}$, the mean rank score and the percentage of rank-1 occurrences are introduced to summarize overall model performance:

$$\text{MeanRank}(\pi, \mathbf{S}') \quad (17a)$$

$$= \frac{1}{|\mathbf{S}'|} \sum_{\mathbf{s} \in \mathbf{S}'} \text{rank}(\pi, \mathbf{s}) \in [1, |\Pi|] \quad (17b)$$

$$\text{RankScore}(\pi, \mathbf{S}') \quad (18a)$$

$$= |\Pi| - \text{MeanRank}(\pi, \mathbf{S}') \in [0, |\Pi| - 1] \quad (18b)$$

$$\mathbf{P}_{\text{rank1}}(\pi, \mathbf{S}') \quad (19a)$$

$$= \frac{1}{|\mathbf{S}'|} \sum_{\mathbf{s} \in \mathbf{S}'} \mathbf{1}_{\{\text{rank}(\pi, \mathbf{s}) = 1\}} \in [0, 1] \quad (19b)$$

Here, $\mathbf{S}'$ is assigned to the *Regular*, *Robustness*, and *Generalization* sets, so that RankScore and $\mathbf{P}_{\text{rank1}}$ reflect performance in the corresponding evaluation tracks. For both metrics, larger values indicate stronger performance.

- **Efficiency** Model efficiency is evaluated in terms of FLOPs, total parameters, and inference time. The efficiency score is defined as the normalized improvement relative to the most resource-demanding model on each metric. Let c denote the cost metric, with $c \in \{\text{FLOPs, Total Params, Inference Time}\}$. The efficiency score is then given by

$$\text{EffScore}(\pi, c) \quad (20a)$$

$$= 1 - \frac{c(\pi)}{\max_{\pi \in \Pi} c(\pi)} \in [0, 1]. \quad (20b)$$

By design, larger values of the efficiency score indicate stronger performance, namely better computational efficiency, consistent with the definitions of RankScore and $\mathbf{P}_{\text{rank1}}$.

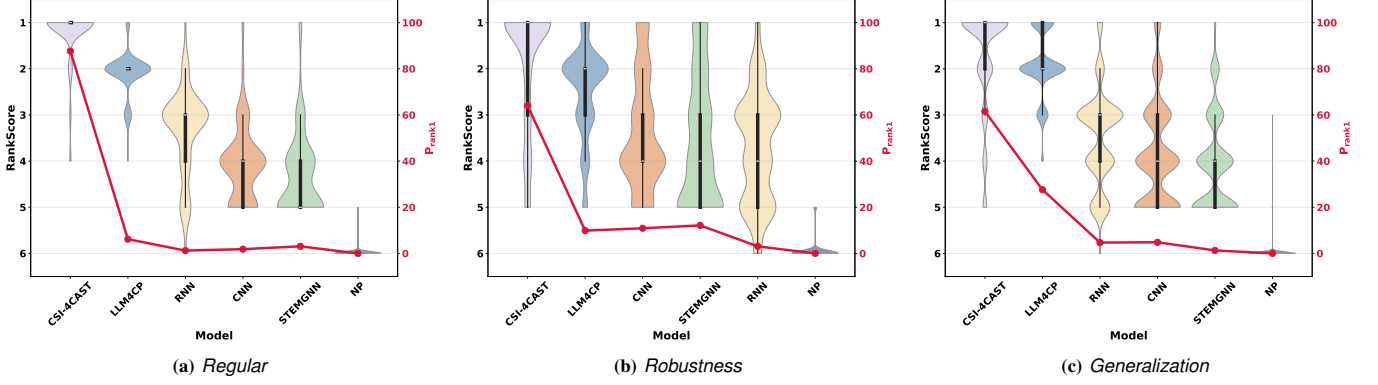


Fig. 5. TDD: NMSE rank distribution of *Regular*, *Robustness*, and *Generalization*. Within each panel, models are ordered left to right by their mean rank, MeanRank in (17) (lower is better). Rank distributions are shown as violin plots, while top-1 percentages, P_{rank1} in (19), are plotted as a red line graph.

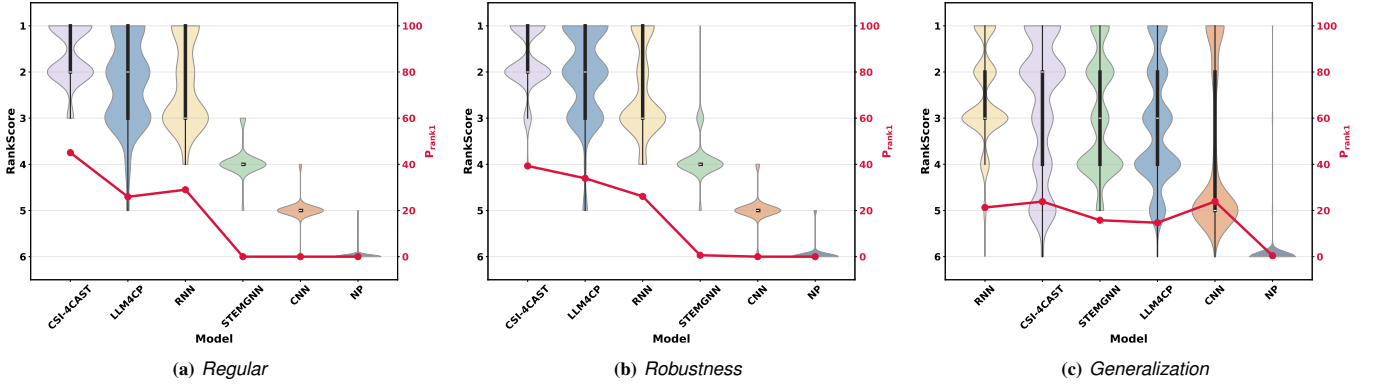


Fig. 6. FDD: NMSE rank distribution of *Regular*, *Robustness*, and *Generalization*. The plotting conventions follow those in Fig. 5.

V. PERFORMANCE EVALUATION

This section presents the experimental results, comparing the proposed model with baseline methods for CSI prediction across a comprehensive set of testing scenarios. As outlined in Section IV-A, the scenarios generated in *CSI-RRG* are grouped into three sets: testing under the same distribution as training (*Regular*), testing under realistic noise conditions (*Robustness*), and testing on unseen scenarios (*Generalization*). Section V-A reports the rank-based evaluation of the models on the *Regular*, *Robustness*, and *Generalization* sets; Section V-B presents the NMSE-based evaluation across varying SNRs, user velocities, delay spreads, and channel models; Section V-C focuses on NMSE under various realistic noise conditions; and Section V-D provides an overall assessment of the models, considering the trade-off between prediction performance and computational overhead.

Both TDD and FDD duplexing modes are evaluated, and the results are reported separately. This separation is necessary due to the substantial differences in the prediction tasks (Section II) and the independent training of models for the two duplexing modes (Section IV-C).

A. Rank-Based Evaluation: Regular, Robustness, & Generalization

This section summarizes the model comparisons across the *Regular*, *Robustness*, and *Generalization* tracks in Fig. 5 for

TDD and Fig. 6 for FDD. For each track, both the scenario-wise rank distribution and the top-1 rank percentage, P_{rank1} in (19), are reported. The MeanRank in (17) is also reflected in each panel, with models arranged from left to right in ascending order.

For TDD system (Fig. 5), the proposed *CSI-4CAST* achieves the best MeanRank and P_{rank1} across all three scenarios, consistently outperforming the baselines. Under *Regular* testing, it achieves a MeanRank of 1.18 and the P_{rank1} of 88.9%. Under *Robustness* testing, the *CSI-4CAST* attains a MeanRank of 1.86 and a P_{rank1} of 64%. For *Generalization* testing, it achieves a MeanRank of 1.72 and a P_{rank1} of 61.2%. The performance margin over LLM4CP (the next-best model) in P_{rank1} is 83.3%, 54.5%, and 33.3% for *Regular*, *Robustness*, and *Generalization*, respectively. Moreover, the proposed *CSI-4CAST* exhibits narrower and more concentrated rank distributions across all three tracks, highlighting its leading position in all tracks. In contrast, while LLM4CP shows stable rank distributions in *Regular* and *Generalization*, its performance in the *Robustness* track is scattered. Other deep learning baselines, including CNN, STEMGNN, and RNN, generally spread across ranks 3-5 in both the *Robustness* and *Generalization* tracks.

For FDD system (Fig. 6), the proposed *CSI-4CAST* achieves the highest MeanRank in both *Regular* and *Robustness* scenarios, with a MeanRank of 1.62 and a P_{rank1} of 43.8% in *Regular* testing, and a MeanRank of 1.69 and a P_{rank1} of

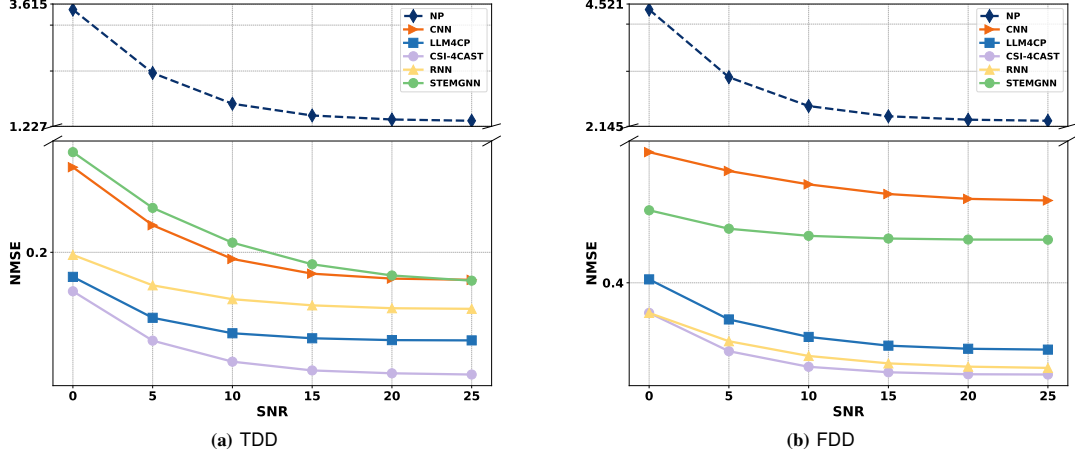


Fig. 7. **NMSE under varying SNR (dB).** Model performance under (a) TDD and (b) FDD is evaluated across SNR from 0 to 25 dB using AWGN.

40.7% in *Robustness* testing. Under *Generalization* testing, however, all models exhibit significant degradation compared to TDD. Due to the inherent difficulty of FDD prediction, which requires inter-band inference, none of the methods achieve reliable generalization to unseen scenarios, resulting in wide rank distributions across all models.

The overall comparison between TDD and FDD underscores the greater complexity of cross-band prediction in FDD and highlights the need for scenario-specific modeling and adaptive training strategies, such as distribution-shift detection and retraining [60,61], to ensure reliable inference performance.

B. NMSE-Based Evaluation: SNR, User Velocity, Delay Spread, & Channel Model

This section reports the NMSE comparisons between the proposed *CSI-4CAST* and the baselines across various factors. Specifically, Sections V-B1–V-B4 analyze sensitivity to (i) AWGN SNR, (ii) user velocity, (iii) channel model, and (iv) delay spread, respectively. The objective is to provide a detailed examination of how each factor influences the CSI prediction performance across different models.

Unless stated otherwise, when one factor is examined, performance is *aggregated over all remaining factors*. For example, in Section V-B1, for each SNR the NMSEs are averaged across all channel models, delay spreads, and user velocities; the same convention applies in the other sections.

1) *Performance Under Varying AWGN SNRs*: Fig. 7 presents NMSE versus AWGN SNR injected into the *input* historical CSI for TDD and FDD. This setup reflects practical conditions in which the observed CSI is corrupted by varying levels of AWGN, requiring prediction models to operate reliably across all such cases. Across all models and both duplexing modes, *CSI-4CAST* consistently achieves the lowest NMSE at every SNR, demonstrating strong robustness to input AWGN.

Comparing TDD and FDD performance, NMSE is consistently higher under FDD. The larger NMSE of NP in FDD directly reflects that prediction under FDD is intrinsically more challenging than under TDD, consistent with the analysis in Section II, where FDD was identified as an inter-band prediction

task requiring prediction across both time and frequency. Similarly, all other deep learning models exhibit higher NMSE in FDD than in TDD. For example, at SNR = 0 dB, the deep learning models (*CSI-4CAST*, *LLM4CP*, *CNN*) achieve roughly 2× higher NMSE in FDD than in TDD.

2) *Performance Under Varying User Velocities*: Fig. 8 reports NMSE versus user velocity for both TDD and FDD. The training dataset and the *Regular* track include velocities of 1, 10, and 30 m/s, marked by the red dashed vertical lines. The *Generalization* track extends to velocities from 3 to 45 m/s, covering both interpolation (velocities within the training range, shaded light green) and extrapolation (velocities outside the training range, shaded light red). Considering user velocity is crucial in the current mobile communication era, as UEs operate across a wide range of speeds, from walking to driving or transporting with various speeds.

Across all duplexing modes and user velocities, *CSI-4CAST* achieves the lowest NMSE or matches the next best baseline. For both TDD and FDD, NMSE increases with user velocity, consistent with high-speed-induced temporal *decorrelation*, which reduces the predictability of future CSI. This interpretation is supported by the autocorrelation analysis in Appendix B.

Comparing duplexing modes, FDD consistently produces higher NMSE than TDD. However, the performance degradation from lowest to highest user speed is more severe in TDD. For *CSI-4CAST* and baselines, NMSE increases by more than 10× between 1 m/s and 45 m/s, whereas in FDD the increase is less than 3× over the same range. The results imply that TDD prediction leverages temporal channel correlation more directly, and as coherence time shortens rapidly with high velocity, its performance becomes significantly degraded. In contrast, the FDD task is inherently more challenging, its prediction accuracy is already limited at low user speeds, so the relative degradation with increasing velocity appears less pronounced.

3) *Performance Under Varying Delay Spreads*: Table IV reports NMSE across various delay spreads: 30, 100, and 300 ns, which are included in training; 50 and 200 ns, used for interpolation within the training range; and 400 ns, used

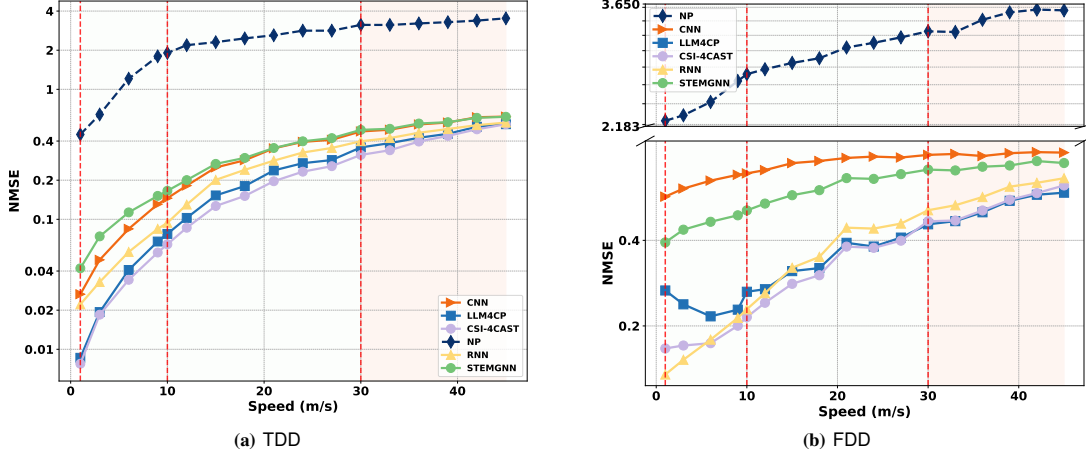


Fig. 8. **NMSE across user velocities.** Red dashed vertical lines mark the velocities included in the *regular* set; all other velocities belong to the *generalization* set. Light green shading denotes the interpolation region (velocities within the *regular* range), whereas light red denotes the extrapolation region (velocities outside that range).

for extrapolation beyond the training range. The delay spread reflects the multipath richness of the channel, which varies significantly across environments (e.g., urban, indoor, rural), motivating the need for evaluation under diverse delay spreads.

For TDD systems, *CSI-4CAST* achieves the lowest NMSE on the *Regular* track and remains the best or a close second-best on the *Generalization* track. The deep learning models generally show degraded performance on unseen delay spreads. In particular, at 400 ns (outside the training range), *CSI-4CAST*, LLM4CP, and RNN experience substantial performance drops compared with 300 ns, the nearest seen condition. For example, NMSE increases by about 50% for LLM4CP and nearly 100% for RNN and *CSI-4CAST*. These results highlight the challenge of generalizing to unseen delay spreads.

For FDD systems, *CSI-4CAST* achieves the second-best NMSE on the *Regular* track for the seen delay spreads. On the *Generalization* track, however, performance degrades substan-

tially for the unseen delay spreads. At 400 ns, the degradation is particularly severe: NMSE increases by more than 5× for *CSI-4CAST*, LLM4CP, and RNN. This sharper degradation in FDD arises from the added challenge of cross-frequency prediction. Since delay spread is inversely proportional to coherence bandwidth, a larger delay spread reduces frequency-domain correlation, making inter-band prediction more difficult in the FDD setting.

The overall results across different delay spreads underscore that generalizing to unseen delay spreads remains challenging for deep learning models. One reason is the complex and intricate influence of delay spread on channel properties. Unlike user velocity, which primarily smoothly affects temporal ACF, delay spread interacts with the channel model to jointly shape the temporal-frequency ACF, as supported by the analysis in Appendix C. The dynamics of multipath structure and scattering geometry are difficult to model and predict, leading to poor generalization. This observation highlights the importance of developing more scenario-specific models, particularly for FDD.

4) *Performance Under Varying Channel Models:* Table V presents NMSE results across different channel models for both TDD and FDD. CDL-A/C/D are included in training (*Regular* track), while CDL-B and CDL-E are used for generalization (*Generalization* track). CDL-A/B/C typically represent *NLOS* conditions with diffuse multipath and rich angular dispersion, often corresponding to dense urban streets or cluttered indoor offices. In contrast, CDL-D and CDL-E represent *LOS* conditions with a strong dominant direct path, commonly found in open outdoor spaces or rural macro scenarios [55]. Including diverse channel models in evaluation is essential to capture real-world environmental dynamics, as mobile communication scenarios span a wide range of conditions that can be abstracted by different channel models.

CSI-4CAST is the overall best performer, ranking within the top two across all channel models for both TDD and FDD, with the only exception being FDD CDL-B. This highlights the strong prediction performance and generalization ability of *CSI-4CAST*. A detailed comparison shows that CDL-A/B/C

TABLE IV
NMSE UNDER VARYING DELAY SPREADS. **BOLD** DENOTES THE BEST VALUE AND UNDERLINE THE SECOND-BEST (LOWER NMSE IS BETTER).

Models	Regular			Generalization		
	30 ns	100 ns	300 ns	50 ns	200 ns	400 ns
TDD						
NP	2.246	1.714	1.526	1.891	1.499	1.466
CNN	0.261	0.211	0.174	0.230	0.173	<u>0.172</u>
STEMGNN	0.253	0.230	0.212	0.222	0.207	0.214
RNN	<u>0.191</u>	0.168	0.156	0.176	0.164	0.282
LLM4CP	0.195	<u>0.145</u>	<u>0.103</u>	<u>0.163</u>	0.110	0.141
<i>CSI-4CAST</i>	0.176	0.125	0.084	0.148	<u>0.119</u>	0.174
FDD						
NP	2.798	2.920	2.488	3.045	2.312	2.025
CNN	0.631	0.800	0.626	0.744	0.797	<u>0.994</u>
STEMGNN	0.493	0.597	0.523	0.562	<u>0.818</u>	0.894
RNN	0.301	0.318	0.255	0.425	0.930	1.330
LLM4CP	0.334	0.467	0.184	0.557	1.003	1.421
<i>CSI-4CAST</i>	<u>0.316</u>	<u>0.342</u>	<u>0.190</u>	<u>0.495</u>	0.976	1.417

TABLE V
NMSE UNDER VARYING CHANNEL MODELS. **BOLD** DENOTES THE BEST VALUE
AND UNDERLINE THE SECOND-BEST (LOWER NMSE IS BETTER).

Models	Regular			Generalization	
	CDL-A	CDL-C	CDL-D	CDL-B	CDL-E
TDD					
NP	2.167	1.948	1.372	1.902	1.353
CNN	0.246	0.335	0.065	0.405	0.074
STEMGNN	0.283	0.364	0.049	0.453	0.058
RNN	0.207	0.263	0.044	0.394	0.057
LLM4CP	<u>0.168</u>	<u>0.245</u>	<u>0.031</u>	0.349	<u>0.043</u>
CSI-4CAST	0.156	0.207	0.022	<u>0.376</u>	0.036
FDD					
NP	3.857	2.935	1.415	3.029	1.419
CNN	0.903	0.925	0.230	1.091	0.278
STEMGNN	0.677	0.855	0.081	<u>1.107</u>	0.106
RNN	0.367	0.443	0.063	1.190	0.091
LLM4CP	0.498	<u>0.431</u>	<u>0.059</u>	1.268	0.106
CSI-4CAST	<u>0.385</u>	0.410	0.052	1.308	<u>0.092</u>

(NLOS) are more challenging than CDL-D/E (LOS) across all models and duplexing modes, with NLOS yielding NMSE values an order of magnitude larger than LOS in both the *Regular* and *Generalization* tracks. This observation aligns with the above analysis of the channel characteristics: complex scattering and rapid fluctuations in NLOS conditions increase prediction difficulty.

In particular, FDD with CDL-B poses an especially difficult

generalization task, as the combination of FDD and NLOS amplifies prediction challenges. Strong models such as RNN, LLM4CP, and *CSI-4CAST*, which perform best under other conditions, show severe degradation. Compared with CDL-A/C, NMSE in CDL-B increases by more than 100% for all three models. The highly complex and distinctive properties of NLOS channels make deep learning models prone to overfitting during training and hinder generalization to unseen conditions.

C. Robustness Analysis Across Various Noise Types and Degrees

This section evaluates the robustness of *CSI-4CAST* and the baselines under three realistic noise types: *phase noise*, *burst-type corruption*, and *packet drops*, which are previously introduced in Section I. Complete definitions, visualizations, parameter ranges, generation procedures, and experiment details for all additional noises are provided in Appendix D.

The NMSE of the models across different noise types are presented in Fig. 9 for both TDD and FDD. Across all noise types and levels, *CSI-4CAST* consistently achieves the lowest NMSE, demonstrating the strongest robustness. Interestingly, CNN performs well under packet drop noise, particularly at high drop probabilities, while *CSI-4CAST*, which has CNN at the front end, shows the best overall robustness. These results support the intuition that CNN-style residual representations are effective for handling channel corruption and extracting structural features [43, 44]. Besides, the performance degradation due to packet drop noise is also more severe in

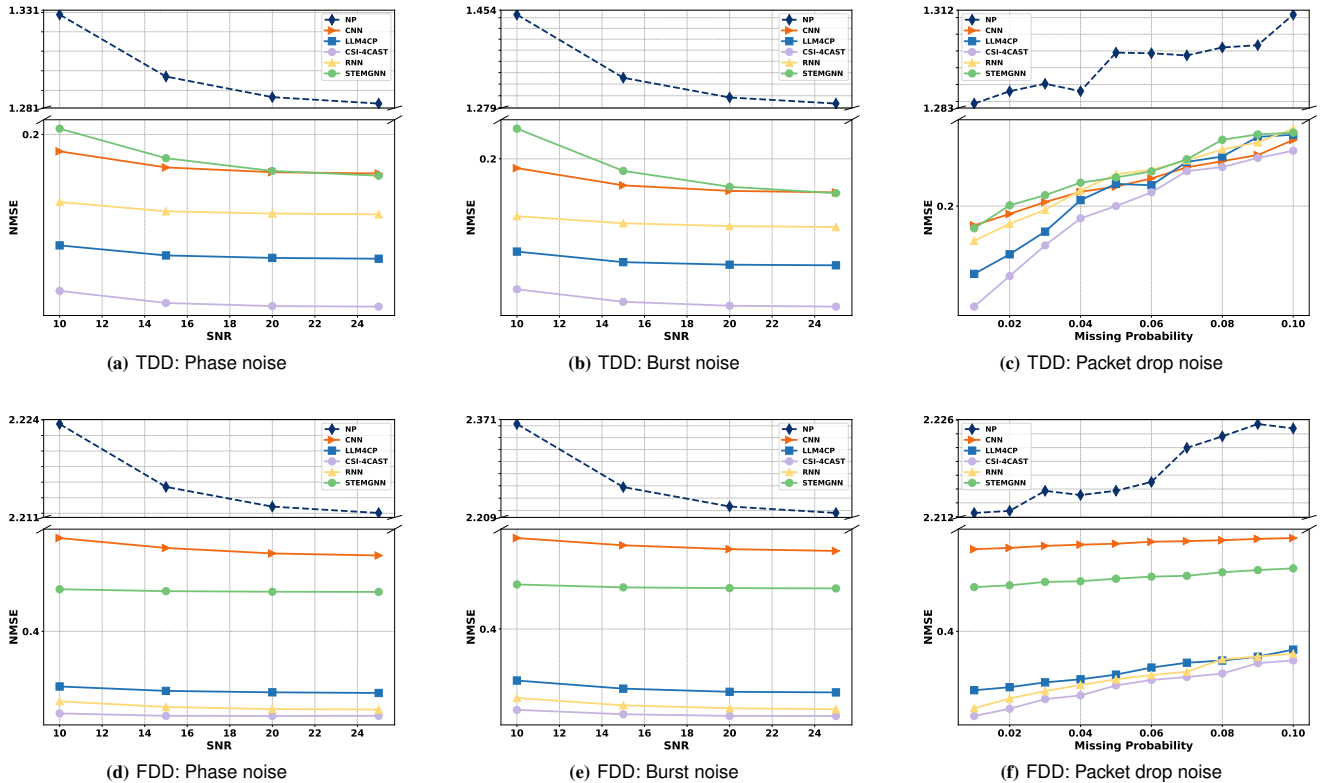


Fig. 9. NMSE under varying realistic additional noises: TDD and FDD.

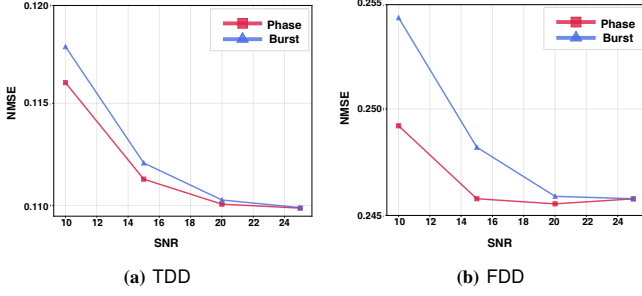


Fig. 10. *CSI-4CAST*: NMSE under Phase and Burst Noise.

TDD than in FDD, consistent with the expectation that temporal correlation is more critical in TDD.

Moreover, Fig. 10 compares *CSI-4CAST*'s performance under phase noise and burst noise at matched SNRs. Despite equal SNR, the two noise types have different effects: burst noise causes greater NMSE degradation than phase noise. Phase noise introduces smooth, continuous perturbations that partly resemble the AWGN used in training, keeping *CSI-4CAST* relatively stable. By contrast, burst noise is abrupt and high-energy over short windows, disrupting the temporal coherence that *CSI-4CAST* relies on and leading to sharper performance losses. These results highlight that robustness depends not only on SNR level but also on the structure of the noise.

D. Overall Performance: Prediction Performance and Computational Considerations

Fig. 11 illustrates the overall performance of *CSI-4CAST* compared to baseline models, jointly evaluating prediction accuracy on *CSI-RRG* and computational overhead. The prediction performance axes are presented based on the RankScore (18). Unlike earlier sections that focused solely on NMSE, both

NMSE-rank and Spectral Efficiency (SE)-rank are included here. This choice is motivated by two factors: (i) high SE is a central goal in wireless communication systems, and (ii) NMSE and SE reflect distinct aspects of model performance, where a low NMSE does not necessarily correspond to high SE. Additional details and a breakdown of SE performance are provided in Appendix E. NMSE-rank and SE-rank are reported across the *Regular*, *Robustness*, and *Generalization* tracks. The orange axes reflect computational efficiency, assessed via the EffScore (20), which accounts for FLOPs, total parameters, and inference time. Notably, RankScore and EffScore are defined such that higher values indicate better performance. Thus, in the figure, larger values along individual axes represent stronger performance in that aspect, and a larger overall polygon reflects a more favorable trade-off between accuracy and efficiency.

For TDD, *CSI-4CAST* achieves the strongest overall performance, dominating all three evaluation tracks and both performance metrics while maintaining significantly lower computational complexity. Only LLM4CP provides competitive performance on *Generalization* w.r.t. NMSE-rank, whereas all other baselines fall consistently behind across every track and both metrics. In terms of efficiency, *CSI-4CAST* is particularly advantageous: its FLOPs are comparable to CNN and only about 1/5 of LLM4CP; its parameter count is reduced to roughly 1/7 of LLM4CP; and its inference time is similar to RNN, requiring only about 1/2 as long as LLM4CP. Detailed efficiency statistics are provided in Appendix A.

For FDD, *CSI-4CAST* continues to lead on the *Regular* and *Robustness* tracks w.r.t. both metrics, although with smaller margins. LLM4CP performs closely, and RNN also shows competitive performance—matching LLM4CP on *Regular* and trailing slightly on *Robustness*. On the *Generalization* track, all models exhibit notable performance degradation due to the inherent difficulty of inter-band prediction. For NMSE-rank, the

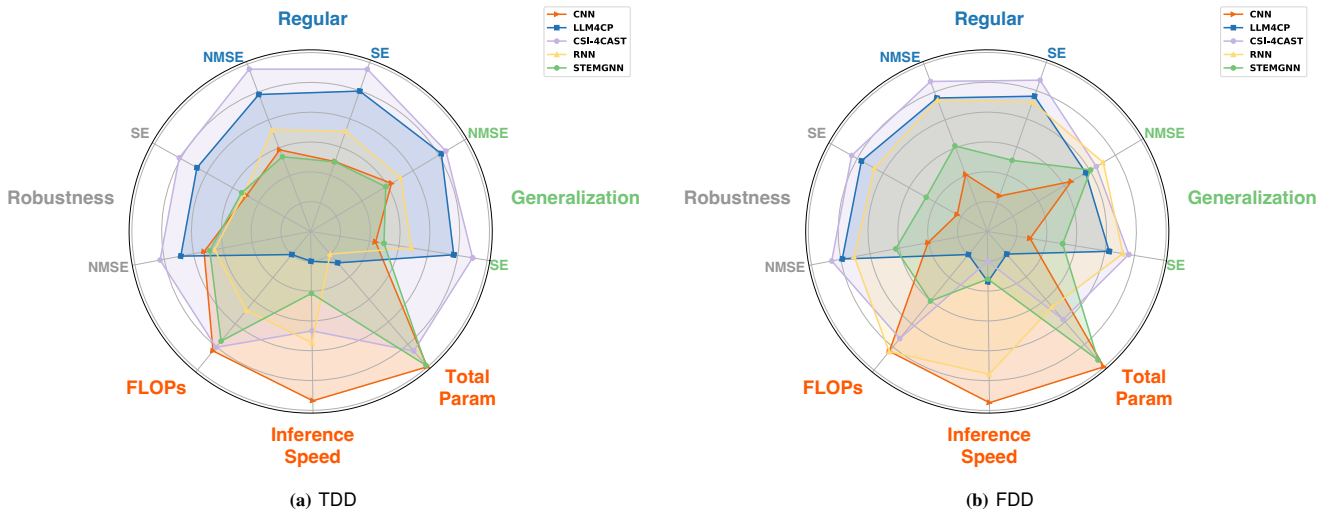


Fig. 11. **Overall performance.** The blue, gray, and green axes represent prediction performance for *Regular*, *Robustness*, and *Generalization* tracks, measured using the RankScore (18), with separate axes for results based on NMSE-rank and Spectral Efficiency (SE)-rank. The orange axes indicate computational cost, quantified by the EffScore (20). Each colored polygon corresponds to a model. By construction, larger scores correspond to better results; hence, values farther from the center indicate stronger performance along that axis. A larger polygon therefore reflects a more favorable overall accuracy-efficiency trade-off. Since axes are scaled independently, comparisons should be made *only along the same axis*.

RankScores are tightly clustered, with CNN as the outlier. For SE-rank, *CSI-4CAST*, LLM4CP, and RNN form a leading group with relatively close performance, clearly separated from the remaining models. These results underscore that generalization in FDD remains a significant challenge, with no model demonstrating clear superiority. Furthermore, the noticeable differences between SE-rank and NMSE-rank—particularly on the *Generalization* track—highlight the distinct characteristics captured by each metric. From an efficiency perspective, *CSI-4CAST* requires about 1/2 the FLOPs of STEMGNN and 1/3 of LLM4CP, with parameter count comparable to RNN and nearly 1/2 of LLM4CP. However, due to the additional subcarrier-wise ACL layer and heavier hyperparameterization in FDD, *CSI-4CAST*'s inference time aligns more closely with LLM4CP and STEMGNN and is slightly slower, in contrast to the clear speed advantage observed in TDD.

VI. CONCLUSION

This paper introduced *CSI-4CAST*, a lightweight hybrid deep learning model for CSI prediction, together with *CSI-RRG*, a large-scale benchmark comprising more than 300,000 instances across 3,060 scenarios for both training and evaluation. Experimental results demonstrated that *CSI-4CAST* consistently outperforms baseline models across diverse testing conditions in both TDD and FDD. Specifically, *CSI-4CAST* achieved the lowest NMSE in 88.9% of TDD scenarios and 43.8% of FDD scenarios, while reducing FLOPs by factors of 5 and 3 compared with the strongest competitor in TDD and FDD, respectively.

The detailed scenario-level analysis provided critical insights into the role of channel parameters in CSI prediction. User velocity and SNR variations produced smooth and predictable performance changes, whereas shifts in channel models had a far more significant effect. Performance clustered clearly by propagation condition, with NLOS channels posing the greatest challenge—exhibiting up to a 10× increase in NMSE due to angular dispersion and the lack of a dominant propagation path. Delay spread was also shown to have a strong influence, with performance at 400 ns (outside the training range) dropping by 50% in TDD and over 500% in FDD.

The comparison between duplexing modes underscored the inherent difficulty of FDD prediction, where inter-band mapping severely limits generalization. All models exhibited a substantial drop from TDD to FDD, and none generalized well to unseen FDD scenarios. This finding highlights the need for future research on adaptive and active learning strategies to detect distribution shifts and dynamically update models in real time.

The robustness analysis further emphasized that prediction performance depends not only on noise level but also on noise structure. Burst noise caused more significant degradation compared to phase noise, due to its abrupt, spike-like distortions. Packet-drop noise affected TDD more severely than FDD, highlighting the importance of temporal continuity in intra-band prediction.

Overall, the results of this work advance the development of robust, efficient, and generalizable CSI prediction for next-generation wireless communication systems. Future research

will extend this line of work by exploring active learning-based frameworks to enable more adaptive, reliable, and self-sustained CSI prediction in practical deployments.

ACKNOWLEDGMENTS

This research was partly supported by NSF award 2112533.

REFERENCES

- [1] R. Chataut and R. Akl, "Massive mimo systems for 5g and beyond networks—overview, recent trends, challenges, and future research direction," *Sensors*, vol. 20, no. 10, 2020. [Online]. Available: <https://www.mdpi.com/1424-8220/20/10/2753>
- [2] T. L. Marzetta, E. G. Larsson, H. Yang, and H. Q. Ngo, *Fundamentals of Massive MIMO*. Cambridge University Press, 2016.
- [3] E. Björnson, J. Hoydis, and L. Sanguinetti, 2017.
- [4] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and spectral efficiency of very large multiuser mimo systems," *IEEE Transactions on Communications*, vol. 61, no. 4, pp. 1436–1449, 2013.
- [5] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive mimo has unlimited capacity," *IEEE Transactions on Wireless Communications*, vol. 17, no. 1, pp. 574–590, 2018.
- [6] E. Björnson, L. Sanguinetti, J. Hoydis, and M. Debbah, "Optimal design of energy-efficient multi-user mimo systems: Is massive mimo the answer?" *IEEE Transactions on Wireless Communications*, vol. 14, no. 6, pp. 3059–3075, 2015.
- [7] K. Huang, R. W. Heath, and J. G. Andrews, "Limited feedback beamforming over temporally-correlated channels," *Trans. Sig. Proc.*, vol. 57, no. 5, pp. 1959–1975, May 2009. [Online]. Available: <https://doi.org/10.1109/TSP.2009.2014272>
- [8] J. Jose, A. Ashikhmin, T. L. Marzetta, and S. Vishwanath, "Pilot contamination and precoding in multi-cell tdd systems," *IEEE Transactions on Wireless Communications*, vol. 10, no. 8, pp. 2640–2651, 2011.
- [9] H. Nguyen, J. Andersen, and G. Pedersen, "Capacity and performance of MIMO systems under the impact of feedback delay." 2004 IEEE 15th International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE Cat. No.04TH8754), 1 2005. [Online]. Available: <https://doi.org/10.1109/pimrc.2004.1370835>
- [10] K. T. Truong and R. W. Heath, "Effects of channel aging in massive MIMO systems," *Journal of communications and networks*, vol. 15, no. 4, pp. 338–351, 8 2013. [Online]. Available: <https://doi.org/10.1109/jcn.2013.000065>
- [11] S. Gao, X. Cheng, and L. Yang, "Estimating doubly-selective channels for hybrid mmwave massive mimo systems: A doubly-sparse approach," *IEEE Transactions on Wireless Communications*, vol. 19, no. 9, pp. 5703–5715, 2020.
- [12] M. Li, M. Lin, W.-P. Zhu, Y. Huang, A. Nallanathan, and Q. Yu, "Performance analysis of MIMO MRC systems with feedback delay and channel estimation error," *IEEE transactions on vehicular technology*, vol. 65, no. 2, pp. 707–717, 2 2016. [Online]. Available: <https://doi.org/10.1109/tvt.2015.2404820>
- [13] T. Eyceoz, A. Duel-Hallen, and H. Hallen, "Deterministic channel modeling and long range prediction of fast fading mobile radio channels," *IEEE communications letters*, vol. 2, no. 9, pp. 254–256, 9 1998. [Online]. Available: <https://doi.org/10.1109/4234.718494>
- [14] Z. Shen, J. Andrews, and B. Evans, "Short range wireless channel prediction using local information," in *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers*, 2003, vol. 1, 2003, pp. 1147–1151 Vol.1.
- [15] 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.
- [16] S. Kashyap, C. Mollén, E. Björnson, and E. G. Larsson, "Performance analysis of (tdd) massive mimo with kalman channel prediction," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 3554–3558.
- [17] 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, 1 2021. [Online]. Available: <https://doi.org/10.1109/tcomm.2020.3027882>
- [18] 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.

- [19] H. Kim, J. Choi, and D. J. Love, "Machine learning for future wireless communications: Channel prediction perspectives," 2025. [Online]. Available: <https://arxiv.org/abs/2502.18196>
- [20] O. Stenhammar, G. Fodor, and C. Fischione, "A comparison of neural networks for wireless channel prediction," 2023. [Online]. Available: <https://arxiv.org/abs/2308.14020>
- [21] W. Gardner, "Simplification of MUSIC and ESPRIT by exploitation of cyclostationarity," *Proceedings of the IEEE*, vol. 76, no. 7, pp. 845–847, 7 1988. [Online]. Available: <https://doi.org/10.1109/5.7152>
- [22] C. Jiang, J. Guo, X. Li, S. Jin, and J. Zhang, "Ai for csi prediction in 5g-advanced and beyond," 2025. [Online]. Available: <https://arxiv.org/abs/2504.12571>
- [23] R. Adeogun, "Toward intelligent fading channel prediction: A comprehensive survey," *IEEE Access*, vol. 13, pp. 111 260–111 281, 2025.
- [24] I. Helmy, P. Tarafder, and W. Choi, "Lstm-gru model-based channel prediction for one-bit massive mimo system," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 8, pp. 11 053–11 057, 2023.
- [25] 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 communications letters*, vol. 10, no. 2, pp. 266–270, 2 2021. [Online]. Available: <https://doi.org/10.1109/lwc.2020.3027774>
- [26] M. S. Safari, V. Pourahmadi, and S. Sodagari, "Deep UL2DL: Data-Driven Channel Knowledge transfer from uplink to Downlink," *IEEE open journal of vehicular technology*, vol. 1, pp. 29–44, 1 2020. [Online]. Available: <https://doi.org/10.1109/ojvt.2019.2962631>
- [27] H. Jiang, M. Cui, D. W. K. Ng, and L. Dai, "Accurate channel prediction based on transformer: making mobility negligible," *IEEE journal on selected areas in communications*, vol. 40, no. 9, pp. 2717–2732, 9 2022. [Online]. Available: <https://doi.org/10.1109/jsac.2022.3191334>
- [28] S. Mourya, P. Reddy, S. Amuru, and K. K. Kuchi, "Spectral Temporal Graph neural network for massive MIMO CSI prediction," *IEEE wireless communications letters*, p. 1, 1 2024. [Online]. Available: <https://doi.org/10.1109/lwc.2024.3372148>
- [29] B. Liu, X. Liu, S. Gao, X. Cheng, and L. Yang, "LLM4CP: Adapting Large Language Models for channel Prediction," 6 2024. [Online]. Available: <https://ieeexplore.ieee.org/document/10582829>
- [30] Z. Zhao, F. Meng, H. Li, X. Li, and G. Zhu, "Mining limited data sufficiently: A bert-inspired approach for csi time series application in wireless communication and sensing," *arXiv preprint arXiv:2412.06861*, 2024, submitted on 9 Dec 2024. [Online]. Available: <https://doi.org/10.48550/arXiv.2412.06861>
- [31] A. Pitarokoulis, S. K. Mohammed, and E. G. Larsson, "Uplink performance of time-reversal mrc in massive mimo systems subject to phase noise," *IEEE Transactions on Wireless Communications*, vol. 14, no. 2, pp. 711–723, 2015.
- [32] I. Zhilin, E. Vinogradov, and I. Akyildiz, "Effect of realistic oscillator phase noise on the performance of cell-free massive mimo systems," 2025. [Online]. Available: <https://arxiv.org/abs/2405.04099>
- [33] A. Li, Y. Wang, W. Xu, and Z. Zhou, "Performance evaluation of mimo systems in a mixture of gaussian noise and impulsive noise," in *APCC/MDMC '04. The 2004 Joint Conference of the 10th Asia-Pacific Conference on Communications and the 5th International Symposium on Multi-Dimensional Mobile Communications Proceeding*, vol. 1, 2004, pp. 292–296 vol.1.
- [34] L. Zhou, J. Dai, W. Xu, and C. Chang, "Uplink channel estimation for massive mimo systems with impulsive noise," *IEEE Communications Letters*, vol. 25, no. 5, pp. 1534–1538, 2021.
- [35] A. Morato, G. Frigo, and F. Tramarin, "Packet losses distributions in 5g networks for pmu-based monitoring systems," in *Proceedings of the 2024 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. Glasgow, United Kingdom: IEEE, May 2024.
- [36] M. Tan, M. A. Merrill, V. Gupta, T. Althoff, and T. Hartvigsen, "Are language models actually useful for time series forecasting?" *arXiv preprint arXiv:2406.16964*, 2024, accepted to NeurIPS 2024 (Spotlight). [Online]. Available: <https://doi.org/10.48550/arXiv.2406.16964>
- [37] A. A. Kalachikov and N. S. Shelkunov, "Construction and validation of analytical wireless MIMO channel models based on channel measurement data," *2018 XIV International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE)*, vol. 2007, pp. 175–179, 10 2018. [Online]. Available: <https://doi.org/10.1109/apeie.2018.8545276>
- [38] H. Li, Y. Li, S. Zhou, and J. Wang, "Static CSI extraction and application in the tomographic channel model," *China Communications*, vol. 16, no. 12, pp. 132–144, 12 2019. [Online]. Available: <https://doi.org/10.23919/jcc.2019.12.010>
- [39] Y. Ma, L. Yang, and X. Zheng, "A geometry-based non-stationary MIMO channel model for vehicular communications," *China Communications*, vol. 15, no. 7, pp. 30–38, 7 2018. [Online]. Available: <https://doi.org/10.1109/cc.2018.8424580>
- [40] "NR; Radio Resource Control (RRC); Protocol specification," 3rd Generation Partnership Project (3GPP), Technical Specification TS 38.331, 2022, release 17, Version V17.2.0. [Online]. Available: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3197>
- [41] H.-W. Liang, W.-H. Chung, and S.-Y. Kuo, "FDD-RT: A simple CSI acquisition technique via channel reciprocity for FDD massive MIMO downlink," *IEEE systems journal*, vol. 12, no. 1, pp. 714–724, 3 2018. [Online]. Available: <https://doi.org/10.1109/jsyst.2016.2556222>
- [42] L. Miretti, R. L. Cavalcante, and S. Stanczak, "FDD Massive MIMO Channel Spatial Covariance Conversion Using Projection Methods." 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 4 2018. [Online]. Available: <https://doi.org/10.1109/icassp.2018.8462048>
- [43] X. Chen, Z. Feng, J. A. Zhang, F. Gao, X. Yuan, Z. Yang, and P. Zhang, "Complex cnn csi enhancer for integrated sensing and communications," *IEEE Journal of Selected Topics in Signal Processing*, vol. 18, no. 5, pp. 752–765, 2024.
- [44] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.
- [45] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Cambridge University Press, 2024.
- [46] D. Tse and P. Viswanath, *Fundamentals of wireless Communication*, 5 2005. [Online]. Available: <https://doi.org/10.1017/cbo9780511807213>
- [47] Z. Lu, J. Wang, and J. Song, "Multi-resolution csi feedback with deep learning in massive mimo system," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.
- [48] S. Ji and M. Li, "Enhancing deep learning performance of massive mimo csi feedback," 2023. [Online]. Available: <https://arxiv.org/abs/2208.11333>
- [49] T. Chen, Y. Wang, H. Chen, Z. Zhao, X. Li, N. Piovesan, G. Zhu, and Q. Shi, "Modelling the 5g energy consumption using real-world data: Energy fingerprint is all you need," 2024. [Online]. Available: <https://arxiv.org/abs/2406.16929>
- [50] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [51] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.
- [52] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd0531c4a845aa-Paper.pdf
- [53] J. Hoydis, S. Cammerer, F. Ait Aoudia, M. Nimier-David, L. Maggi, G. Marcus, A. Vem, and A. Keller, "Sionna," 2022, <https://nvlabs.github.io/sionna/>.
- [54] J. M. Meredith, "NR; Physical channels and modulation (3GPP TS 38.211 version 15.3.0 Release 15)," 3rd Generation Partnership Project (3GPP), Tech. Rep. TS 38.211, October 2018, eTSI TS 138 211 V15.3.0 (2018-10). [Online]. Available: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3213>
- [55] "Study on channel model for frequencies from 0.5 to 100 GHz (3GPP TR 38.901 version 16.1.0 Release 16)," 3rd Generation Partnership Project (3GPP), Tech. Rep. TR 38.901, Nov. 2020, version 16.1.0. [Online]. Available: <https://cdn.standards.itech.ai/samples/59772/4d8229a2e4c149c78a8a0847e61a78f6/ETSI-TR-138-901-V16-1-0-2020-11-.pdf>
- [56] S. Mourya, S. Amuru, and K. K. Kuchi, "A spatially separable attention mechanism for massive mimo csi feedback," *IEEE Wireless Communications Letters*, vol. 12, no. 1, p. 40–44, Jan. 2023. [Online]. Available: <http://dx.doi.org/10.1109/LWC.2022.3216352>
- [57] W. Jiang and H. D. Schotten, "Neural Network-Based Fading Channel Prediction: A Comprehensive Overview," *IEEE access*, vol. 7, pp. 118 112–118 124, 1 2019. [Online]. Available: <https://doi.org/10.1109/access.2019.2937588>
- [58] —, "A comparison of Wireless Channel Predictors: Artificial Intelligence versus Kalman Filter." 2019 IEEE International

- Conference on Communications (ICC), 5 2019. [Online]. Available: <https://doi.org/10.1109/icc.2019.8761308>
- [59] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A next-generation hyperparameter optimization framework,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.
 - [60] C. Jiang, J. Guo, C.-K. Wen, and S. Jin, “Enhancing reliability in ai-based csi prediction: A proxy-based performance monitoring approach,” *IEEE Transactions on Communications*, vol. 73, no. 4, pp. 2602–2615, 2025.
 - [61] Y. Yang, F. Gao, G. Y. Li, and M. Jian, “Deep learning-based downlink channel prediction for fdd massive mimo system,” *IEEE Communications Letters*, vol. 23, no. 11, pp. 1994–1998, 2019.
 - [62] M. E. Rasekh, M. Abdelghany, U. Madhow, and M. Rodwell, “Phase noise in modular millimeter wave massive MIMO,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 10, pp. 6522–6535, 4 2021. [Online]. Available: <https://doi.org/10.1109/twc.2021.3074911>
 - [63] H. Mehrpouyan, A. A. Nasir, S. D. Blostein, T. Eriksson, G. K. Karagiannidis, and T. Svensson, “Joint estimation of channel and oscillator phase noise in MIMO systems,” *IEEE Transactions on Signal Processing*, vol. 60, no. 9, pp. 4790–4807, 6 2012. [Online]. Available: <https://doi.org/10.1109/tsp.2012.2202652>
 - [64] C. Shannon, “Communication in the presence of noise,” *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1 1949. [Online]. Available: <https://ieeexplore.ieee.org/document/1697831>

APPENDIX A COMPUTATIONAL OVERHEAD

Table VI presents the computational overhead of the proposed model in comparison with baseline models. The reported metrics include trainable parameters (M), total parameters (M), FLOPs (G), inference time (ms), and training time (ms) for both TDD and FDD duplexing modes. The FLOPs, inference time, and training time are accounted for one whole CSI sequence. Because hyperparameter tuning is performed independently for models trained on the TDD and FDD datasets, the same model may yield different results across the two duplexing modes.

TABLE VI
COMPUTATIONAL OVERHEAD ANALYSIS.

Duplexing Mode	Model	Trainable Params (M)	Total Params (M)	FLOPs (G)	Inference Time (ms)	Training Time (ms)
TDD						
	NP	0	0	0	0.042	0.042
	CNN	0.197	0.197	60.45	0.968	0.976
	RNN	156.312	156.312	190.04	6.881	6.936
	STEMGNN	2.345	2.345	91.62	11.918	12.105
	LLM4CP	4.532	144.140	366.96	15.194	15.528
	<i>CSI-4CAST</i>	21.914	21.914	71.90	8.099	10.668
FDD						
	NP	0	0	0	0.042	0.042
	CNN	0.197	0.197	60.45	0.968	0.976
	RNN	48.981	48.981	59.54	4.539	5.220
	STEMGNN	5.754	5.754	222.34	16.469	16.677
	LLM4CP	3.811	92.002	372.21	16.118	16.466
	<i>CSI-4CAST</i>	38.636	38.636	101.64	18.698	20.486

APPENDIX B

AUTOCORRELATION FUNCTION (ACF) ACROSS DIFFERENT USER VELOCITIES

This section presents the autocorrelation function (ACF) of CSI across user velocities. Figs. 12–13 report results for FDD and TDD under CDL-A with a 30 ns delay spread. For each duplexing mode and velocity, the test tensor has shape $100 \times 32 \times 20 \times 300$ (samples $\times$ antennas $\times$ timestamps $\times$ subcarriers). The data are decomposed into per-(antenna, subcarrier) time series of length 20; the sample ACF is computed for each series and then averaged across all samples, antennas, and subcarriers. The resulting mean ACF is shown as a stem plot.

Two patterns emerge. (i) At low speed, both TDD and FDD exhibit pronounced temporal correlation (slow ACF decay), with TDD showing stronger correlation than FDD. (ii) At high speed, temporal correlation diminishes rapidly in both duplexing modes.

APPENDIX C

AUTOCORRELATION FUNCTION (ACF) ACROSS DIFFERENT DELAY SPREADS

Similar to the previous section, which reported temporal ACF across different user velocities, here the temporal-frequency ACF across different delay spreads is presented. Compared with the temporal ACF in Fig. 12, the frequency ACF in Fig. 14 exhibits more significant variation across delay spreads.

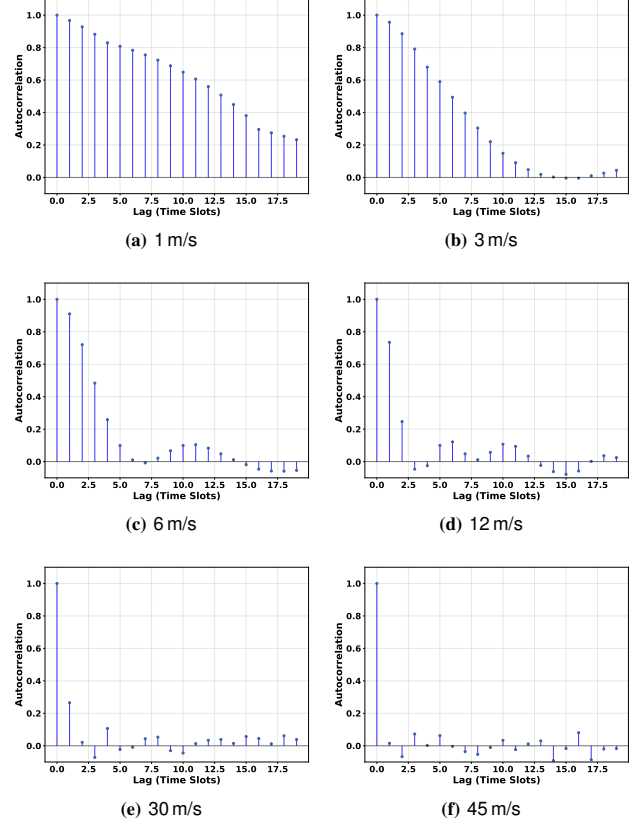


Fig. 12. ACF across different user velocities (FDD | CDL-A | 30ns)

Furthermore, as illustrated in Fig. 15 and Fig. 16, different channel models lead to distinct temporal-frequency ACF patterns across delay spreads. These results highlight that temporal-frequency ACF variation follows a more complex mechanism, jointly influenced by both the channel model and the delay spread.

APPENDIX D

ADDITIONAL NOISE

In the following, the different types of additional noise are defined. For the accuracy of the definition, the element-wise noisy CSI is defined as follows:

$$\begin{aligned} \tilde{h}_{m,k}^t &= h_{m,k}^t + e_{m,k}^t \\ \text{where } \tilde{h}_{m,k}^t &= \tilde{\mathbf{H}}^t[m, 1, k] \\ \text{and } h_{m,k}^t &= \mathbf{H}^t[m, 1, k], \end{aligned} \quad (21)$$

with m and k denoting the BS antenna index and subcarrier index, 1 denotes the receiver antenna index (the single omnidirectional receive antenna is considered in this work), respectively. The variable t indicates the time index, and $e_{m,k}^t$ represents the additional noise.

A. Additive White Gaussian Noise (AWGN)

AWGN is a common noise model in wireless communication systems. It describes the noise as a Gaussian random variable with zero mean and variance σ^2 , yielding,

$$e_{m,k}^t \sim \mathcal{N}(0, \sigma^2). \quad (22)$$

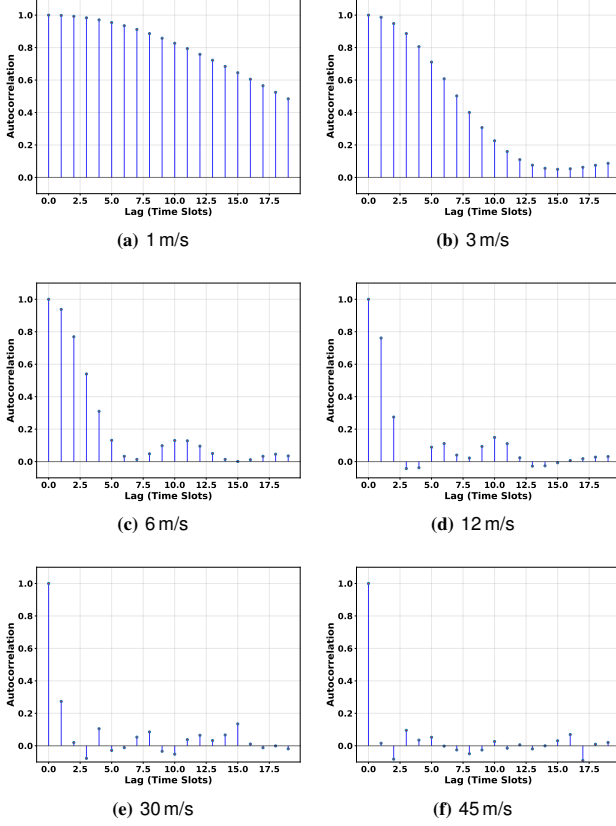


Fig. 13. ACF across different user velocities (TDD | CDL-A | 30ns)

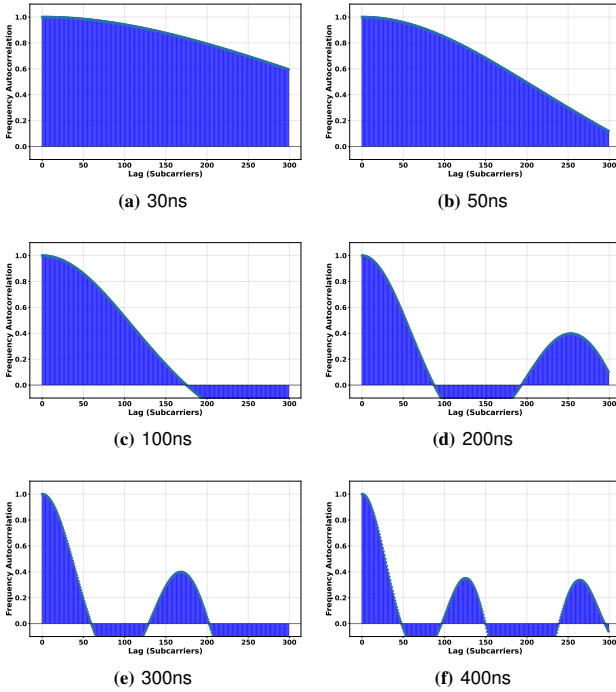


Fig. 14. Frequency ACF across different delay spreads. (FDD | CDL-C | 1m/s)

where σ^2 is the variance of the noise. The relationship between the SNR and the σ^2 is given by:

$$\text{SNR} = 10 \log_{10} \left(\frac{\|\mathbf{H}\|_F^2}{\sigma^2} \right) \quad (23)$$

$$\sigma^2 = \|\mathbf{H}\|_F^2 \cdot 10^{-\text{SNR}/10}$$

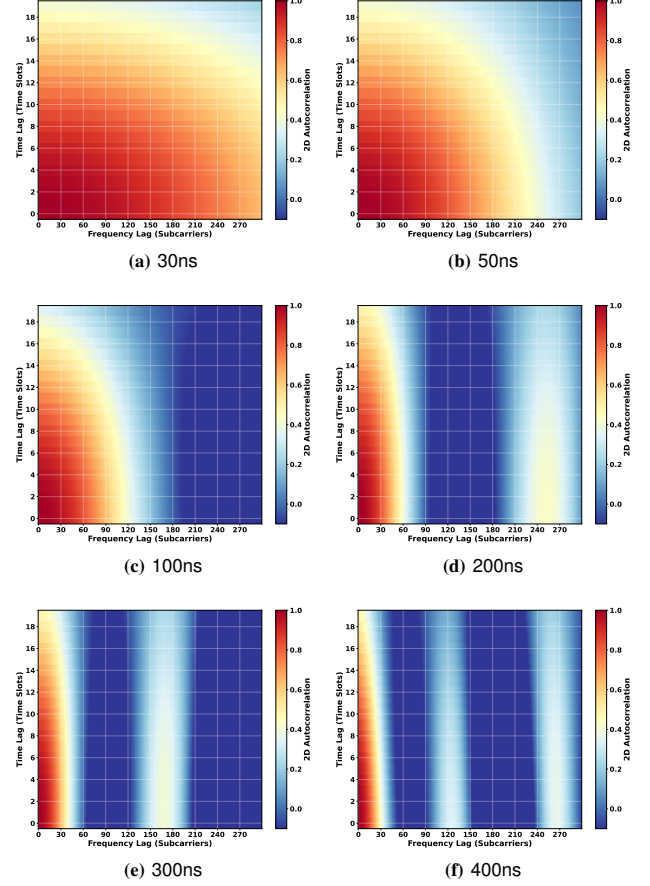


Fig. 15. 2D Frequency ACF across different delay spreads. (TDD | CDL-C | 1m/s)

The above explicit expression of the σ^2 is used to generate the AWGN noise with the target SNRs.

B. Definition of Realistic Additional Noise

1) *Phase Noise*: Phase noise (Fig. 17a) is pervasive in practical communication systems and is a crucial factor limiting the performance of high-speed communications, thus requiring model robustness to phase fluctuations [62,63]. The complex element CSI can be represented by gain and phase, namely,

$$h_{m,k}^t = |h_{m,k}^t| e^{j\theta_{m,k}^t}. \quad (24)$$

Accordingly, the Gaussian-like perturbation $\Delta_{m,k}^t$ is introduced to the phase part of the element CSI $\theta_{m,k}^t$, the element-wise phase noise is formulated as follows:

$$\begin{aligned} e_{m,k}^t &= |h_{m,k}^t| \left(e^{j\tilde{\theta}_{m,k}^t} - e^{j\theta_{m,k}^t} \right) \\ &= |h_{m,k}^t| \left(e^{j(\theta_{m,k}^t + \Delta_{m,k}^t)} - e^{j\theta_{m,k}^t} \right), \end{aligned} \quad (25)$$

where $\Delta_{m,k}^t \sim \mathcal{N}(0, \sigma^2)$

The resulting noisy CSI is given by:

$$\tilde{h}_{m,k}^t = |h_{m,k}^t| e^{j\tilde{\theta}_{m,k}^t} \quad (26)$$

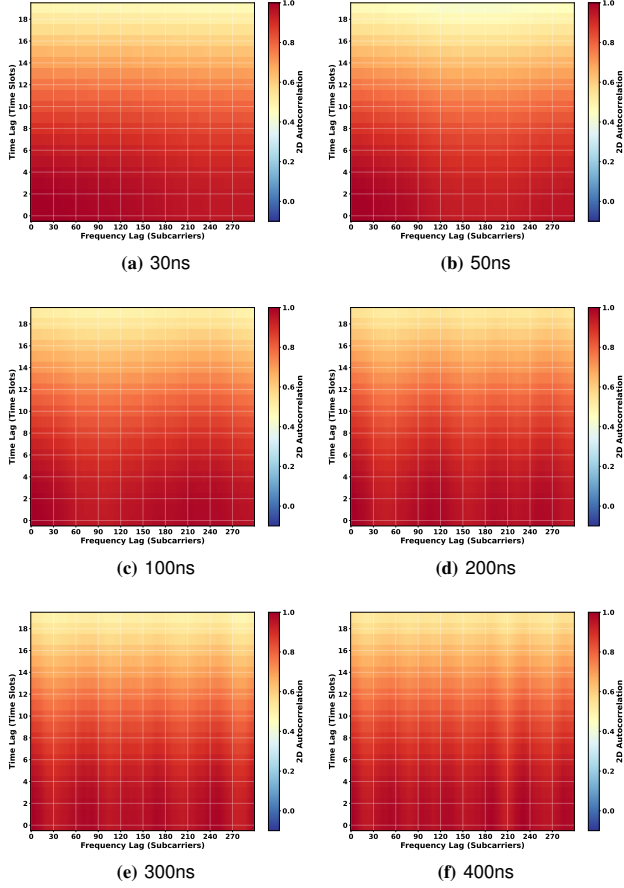


Fig. 16. 2D Frequency ACF across different delay spreads. (TDD | CDL-D | 1m/s)

2) *Burst Noise*: To better mimic practical channel conditions, burst noise (Fig. 17b) is introduced to simulate sudden spike-like disturbances that may result from abrupt environmental changes or unexpected obstacles between the BS and UE [33, 34]. Currently, burst noise is modeled as a bell-shaped perturbation spanning L_{burst} consecutive slots. The amplitude of this bell shape is A_{burst} and the probability of burst noise occurring in any given slot is characterized by the Bernoulli trial with burst probability P_{burst} . One additional limitation is that in a single historical CSI input (L packets), there is at most one burst noise.

Accordingly, the starting time index of the burst noise is formulated as a truncated geometric distribution, yielding,

$$\mathbb{P}(s_{m,k} = t) = (1 - P_{\text{burst}})^{t-1} P_{\text{burst}}, \quad t = 1, 2, \dots, L \quad (27)$$

The bell-shaped perturbation is formulated as follows:

$$\begin{aligned} \text{bell}_{m,k}^t &= A_{\text{burst}} g(t - s_{m,k}) o(t, s_{m,k}) \\ \text{where} \\ o(t, s_{m,k}) &= \mathbf{1} \{0 \leq t - s_{m,k} \leq \min\{L, L_{\text{burst}} - 1\}\} \\ g(\tau) &= \exp\left(-\frac{(\tau - c)^2}{2}\right) \\ c &= \frac{L_{\text{burst}} - 1}{2}, \quad \tau \in [0, \min\{L, L_{\text{burst}} - 1\}] \end{aligned} \quad (28)$$

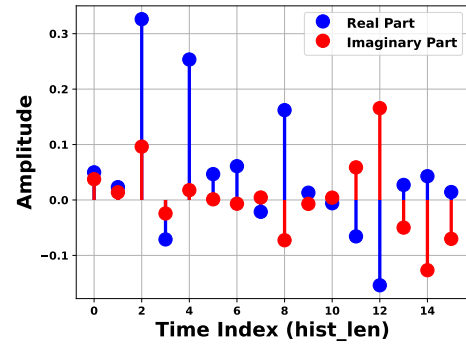
$$e_{m,k}^t = \text{bell}_{m,k}^t \cdot \epsilon_{m,k}^t, \quad \text{where } \epsilon_{m,k}^t \sim \mathcal{N}(0, 1) \quad (29)$$

3) *Packet drop Noise*: Packet drop noise (Fig. 17c) refers to the random omission of CSI packets [35]. For each time step t , whether a packet is dropped is modeled as a realization of a Bernoulli random variable with parameter p_d , i.e., $d^t \sim \text{Bernoulli}(p_d)$. The packet drop noise is then defined as:

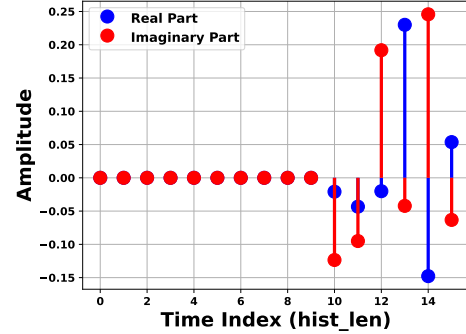
$$e_{m,k}^t = 0 - h_{m,k}^t \cdot d^t. \quad (30)$$

This implies that if $d^t = 1$, all CSI elements at time t are dropped; otherwise, the CSI remains unchanged. Consequently, the resulting noisy CSI is given by:

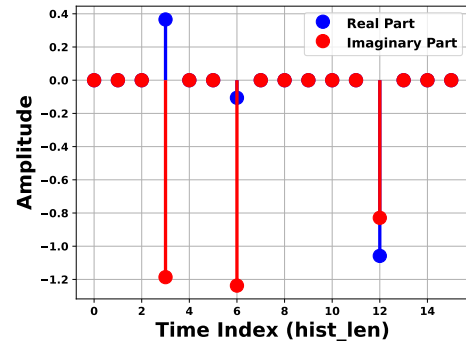
$$\tilde{h}_{m,k}^t = \begin{cases} h_{m,k}^t, & \text{if } d^t = 0 \\ 0, & \text{if } d^t = 1 \end{cases}. \quad (31)$$



(a) Phase noise



(b) Burst noise



(c) Packet drop noise

Fig. 17. Visualization of realistic additive noises. For the first dataset sample, we plot the real and imaginary parts of the injected noise on antenna index 0 and subcarrier index 1 for each additional noise type (phase, burst, and packet drop).

C. Experiment Details

a) *Noise calibration and SNR definition.*: For *phase noise*, the standard deviation σ of the Gaussian perturbation is controlled; larger σ yields larger phase excursions. For *burst noise*, both the pulse amplitude A_{burst} and the occurrence probability P_{burst} are set proportional to a controllable *noise-degree* parameter nd (i.e., $A_{\text{burst}} \propto nd$ and $P_{\text{burst}} \propto nd$), so higher nd produces stronger and more frequent bursts.

The simulation experiments are conducted to empirically calibrate (σ, nd) against the resulting SNR. Throughout, SNR is defined as the signal-to-noise power ratio as follows:

$$\text{SNR} = 10 \log_{10} \left(\frac{\|\mathbf{H}\|_F^2}{\sum_t \sum_m \sum_k |h_{m,k}^t|^2} \right). \quad (32)$$

Then the (σ, nd) are selected to realize matched SNR targets $\{10, 15, 20, 25\}$ dB for fair comparison. Very low SNRs (e.g., 0–5 dB) would require extreme parameter values (e.g., unusually large phase excursions) and are uncommon in practice (e.g., 0.59 radians on the phase perturbation corresponds to the SNR equals to 5 dB). For *packet-drop* noise, we use per-step Bernoulli erasures with drop probabilities $\{0.01, 0.02, \dots, 0.10\}$ to reflect realistic operating conditions.

b) *Imputation for the packet drop noise.*: Following [30], packet drops can be detected by monitoring inter-packet intervals of consecutive CSI frames; accordingly, indices of dropped packets are treated as known. In this work, missing CSI samples are handled via simple *imputation*: each missing CSI sample is replaced by the last available/(observed) sample. To maintain a simple protocol and isolate the intrinsic robustness of the prediction models, more sophisticated imputers are intentionally omitted.

APPENDIX E

SPECTRAL EFFICIENCY (SE)

While NMSE quantifies the accuracy of CSI prediction by measuring the element-wise deviation between the predicted and ground truth CSI, it does not directly reflect the impact of prediction quality on overall system performance. To ensure practical relevance, this study also evaluates the predicted DL CSI using end-to-end performance metrics, specifically spectral efficiency (SE).

SE is defined as the maximum achievable data rate per unit bandwidth and is a key performance metric in wireless communication systems. Maximizing SE is a central objective in system design. Its derivation is based on Shannon's capacity formula [64], applied to the signal model at subcarrier k :

$$y_k = \mathbf{h}_k^\dagger \omega_k x_k + n_k, \quad (33)$$

where x_k and y_k are the transmitted and received signals, respectively, $\mathbf{h}_k = \mathbf{H}^t[k]$ is the channel state information (CSI), ω_k is the precoding vector, and n_k represents additive white Gaussian noise (AWGN) with zero mean and variance σ_n^2 .

The theoretical SE is computed as:

$$\text{SE} = \frac{1}{N_{\text{sc}}} \sum_{k=1}^{N_{\text{sc}}} \log_2 \left(1 + \frac{\mathbf{h}_k^\dagger \omega_k \omega_k^\dagger \mathbf{h}_k}{\sigma_n^2} \right), \quad (34)$$

achieved when $\omega_k = \frac{\mathbf{h}_k}{\|\mathbf{h}_k\|}$.

For predicted DL CSI, the SE is estimated using:

$$\widehat{\text{SE}} = \frac{1}{N_{\text{sc}}} \sum_{k=1}^{N_{\text{sc}}} \log_2 \left(1 + \frac{|\hat{\mathbf{h}}_k^\dagger \mathbf{h}_k|^2}{\sigma_n^2 \|\hat{\mathbf{h}}_k\|^2} \right), \quad (35)$$

where $\hat{\mathbf{h}}_k = \hat{\mathbf{H}}[t, k]$ and $\mathbf{h}_k$ denote the predicted and ground truth DL CSI, respectively (the subscript f is omitted for simplicity). This formulation reflects that the BS configures the precoding vector ω_k using the predicted CSI $\hat{\mathbf{h}}_k$, while the resulting SE is evaluated with respect to the actual channel $\mathbf{h}_k$.

The following presents a comprehensive performance analysis based on Spectral Efficiency (SE), including SE rank distributions in Figs. 18–19; SE across SNR levels under AWGN (Fig. 20); SE across user velocities (Fig. 21); delay spreads (Table VIII); channel models (Table VIII); and SE under various types of additional noise (Fig. 22).

TABLE VII
SE UNDER VARYING DELAY SPREADS. **BOLD** DENOTES THE BEST VALUE AND UNDERLINE THE SECOND-BEST (HIGHER SE IS BETTER).

Models	Regular			Generalization		
	30 ns	100 ns	300 ns	50 ns	200 ns	400 ns
TDD						
NP	7.761	7.739	7.735	7.776	7.782	7.743
CNN	7.944	7.905	7.924	7.938	7.958	7.930
STEMGNN	7.962	7.912	7.871	7.970	7.919	7.863
RNN	8.005	8.011	8.048	8.017	8.024	7.824
LLM4CP	8.030	8.054	<u>8.095</u>	<u>8.061</u>	8.106	8.039
CSI-4CAST	8.081	8.100	8.130	8.088	8.089	8.003
FDD						
NP	7.613	7.078	7.313	7.408	7.410	7.370
CNN	7.212	6.395	7.012	6.932	6.707	6.985
STEMGNN	7.584	6.876	7.157	7.282	6.682	6.765
RNN	7.891	7.830	7.914	7.738	7.238	6.990
LLM4CP	7.898	7.747	<u>8.018</u>	7.580	7.182	<u>7.200</u>
CSI-4CAST	7.912	<u>7.823</u>	8.019	<u>7.654</u>	<u>7.285</u>	6.922

TABLE VIII
SE UNDER VARYING CHANNEL MODELS. **BOLD** DENOTES THE BEST VALUE AND UNDERLINE THE SECOND-BEST (HIGHER SE IS BETTER).

Models	Regular			Generalization	
	CDL-A	CDL-C	CDL-D	CDL-B	CDL-E
TDD					
NP	7.720	7.504	8.011	7.297	7.995
CNN	7.923	7.634	8.216	7.295	8.196
STEMGNN	7.891	7.617	8.237	7.260	8.220
RNN	7.970	7.845	8.250	7.394	8.232
LLM4CP	<u>8.048</u>	<u>7.873</u>	<u>8.258</u>	7.520	<u>8.239</u>
CSI-4CAST	8.060	7.986	8.265	<u>7.507</u>	8.246
FDD					
NP	7.427	6.623	7.955	5.910	7.919
CNN	6.731	5.852	8.035	5.020	7.952
STEMGNN	7.224	6.175	8.218	5.335	8.175
RNN	<u>7.778</u>	7.613	8.244	<u>5.895</u>	<u>8.201</u>
LLM4CP	7.751	<u>7.668</u>	<u>8.245</u>	5.747	8.188
CSI-4CAST	7.822	7.678	8.253	5.767	8.202

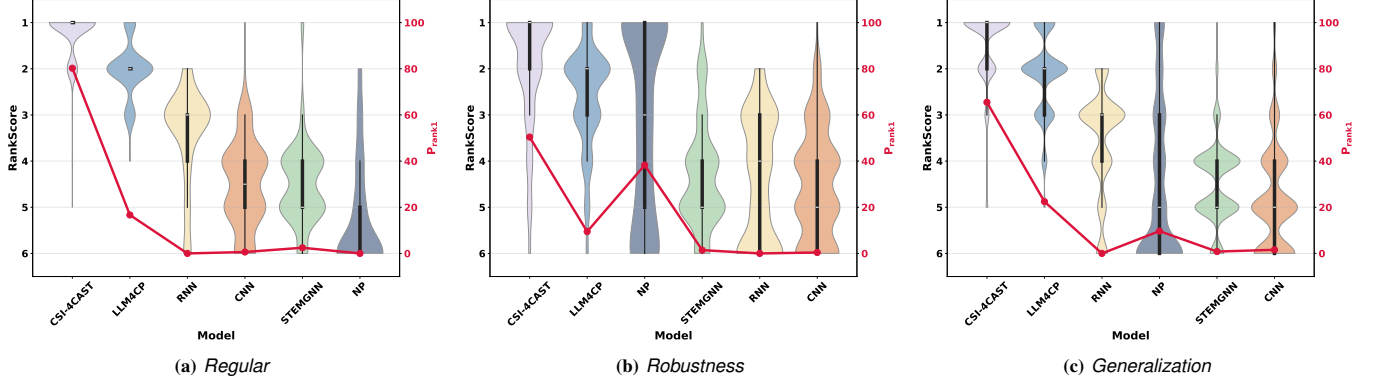


Fig. 18. **TDD: SE rank distribution of Regular, Robustness, and Generalization.** Within each panel, models are ordered left to right by their mean rank, MeanRank in (17) (lower is better). Rank distributions are shown as violin plots, while top-1 percentages, P_{rank1} in (19), are plotted as a line graph.

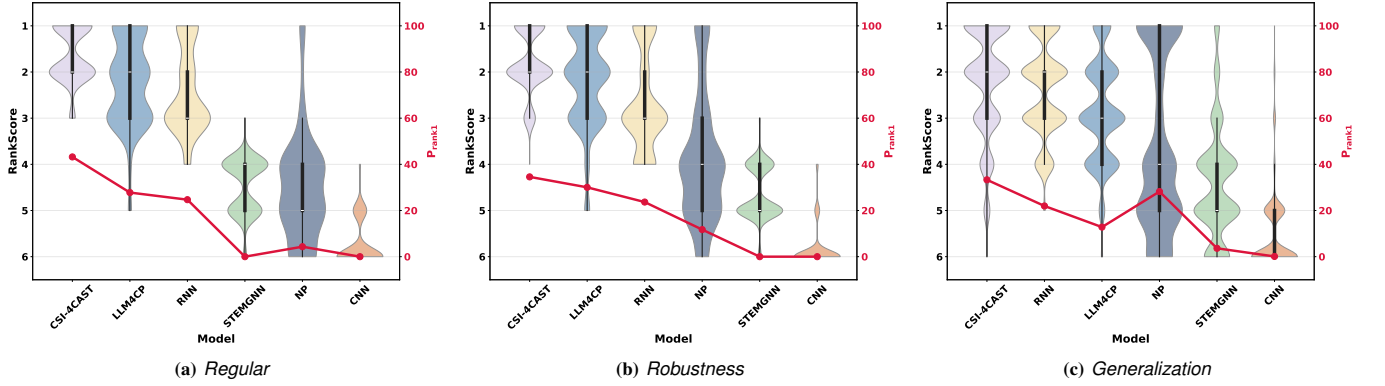


Fig. 19. **FDD: SE rank distribution of Regular, Robustness, and Generalization.**

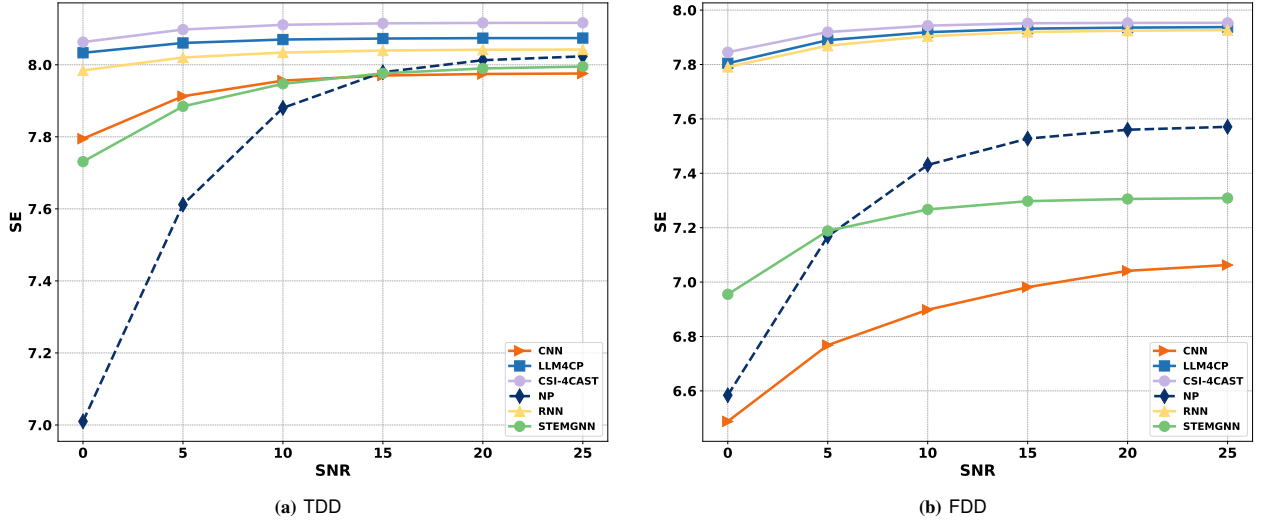


Fig. 20. **SE under varying SNR of noises.**

APPENDIX F TRAINING CONFIGURATIONS

Table IX outlines the defined hyperparameter search space and the trainer settings used with the Optuna framework for automated tuning. The optimizer, scheduler, and training settings are shared across models, while architecture-specific

hyperparameters are listed separately.

Each model undergoes one round of hyperparameter tuning for both the TDD and FDD datasets, as the distinct characteristics of the two duplexing modes demand separate configurations. The subcarrier-wise ACL layer in *CSI-4CAST* is enabled only under the FDD setting.

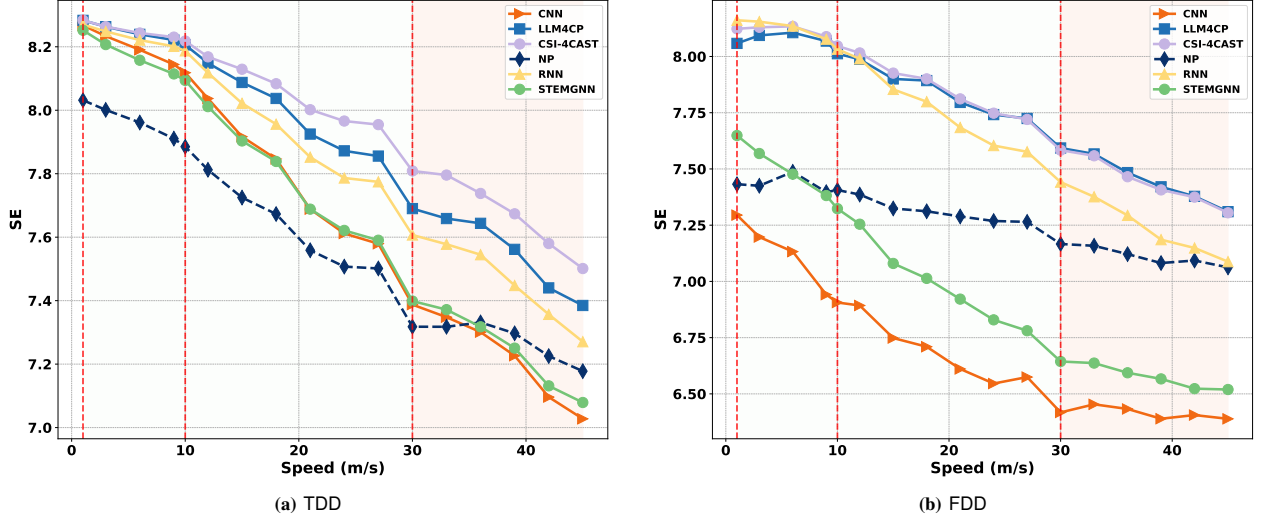


Fig. 21. **SE across user velocities.** Red dashed vertical lines mark the velocities included in the *regular* set; all other velocities belong to the *generalization* set. Light green shading denotes the interpolation region (velocities within the *regular* range), whereas light red denotes the extrapolation region (velocities outside that range). The annotation has the same meaning in the subsequent figures will not be reiterated.

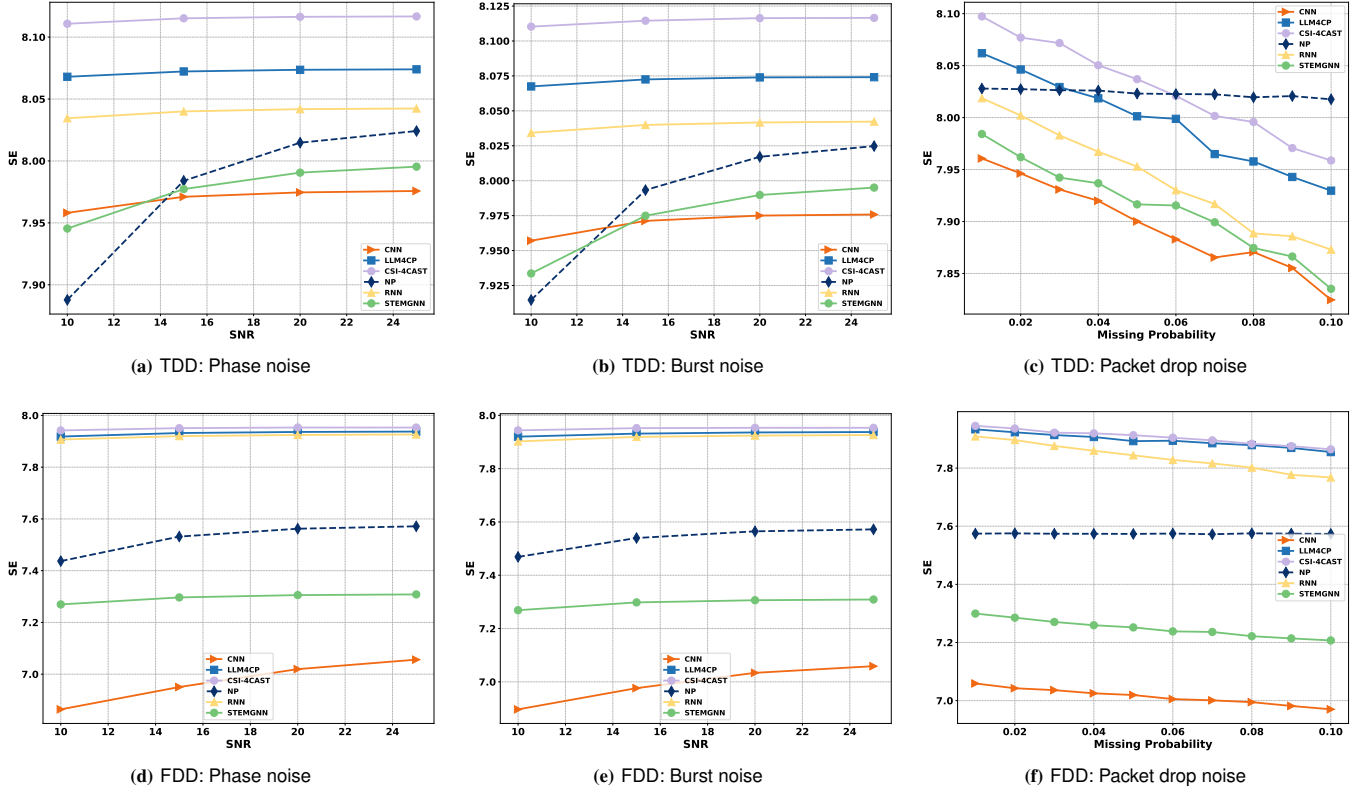


Fig. 22. **SE under varying realistic additional noises: TDD and FDD.**

Baseline models follow the official implementations provided by their authors [28,29]. Although a flexible search space is applied, some constraints remain due to the limited configurability of the original codebases.

TABLE IX
HYPERPARAMETER SEARCH SPACE (DOMAINS ARE INCLUSIVE).

Module	Hyperparameter	Domain	Type	Notes
Optimizer				
	name	{Adam, AdamW}	categorical	–
	lr	$[10^{-5}, 5 \times 10^{-3}]$	log-uniform	–
	weight_decay	$[10^{-6}, 10^{-2}]$	log-uniform	–
	beta_1	[0.85, 0.95]	uniform	step = 0.005
	beta_2	[0.98, 0.999]	uniform	step = 0.001
Scheduler				
	<i>ReduceLROnPlateau.factor</i>	[0.1, 0.7]	uniform	mode = <i>min</i>
	<i>ReduceLROnPlateau.patience</i>	{5, 10, 20}	categorical	threshold = 10^{-4}
	<i>ReduceLROnPlateau.cooldown</i>	[0]	integer	–
	<i>ReduceLROnPlateau.min_lr</i>	$[10^{-8}, 10^{-5}]$	log-uniform	–
Training (ensure all models have sufficiently large effective batch size)				
	batch_size	{4, 8, 16}	categorical	–
	accumulate_grad_batches	{1, 2, 4}	categorical	–
CSI-4CAST				
<i>CNN-based Residual Representation</i>				
	num_filters_2d	[1, 5]	integer	step = 1
	filter_size_2d	{3, 5}	categorical	–
	filter_size_1d	{3, 5}	categorical	–
	is_residual	{True, False}	categorical	–
	activation	{tanh, relu, gelu}	categorical	–
<i>Adaptive Correction Layers (time)</i>				
	layers	[2, 4]	integer	step = 1
	hidden_dim	{128, 256, 512}	categorical	–
	out_act	{sigmoid, tanh, relu, none}	categorical	–
	arl_op	{add, multiply}	categorical	–
<i>Adaptive Correction Layers (subcarrier)</i>				
	layers	[2, 4]	integer	step = 1
	hidden_dim	{128, 256, 512, 1024, 2048}	categorical	–
	out_act	{sigmoid, tanh, relu, none}	categorical	–
	arl_op	{add, multiply}	categorical	–
<i>Shuffle Blocks</i>				
	res_layers	[4, 6]	integer	step = 1
	res_dim	{64, 128, 256}	categorical	–
	groups	[4, 8]	categorical	–
	dropout	{0.1, 0.2, 0.3}	categorical	–
<i>Transformer Encoder</i>				
	d_{model}	{512, 768, 1024, 2048}	categorical	–
	num_layers	[4, 6]	integer	step = 1
	num_heads	[4, 8]	integer	step = 1
	hidden_dim	{512, 1024, 2048}	categorical	–
	dropout_prob	{0.1, 0.2, 0.3}	categorical	–
STEGMNN (follows [28]’s implementation)				
	n_stacks	{2}	integer	fixed by authors
	multi_layer	{2, 4, 8, 16}	categorical	–
LLM4CP (follows [29]’s implementation)				
	res_layers	[2, 8]	integer	step = 1
	res_dim	{64, 128, 256, 512, 1024, 2048, 4096}	categorical	–
	gpt_type	{gpt2, gpt2-medium, gpt2-large}	categorical	–
	gpt_layers	[2, 8]	integer	step = 1
RNN (follows [29]’s implementation)				
	rnn_hidden_dim	{128, 256, 512, 1024, 2048, 4096}	–	–
	rnn_num_layers	[1, 8]	integer	step = 1
CNN (follows [29]’s implementation)				
	num_filters	[3, 10]	integer	step = 1