Improved Downlink Rates for FDD Massive MIMO Systems through Bayesian Neural Networks-Based Channel Prediction

Zhihao Tao and Shaowei Wang, *Senior Member, IEEE*

***Abstract*—In frequency-division-duplex (FDD) massive MIMO systems, channel state information (CSI) feedback waiting phase does not get fully exploited since base station needs to wait for the CSI feedback before transmitting downlink data. The proportion of the CSI feedback waiting phase during the downlink transmission would be high as the MIMO system scales up, which sacrifices downlink rates of the FDD massive MIMO systems significantly. In this paper, we first present a channel prediction-aided FDD scheme to utilize the idle waiting time efficiently. Then we propose a novel channel prediction method based on Bayesian neural network (BNN), which can handle the uncertainty in a natural manner and learn regularization from data without painstaking manual pre-tuning of network hyperparameters. Numerical results show that our proposed channel prediction-aided FDD scheme can achieve remarkable performance gains in terms of either achievable downlink rates or bit error rate. Moreover, our proposed BNN-based channel predictor is much more effective and robust in contrast to the state-of-the-art channel prediction techniques such as autoregressive model and recurrent neural network.**

***Index Terms*—Bayesian neural network, channel prediction, frequency-division-duplex massive MIMO, probabilistic models.**

I. Introduction

Multiple-input-multiple-output (MIMO) has been deemed as an enabling technology for mobile networks to improve spectral efficiency and link reliability in the past two decades. To meet the exponentially increasing mobile data traffic, massive MIMO has been proposed for the future generation mobile communication systems [2]. Theoretically, massive MIMO systems can offer high spectral and energy efficiency with simple linear signal processing approaches by employing large arrays of antennas at base stations (BSs) [3]. Besides, massive MIMO can reduce the capacity losses caused by small-scale fading and uncorrelated noise, simplify signal processing and provide effective power control [4].

However, these potential massive MIMO gains rely heavily on the availability of accurate channel state information (CSI) estimation at BSs. If massive MIMO systems operate in time-division-duplex (TDD) mode, only the uplink CSI sent by single-antenna users needs to be estimated due to channel reciprocity [4]. The CSI overhead of TDD massive MIMO systems is proportional to the number of users, not to the number of antennas at BSs. However, this is not the case for the frequency-division-duplex (FDD) massive MIMO systems, for which channel reciprocity does not hold since the uplink and downlink channels of the FDD systems transmit data over different frequency bands. In other words, both uplink CSI and downlink CSI are required for the FDD massive MIMO systems, which will lead to prohibitively heavy signaling overhead as the number of antennas at the BSs scales up [5]. Nonetheless, most of the contemporary cellular networks are operated in FDD mode [6], and it is cost-efficient for the conventional mobile communications providers to upgrade to massive MIMO with FDD mode. Additionally, FDD systems generally provide lower transmission delay due to simultaneous uplink and downlink transmissions, and are free of system interference [7]. In contrast with FDD mode, TDD massive MIMO also suffers system imperfections such as pilot contamination, calibration error and hardware impairments [8]. Thus, it is meaningful to design schemes to reap the massive MIMO gains in FDD mode from the viewpoint of both system performance and capital expenditure.

To mitigate the unfavorable effects of deploying large number of antennas at the BSs in FDD massive MIMO systems, current researches mainly focus on conceiving the available downlink training techniques and uplink CSI feedback strategies based on channel sparsity [8]–[12]. Though these proposed schemes can reduce signaling overhead, the channel sparsity hypothesis, which is the cornerstone of these proposals, is yet questionable and needs to be further validated [6]. Moreover, these schemes do not exploit the potential of the CSI feedback waiting phase in downlink transmission, which is introduced by the uplink CSI feedback in FDD systems. The basic frame structure of the FDD massive MIMO system is shown in Fig. 1, where one downlink transmission slot generally includes three phases: training, waiting and data transmission. As can be seen in Fig. 1, the BS needs to wait until the uplink feedback completes so as to get the estimated downlink CSI, based on which the BS transmits downlink data. The waiting period is called the CSI feedback waiting phase in Fig. 1. With the number of BS antennas scaling up, the pilot and CSI feedback overhead will become overwhelming, prolonging the duration of the CSI feedback waiting phase accordingly. On this account, the proportion of the CSI feedback waiting phase occupied in the downlink

Manuscript received February 28, 2021; revised July 9, 2021; accepted August 30, 2021. This work was partially supported by the National Natural Science Foundation of China under Grants 61931023 and U1936202. Part of this work has been presented at the IEEE Globecom 2019 [1], Waikoloa, HI, USA, December 9-13, 2019. The associate editor coordinating the review of this article and approving it for publication was D. Gunduz. *(Corresponding author: Shaowei Wang.)*

The authors are with the School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China (e-mail: mg1823078@smail.nju.edu.cn; wangsw@nju.edu.cn).

transmission gets higher. It underutilizes spectrum resources and sacrifices the possible downlink rates of the FDD massive MIMO system to a greater extent. The motivation of this work is to exploit the CSI feedback waiting phase to enhance the performance of FDD massive MIMO systems at the least cost.

In this paper, we propose a channel prediction-aided FDD scheme to harness the idle time of the CSI feedback waiting phase, and introduce Bayesian neural network (BNN) to perform channel prediction, based on which the BS can precode and transmit downlink data in the waiting phase. Early channel prediction methods are model-based such as autoregressive (AR), sum of sinusoids, and band-limited processes [13]–[15]. These methods have achieved attractive performance in specific channels but lack of robustness to diverse scenarios, and are vulnerable to non-stationary and fast-varying environments facing in massive MIMO [16]. Recently, machine learning is introduced to deal with challenging communication tasks such as massive access, spectrum sensing, UAV-assisted emergency communications and antenna selection [17]–[21], and deep neural network (DNN) has also been proposed to address the channel prediction problem. In [22], a multilayer perceptron (MLP)-based channel predictor is developed for realistic massive MIMO channels. In [23] and [24], convolutional neural network (CNN) is introduced to infer the target CSI. In [16] and [25], recurrent neural network (RNN) is proposed to predict the future CSI based on the known CSI in sequences. Though the DNN is mighty for addressing the nonlinearity issues by learning on data with no need for prior knowledge about the task, the neural network hyperparameters are typically set based on rules of thumb, which is time-consuming and inconvenient for real-time channel prediction, especially in practical communication environments having rapid and irregular channel changes. Another concern is that the traditional DNN is prone to overfitting the training data and providing over-confident predictions when input samples are out of the training distribution or corrupted by noise, since it is often incapable of expressing the uncertainty in the collected data or the model effectively [26], [27]. The uncertainty in the data measures the noise inherent in the obtained CSI samples, which is inevitably introduced by the channel estimation error, hardware impairments and inter-cell interference, etc. The uncertainty in the model represents the epistemic uncertainty of channel prediction model, which derives from imbalances in the training data distribution due to that the training sets cannot be expected to contain all possible cases. The complex propagation environment and unpredictable user mobility also intensify the model uncertainty. The traditional DNN using point estimates as weights does not take these uncertainties into consideration, so it overfits almost unavoidably and lacks of a good generalization capability for novel samples [28]. Promisingly, the BNN points to a potential remedy for these concerns by adding a measure of uncertainty and regularization in predictions, which has been shown in recent advances [27]–[29].

The BNN is a unique combination of a neural network and a stochastic model, which exploits the strengths of both of them. Neural networks exhibit universal and flexible function approximation capabilities, and stochastic models allow the networks to represent uncertainty via its parameters. In this work, we introduce a standard Bayesian framework to neural network by placing probability distributions over the network parameters and outputs, based on which the network can capture the uncertainties naturally and produce reliable predictions. The probabilistic setting also facilitates the inference of optimal regularization hyperparameters in an automated fashion within the training process, which formulates an adaptive network structure for the prediction of rapidly changing channels. Instead of employing the fairly complex and time-consuming Bayesian CNN or Bayesian RNN, which can cause severe model aging in real-time channel prediction, we explore a Bayesian scheme with Gaussian approximation for the MLP. Specifically, the main contributions of this paper are summarized as follows:

- We analyze the basic frame structure of the FDD massive MIMO systems and propose a channel prediction-aided downlink transmission scheme, by which the BS can exploit the CSI feedback waiting phase and achieve higher downlink rates.
- We provide a novel BNN-based channel prediction method by introducing Bayesian learning to neural network so as to incorporate uncertainties into predictions, which can not only control model complexity well to yield better predictions but also be implemented online to track the rapidly-changing channels. Furthermore, we analyze the predictive distribution and the prediction error of BNN.
- We evaluate the proposed FDD schemes and the BNN-based channel prediction method with extensive experiments. Numerical results show that the channel prediction-aided FDD schemes can achieve significant performance gains. Moreover, the proposed BNN-based channel predictor outperforms the state-of-the-art channel prediction techniques for both prediction accuracy and system performance. In brief, our proposed schemes shed insights on exploiting the full potential of FDD massive systems efficiently.

The remainder of the paper is organized as follows. In Section II, we describe system model considered in this work. In Section III, we specify how to exploit the CSI feedback waiting phase by channel prediction techniques. In Section IV, we first illustrate the main steps of massive MIMO channel prediction. Then, we present the architecture of BNN and how it works. Section V includes numerical results and analyses. Finally, we conclude our work in Section VI and proofs are relegated to Appendices.

Notations: Throughout the paper, we use boldface uppercase letters, boldface lowercase letters and lowercase letters to denote matrices, column vectors and scalars, respectively. $\boldsymbol{X}^T$, $\boldsymbol{X}^*$, $\boldsymbol{X}^\dagger$, $\boldsymbol{X}^{-1}$, tr($\boldsymbol{X}$), $|\boldsymbol{X}|$, and $\|\boldsymbol{X}\|$ correspond to the transpose, complex conjugate, complex conjugate transpose, inverse, trace, modulus, two-norm of $\boldsymbol{X}$, respectively. The notation $\mathbb{E}[\cdot]$ denotes expectation operation.

II. SYSTEM MODEL

Without loss of generality, we consider an FDD massive MIMO system with one BS, where the BS equipped with M

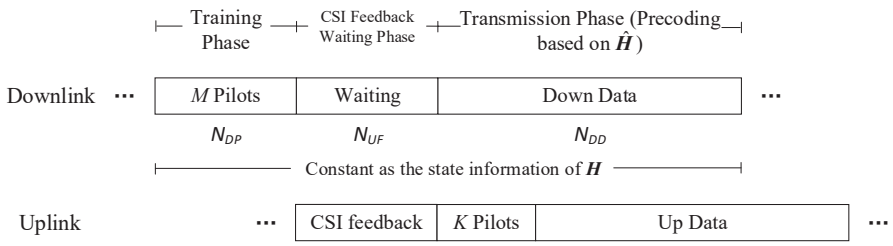

Fig. 1. The basic frame structure for the FDD massive MIMO system where the BS is equipped with M antennas and serves K users. $\hat{\boldsymbol{H}}$ is the channel estimation of actual channel $\boldsymbol{H}$ in current transmission slot.

antennas serves K single-antenna users. q_k is the downlink transmission symbol to the kth user and the source information vector $\boldsymbol{q} \triangleq [q_1, \ldots, q_k, \ldots, q_K]^T$ satisfies $\mathbb{E}[\boldsymbol{q}\boldsymbol{q}^\dagger] = \boldsymbol{I}_K$, where $\boldsymbol{I}_K$ denotes a $K \times K$ identity matrix. The precoded transmission signal vector is $\boldsymbol{x} = \sqrt{\eta}\boldsymbol{W}\boldsymbol{q}$, where $\boldsymbol{W} \in \mathbb{C}^{M\times K}$ is the precoding matrix. η is a normalization coefficient satisfying the power constraint $\mathbb{E}[\|\boldsymbol{x}\|^2] = 1$ and can be calculated as

$$\eta = \frac{1}{\mathbb{E}[\mathrm{tr}(\boldsymbol{W}\boldsymbol{W}^\dagger)]}. \tag{1}$$

We employ the zero-forcing (ZF) precoding, by which the inter-user interference can be cancelled at receivers [30]. The precoding matrix $\boldsymbol{W}$ needs to be calculated by the pseudo-inverse of channel matrix $\boldsymbol{H} \in \mathbb{C}^{M\times K}$, i.e.,

$$\boldsymbol{W} = \boldsymbol{H}^*(\boldsymbol{H}^T\boldsymbol{H}^*)^{-1}. \tag{2}$$

Then the received signal vector of users is given by

$$\boldsymbol{y} = \sqrt{\rho_d}\boldsymbol{H}^T\boldsymbol{x} + \boldsymbol{z}, \tag{3}$$

where $\boldsymbol{y} \triangleq [y_1, \ldots, y_k, \ldots, y_K]^T$. ρ_d is the average downlink SNR. $\boldsymbol{z} \triangleq [z_1, \ldots, z_k, \ldots, z_K]^T$, where z_k is the noise at the kth user and is a Gaussian random variable with zero mean and unit variance. Consequently, the received signal at the kth user is

$$\begin{aligned} y_k &= \sqrt{\rho}_d\boldsymbol{h}_k^T\boldsymbol{x} + z_k \\ &= \sqrt{\eta\rho_d}\boldsymbol{h}_k^T\boldsymbol{w}_k q_k + \sqrt{\eta\rho_d}\sum_{k'\neq k}^{K}\boldsymbol{h}_k^T\boldsymbol{w}_{k'}q_{k'} + z_k, \end{aligned} \tag{4}$$

where $\boldsymbol{h}_k$ and $\boldsymbol{w}_k$ are the kth column of the channel matrix $\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_k, \ldots, \boldsymbol{h}_K]$ and the kth column of the precoding matrix $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k, \ldots, \boldsymbol{w}_K]$, respectively. Therefore, the SINR at the receiver of the kth user is

$$\mathrm{SINR}_k = \frac{\eta\rho_d|\boldsymbol{h}_k^T\boldsymbol{w}_k|^2}{\eta\rho_d\sum_{k'\neq k}^{K}|\boldsymbol{h}_k^T\boldsymbol{w}_{k'}|^2 + 1}. \tag{5}$$

The achievable downlink sum rate of the FDD massive MIMO system can be calculated as

$$C = \sum_{k=1}^{K}\log_2(1 + \mathrm{SINR}_k). \tag{6}$$

Considering that fading makes the channel responses vary over time-frequency plane, to conveniently compare the performance of our designed channel prediction-aided FDD scheme with the conventional one, the estimation and payload transmission need to be fitted into one time/frequency block where the channels are static or quasi-static. Therefore, we assume that the channel is constant during each transmission slot and varies from slot to slot in this work. Meanwhile, the channel in a given time slot is correlated with the channels in previous and future time slots, which is known as temporal channel correlation. We also adopt multi-carrier orthogonal frequency-division multiplexing (OFDM) modulation. In this case, the channel is constant for τ symbols transmitted on the time-frequency plane, where the channel coherence block length τ is a dimensionality given by the coherence bandwidth B_c and the coherence time T_c, i.e. $\tau = B_cT_c$ transmission symbols. The channel coherence depends on the propagation environment, user mobile velocity, carrier frequency and so on. Furthermore, we consider analog feedback in the uplink of FDD massive MIMO system model, which allows the users directly send the downlink CSI to the BS in an unquantized and uncoded fashion [31].

III. CHANNEL PREDICTION-AIDED FDD SCHEME DESIGN

A. Conventional FDD Scheme

For the downlink transmission in the conventional FDD scheme, the BS first transmits N_{DP} orthogonal training pilots to users for downlink CSI acquisition as illustrated in Fig. 1. Next, the users send the feedback of CSI to the BS over uplink channel, during which the BS cannot transmit downlink data. The symbol length of the CSI feedback in one transmission slot is defined as N_{UF}. When the feedback completes, the BS can precode and transmit the downlink data then. We denote the symbol length of downlink data by N_{DD}. The downlink channel in one given time slot is regarded as quasi-static and denoted by $\boldsymbol{H}$. Considering channel estimation, the BS actually employs the estimated state information of $\boldsymbol{H}$, i.e. $\hat{\boldsymbol{H}}$, to compute the precoding matrix and transmit downlink data in the transmission phase, where the precoding matrix is defined as $\hat{\boldsymbol{W}}$. Therefore, the SINR at the receiver of the kth user in the conventional FDD scheme is

$$\mathrm{SINR}_{conv,k} = \frac{\eta_{conv}\rho_d|\boldsymbol{h}_k^T\hat{\boldsymbol{w}}_k|^2}{\eta_{conv}\rho_d\sum_{k'\neq k}^{K}|\boldsymbol{h}_k^T\hat{\boldsymbol{w}}_{k'}|^2 + 1}, \tag{7}$$

$$\eta_{conv} = \frac{1}{\mathbb{E}[\mathrm{tr}(\hat{\boldsymbol{W}}\hat{\boldsymbol{W}}^\dagger)]}, \tag{8}$$

where $\boldsymbol{h}_k$ and $\hat{\boldsymbol{w}}_k$ are the kth column of the actual channel matrix $\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_k, \ldots, \boldsymbol{h}_K]$ and the kth column of the estimated precoding matrix $\hat{\boldsymbol{W}} = [\hat{\boldsymbol{w}}_1, \ldots, \hat{\boldsymbol{w}}_k, \ldots, \hat{\boldsymbol{w}}_K]$, respectively. Note that the BS only transmits data during the transmission phase in the conventional scheme, so the downlink rate needs to be multiplied by a transmission coefficient λ, which can be written as

$$\lambda = \frac{N_{DD}}{N_{DP} + N_{UF} + N_{DD}}. \tag{9}$$

For a typical OFDM system, the sum of N_{DP}, N_{UF} and N_{DD} is actually equal to the number of useful symbols within one coherence block considering the cyclic prefix [30]. The achievable rate of the conventional FDD scheme can be calculated as

$$C_{conv} = \lambda \sum_{k=1}^{K} \log_2(1 + \mathrm{SINR}_{conv,k}). \tag{10}$$

B. Our Proposed Channel Prediction-Aided FDD Scheme

Recall that the duration of the waiting phase is equal to the duration of CSI feedback, so the proportion of the CSI feedback waiting phase occupied in the downlink transmission, denoted by δ, can be given by

$$\delta = \frac{N_{UF}}{N_{DP} + N_{UF} + N_{DD}}. \tag{11}$$

For a basic FDD scheme, the massive MIMO system generally requires M pilot symbols per coherence block in the downlink frequency band and feedback of M channel coefficients per terminal on the uplink frequency band. We do not consider CSI compression in this work for simplicity, and use M symbols and multiplex of K coefficients per symbol for uplink feedback based on the analog channel feedback technique, which is also applied in [6]. As a result, we are able to obtain $N_{DP} = N_{UF} = M$. Hence, δ can be considerably high when the BS is equipped with a large number of antenna arrays. Since the BS cannot transmit any downlink data in the CSI feedback waiting phase in the conventional FDD scheme, we propose a channel prediction-aided downlink transmission scheme to utilize the idle waiting time sufficiently so as to exploit the potential of the FDD massive MIMO systems.

Based on the frame structure shown in Fig. 1, we consider fetching the predicted CSI inferred from the historical channel information to compute the ZF precoding matrix, and precode and transmit data in the CSI feedback waiting phase in the current transmission slot. Then the BS takes the estimated CSI fed back via uplink channel for precoding to transmit data in the original transmission phase as the conventional FDD scheme does. In this case, the BS does not need to wait for the CSI feedback and can transmit data in both the CSI feedback waiting and the transmission phase. Since the BS can only obtain historical channel information of $\hat{\boldsymbol{H}}$ by means of uplink feedback and channel estimation techniques, the predicted CSI refers to predicting the channel information of $\boldsymbol{H}$ for the next transmission slots based on the collected samples of $\hat{\boldsymbol{H}}$. Our proposed channel prediction-aided FDD scheme is depicted in Fig. 2, where the virtual transmission phase includes the CSI feedback waiting phase and the original transmission phase. The predicted CSI is defined as $\tilde{\boldsymbol{H}}$ and the precoding matrix evaluated by $\tilde{\boldsymbol{H}}$ is denoted as $\tilde{\boldsymbol{W}}$. Hence the SINR of the kth user using the predicted CSI is

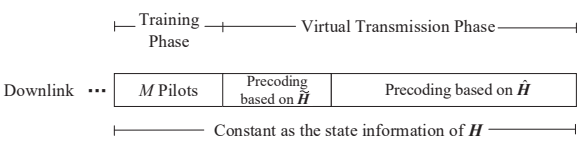

Fig. 2. Illustration of the proposed channel prediction-aided FDD scheme.

$$\mathrm{SINR}_{pred,k} = \frac{\eta_{pred}\rho_d |\boldsymbol{h}_k^T \tilde{\boldsymbol{w}}_k|^2}{\eta_{pred}\rho_d \sum_{k' \neq k}^{K} |\boldsymbol{h}_k^T \tilde{\boldsymbol{w}}_{k'}|^2 + 1}, \tag{12}$$

$$\eta_{pred} = \frac{1}{\mathbb{E}[\mathrm{tr}(\tilde{\boldsymbol{W}}\tilde{\boldsymbol{W}}^{\dagger})]}, \tag{13}$$

where $\tilde{\boldsymbol{w}}_k$ is the kth column of the predicted precoding matrix $\tilde{\boldsymbol{W}} = [\tilde{\boldsymbol{w}}_1, \ldots, \tilde{\boldsymbol{w}}_k, \ldots, \tilde{\boldsymbol{w}}_K]$. Thus, the achievable downlink rate of proposed channel prediction-aided FDD scheme can be written as

$$C_{proposed} = \delta \sum_{k=1}^{K} \log_2(1 + \mathrm{SINR}_{pred,k}) + C_{conv}. \tag{14}$$

When the predicted CSI is perfect, i.e. $\tilde{\boldsymbol{H}} = \boldsymbol{H}$, we can get the upper bound of the achievable downlink rate of the proposed scheme:

$$C_{upperbound} = \frac{\delta + \lambda}{\lambda} C_{conv}. \tag{15}$$

IV. Massive MIMO Channel Prediction

A. Downlink Channel Prediction with Model Aging

Downlink channel prediction in FDD massive MIMO systems consists of three principal steps:

- Model: conceive an appropriate channel prediction model to match practical radio propagation scenarios.
- Estimate: figure out the relevant parameters of the selected channel prediction model through observed CSI samples.
- Predict: yield future CSI for the next downlink transmission slots.

According to the number of samples per prediction, the channel prediction model can be categorized into the single-step prediction model and the multi-step one. We consider single-step prediction model in this work to simplify analysis[1]. In the downlink transmission, the BS first gathers historical CSI samples to obtain a training set, and then uses the data set to train the prediction model, which inevitably takes some time. Next, the BS yields future CSI samples with the prediction model, and upgrades the training set and re-estimates the model to track time-varying channel characteristics. The

[1]Although multi-step channel prediction model can further utilize the CSI feedback waiting phase and may yield higher downlink rates, it sacrifices much more BER performance due to the error propagation issue. The error propagation problem and how to determine the number of samples per prediction in multi-step model are yet intractable and remain mostly open.

process of the rolling prediction is illustrated in Fig. 3, where the size of the training set and the testing set are defined as N_t, N_p, respectively. As shown in Fig. 3, we consider the time period for model training, which gives rise to a phenomenon known as model aging. Specifically, the model aging refers to that the trained prediction model is not up to date for the future CSI samples to be predicted in continuously changing channel environments due to a time delay caused by the model training. This phenomenon will inevitably lead to a mismatch between the trained model and the most timely one and accordingly degrade the prediction performance to a certain degree. The model training will take more time with the complexity of channel prediction algorithm increasing, so it is necessary to take both the prediction accuracy and the complexity of a prediction model into consideration in practical scenarios.

In this work, we process a massive MIMO channel as a series of parallel single-input-single-output (SISO) channels. That is to say, instead of using joint processing with exploiting the spatial correlation across antennas at the BS, we design channel prediction algorithms for each SISO channel. The reason is that the complexity of joint processing prediction algorithms is far greater than the SISO processing in massive MIMO systems, e.g., the 2D-MMSE algorithm for the vector AR model is much more complex than the SISO algorithm with the number of antennas at the BS increasing [32]. Moreover, the techniques for reducing the complexity of joint processing are still expected to be further studied. Therefore, it is more reasonable to adopt the SISO processing when considering the training time and the model aging in massive MIMO systems, and it is also fair for comparison that our designed algorithms are all applied for per-SISO prediction[2]. A SISO channel between the mth (m = 1, 2, $\cdots$, M) transmitting antenna and the kth (k = 1, 2, $\cdots$, K) user at moment n can be denoted as $h_{mk}(n)$. We first predict each SISO channel and then combine them to get the predicted massive MIMO channel matrix. Hereafter, the antenna index and the user index are omitted for simplicity.

B. Bayesian Neural Networks-Based Channel Prediction

The BNN-based channel prediction model we propose is comprised of an MLP network and a Bayesian framework. For the MLP network, it is a fully connected neural network that maps a set of input data onto a set of appropriate outputs by feedforward propagation. Fig. 4 shows an MLP with an input layer, an output layer and two hidden layers. We denote the network inputs, outputs and connection weights (including the network biases) as $\boldsymbol{u}$, $\boldsymbol{v}$ and $\boldsymbol{\theta}$, respectively. Except for the input layer, other layers all contain a number of computing nodes, also referred to as neurons, which calculate the weighted sum of inputs and leverage a nonlinear activation function to transform the sum. In massive MIMO channel

[2]In addition, we can see from [32] that the advantage of using joint processing exists only in high spatial correlation. When the spatial correlation is low, there is just slight improvement on the channel prediction quality for the joint processing as compared to the SISO processing. This means that using joint processing may not yield better predictions at the cost of considerably high complexity in massive MIMO scenarios. We prefer to leave this to future study.

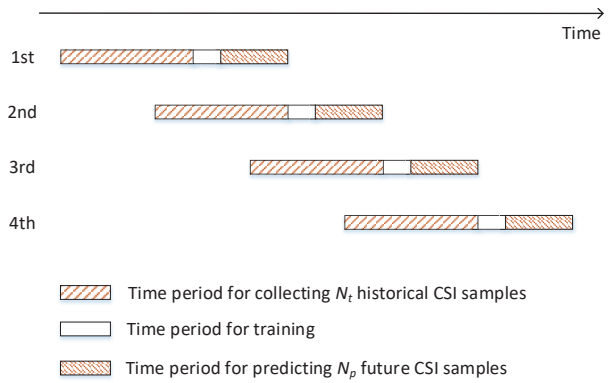

Fig. 3. The process of rolling channel prediction.

prediction, the input data are the observed historical CSI samples while the outputs are the predicted CSI ones. The input vector $\boldsymbol{u}$ includes the real and the imaginary parts of input samples. Assume the number of input samples as U, the length of $\boldsymbol{u}$ is $2U$, and the output layer needs two nodes to output the real and the imaginary values of predicted channels based on the aforementioned single-step prediction. Denote the network architecture by R, $\boldsymbol{v}(\boldsymbol{u}|\boldsymbol{\theta}, R)$ represents mapping input $\boldsymbol{u}$ to output $\boldsymbol{v}$ through network R with weight $\boldsymbol{\theta}$. The optimization task of the neural network training process is generally to minimize the sum of squared errors:

$$E_D(D|\boldsymbol{\theta}, R) = \sum_{i}^{N_D} \frac{1}{2} \|\boldsymbol{v}_i(\boldsymbol{u}_i|\boldsymbol{\theta}, R) - \boldsymbol{t}_i\|^2, \tag{16}$$

where $D = \{\boldsymbol{u}_i, \boldsymbol{t}_i\}_{i=1,2,\ldots,N_D}$ represents the training set. N_D is the total number of input-target pairs, and $\boldsymbol{t}_i$ represents the training target. To improve the generalization capability of model, it is common to add an extra regularization term E_θ to (16), and get a combined objective function:

$$F = \beta E_D(D|\boldsymbol{\theta}, R) + \alpha E_\theta(\boldsymbol{\theta}|R), \tag{17}$$

where $E_\theta(\boldsymbol{\theta}|R) = \sum_{j}^{N_\theta} \frac{1}{2} {\theta_j}^2$ is the sum of squares of network weights, N_θ is the total number of weights in the network, and α and β are the regularization hyperparameters. With the regularization term and apposite hyperparameters, the network can control network size well so as to achieve smoother mapping and good generalization while avoid underfitting.

The universality and flexibility of traditional neural networks including the MLP network and the DNN make them able to discover more general relationships in data than statistical models, and they do not require prior knowledge of channel characteristics for channel prediction. However, it is yet laborious and tedious to optimize the hyperparameters of traditional neural networks by any orthodox search technique (e.g. rules of thumb, trial and error, using reserved data to evaluate generalization ability, and so forth) [33]. The most popular way of setting hyperparameters is to compare the performance of networks trained with different parameter values by the cross-validation technique. In this case, it is required to do multiple learning runs with different values of hyperparameters and compare their performance on validation

sets in order to find the best hyperparameters. Nevertheless, repeating the learning with all possible values is impractical for the massive MIMO channel prediction, since this procedure is time consuming which will certainly lead to severe model aging in actual communication scenarios marked by rapid and irregular channel changes. For example, suppose the network has 3 hyperparameters to be set and assign 10 possible values for them in each. By means of the cross-validation technique, we will run 10^3 neural network models to determine the best values for those three hyperparameters. Obviously, this task is massive even with reasonable training dataset size, and this manual pre-tuning of network hyperparameters is inefficient and lacks of adaptiveness in time-varying channel environments. Hence, the network hyperparameters need to be optimized automatically and timely for adaptive channel prediction. Moreover, traditional neural networks using point estimates as weights can not effectively express the prediction uncertainty from a probability theory perspective [28]. The prediction uncertainty in neural networks generally derives from data uncertainty and model uncertainty (also known as aleatoric uncertainty and epistemic uncertainty, respectively). The former usually arises because of noise in the data caused by the channel estimation error, hardware impairments and signal interference, etc., while the latter is resulted from imbalances in the training data distribution. In actual channel environments, the obtained CSI samples cannot be expected to be noise free and contain all possible cases. As a consequence, the traditional neural network trained on the noisy or insufficient data are prone to overfitting and making overly confident predictions.

To address the above issues, we introduce Bayesian learning to the MLP network by compensating randomness characterized by Gaussian distribution for network weights and outputs. In principle, all hyperparameters in a neural network including the model hyperparameters (e.g. the number of network layers, size of each layer, activation function, and etc.) and the regularization hyperparameters (α and β defined in (17)) can be inferred automatically within the network training by using Bayesian methods [34]. However, the Bayesian inferences of model hyperparameters are yet intractable since we need to incorporate them into the network architecture R and evaluating the evidence for R is overly complex. On the other hand, the regularization hyperparameters, which control the network weights and the network output errors, are two pivotal hyperparameters over the network prediction performance. Therefore, we focus on the Bayesian choice of α and β in this work.

In the Bayesian view of training the MLP neural network, the connection weights and the target outputs are considered as random variables [33]–[36]. The architecture of BNN is depicted in Fig. 4. For the sake of simplifying the following derivations and proofs, we consider the case of predicting a single target variable t form an input vector $\boldsymbol{u}$ (the extension to multidimensional case is straightforward and ready), so the total number of network outputs on the training set is equal to N_D. Given the network and the training set, the posterior probability for the weights can be formulated according to

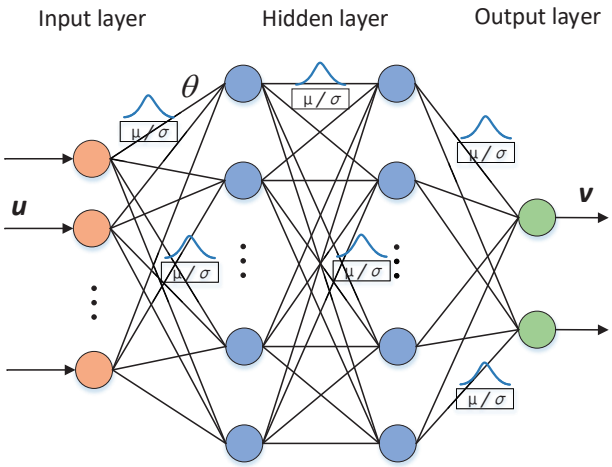

Fig. 4. The architecture of BNN-based channel prediction model.

Bayes' rule as

$$P(\boldsymbol{\theta}|D,\alpha,\beta,R)=\frac{P(D|\boldsymbol{\theta},\beta,R)P(\boldsymbol{\theta}|\alpha,R)}{P(D|\alpha,\beta,R)}, \tag{18}$$

where $P(D|\boldsymbol{\theta},\beta,R)$ is the likelihood probability[3], which represents the probability of the observed data given network weights $\boldsymbol{\theta}$. $P(\boldsymbol{\theta}|\alpha,R)$ is the prior probability which represents what we know about the weights before the data is taken into account. $P(D|\alpha,\beta,R)$ is the evidence, also called as the normalization factor guaranteeing that the total probability is 1. In the likelihood function, α is omitted from the condition variables since the data distribution does not depend on the regularization term. Similarly, the prior probability does not depend on β. Generally, the network output errors on the training data and the prior distribution on the weights are both modelled as zero-mean Gaussian i.i.d., by which the likelihood probability and the prior probability can be written respectively as[4]

$$\begin{aligned}P(D|\boldsymbol{\theta},\beta,R)&=\prod_{i=1}^{N_D}P(t_i|\boldsymbol{u}_i,\boldsymbol{\theta},\beta,R)\\&=\prod_{i=1}^{N_D}\mathcal{N}(t_i|v_i(\boldsymbol{u}_i|\boldsymbol{\theta},R),\beta^{-1})\\&=\prod_{i=1}^{N_D}(\frac{\beta}{2\pi})^{1/2}\exp(-\frac{\beta}{2}(v_i(\boldsymbol{u}_i|\boldsymbol{\theta},R)-t_i)^2)\\&=\frac{1}{Z_D(\beta)}\exp(-\beta E_D),\end{aligned} \tag{19}$$

$$P(\boldsymbol{\theta}|\alpha,R)=\mathcal{N}(\boldsymbol{\theta}|\mathbf{0},\alpha^{-1}\boldsymbol{I})$$

[3]This probability is a joint probability of all input-target pairs, which can also be written as $P(\{t_i\}|\{\boldsymbol{u}_i\},\boldsymbol{\theta},\beta,R)$ since the network does not predict the distribution of input variables $\{\boldsymbol{u}_i\}$. Likewise, $P(D|\alpha,\beta,R)$ can be written in the form of $P(\{t_i\}|\{\boldsymbol{u}_i\},\alpha,\beta,R)$.

[4]Since the hyperparameter α can control the expected weight magnitude and β can control the network output errors, which signifies that α and β have a bearing on the distribution of the likelihood probability and the prior probability, respectively, we can view α and β as the inverse variance of these two distributions, respectively.

$$= \frac{1}{Z_\theta(\alpha)} \exp(-\alpha E_\theta), \tag{20}$$

where '$\mathcal{N}$' denotes the Gaussian distribution, and $Z_D(\beta) = (2\pi/\beta)^{N_D/2}$ and $Z_\theta(\alpha) = (2\pi/\alpha)^{N_\theta/2}$. Substitute (19) and (20) into (18), we can obtain

$$\begin{aligned} P(\boldsymbol{\theta}|D,\alpha,\beta,R) &= \frac{\frac{1}{Z_\theta(\alpha)}\frac{1}{Z_D(\beta)}\exp(-(\beta E_D + \alpha E_\theta))}{normalization\ factor} \\ &= \frac{1}{Z_F(\alpha,\beta)}\exp(-F(\boldsymbol{\theta})), \end{aligned} \tag{21}$$

where $Z_F(\alpha,\beta) = \int e^{-F(\boldsymbol{\theta})} d\boldsymbol{\theta}$. Recall that $F(\boldsymbol{\theta})$ is the combined objective function defined in (17), minimizing regularized objective function F is identical to finding the (local) maximum a posteriori parameters $\boldsymbol{\theta}_{\mathrm{MAP}}$. Assuming for the moment that α and β are fixed, we can find $\boldsymbol{\theta}_{\mathrm{MAP}}$ by nonlinear optimization algorithms and error backpropagation. In this work, we adopt the Levenberg-Marquardt algorithm [37] to solve for $\boldsymbol{\theta}_{\mathrm{MAP}}$.

Once again, using Bayes' rule to infer the optimal values of α and β from data, we have the posterior probability for them as

$$P(\alpha,\beta|D,R) = \frac{P(D|\alpha,\beta,R)P(\alpha,\beta|R)}{P(D|R)}. \tag{22}$$

Now we assign a uniform prior density to (α, β), hence maximizing the posterior is equivalent to maximizing the likelihood function $P(D|\alpha,\beta,R)$, which is the normalization factor in (18). According to (21), the normalization factor can be derived as

$$P(D|\alpha,\beta,R) = \frac{Z_F(\alpha,\beta)}{Z_\theta(\alpha)Z_D(\beta)}, \tag{23}$$

where $Z_D(\beta)$ and $Z_\theta(\alpha)$ are defined earlier in (19) and (20), respectively. Then, the only part that needs to be calculated is $Z_F(\alpha,\beta)$. Assume that the posterior distribution formulated in (21) has a relatively small variance and the network function $v(\boldsymbol{u}_i,\boldsymbol{\theta})$ does not vary too much over the region of significant probability density. This allows us to make a Taylor series expansion of the network function around $\boldsymbol{\theta}_{\mathrm{MAP}}$ and construct a local linearization of the output [38]:

$$v(\boldsymbol{u}_i,\boldsymbol{\theta}) \simeq v(\boldsymbol{u}_i,\boldsymbol{\theta}_{\mathrm{MAP}}) + \boldsymbol{g}^T(\boldsymbol{\theta}-\boldsymbol{\theta}_{\mathrm{MAP}}), \tag{24}$$

where

$$\boldsymbol{g} = \nabla_\theta v(\boldsymbol{u}_i,\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\mathrm{MAP}}}. \tag{25}$$

Substitute (24) into (16), we can find $E_D(\boldsymbol{\theta})$ is quadratic around $\boldsymbol{\theta}_{\mathrm{MAP}}$. Since E_θ is also a quadratic function of $\boldsymbol{\theta}$, the objective function F can be locally approximated as quadratic around $\boldsymbol{\theta}_{\mathrm{MAP}}$, and we can estimate $F(\boldsymbol{\theta})$ around there by Taylor series expansion as

$$F(\boldsymbol{\theta}) \simeq F(\boldsymbol{\theta}_{\mathrm{MAP}}) + \frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}_{\mathrm{MAP}})^T \boldsymbol{G}_{\mathrm{MAP}}(\boldsymbol{\theta}-\boldsymbol{\theta}_{\mathrm{MAP}}), \tag{26}$$

where $\boldsymbol{G} = \beta\nabla^2 E_D + \alpha\nabla^2 E_\theta$ is the Hessian matrix of the objective function F, and $\boldsymbol{G}_{\mathrm{MAP}}$ is evaluated at $\boldsymbol{\theta}_{\mathrm{MAP}}$. Note that the first-order term in (26) does not appear since the gradient $\nabla_\theta F(\boldsymbol{\theta})$ will vanish at $\boldsymbol{\theta}_{\mathrm{MAP}}$. As a consequence, Z_F is the Gaussian integral:

$$Z_F = \int \exp(-F(\boldsymbol{\theta}))d\boldsymbol{\theta} \simeq \exp(-F(\boldsymbol{\theta}_{MAP}))\times$$

Algorithm 1 Iterative Algorithm for BNN

1: Initialize α, β and the network weights $\boldsymbol{\theta}$;
2: **repeat**
3: Take the Levenberg-Marquardt algorithm to minimize the combined objective function $F(\boldsymbol{\theta}) = \beta E_D + \alpha E_\theta$ and obtain the maximum a posteriori weights $\boldsymbol{\theta}_{\mathrm{MAP}}$;
4: Compute the effective number of network weights γ according to (32);
5: Use (30) and (31) to update the estimations for α and β;
6: **until** Convergence.

$$\begin{aligned} &\int \exp(-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}_{\mathrm{MAP}})^T \boldsymbol{G}_{\mathrm{MAP}}(\boldsymbol{\theta}-\boldsymbol{\theta}_{\mathrm{MAP}}))d\boldsymbol{\theta} \\ &= \exp(-F(\boldsymbol{\theta}_{\mathrm{MAP}}))(2\pi)^{N_\theta/2}\det{}^{-\frac{1}{2}}\boldsymbol{G}_{\mathrm{MAP}}, \end{aligned} \tag{27}$$

where 'det' represents the determinant of a matrix. A Gauss-Newton approximation to the Hessian matrix is available in the Levenberg-Marquardt algorithm [39]:

$$\boldsymbol{G} \simeq \beta\boldsymbol{J}^T\boldsymbol{J} + \alpha\boldsymbol{I}_{N_\theta}, \tag{28}$$

where $\boldsymbol{J}$ is the Jacobian matrix of the training error function E_D, and $\boldsymbol{I}_{N_\theta}$ denotes a $N_\theta \times N_\theta$ identity matrix.

Next, we place $Z_\theta(\alpha)$, $Z_D(\beta)$ and (27) into (23) and write the logarithms of both sides of (23) as

$$\begin{aligned} \ln P(D|\alpha,\beta,R) = &-\alpha E_\theta^{\mathrm{MAP}} - \beta E_D^{\mathrm{MAP}} - \frac{1}{2}\ln\det\boldsymbol{G}_{\mathrm{MAP}} \\ &+ \frac{N_\theta}{2}\ln\alpha + \frac{N_D}{2}\ln\beta - \frac{N_D}{2}\ln 2\pi. \end{aligned} \tag{29}$$

We can make estimates for α and β by maximizing $\ln P(D|\alpha,\beta,R)$. Take the derivatives of (29) with respect to α and β, respectively, and let them equal to zero, we obtain the re-estimation formulas of α and β as

$$\alpha = \frac{\gamma}{2E_\theta^{\mathrm{MAP}}}, \tag{30}$$

$$\beta = \frac{N_D - \gamma}{2E_D^{\mathrm{MAP}}}, \tag{31}$$

where

$$\gamma = \sum_{j=1}^{N_\theta} \frac{\xi_j}{\xi_j + \alpha}. \tag{32}$$

$\{\xi_j\}_{j=1,2,\ldots,N_\theta}$ are the eigenvalues of $\beta\boldsymbol{J}^T\boldsymbol{J}$, and γ represents the effective number of network weights, which can range from zero to N_θ. Detailed derivations for the re-estimation formulas of α and β are shown in Appendix A. We summarize the main steps required for Bayesian optimization of MLP in Algorithm 1.

Given the Bayesian inferences of the most probable values of α, β and $\boldsymbol{\theta}$ described above, we then evaluate the predictive distribution, which gives the prediction error of BNN. Define the predictive distribution of a new datum t_{N_D+1} by marginalizing with respect to the posterior distribution in (21) as

$$P(t_{N_D+1}|\boldsymbol{u}_{N_D+1}, D, \alpha, \beta, R) =$$

$$\int P(t_{N_D+1}|\boldsymbol{u}_{N_D+1},\boldsymbol{\theta},\beta,R)P(\boldsymbol{\theta}|D,\alpha,\beta,R)d\boldsymbol{\theta}. \quad (33)$$

Combine (21), (26) with (27), we can obtain a Gaussian distribution for the posterior probability:

$$\begin{aligned} P(\boldsymbol{\theta}|D,\alpha,\beta,R) &= \frac{(\det \boldsymbol{G}_{\mathrm{MAP}})^{\frac{1}{2}}}{(2\pi)^{\frac{N_\theta}{2}}}\times \\ &\exp(-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}_{\mathrm{MAP}})^T\boldsymbol{G}_{\mathrm{MAP}}(\boldsymbol{\theta}-\boldsymbol{\theta}_{\mathrm{MAP}})) \\ &= \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\theta}_{\mathrm{MAP}},\boldsymbol{G}^{-1}_{\mathrm{MAP}}), \end{aligned} \quad (34)$$

With the linear approximation in (24) and the likelihood function in (19), $P(t_{N_D+1}|\boldsymbol{u}_{N_D+1},\boldsymbol{\theta},\beta,R)$ can be written as

$$\begin{aligned} &P(t_{N_D+1}|\boldsymbol{u}_{N_D+1},\boldsymbol{\theta},\beta,R) = \\ &\mathcal{N}(t_{N_D+1}|v(\boldsymbol{u}_{N_D+1},\boldsymbol{\theta}_{\mathrm{MAP}})+\boldsymbol{g}^T(\boldsymbol{\theta}-\boldsymbol{\theta}_{\mathrm{MAP}}),\beta^{-1}). \end{aligned} \quad (35)$$

Therefore, the predictive distribution defined in (33) becomes a marginal Gaussian distribution and can be evaluated analytically with the result

$$\begin{aligned} &P(t_{N_D+1}|\boldsymbol{u}_{N_D+1},D,\alpha,\beta,R) = \\ &\frac{1}{(2\pi\sigma_t^2)^{1/2}}\exp(-\frac{(t_{N_D+1}-v(\boldsymbol{u}_{N_D+1},\boldsymbol{\theta}_{\mathrm{MAP}}))^2}{2\sigma_t^2}) \\ &= \mathcal{N}(t_{N_D+1}|v(\boldsymbol{u}_{N_D+1},\boldsymbol{\theta}_{\mathrm{MAP}}),\sigma_t^2), \end{aligned} \quad (36)$$

where the variance is given by

$$\sigma_t^2 = \beta^{-1} + \boldsymbol{g}^T\boldsymbol{G}^{-1}_{\mathrm{MAP}}\boldsymbol{g}. \quad (37)$$

We can see that the mean of the predictive distribution is given by the network output $v(\boldsymbol{u}_{N_D+1},\boldsymbol{\theta}_{\mathrm{MAP}})$. The variance σ_t^2 includes two terms, the first of which occurs from the intrinsic noise on the data and is dominated by the hyperparameter β, while the second term arises from the uncertainty in the model parameters $\boldsymbol{\theta}$ and would go to zero with the limit $N_D \to \infty$. We generally use the standard deviation σ_t to measure the prediction error of BNN, so it is clear that the uncertainty has an impact on the prediction quality.

As elaborated above, we first use probability distributions to quantify the uncertainties. Specifically, we place a prior Gaussian distribution over the network's weights to model the epistemic uncertainty and place Gaussian distributions over the network outputs to model the aleatoric uncertainty. Then the optimal values for the regularization hyperparameters α, β and the network weights $\boldsymbol{\theta}$ in Bayesian neural networks can be directly inferred from the data without manually pre-tuning of the network hyperparameters. As compared to the traditional DNN, the BNN can capture the uncertainties naturally and update the regularization hyperparameters timely so as to keep an adaptive network structure, which further yields better channel predictions in actual mobile communication scenarios.

V. NUMERICAL RESULTS

We consider an FDD massive MIMO system where the BS is equipped with M = 64 antennas and serves K = 6 users. A typical OFDM modulation is employed, where the channel delay spread is equal to the duration of the cyclic prefix [30]. The duration of one OFDM symbol is 1/14 ms, while the duration of useful part and cyclic prefix occupied in one OFDM symbol are 1/15 ms and $1/(14 \times 15)$ ms, respectively. Other important OFDM parameters are shown in Table I. Unless otherwise specified, the signal-to-noise ratio (SNR) of channel estimation is 15 dB, and the mobile user speed is 5 m/s. The channel is estimated by the least-squares estimator in the simulations. The CSI samples of training sets and test sets are generated by employing the COST 2100 outdoor channel model, which supports non-stationary characteristics of radio channel and has close agreements with realistic massive MIMO channel measurements [40], [41]. The BS is fixed in the center of a circle area of radius 500 m, and the users are randomly distributed in the area. The scattering environment follows the default setting in [40]. We choose N_t = 600, N_p = 200, and the total number of CSI samples is 50,000. For the BNN, the number of input samples is 4, and other important parameters related to the network are defined in Table I. For the purpose of performance comparison, the state-of-the-art channel prediction techniques, including the traditional AR model [13] and the long short-term memory (LSTM) network [16], are discussed in our simulations. For the AR model, we set the AR order as 4 and 8, respectively, and specify the Wiener filter for estimating the AR model coefficients in Appendix B. The main parameters used to train the LSTM network, a prevailing variant of RNN, can be found in [16]. For the prediction model training, we use one Nvidia GeForce GTX 1080 GPU and one Intel Xeon Processor E5 V4 CPU with 8 cores for acceleration. For the performance metrics, we employ the normalized mean square error (NMSE) to evaluate the prediction accuracy, which is given by

$$\mathrm{NMSE} = \mathbb{E}[\frac{1}{K}\sum_{k=1}^{K}\frac{\|\boldsymbol{h}_k(n)-\tilde{\boldsymbol{h}}_k(n)\|^2}{\|\boldsymbol{h}_k(n)\|^2}]. \quad (38)$$

We also adopt the achievable downlink rate and the BER performance to assess the prediction effectiveness and reliability, respectively.

To evaluate the prediction capability of BNN with different networks, we first summarize the training results for a number

TABLE I
SIMULATION PARAMETERS

OFDM Parameter	Value
Slot duration	2 ms
Number of OFDM symbols within one slot	28
Number of occupied subcarriers	16
Subcarrier spacing	15 kHz
Number of useful samples within one slot	$28 \times 16 = 448$
Central frequency	2.0 GHz
Total bandwidth	20 MHz
Modulation order	16QAM
BNN Parameter	**Value**
Number of hidden layers	3
Unit number in hidden layers	[10, 4, 10]
Unit number in input layer	$4 \times 2 = 8$
Epochs in training	1000
Marquardt adjustment parameter	1e-4

TABLE II
THE TRAINING RESULTS OF BNN WITH DIFFERENT NETWORKS

S	E_D	E_θ	N_θ	γ	NMSE (dB)
1	124.60	2.84	143	86.32	4.62
2	3.40	17.28	164	106.07	-11.08
3	2.79	16.48	185	120.34	-11.67
4	2.67	16.51	206	120.06	-11.67
5	2.67	16.51	227	120.06	-11.61
6	2.67	16.51	248	120.06	-11.02
7	2.67	16.51	269	120.06	-10.66
8	2.67	16.51	290	120.06	-9.870
10	2.67	16.51	332	120.06	-9.21
12	2.67	16.51	374	120.06	-8.54
18	2.67	16.51	500	120.06	-7.57

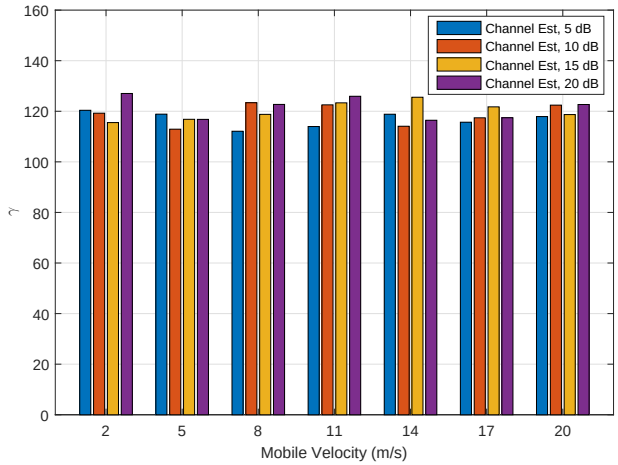

Fig. 5. Effective number of BNN weights with respect to mobile velocity and channel estimation SNR.

of networks of the 10-S-10 architecture in Table II. The 10-S-10 architecture refers to the three hidden layers we set in the BNN, and 10, S, 10 are the number of units in the first, second and third hidden layer, respectively. For simplicity, we just consider different numbers of units in the second hidden layer to represent the different networks, and pick out some representative results from a group of BNN-based predictors. The results shows that the training error E_D and the sum of squares of network weights E_θ are constant for any network with $S \geq 4$. This indicates that these networks with $S \geq 4$ are large enough to properly represent the true prediction function. Also, the effective number of network weights γ reaches a maximum with the 10-4-10 network and remains constant as the total number of parameters N_θ continues to increase, indicating that the 10-4-10 network is the smallest network required to fit the data. Besides, we notice that the prediction NMSE is roughly constant when S is around 4. As S is increased, the networks do not produce consistent results and the prediction accuracy begins to decline. The reason is that more units in the hidden layer means more complex network and taking more time on network training, which will perforce cause a more severe model aging and

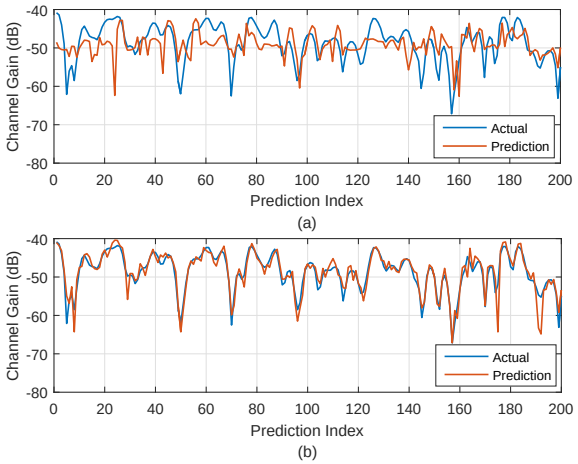

Fig. 6. Channel gains of actual and predicted CSI: a) prediction without regularization; b) prediction with Bayesian regularization.

reduce the prediction quality. So it is important to choose an appropriate network structure and avoid using over-complex networks in actual channel prediction scenarios. Moreover, to further illustrate how the effective network size varies with the user mobile velocity and the channel estimation SNR, we depict the training results of γ using the 10-4-10 network with respect to them in Fig. 5. We can observe that the effective number of network weights does not vary too much with different mobile velocities or channel estimation SNR. Even though with high velocities or low SNR, γ does not become quite large, which demonstrates that Bayesian optimization is a robust method for pruning the network size.

Then, we depict the channel gains of actual and predicted CSI in Fig. 6 to illustrate the generalization ability of BNN intuitively. The channel gain of a channel matrix $\boldsymbol{H}$ is calculated by $10\lg(\mathrm{tr}(\boldsymbol{H}\boldsymbol{H}^\dagger))$. In Fig. 6a, we just adopt the Levenberg-Marquardt algorithm to optimize the training error function and do not consider regularization in predictions, which is in contrast with the predicted CSI with Bayesian regularization in Fig. 6b. As observed from Figs. 6a-6b, we can find that the predicted channels without regularization have much more deviations from the actual channels, whereas the predicted channels with Bayesian regularization follow closely to the trends of the actual ones. This indicates that the proposed BNN-based predictor can generalize well in our designed channel prediction scheme.

Next, we take the AR predictors including the AR(4) and AR(8) predictor, the LSTM predictor and the non-prediction strategy as benchmarks to compare the prediction performance with our proposed BNN-based channel predictor. The non-prediction strategy means that we just utilize the obtained CSI of last slot to replace the predicted CSI for precoding and transmitting data. This is a convenient and cost-efficient idea for our proposed FDD scheme since it does not get involved in processing data and training the prediction model. Fig. 7 shows the NMSE of different channel predictors against the channel estimation SNR. We can observe from Fig. 7 that the BNN predictor has the lowest NMSE among all algorithms for all SNR, indicating that the BNN performs better than

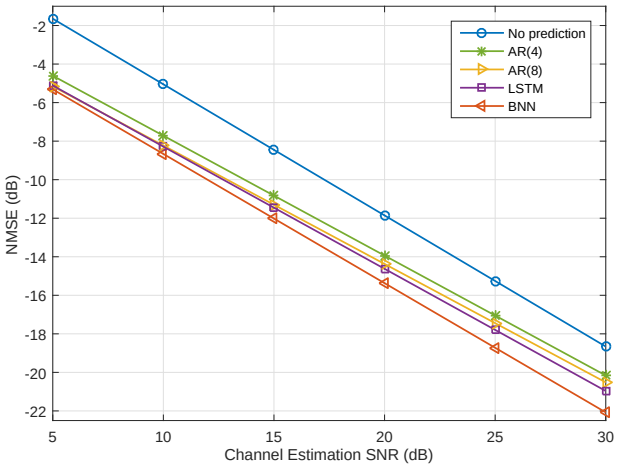

Fig. 7. Prediction NMSE among non-prediction strategy, AR predictors, LSTM predictor and BNN predictor with respect to channel estimation SNR.

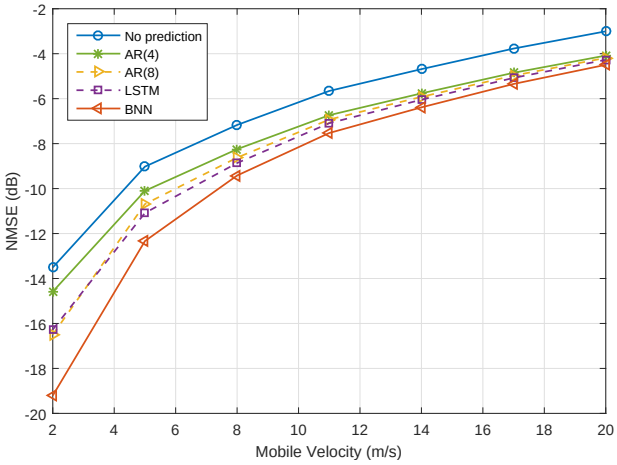

Fig. 8. Prediction NMSE among non-prediction strategy, AR predictors, LSTM predictor and BNN predictor with respect to mobile velocity.

other predictors and the non-prediction strategy in both the low and high SNR scenarios. We can also see from Fig. 7 that the accuracy improvement of the BNN predictor is more remarkable in cases where the channel estimation SNR is high as compared to low SNR. This can be explained by the fact that low SNR makes the estimated channels more noisy and further creates more uncertainty in the collected data, which makes the prediction more difficult. In Fig. 8, we depict the prediction NMSE of different predictors against the mobile velocity. It can be seen that the corollaries of Fig. 8 are akin to the ones in Fig. 7. The BNN predictor outperforms other channel prediction methods in both low and high mobility scenarios, and its accuracy improvement is more prominent for low mobile velocities as compared to high ones, which is consistent with intuition since fast varying channels introduce more model uncertainty and hence make the prediction model harder to characterize the relationship between past and future CSI. From Figs.7-8, we can also observe that the non-prediction strategy performs much worse than channel predictors for all parameter configurations. So we

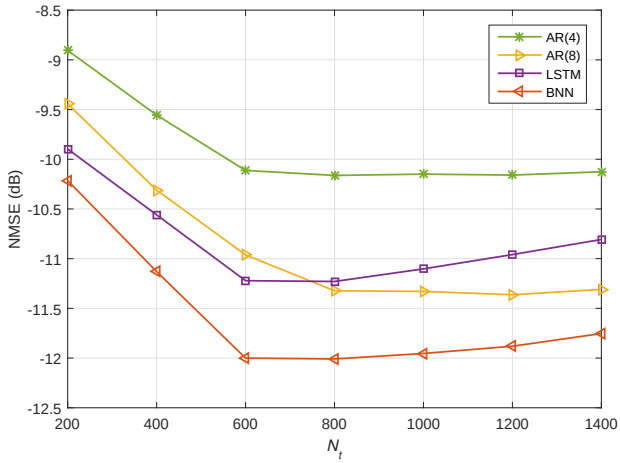

Fig. 9. Prediction NMSE among AR predictors, LSTM predictor and BNN predictor with respect to the number of training samples. $N_p = 200$.

TABLE III
AVERAGE TRAINING TIME OF DIFFERENT PREDICTORS (IN MILLISECONDS)

N_t	AR(4)	AR(8)	LSTM	BNN
200	5.23	5.35	38.18	36.92
400	5.25	5.43	47.58	45.10
600	5.27	5.44	59.39	55.08
800	5.35	5.46	73.98	67.28
1000	5.43	5.52	92.18	82.17
1200	5.54	5.61	114.87	100.36
1400	5.89	6.39	143.13	122.59

can conclude that it is necessary for the FDD massive MIMO systems to perform channel prediction to reap performance gains while the non-prediction strategy is much more simple and convenient to implement, and the proposed BNN-based predictor is superior to the AR model and the LSTM network for our designed channel prediction-aided FDD scheme.

Now we illustrate the impact of model aging on the accuracy of channel prediction models more specifically by comparing the prediction NMSE among the AR predictors, the LSTM predictor and the BNN predictor with respect to the number of training samples N_t. Since N_t can affect the model training time and cause varying degrees of model aging, we choose it as an independent variable and set the number of prediction samples N_p fixedly as 200 to evaluate the effect of model aging. The results are exhibited in Fig. 9, from which we can see that the prediction errors of all predictors have a declining trend as N_t increases at the beginning. Then the NMSE for the AR predictors gets convergent while it has a slight rise for the LSTM predictor and the BNN predictor when the number of training samples becomes larger. This is reasonable since the LSTM network and the BNN model both expend much more time on training as compared to the AR model, which makes the model aging of the former more severe and accordingly increases the prediction error. To further demonstrate this point, using the aforementioned

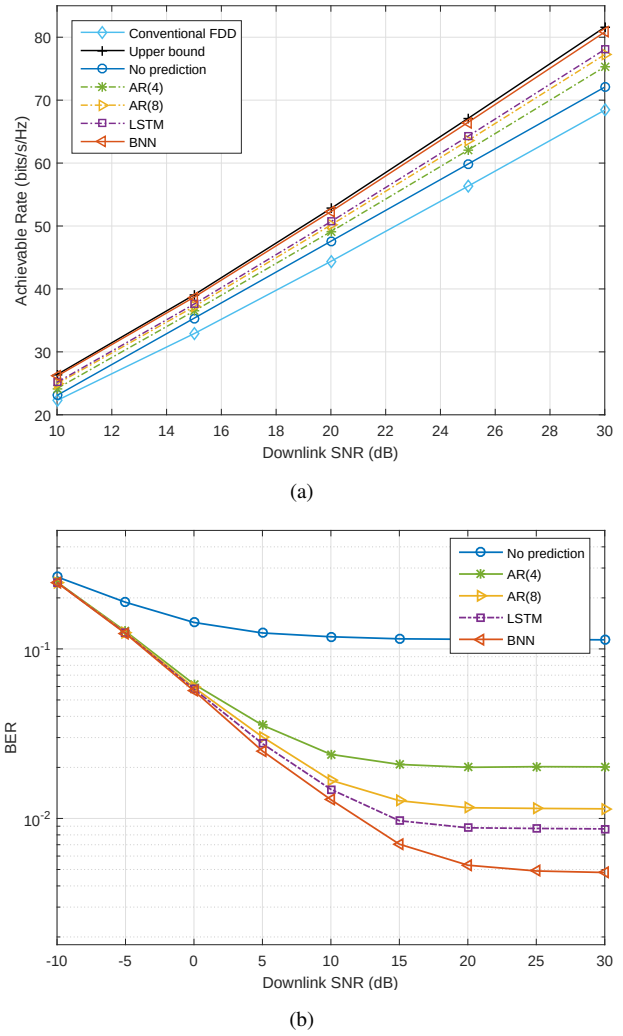

Fig. 10. Performance among conventional FDD scheme, upper bound, non-prediction strategy, AR predictors, LSTM predictor and BNN predictor with respect to downlink transmission SNR: a) achievable downlink rates; b) BER.

GPU and CPU devices we give the average training time (in milliseconds) of these channel predictors in Table III. From Table III, we can find that the training time of neural network-based predictors is tens of times longer than the one of AR predictors for large N_t, which is consistent with the results in Fig. 9. In conclusion, it is essential to set moderate values of N_t for the BNN to avoid under-fitting or severe model aging, while the BNN predictor has a better prediction capability in real-time channel prediction [5].

Finally, to showcase the performance gains provided by the channel prediction-aided FDD massive MIMO systems based on different predictors, we depict the achievable downlink rates and the BER performance among the AR predictors, the LSTM predictor and the BNN predictor against the downlink transmission SNR in Fig. 10, and the results of the non-prediction strategy and the performance upper bound are also included in this figure as benchmarks. As mentioned earlier, the performance upper bound can be reached when the predicted CSI is perfectly accurate. Furthermore, we give the achievable rate of the conventional FDD scheme in Fig. 10a to illustrate the downlink rate improvement obtained by our designed FDD scheme. It is more reasonable to have users with diverse mobility in the same cell, so we assume that the velocities of users are distributed randomly in the range of [0, 20] m/s. From Figs. 10a-b, we can observe that the conventional FDD scheme has the lowest downlink rate as compared to the channel prediction-aided FDD ones, and the non-prediction strategy performs very poorly in the BER simulation even with higher rates than the conventional FDD scheme. We also observe that the BNN predictor outperforms other channel predictors from the viewpoint of both achievable rates and BER performance. More specifically, the BNN predictor always achieves higher downlink transmission rates with sacrificing the least BER performance relatively to the AR model and the LSTM network, especially for high downlink SNR. These results demonstrate that our proposed BNN-based channel prediction method is more promising to facilitate the FDD massive MIMO systems to reap more performance gains.

VI. CONCLUSION

In this paper, we designed a channel prediction-aided downlink transmission scheme for FDD massive MIMO systems to utilize the idle CSI feedback waiting phase, which is usually neglected in current researches despite its importance. Then we proposed a novel BNN-based channel prediction method by introducing a Bayesian learning framework to the neural network. Different from traditional neural networks, the BNN can express the uncertainty efficaciously and optimize the regularization hyperparameters automatically within the network training, which endows it with better prediction capability in practice time-varying channel environments. Also, we investigated the performance of the conventional FDD scheme and the channel prediction-aided ones based on different prediction approaches. Numerical results show that the channel prediction-aided FDD schemes achieve higher downlink rates than the conventional one, and the BNN predictor can reach the state-of-the-art performance for both the prediction quality and the system gains as compared to the AR model and the LSTM network. These results showcase the great potential of the proposed BNN-based channel predictor for enhancing FD-D massive MIMO systems. Note that the Bayesian framework can be extended to compare different models by evaluating the evidence for network architecture. In this way, the BNN can also automatically optimize the model hyperparameters in addition to the regularization hyperparameters, which will further improve its prediction performance. This could be an interesting topic for future work.

ACKNOWLEDGEMENT

The authors would like to thank the editors and the anonymous reviewers, whose invaluable comments helped improve the presentation of this paper substantially.

[5]We can also use more powerful hardware acceleration devices to mitigate severe model aging caused by excess training data. Moreover, it is interesting to develop a light and efficient neural network structure for real-time signal processing, by which the BNN predictor can further improve prediction quality with less training time.

APPENDIX A
DERIVATIONS FOR THE RE-ESTIMATION FORMULAS OF α AND β

We first make estimation for α by maximizing $\ln P(D|\alpha,\beta,R)$ with respect to α. Define the following eigenequation

$$(\beta \boldsymbol{J}^T \boldsymbol{J})\boldsymbol{\psi}_j = \xi_j \boldsymbol{\psi}_j, \tag{39}$$

where $\boldsymbol{\psi}_j$ and ξ_j are the eigenvectors and the eigenvalues of $\beta \boldsymbol{J}^T \boldsymbol{J}$, respectively. From (28), we can know that $\boldsymbol{G}$ has eigenvalues $\alpha + \xi_j$. Now consider the derivative of the term $\ln\det \boldsymbol{G}_{\rm MAP}$ in (29) with respect to α, we obtain

$$\begin{aligned}\frac{d}{d\alpha}\ln\det \boldsymbol{G}_{\rm MAP} &= \frac{d}{d\alpha}\ln\prod_{j=1}^{N_\theta}(\xi_j+\alpha)\\ &= \frac{d}{d\alpha}\sum_{j=1}^{N_\theta}\ln(\xi_j+\alpha)\\ &= \sum_{j=1}^{N_\theta}\frac{1}{\xi_j+\alpha}.\end{aligned} \tag{40}$$

Then the stationary points of (29) with respect to α can be derived as

$$-E_\theta^{\rm MAP} - \frac{1}{2}\sum_{j=1}^{N_\theta}\frac{1}{\xi_j+\alpha} + \frac{N_\theta}{2\alpha} = 0. \tag{41}$$

Multiplying both sides of (41) by 2α and rearranging, we have

$$2\alpha E_\theta^{\rm MAP} = N_\theta - \alpha\sum_{j=1}^{N_\theta}\frac{1}{\xi_j+\alpha} = \sum_{j=1}^{N_\theta}\frac{\xi_j}{\xi_j+\alpha} = \gamma. \tag{42}$$

Thus, we can solve for α from (42) as

$$\alpha = \frac{\gamma}{2E_\theta^{\rm MAP}}. \tag{43}$$

Note that (43) is an implicit solution for α since γ and $\theta_{\rm MAP}$ both depend on α, we can adopt an iterative procedure to determine α. Specifically, we first choose an initial value for α, and use this to find $\boldsymbol{\theta}_{\rm MAP}$ by the Levenberg-Marquardt algorithm and evaluate γ using (42). Then these values are used to re-estimate α by (43). We repeat the process until convergence. Similarly, we can estimate β by maximizing $\ln P(D|\alpha,\beta,R)$ with respect to β. Consider that the eigenvalues ξ_j defined by (39) are proportional to β, that is to say, $d\xi_j/d\beta = \xi_j/\beta$. Therefore, we can obtain

$$\frac{d}{d\beta}\ln\det \boldsymbol{G}_{\rm MAP} = \frac{d}{d\beta}\sum_{j=1}^{N_\theta}\ln(\xi_j+\alpha) = \frac{1}{\beta}\sum_{j=1}^{N_\theta}\frac{\xi_j}{\xi_j+\alpha} = \frac{\gamma}{\beta}, \tag{44}$$

and the stationary points of (29) with respect to β satisfy

$$-E_D^{\rm MAP} - \frac{\gamma}{2\beta} + \frac{N_D}{2\beta} = 0. \tag{45}$$

Obviously, the re-estimation formula of β can be derived from (45) as

$$\beta = \frac{N_D-\gamma}{2E_D^{\rm MAP}}. \tag{46}$$

Likewise, this is an implicit solution for β and can be solved by an iterative method.

APPENDIX B
THE AUTOREGRESSIVE MODEL

In the AR model, the predicted channel $\tilde{h}(n)$ is represented by a linear combination of the previous P channel samples:

$$\tilde{h}(n) = \sum_{k=1}^{P}\phi_k h(n-k), \tag{47}$$

where P is the AR order and $\{\phi_k\}_{k=1}^P$ represent the model coefficients. We use Wiener filter, also known as linear minimum mean square error estimator, to estimate the AR model coefficients:

$$\boldsymbol{\Phi} = \boldsymbol{A}^{-1}\boldsymbol{r}, \tag{48}$$

where $\boldsymbol{\Phi} = [\phi_1, \phi_2, ..., \phi_P]^T$ is the AR model coefficient vector, and $\boldsymbol{r} = [r_1, r_2, ..., r_P]^T$ is the autocorrelation vector with coefficient $r_k = E\{h(n)h^*(n-k)\}$. $\boldsymbol{A}$ is the autocorrelation matrix, which is a Toeplitz matrix given as

$$\boldsymbol{A} = \begin{bmatrix} r_0 & r_1 & r_2 & \cdots & r_{P-2} & r_{P-1} \\ r_1 & r_0 & r_1 & \cdots & r_{P-3} & r_{P-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ r_{P-2} & r_{P-3} & r_{P-4} & \cdots & r_0 & r_1 \\ r_{P-1} & r_{P-2} & r_{P-3} & \cdots & r_1 & r_0 \end{bmatrix}, \tag{49}$$

where the coefficient $A_{ij} = E\{h(n-i)h^*(n-j)\}$ (i, j = 1,2,$\cdots$,P). The coefficients A_{ij} and r_k can be estimated by means of calculating the autocorrelation function of past channel samples, which does not require the maximum Doppler shift or the number of scatterers.

REFERENCES

[1] Z. Tao, T. Wang, and S. Wang, "Improve Downlink Rates of FDD Massive MIMO Systems by Exploiting CSI Feedback Waiting Phase," in *Proc. IEEE Globecom'19*, Waikoloa, HI, 2019.

[2] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, and F. Tufvesson, "Scaling Up MIMO: Opportunities and Challenges with Very Large Arrays," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40–60, Jan. 2013.

[3] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and Spectral Efficiency of Very Large Multiuser MIMO Systems," *IEEE Trans. Commun.*, vol. 61, no. 4, pp. 1436–1449, Apr. 2013.

[4] T. L. Marzetta, "Massive MIMO: An Introduction," *Bell Labs Tech. J.*, vol. 20, no. 20, pp. 11–22, Mar. 2015.

[5] S. Noh, M. D. Zoltowski, and D. J. Love, "Training Sequence Design for Feedback Assisted Hybrid Beamforming in Massive MIMO Systems," *IEEE Trans. Commun.*, vol. 64, no. 1, pp. 187–200, Jan. 2016.

[6] E. Bjornson, E. G. Larsson, and T. L. Marzetta, "Massive MIMO: ten myths and one critical question," *IEEE Commun. Mag.*, vol. 54, no. 2, pp. 114–123, Feb. 2016.

[7] P. W. Chan, E. S. Lo, R. Wang, E. K. S. Au, V. K. N. Lau, R. S. Cheng, W. H. Mow, R. D. Murch, and K. B. Letaief, "The evolution path of 4G networks: FDD or TDD?" *IEEE Commun. Mag.*, vol. 44, no. 12, pp. 42–50, Dec. 2006.

[8] J. Choi, D. J. Love, and P. Bidigare, "Downlink Training Techniques for FDD Massive MIMO Systems: Open-Loop and Closed-Loop Training With Memory," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 802–814, Oct. 2014.

[9] S. Noh, M. D. Zoltowski, Y. Sung, and D. J. Love, "Pilot Beam Pattern Design for Channel Estimation in Massive MIMO Systems," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 787–801, May 2014.

[10] S. Bazzi and W. Xu, "Downlink Training Sequence Design for FDD Multiuser Massive MIMO Systems," *IEEE Trans. Signal Process.*, vol. 65, no. 18, pp. 4732–4744, Sept. 2017.

[11] M. B. Mashhadi, Q. Yang, and D. Gündüz, "Distributed Deep Convolutional Compression for Massive MIMO CSI Feedback," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2621–2633, Apr. 2021.

[12] X. Ma, Z. Gao, F. Gao, and M. Di Renzo, "Model-Driven Deep Learning Based Channel Estimation and Feedback for Millimeter-Wave Massive Hybrid MIMO Systems," *IEEE J. Sel. Areas Commun.*, Jun. 2021, DOI: 10.1109/JSAC.2021.3087269.

[13] A. Duel-Hallen, S. Hu, and H. Hallen, "Long Range Prediction of Fading Signals: Enabling Adaptive Transmission for Mobile Radio Channels," *IEEE Signal Process. Mag.*, vol. 17, no. 3, pp. 62–75, May 2000.

[14] I. C. Wong and B. L. Evans, "Low-complexity adaptive high-resolution channel prediction for OFDM systems," in *Proc. IEEE GLOBECOM'06*, San Francisco, CA, USA, 2006.

[15] T. Zemen, C. F. Mecklenbrauker, F. Kaltenberger, and B. H. Fleury, "Minimum-Energy Band-Limited Predictor With Dynamic Subspace Selection for Time-Variant Flat-Fading Channels," *IEEE Trans. Signal Process.*, vol. 55, no. 9, pp. 4534–4548, Sept. 2007.

[16] Y. Zhu, X. Dong, and T. Lu, "An Adaptive and Parameter-Free Recurrent Neural Structure for Wireless Channel Prediction," *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 8086–8096, Aug. 2019.

[17] M. Ke, Z. Gao, Y. Wu, X. Gao, and R. Schober, "Compressive Sensing-Based Adaptive Active User Detection and Channel Estimation: Massive Access Meets Massive MIMO," *IEEE Trans. Signal Process.*, vol. 68, pp. 764–779, Jan. 2020.

[18] M. Ke, Z. Gao, Y. Wu, X. Gao, and K.-K. Wong, "Massive Access in Cell-Free Massive MIMO-Based Internet of Things: Cloud Computing and Edge Computing Paradigms," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 3, pp. 756–772, Mar. 2021.

[19] M. Zhou, T. Wang, and S. Wang, "Spectrum Sensing Across Multiple Service Providers: A Discounted Thompson Sampling Method," *IEEE Commun. Lett.*, vol. 23, no. 12, pp. 2402–2406, Dec. 2019.

[20] Y. Lin, T. Wang, and S. Wang, "UAV-Assisted Emergency Communications: An Extended Multi-Armed Bandit Perspective," *IEEE Commun. Lett.*, vol. 23, no. 5, pp. 938–941, May 2019.

[21] Z. Kuai and S. Wang, "Thompson Sampling Based Antenna Selection with Partial CSI for TDD Massive MIMO Systems," *IEEE Trans. Commun.*, vol. 68, no. 12, pp. 7533–7546, Dec. 2020.

[22] H. Kim, S. Kim, H. Lee, C. Jang, Y. Choi, and J. Choi, "Massive MIMO Channel Prediction: Kalman Filtering vs. Machine Learning," *IEEE Trans. Commun.*, vol. 69, no. 1, pp. 518–528, Jan. 2021.

[23] S. Chen, Z. Jiang, S. Zhou, Z. Niu, Z. He, A. Marinescu, and L. A. Dasilva, "Learning-Based Remote Channel Inference: Feasibility Analysis and Case Study," *IEEE Trans. Wireless Commun.*, vol. 18, no. 7, pp. 3554–3568, 2019.

[24] J. Yuan, H. Q. Ngo, and M. Matthaiou, "Machine Learning-Based Channel Prediction in Massive MIMO With Channel Aging," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 2960–2973, May 2020.

[25] Z. Jiang, S. Chen, A. F. Molisch, R. Vannithamby, S. Zhou, and Z. Niu, "Exploiting Wireless Channel State Information Structures Beyond Linear Correlations: A Deep Learning Approach," *IEEE Commun. Mag.*, vol. 57, no. 3, pp. 28–34, Mar. 2019.

[26] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight Uncertainty in Neural Networks," in *Proc. ICML'15*, vol. 37, Lille, France, 2015, pp. 1613–1622.

[27] A. Loquercio, M. Segu, and D. Scaramuzza, "A General Framework for Uncertainty Estimation in Deep Learning," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 3153–3160, 2020.

[28] E. Goan and C. Fookes, "Bayesian Neural Networks: An Introduction and Survey," in *Case Studies in Applied Bayesian Data Science*. Springer, 2020, pp. 45–87.

[29] M. Elnaggar, K. Whitehouse, and C. H. Fleming, "Bayesian Wireless Channel Prediction for Safety-Critical Connected Autonomous Vehicles," in *Proc. NIPS'18*, Montreal, Canada, 2018.

[30] T. L. Marzetta, E. G. Larsson, H. Yang, and H. Q. Ngo, *Fundamentals of Massive MIMO*. Cambridge, U.K.: Cambridge University Press, 2016.

[31] M. B. Mashhadi, Q. Yang, and D. Gündüz, "CNN-Based Analog CSI Feedback in FDD MIMO-OFDM Systems," in *Proc. IEEE ICASSP'20*, Barcelona, Spain, 2020.

[32] I. C. Wong and B. L. Evans, "Exploiting Spatio-Temporal Correlations in MIMO Wireless Channel Prediction," in *Proc. IEEE GLOBECOM'06*, San Francisco, CA, USA, 2006.

[33] D. J. C. Mackay, "A Practical Bayesian Framework for Backprop Networks," *Neural Comput.*, vol. 4, no. 3, pp. 448–472, May 1992.

[34] D. J. C. MacKay, "Bayesian Interpolation," *Neural Comput.*, vol. 4, no. 3, pp. 415–447, May 1992.

[35] R. M. Neal, "Bayesian Learning for Neural Networks," Ph.D. dissertation, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 1995.

[36] F. D. Foresee and M. T. Hagan, "Gauss-Newton Approximation to Bayesian Learning," in *Proc. ICNN'97*, Houston, TX, USA, 1997.

[37] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Networks*, vol. 5, no. 6, pp. 989–993, Nov. 1994.

[38] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer, 2006.

[39] M. T. Hagan, H. B. Demuth, and M. H. Beale, *Neural Network Design*. Boston, U.S.: PWS Publishing Co., 1996.

[40] L. Liu, C. Oestges, J. Poutanen, K. Haneda, P. Vainikainen, F. Quitin, F. Tufvesson, and P. De Doncker, "The COST 2100 MIMO channel model," *IEEE Wireless Commun.*, vol. 19, no. 6, pp. 92–99, Dec. 2012.

[41] M. Zhu, G. Eriksson, and F. Tufvesson, "The COST 2100 Channel Model: Parameterization and Validation Based on Outdoor MIMO Measurements at 300 MHz," *IEEE Trans. Wireless Commun.*, vol. 12, no. 2, pp. 888–897, Feb. 2013.

Zhihao Tao received the BS degree in electronics and information engineering from Sichuan University, Chengdu, China, in 2018, and the MS degree in communication and information systems from Nanjing University, Nanjing, China, in 2021. His research interests include wireless communications and machine learning.

Shaowei Wang (S'06-M'07-SM'13) received the PhD degree from Wuhan University, Wuhan, China, in 2006, and joined the School of Electronic Science and Engineering at Nanjing University, Nanjing, China, as a faculty member in the same year, where he is currently a Full Professor. From 2012 to 2013, he was a Visiting Scholar/Professor with Stanford University, Stanford, CA, USA, and The University of British Columbia, Vancouver, BC, Canada. His research interests include communications and networking, operations research and machine learning.