IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 29, 2020, accepted June 29, 2020, date of publication July 2, 2020, date of current version July 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3006518

Deep Learning Based Channel Estimation for MIMO Systems With Received SNR Feedback

JAE-MO KANG[1], (Member, IEEE), CHANG-JAE CHUN[2], (Member, IEEE), AND IL-MIN KIM[3], (Senior Member, IEEE)

[1]Department of Artificial Intelligence, Kyungpook National University, Daegu 41566, South Korea
[2]Industry Applications Research Division, Korea Electrotechnology Research Institute, Changwon 51543, South Korea
[3]Department of Electrical and Computer Engineering, Queen's University, Kingston, ON K7L 3N6, Canada

Corresponding author: Il-Min Kim (ilmin.kim@queensu.ca)

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), and in part by the Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education under Grant 2020R1I1A3073651.

ABSTRACT Channel estimation with received signal-to-noise ratio (SNR) feedback is promising and effective for practical wireless multiple-input multiple-output (MIMO) systems. In this paper, we investigate the channel estimation problem for the MIMO system with received SNR feedback, of which goal is to estimate the MIMO channel coefficients at a transmitter based on the received SNR feedback information from a receiver in the sense of minimizing the mean square error (MSE) of the channel estimation. For analysis, we consider two very common and widely adopted scenarios of fading: (i) quasi-static block fading and (ii) time-varying fading. In both fading scenarios, it is generally challenging to analytically tackle the channel estimation problem due to its nonlinearity and nonconvexity. To intelligently and effectively address this issue, deep learning is exploited in this paper. First, in the quasi-static block fading scenario, we propose a novel learning scheme for joint channel estimation and pilot signal design by constructing a deep autoencoder via a convolutional neural network (CNN). Also, in the time-varying fading scenario, a novel channel estimation scheme is developed by connecting a recurrent neural network (RNN) to a CNN. Moreover, in both fading scenarios, we present new and effective ways to train the proposed schemes using generative adversarial networks (GANs) to address the practical issue of a limited number of actual channel samples (i.e., real-world data) required for training. Through extensive numerical simulations, we demonstrate effectiveness and superior performance of the proposed schemes.

INDEX TERMS Autoencoder, channel estimation, convolutional neural network (CNN), deep learning, generative adversarial network (GAN), recurrent neural network (RNN), received SNR feedback, pilot signal design.

I. INTRODUCTION

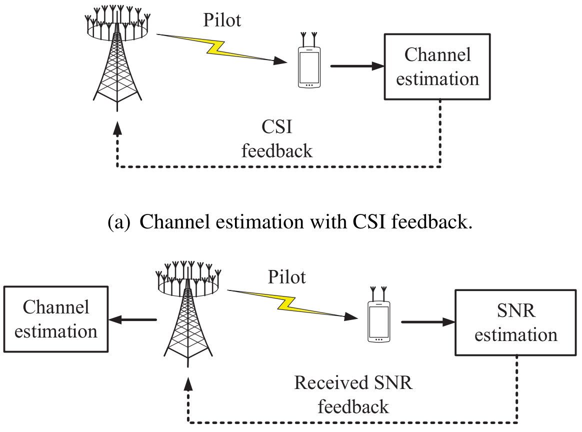
A. MOTIVATIONS

Multiple-input multiple-output (MIMO) is one of the widely adopted technologies in many wireless standards (e.g., in IEEE 802.11 and 802.16 families) due to its outstanding capability of achieving high data rates through multiplexing gain and providing low error rates through spatial diversity [1], [2]. In any wireless communication systems, the channel estimation is crucial and essential for coherent data transmission and detection [2]. In MIMO wireless communication systems, acquiring accurate channel state information (CSI) at the transmitter is particularly important, because the system performance (e.g., capacity or bit error rate) critically depends on the accuracy of the CSI at the transmitter [1], [2]. In particular, the accurate CSI at the transmitter can be used to substantially improve the performance of the MIMO system through the transmit beamforming or precoding. Thus, the accurate channel estimation at the transmitter is a practically very important problem for the MIMO system.

1) MOTIVATIONS FOR STUDYING THE CHANNEL ESTIMATION WITH RECEIVED SNR FEEDBACK

In the literature, several methods have been developed to provide the CSI at the transmitter [3]–[8]. Among those,

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



(a) Channel estimation with CSI feedback.

(b) Channel estimation with received SNR feedback.

FIGURE 1. Two different channel estimation schemes at the transmitter.

one obvious and common method is to first estimate the CSI at the receiver using downlink pilot signals sent from the transmitter, and then, to feed back the estimated CSI (which is complex-valued in general) from the receiver to the transmitter via a feedback link [3], [4], as shown in Fig. 1. This method is referred to as the *channel estimation with CSI feedback*. Let M_t and M_r denote the number of transmit and receive antennas, respectively, in the downlink of MIMO system. In the channel estimation with CSI feedback, M_tM_r complex numbers (or 2 M_tM_r real numbers) corresponding to all the MIMO channel coefficients need to be fed back for the CSI acquisition at the transmitter. The channel estimation with CSI feedback works well for the small to moderate number M_t of transmit antennas. However, it does not scale well when M_t becomes (very) larger (e.g., for the massive MIMO system with $M_t \gg 1$), because the signaling overheads (i.e., 2 M_tM_r) required for transmission of the CSI feedback overwhelmingly increase proportionally to M_t [5], [6].

Another method that addresses the scalability issue is to estimate the CSI at the transmitter directly using uplink pilot signals sent from the receiver [5], [6]. This method is referred to as the *uplink channel estimation*. However, the uplink channel estimation assumes the reciprocity between the downlink and uplink channels, and thus, its applicability is limited only to the time-division duplex (TDD) scenario. For the general scenario such as in the frequency-division duplex (FDD) case where the downlink and uplink channels are not reciprocal, the uplink channel estimation is generally not possible to use for the CSI acquisition at the transmitter.

In order to break through the above limitations (i.e., the scalability and applicability), recently, a very promising and effective method has been developed [7], [8]. As depicted in Fig. 1(b), the transmitter first sends the pilot signal to the receiver, and then, the receiver feeds back to the transmitter the signal-to-noise ratio (SNR) estimated (or measured) at each of receive antennas (which is real-valued,[1] not complex-valued). Thereafter, the transmitter estimates the CSI based on the received SNR feedback information. This method is referred to as the *channel estimation with received SNR feedback*. Compared to the channel estimation with CSI feedback, the channel estimation with received SNR feedback requires (much) lower signaling overheads for the feedback transmission since the amount of feedback is proportional to the number M_r of receive antennas (not the number of M_t transmit antennas). Thus, it scales well, even when M_t is (very) large, which is in sharp contrast to the channel estimation with CSI feedback. Note that in practice, the required amount of feedback is indeed (very) small because the receiver is a mobile device in the real-world scenarios of downlink MIMO, which is typically equipped with a (very) small number of antennas due to its limited hardware size (i.e., M_r is (very) small) [1], [2], [8].

Another merit of the channel estimation with received SNR feedback is that it can be easily implemented in practice because the mechanism for received signal strength indicator (SSI) or channel quality indicator (CQI) feedback (which is essentially the same as the received SNR feedback) is already incorporated into the most existing and emerging wireless communication systems/standards at the hardware and system levels [7], [8]. Furthermore, it can be used for either the TDD or FDD case (i.e., whether the downlink and uplink channels are reciprocal or not).

Motivated by its significant benefits of low signaling overhead, diverse applicability, and easy implementation, in this paper, we study the channel estimation with received SNR feedback.

2) MOTIVATIONS FOR DEVELOPING A NEW CHANNEL ESTIMATION SCHEME WITH RECEIVED SNR FEEDBACK

Despite the above benefits, a critical challenge involved in the channel estimation with received SNR feedback is that it is a nonlinear estimation problem since the received SNR (or SSI/CQI) values are all nonlinear functions of the channel coefficients to be estimated. Also, the estimation problem is even nonconvex. Due to these challenges, it is very difficult to analytically develop any channel estimation algorithms for the case of received SNR feedback. One possible approach to overcome these challenges is to exploit the machine learning technique, which is one of the widely adopted traditional (yet effective) techniques to tackle the nonlinear and nonconvex problems [9], [10]. Specifically, as studied in [8], the channel estimation with received SNR feedback can be considered as the phase retrieval problem from the estimation-theoretic perspective [11]. One of the widely used and popular algorithms to tackle the phase retrieval problem is the Gerchberg-Saxton (GS) algorithm in [12], which can be considered as a machine

[1] Although the received SNR value is real-valued (i.e., phaseless), it contains enough information about phases (as well as magnitudes) of the fading channel coefficients, as can be seen from (3). Thus, even if the received SNR value is used for the channel estimation, it is possible to extract both the phase and magnitude information of the fading channel coefficients.

learning technique [11]. The GS algorithm can be applied to the channel estimation with received SNR feedback. However, the channel estimation performance of the GS algorithm turns out to be unsatisfactory due to global phase ambiguities in the obtained channel estimates [8], [11]. Furthermore, the GS algorithm generally requires a considerable number of iterations to compute the channel estimates, which significantly limits its applicability to real-time applications and/or time-varying environments.

To the best of our knowledge, all the aforementioned critical issues have not been addressed in the literature. In addition, for the MIMO system, an effective and high-performing channel estimation scheme with received SNR feedback has not been developed yet. These motivated our work.

3) MOTIVATIONS FOR EXPLOITING DEEP LEARNING TECHNIQUES

In this paper, the channel estimation with received SNR feedback is studied for the MIMO system. Particularly, to address all above issues intelligently and effectively, we exploit the deep learning technique, which has recently gained an upsurge of research interest including in the area of wireless communications [13]–[21]. One of the key motivations of adopting the deep learning technique in this paper is its excellent ability to significantly outperform the traditional machine learning techniques and to even surpass the other analytical algorithms in solving the nonlinear and nonconvex problems [9], [10].

B. CONTRIBUTIONS

The main contributions of our work lie in that based on deep learning, we develop effective and high-performing channel estimation techniques for the MIMO system with received SNR feedback. Clearly, the proposed scheme possesses several critical benefits over the traditional GS algorithm. To the best of our knowledge, our work is the first to utilize deep learning for the channel estimation with received SNR feedback in the MIMO system. We conduct the analysis for two very common and widely adopted (yet distinct) scenarios of fading: (i) quasi-static block fading (assuming negligible receiver mobility) and (ii) time-varying fading (assuming non-negligible receiver mobility).

The main contributions of our work are as follows.

- In the quasi-static block fading scenario, we develop a novel scheme for joint channel estimation and pilot signal design by constructing a deep autoencoder, which learns (or estimates) the MIMO channel coefficients at the transmitter while optimizing the pilot signal based on the received SNR feedback information in the sense of minimizing the mean square error (MSE) of the channel estimation. In this scheme, the encoder is constructed using a two-layer (i.e., shallow) feedforward neural network (FNN) such that the received SNR feedback information is learnable and the pilot signal is optimizable. Also, the decoder is constructed based on a convolutional neural network (CNN) such that the spatial (and the other useful) features of the MIMO channel coefficients are effectively and fully exploitable for the channel estimation and the MIMO channel coefficients are accurately predictable from the received SNR feedback information.
- As another very important fading scenario, we also study the time-varying fading scenario. In this fading scenario, we propose a new channel estimation scheme, which sequentially estimates the time-varying MIMO channel coefficients at the transmitter from the received SNR feedback information in the sense of minimizing the channel estimation MSE. In this scheme, considering the temporal dependency of the channel variation, we further utilize the previous channel estimates for effective learning in addition to the received SNR feedback information.
  Different from the quasi-static block fading scenario, in the time-varying fading scenario, additional features (e.g., the temporal features) of the MIMO channels can be further utilized for learning. To this end, we take an innovative and effective approach: we combine a recurrent neural network (RNN) and a CNN to effectively learn and extract the temporal features of the time-varying MIMO channels along with their spatial features. The key idea of employing the RNN is that we regard the time-varying MIMO channels as a sequence to be predicted (like natural language) and we extract their temporal features through the RNN. To address the issue of vanishing/exploding gradients involved in the naive RNN, we utilize the long short-term memory (LSTM), which is an advanced RNN that effectively addresses the vanishing/exploding gradient problem. Indeed, the LSTM is very useful to learn the temporal features of the time-varying MIMO channels. However, it does not properly capture the other useful features (such as the spatial features). To address this issue, in the proposed scheme, we further utilize the CNN and combine it with the LSTM. Note that our approach of combining the RNN and the CNN enables to extract the spatial features and the other high-level features of the time-varying MIMO channels (which cannot be effectively captured only with the RNN) and to effectively utilize those features for the channel estimation in addition to the temporal features.
- The proposed channel estimation (or learning) schemes generally require a considerable number of actual channel samples (i.e., transfer functions or impulse responses of the real-world wireless environment) for training. In practice, however, it may be very difficult to acquire enough actual channel samples by field measurements. To address this critical issue, for both fading scenarios, we present novel and effective ways to train the proposed schemes given the limited number of actual channel samples. The key idea is to artificially generate additional channel samples using deep generative models, namely, the generative adversarial network (GAN) and

the conditional GAN (CGAN), respectively.[2] We also present how to train the proposed scheme using the artificial channel samples. From the numerical results, it turns out that the proposed training approach using the GAN or CGAN is very useful and effective for the practical applications.

- Extensive numerical simulations and experimental validations are conducted to demonstrate the performance and effectiveness of the proposed schemes. Through the numerical results, it turns out that the proposed scheme considerably outperforms baseline schemes in terms of the channel estimation MSE and mutual information. The performance improvement is indeed very significant. Therefore, the proposed schemes are very useful and meaningful in the practical aspect. Also, from the numerical results, we draw useful insights.

C. RELATED WORKS

In the literature, several channel estimation techniques have been developed for the MIMO system based on deep learning [16]–[21]. However, in the schemes of [16]–[21], the channels were estimated *without* considering the received SNR feedback. Thus, the extension of [16]–[21] to the MIMO channel estimation with received SNR feedback is never straightforward.

In [22]–[24], several channel modeling approaches have been suggested for the MIMO system. Also, in [25] and [26], deep learning based direction-of-arrival (DOA) estimation technique and hybrid precoding design technique have been developed for the massive MIMO system, respectively. However, the techniques of [22]–[26] assume some specific channel models that are parameterized by a set of parameters such as array response, angle-of-departure, and direction-of-arrival (i.e., parametric channel models). Different from [22]–[26], in this paper, we do not make such assumptions and we consider the general non-parametric channel model, which does not rely on the specific parameters. Thus, our proposed scheme can be used regardless of the channel model (i.e., whether the channel model is parametric or not).

Also, in [22]–[24], the goal was to estimate the statistics of the channel parameters (rather than estimating the instantaneous values of the channel coefficients) using certain probability distribution. Therefore, the extensions of [22]–[24] to the channel estimation with received SNR feedback are never straightforward. Moreover, the techniques of [25] and [26] are confined only to the quasi-static block fading. Thus, the extensions of [25] and [26] to the time-varying fading are never straightforward.

D. ORGANIZATION

This paper is organized as follows. In Section II, the system model and the problem are described. The proposed schemes for the scenarios of quasi-static block fading and time-varying fading are presented in Sections III and IV, respectively. The numerical results are presented in Section V. This paper is concluded in Section VI.

Notation: We use $(\cdot)^T$, $(\cdot)^H$, $\|\cdot\|$, $\|\cdot\|_F$, $\otimes$, $\mathbb{E}[\cdot]$, ∇, $\%$, and $Re[\cdot]$ to denote the transpose, conjugate transpose, Euclidean norm, Frobenious norm, Kronecker product, expectation, gradient operator, modulo operator, and real part of the argument, respectively. Also, $|\cdot|$ denotes the element-wise absolute value for a vector or the cardinality for a set. The $p \times p$ identity matrix and the imaginary unit are denoted by $\mathbf{I}_p$ and $\jmath = \sqrt{-1}$, respectively.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. SYSTEM MODEL

We consider a MIMO system composed of a transmitter and a receiver. The transmitter is equipped with M_t antennas and the receiver is equipped with M_r antennas. Let $\mathbf{H}(n) \in \mathbb{C}^{M_r \times M_t}$ denote the MIMO channel matrix from the transmitter to the receiver at the nth symbol time, where the (k, j)th element of $\mathbf{H}(n)$ is given by $h_{k,j}(n)$, which denotes the channel coefficient from the jth antenna of the transmitter to the kth antenna of the receiver.

For the purpose of channel estimation, at each symbol time, the transmitter sends pilot symbols that are priori known to the receiver, and then, the receiver feeds back the received SNR values estimated at its antennas to the transmitter.[3] It is assumed that in each symbol time, there are several mini-time slots, among which the first N_p consecutive mini-time slots are dedicated to the transmission of pilot symbols. At the nth symbol time, the received signal at the receiver is given by

$$\mathbf{R}(n) = \mathbf{H}(n)\mathbf{S}(n) + \mathbf{Q}(n) \tag{1}$$

where $\mathbf{Q}(n) \in \mathbb{C}^{M_r \times N_p}$ is the matrix of received noises. Also, $\mathbf{S}(n) \triangleq [\mathbf{s}_1(n), \cdots, \mathbf{s}_{N_p}(n)] \in \mathbb{C}^{M_t \times N_p}$ is the matrix of pilot symbols, where $\mathbf{s}_m(n) \in \mathbb{C}^{M_t \times 1}$ is the vector of pilot symbols sent at the mth mini-time slot, $m = 1, \cdots, N_p$. In this paper, we will refer to the pilot symbol matrix $\mathbf{S}$ simply as the pilot signal. The transmit power of the pilot signal is constrained by $\|\mathbf{S}\|_F^2 = \mathrm{Tr}(\mathbf{S}\mathbf{S}^H) \leq P$, where P denotes the maximum amount of transmit power at the transmitter.[4]

[2] The GAN and CGAN proposed in this paper can be effectively used for predicting (or estimating) the probability distributions as well as the spatial/temporal correlations of the MIMO channels. One simple, yet effective, approach is to directly extract the correlation information from the artificial channel samples (along with the actual channel samples) based on the sample mean approach.

[3] In practice, the SNR values must be estimated, and thus, it will be noisy [28]. Fortunately, as shown in [28], the SNR estimation error gets smaller (and asymptotically, it approaches zero) when the transmit power of the pilot signal becomes larger. In practice, therefore, the transmit power of the pilot signal should be large enough such that the impact of the SNR estimation error becomes negligible.

[4] Even when another type of (power) constraint, such as the peak-to-average-power ratio (PAPR) constraint, is taken into account, the proposed scheme can be readily extended. Specifically, the pilot signals can be designed (or learned) to satisfy the constraint with only some minor modification on the projection operator in (15).

By vectorizing $\mathbf{R}(n)$ of (1) and using the result of $vec(\mathbf{AXB}) = (\mathbf{B}^T \otimes \mathbf{A})vec(\mathbf{X})$ [27], we have

$$\mathbf{r}(n) = \left(\mathbf{S}^T(n) \otimes \mathbf{I}_{M_r}\right)\mathbf{h}(n) + \mathbf{q}(n) \tag{2}$$

where $\mathbf{h}(n) \triangleq vec(\mathbf{H}(n))$, $\mathbf{r}(n) \triangleq vec(\mathbf{R}(n))$, and $\mathbf{q}(n) \triangleq vec(\mathbf{Q}(n))$. During the pilot signal transmission, the received SNR (or SSI/CQI) values estimated at the antennas of the receiver are given by the elements of $g^2(n)\left|\left(\mathbf{S}^T(n) \otimes \mathbf{I}_{M_r}\right)\mathbf{h}(n)\right|^2$, where $g^2(n) \triangleq \frac{1}{\xi^2(n)}$ and $\xi^2(n)$ is the variance of each element of $\mathbf{q}(n)$. In this paper, it is assumed that only the magnitudes of the received SNR values (i.e., the elements of $g(n)\left|\left(\mathbf{S}^T(n) \otimes \mathbf{I}_{M_r}\right)\mathbf{h}(n)\right|$) are fed back from the receiver to the transmitter.[5] Thus, at the nth symbol time, the feedback signal received by the transmitter is given by

$$\boldsymbol{\gamma}(n) = g(n)\left|\left(\mathbf{S}^T(n) \otimes \mathbf{I}_{M_r}\right)\mathbf{h}(n)\right| + \boldsymbol{\eta}(n) \tag{3}$$

where $\boldsymbol{\eta}(n)$ denotes the noise-plus-error term involved in the SNR feedback, which takes into account the impairments (including the antenna noises, the estimation errors, the measurement errors, the feedback quantization errors, etc) involved in the SNR estimation, measurement, quantization processes, etc.

B. PROBLEM FORMULATION

In this paper, the goal of the channel estimation is to estimate the MIMO channels $\{\mathbf{h}(n)\}$ as accurately as possible at the transmitter based on the received SNR feedback information $\{\boldsymbol{\gamma}(n)\}$ from the receiver. For the channel estimation, we consider two most common scenarios of fading: (i) the quasi-static block fading and (ii) the time-varying fading. In the following, the optimization problems for both fading scenarios are formulated.

1) QUASI-STATIC BLOCK FADING

We first consider the most common and important scenario of quasi-static block fading, which corresponds to the case when the receiver mobility is negligible (and thus, the channel coherence time is much longer than the duration of a codeword) [1], [2]. In this fading scenario, the MIMO channels remain constant over several consecutive symbol times (i.e., a block) and those vary from one block to another block independently [1], [2]. Therefore, we drop the time index n from the relevant expressions and we focus on the estimation of $\mathbf{h}(n)$ for a particular n. In the quasi-static block fading scenario, we aim to find a channel estimator $\hat{\mathbf{h}} = F(\boldsymbol{\gamma}; \mathbf{S})$, and at the same time, we aim to design the pilot signal $\mathbf{S}$ in the sense of minimizing the MSE of the channel estimation. Mathematically, this problem can be formulated as

$$\begin{aligned} \text{(P1):}\ & \underset{F(\cdot),\mathbf{S}}{min}\ \mathbb{E}\left[\left\|\mathbf{h} - \hat{\mathbf{h}}\right\|^2\right] \\ & s.t.\ \hat{\mathbf{h}} = F\left(\boldsymbol{\gamma}; \mathbf{S}\right). \end{aligned} \tag{4}$$

2) TIME-VARYING FADING

As another very important fading scenario, we also consider the time-varying fading scenario, in which the MIMO channels $\{\mathbf{h}(n)\}$ dependently vary within each block [1], [2]. It also corresponds to the case when the receiver mobility is non-negligible (and thus, the channel coherence time is much shorter than the duration of a codeword) [1], [2]. Let $\hat{\mathbf{h}}(n)$ denote an estimate of $\mathbf{h}(n)$. In the time-varying fading scenario, we utilize the received SNR feedback information from the $(n - T + 1)$th symbol time to the nth symbol time (i.e., $\{\boldsymbol{\gamma}(l)\}_{l=n-T+1}^{n}$) for estimating the MIMO channel at the nth symbol time (i.e., $\mathbf{h}(n)$). Also, considering the dependency of the channel variation, we further utilize the estimates of the MIMO channels from the $(n - T)$th symbol time to the $(n - 1)$th symbol time (i.e., $\{\hat{\mathbf{h}}(l)\}_{l=n-T}^{n-1}$) for the channel estimation. We aim to find a channel estimator $\mathcal{F}(\{\boldsymbol{\gamma}(l)\}_{l=n-T+1}^{n}, \{\hat{\mathbf{h}}(l)\}_{l=n-T}^{n-1})$ in the sense of minimizing the MSE, which can be formulated as[6]

$$\begin{aligned} \text{(P2):}\ & \underset{\mathcal{F}(\cdot)}{min}\ \mathbb{E}\left[\left\|\mathbf{h}(n) - \hat{\mathbf{h}}(n)\right\|^2\right] \\ & s.t.\ \hat{\mathbf{h}}(n) = \mathcal{F}\left(\{\boldsymbol{\gamma}(l)\}_{l=n-T+1}^{n}, \{\hat{\mathbf{h}}(l)\}_{l=n-T}^{n-1}\right) \end{aligned} \tag{5}$$

for $\forall n$.

Note that in both problems (P1) and (P2), the channel estimators $F(\cdot)$ and $\mathcal{F}(\cdot)$ are generally all nonlinear even for a simple channel model, such as the Rayleigh or Rician channel model, because the observations $\{\boldsymbol{\gamma}(n)\}$ of (3) are all nonlinear functions of the variables $\{\mathbf{h}(n)\}$ to be estimated [29]. Thus, it is generally very challenging to analytically derive the channel estimators. Moreover, the problems (P1) and (P2) are even nonconvex. Therefore, it is generally very difficult to tackle both the problems (P1) and (P2) analytically and even numerically.

C. A MACHINE LEARNING APPROACH: GERCHBERG-SAXTON (GS) ALGORITHM

Estimation-theoretically, the problem of estimating the MIMO channels $\{\mathbf{h}(n)\}$ from the received SNR feedback information $\{\boldsymbol{\gamma}(n)\}$ can be considered as the phase retrieval problem [8]. In the literature, there are several methods to tackle this problem [30]. Among the methods, the most well-known and popular method is the GS algorithm in [12]. In fact, this algorithm can be considered as a machine learning (or data clustering) technique due to the algorithmic

[5] It is possible to feedback the received signal to the transmitter (rather than the received SNR). However, feeding back the received signal needs to send M_r complex numbers (i.e., $2M_r$ real numbers), whereas feeding back the received SNR needs to send only M_r real numbers. In this sense, therefore, the latter can be considered to be more efficient than the former because a lower signaling overhead is required. Even when the received signal is fed back to the transmitter, our proposed scheme for the received SNR feedback can be directly extended only with very minor modifications on the constructed networks in Figs. 2 and 4.

[6] In the time-varying fading scenario, we assume that the pilot signals $\{\mathbf{S}(n)\}$ are pre-determined because jointly finding the channel estimator $\mathcal{F}(\cdot)$ and designing the pilot signals $\{\mathbf{S}(n)\}$ is indeed extremely difficult due to the dependencies among the optimizations of $\{\mathbf{S}(n)\}$.

similarity to the generalized expectation-maximization (EM) algorithm [11].

For the two fading scenarios considered in this paper, the GS algorithm can be used to estimate the MIMO channel elements with received SNR feedback. For exposition, let us consider the time-varying fading scenario without loss generality. In this case, given the pilot signals $\{\mathbf{S}(n)\}$, the estimates of the MIMO channels $\{\mathbf{h}(n)\}$ can be computed iteratively as follows [12] (refer to Appendix A for details):

$$\hat{\mathbf{h}}(n) \leftarrow \left(\left(\mathbf{S}^*(n)\mathbf{S}^T(n)\right)^{-1}\mathbf{S}(n) \otimes \mathbf{I}_{M_r}\right) \times \left|\boldsymbol{\gamma}(n)\right| \exp\left(\jmath \angle\left\{\left(\mathbf{S}^T(n) \otimes \mathbf{I}_{M_r}\right)\hat{\mathbf{h}}(n)\right\}\right) \qquad (6)$$

for $\forall n$.

Despite its popularity, there are several major drawbacks in the GS algorithm. First of all, it is strictly suboptimal in the sense of minimizing the channel estimation MSE. In particular, $\{\mathbf{h}(n)\}$ and $\{\mathbf{h}(n)e^{\jmath\omega}\}$ for any ω yield the same observations $\{\boldsymbol{\gamma}(n)\}$ as in (3). Thus, there always exist the global phase ambiguities in the estimates obtained by GS algorithm, which might severely deteriorate the channel estimation performance. Also, it generally requires a considerable number of iterations to compute the channel estimates, which might be practically very difficult to carry out in the stringent real-time applications and/or in the time-varying environments.

In order to break through all the aforementioned challenges and shortcomings intelligently and effectively, in the next two sections, we exploit the deep learning techniques.

III. DEEP LEARNING BASED JOINT CHANNEL ESTIMATION AND PILOT SIGNAL DESIGN IN QUASI-STATIC BLOCK FADING

In this section, we develop a novel learning scheme for the joint channel estimation and pilot signal design in the quasi-static block fading scenario by constructing a deep autoencoder. First, the network architecture of the proposed scheme is presented. Then we present the training procedure of the proposed scheme.

A. NETWORK ARCHITECTURE

The network architecture of the proposed scheme for the joint channel estimation and pilot signal design in the quasi-static block fading scenario (i.e., the optimization of (P1)) is shown in Fig. 2, which is based on the (deep) autoencoder. It is composed of two parts: 1) the encoder and 2) the decoder. We utilize the encoder and the decoder for the two different purposes: to design (or optimize) the pilot signal $\mathbf{S}$ and to represent (or approximate) the channel estimator $F(\cdot)$, respectively. In the following, the encoder and the decoder of the proposed scheme in Fig. 2 are explained in detail.

1) ENCODER

In order to carry out the joint optimization of (P1) using the autoencoder, the most important issue is how to construct the encoder. This, however, is particularly very challenging because a neural network must be constructed such that the pilot signal $\mathbf{S}$ is optimizable. In the existing literature for the deep learning in the area of wireless communications such as in [13]–[15], the naive approach was commonly taken to construct the encoder. Specifically, in [13]–[15], the encoder was constructed using a (deep) feedforward neural network (FNN) with arbitrary activation function, arbitrary weights, and arbitrary biases to learn an arbitrary feature of the input (in our case, the input is the MIMO channel $\mathbf{h}$). However, this naive approach is never effective for the pilot signal design (and thus, for the optimization of (P1)), because the constructed encoder has no capability of optimizing the pilot signal $\mathbf{S}$. To address such critical issue, in this paper, we construct the encoder using a two-layer (i.e., shallow rather than deep) FNN with specific activation function, specific weights, and specific biases (rather than arbitrary ones) to learn a specific feature (rather than arbitrary feature) of the MIMO channel $\mathbf{h}$.

The key idea of our approach is that *it is possible to exactly model the physical mechanisms for the pilot signal transmission and received SNR feedback in (3) via a two-layer FNN.* To show this, let us consider an FNN with two layers: the input and output layers, where the number of the input nodes is given by M_tM_r (i.e., the dimension of $\mathbf{h}$) and the number of the output nodes is given by M_rN_p (i.e., the dimension of $\boldsymbol{\gamma}$). Also, suppose that the operation at each output node is performed as follows. First, the weighted sum of its inputs is computed. Then the result is fed into an activation function. Finally, a bias is added. Note that this operation is slightly different from the operation of the standard artificial neuron in the sense that the bias is added after the activation (rather than before the activation). The output of this two-layer FNN can be mathematically written as

$$\mathbf{y}_m = \phi\left(\mathbf{W}_m\mathbf{x}\right) + \mathbf{b}_m, \quad m = 1, \cdots, N_p. \qquad (7)$$

In (7), $\mathbf{x}$ denotes the input vector of size M_tM_r and $\mathbf{y}_m = [y_{(m-1)M_r+1}, \cdots, y_{mM_r}]^T$ denotes the mth sub-output vector of size M_r, where y_k denotes the output of the kth output node. Also, $\phi(\cdot)$ is the activation function, $\mathbf{W}_m$ is the matrix of weights, and $\mathbf{b}_m$ is the vector of biases. Meanwhile, from (3), the received SNR feedback information at the mth mini-time slot can be written as

$$\boldsymbol{\gamma}_m = g\left|\left(\mathbf{s}_m^T \otimes \mathbf{I}_{M_r}\right)\mathbf{h}\right| + \boldsymbol{\eta}_m, \quad m = 1, \cdots, N_p. \qquad (8)$$

In (8), $\boldsymbol{\gamma}_m = [\gamma_{(m-1)M_r+1}, \cdots, \gamma_{mM_r}]^T$ and $\boldsymbol{\eta}_m = [\eta_{(m-1)M_r+1}, \cdots, \eta_{mM_r}]^T$, where γ_k and η_k are the kth elements of $\boldsymbol{\gamma}$ and $\boldsymbol{\eta}$, respectively. It is very important to note that the expressions of (7) and (8) are exactly the same (i.e., $\mathbf{y}_m = \boldsymbol{\gamma}_m$, $\forall m$) if we set $\mathbf{x} = \mathbf{h}$, $\phi(\cdot) = |\cdot|$, $\mathbf{W}_m = \mathbf{s}_m^T \otimes \mathbf{I}_{M_r}$, $\forall m$, and $\mathbf{b}_m = \boldsymbol{\eta}_m$, $\forall m$. This means that the two-layer FNN in (7) can exactly model the mechanisms for the pilot signal transmission and received SNR feedback in (8) when the specific activation function (i.e., $\phi(\cdot) = g|\cdot|$), specific weights (i.e., $\mathbf{W}_m = \mathbf{s}_m^T \otimes \mathbf{I}_{M_r}, \forall m$), and specific biases (i.e., $\mathbf{b}_m = \boldsymbol{\eta}_m$, $\forall m$) are used.

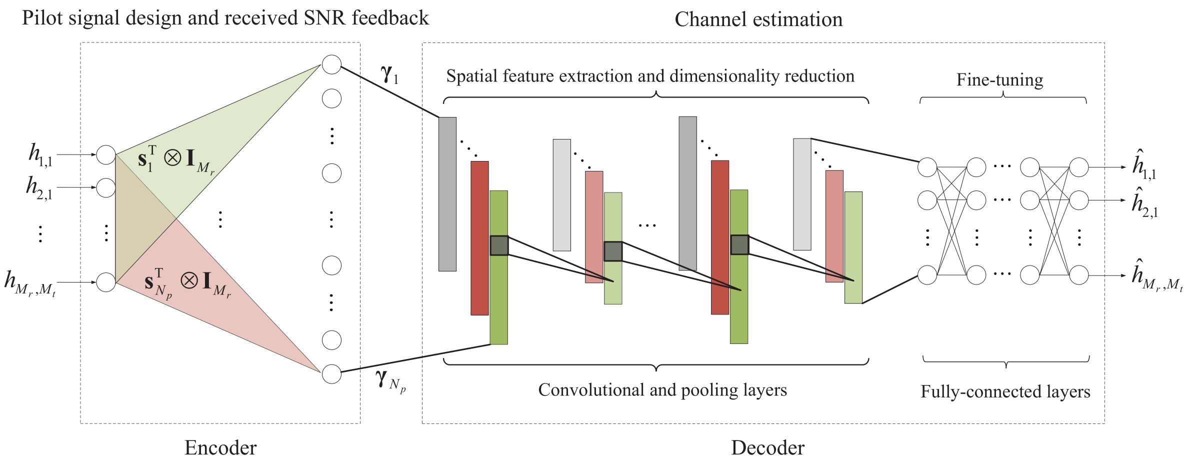

Pilot signal design and received SNR feedback
Channel estimation
$h_{1,1}$
$h_{2,1}$
⋮
h_{M_r,M_t}
$\mathbf{s}_1^{\mathrm{T}} \otimes \mathbf{I}_{M_r}$
$\mathbf{s}_{N_p}^{\mathrm{T}} \otimes \mathbf{I}_{M_r}$
γ_1
γ_{N_p}
Spatial feature extraction and dimensionality reduction
Fine-tuning
$\hat{h}_{1,1}$
$\hat{h}_{2,1}$
$\hat{h}_{M_r,M_t}$
Convolutional and pooling layers
Fully-connected layers
Encoder
Decoder

FIGURE 2. Network architecture of the proposed scheme for the joint channel estimation and pilot signal design in the quasi-static block fading scenario.

Motivated by the discussions above, we construct the encoder using the two-layer FNN to learn the received SNR feedback information $\{\boldsymbol{\gamma}_m\}$ (i.e., the specific features). To this end, we propose to use the Kronecker products between the pilot symbol vectors $\{\mathbf{s}_m\}$ and the $M_r \times M_r$ identity matrix $\mathbf{I}_{M_r}$ as the weights of the encoder (i.e., $\mathbf{W}_m = \mathbf{s}_m^T \otimes \mathbf{I}_{M_r}$, $\forall m$) and to use the noise-plus-error terms $\{\eta_k\}$ as the biases of the encoder (i.e., $\mathbf{b}_m = \boldsymbol{\eta}_m$, $\forall m$). Also, we propose to use a scaled magnitude function as the activation function at the output nodes (i.e., $\phi(\cdot) = g|\cdot|$). It is important to note that the proposed encoder enables the optimization of the pilot signal $\mathbf{S}$ (and thus, the optimization of (P1)) possible, because its weights (i.e., pilot signal $\mathbf{S}$) can be optimized through the training, which is in sharp contrast to the naive approach of [13]–[15]. For the purpose of learning, the proposed encoder takes the MIMO channel coefficients $\{h_{k,j}\}$ as its input.

2) DECODER

In (P1), analytically finding the channel estimator $F(\cdot)$ (i.e., a function) is very difficult because it is nonlinear and non-convex. To address this issue, we construct the decoder using a deep neural network to represent (or approximate) the channel estimator $F(\cdot)$. This is motivated by the universal function approximation theorem: a (deep) neural network is a universal function approximator, which represent any function arbitrarily well within some accuracy [32]. One might take the naive approach of using only the FNN to construct the decoder, such as in [21] and [29]. However, such naive approach is never effective because all useful features (e.g., the spatial features or correlations) of the MIMO channel coefficients cannot be learned satisfactorily solely by the FNN [9], [10]. To address this issue, in the proposed scheme in Fig. 2, we construct the decoder using a CNN. The key idea of using the CNN is to effectively extract the spatial and the other useful features of the MIMO channel coefficients and to fully utilize those for learning by treating the MIMO channel matrix as an image.

The proposed decoder is composed of two parts: 1) the convolutional and pooling layers, 2) followed by the fully-connected layers. In the former, a convolutional layer and a pooling layer are sequentially stacked to extract the spatial and the other useful features of the MIMO channel coefficients and to reduce the dimensions of the extracted features, respectively. In the fully-connected layers, we stack several FNNs to fine-tune the obtained features for the accurate channel estimation. To learn (or estimate) the MIMO channel coefficients $\{h_{k,j}\}$ from the received SNR feedback information, the proposed decoder takes the values of $\{\boldsymbol{\gamma}_m\}$ (which is learned by the encoder) as the input. Also, it outputs the estimates $\{\hat{h}_{k,j}\}$ of the MIMO channel coefficients.

In the convolutional layers, we employ a number of different spatial filters with a small number of learnable parameters. This enables to effectively and efficiently learn the spatial features as well as the other useful high-level features. In each convolutional layer, the input is first convolved with the filters in a sliding manner (in the spatial domain), and then, the results are fed into the activation functions. The output of a filter is given by

$$y(n) = \varsigma\left(\sum_{l=0}^{N_f-1} w_l x(n-l) + b\right) \tag{9}$$

for $n = 1, \cdots, \frac{N_i - N_f + 2\,N_0}{N_s} + 1$, where $\{x(n)\}$ denotes the input of size N_i, $\varsigma(\cdot)$ is the activation function, and N_s denotes the stride rate. Also, $\{w_l\}$ and b are the weights and bias of the filter, respectively. The number of weights (i.e., size of the filter) and the number of zeros padded to the input are denoted by N_f and N_0, respectively.

In the pooling layers, the features extracted in the convolutional layers are down-sampled in a sliding manner, and thus, the dimensionality is reduced. Let N_i' denote the size of the input, N_f' denote the size of a pooling filter, N_s' denote the stride rate, and N_0' denote the number of zeros padded to the input. Then the output of the pooling filter is given by

$$d(n) = \psi\big(y(n), \cdots, y(n+N_f')\big) \tag{10}$$

for $n = 1, \cdots, \frac{N_i'-N_f'+2\,N_0'}{N_s'}+1$, where $\psi(\cdot)$ denotes a pooling operation such as the max pooling, the average pooling, or the $L2$-norm pooling.

In the fully-connected layers, the operation at each node is performed as follows. First, the weighted sum of its inputs is computed and a bias is added. Then the result is fed into an activation function. This operation is the same as that of the standard artificial neuron. Let $\mathbf{x}$ denote the input to a layer. Then the output of the next layer is given by

$$\mathbf{y} = \varphi\left(\mathbf{W}\mathbf{x}\right) + \mathbf{b} \tag{11}$$

where $\varphi(\cdot)$ is the activation function. Also, $\mathbf{W}$ and $\mathbf{b}$ denote the weight matrix and the bias vector, respectively.

B. TRAINING PROCEDURE

In order to carry out the optimization of (P1), the proposed scheme in Fig. 2 has to be trained such that the MIMO channel is estimated as accurately as possible with the minimal amount of estimation errors. To achieve this goal, how to select the loss function is particularly important, because unless the loss function is properly selected, the channel estimation performance might not be satisfactory. In order to address this issue, we directly use the cost function of the problem (P1) (i.e., the MSE of the channel estimation) as the loss function for the training.

Let $G_{enc}(\cdot;\mathbf{S})$ and $G_{dec}(\cdot;\theta)$ denote the mathematical operations of the encoder and the decoder of the proposed scheme in Fig. 2, respectively, where the elements of $\mathbf{S}$ are the weights of the encoder (i.e., pilot signal) and θ denotes the set of the parameters (i.e., the weights and biases) of the decoder. Then the overall operation of the proposed scheme in Fig. 2 can be written as $\hat{\mathbf{h}} = G_{dec}(G_{enc}(\mathbf{h};\mathbf{S});\theta)$. From this, the cost function of (P1) can be written as

$$\begin{aligned} C\big(\mathbf{S},\theta\big) &= \mathbb{E}\Big[\big\|\mathbf{h}-\hat{\mathbf{h}}\big\|^2\Big] \\ &= \mathbb{E}\Big[\big\|\mathbf{h}-G_{dec}\big(G_{enc}\big(\mathbf{h};\mathbf{S}\big);\theta\big)\big\|^2\Big]. \end{aligned} \tag{12}$$

Note that the above cost function involves the statistical expectation $\mathbb{E}[\cdot]$, of which analytical expression is very difficult to obtain in closed form. To address this issue, we take the Monte-Carlo integration approach to compute the cost function $C(\cdot)$ of (12). Specifically, replacing the expectation by the sample mean, the cost function can be approximately written as

$$C\big(\mathbf{S},\theta\big) \simeq L\big(\mathbf{S},\theta\big) = \frac{1}{|T|}\sum_{\mathbf{h}\in T}\big\|\mathbf{h}-G_{dec}\big(G_{enc}\big(\mathbf{h};\mathbf{S}\big);\theta\big)\big\|^2 \tag{13}$$

where T denotes the set of training samples. Note that due to the law of large numbers, as the number of training samples becomes larger (i.e., $|T|\to\infty$), $L(\cdot)$ of (13) converges to the actual cost function $C(\cdot)$ of (12) (i.e., the MSE of the channel estimation) as follows: $L(\mathbf{S},\theta) \to C(\mathbf{S},\theta)$. Therefore, we choose $L\big(\mathbf{S},\theta\big)$ as the loss function for the training of the proposed scheme in Fig. 2.

In order to train the proposed scheme in Fig. 2, the parameters $\mathbf{S}$ and θ need to be optimized to minimize the loss function $L(\cdot)$ of (13). To this end, we take two different gradient descent approaches. First, since there is no constraint for the parameters θ of the decoder, we update the value of θ according to the naive (or standard) gradient method as follows:

$$\theta \leftarrow \theta - \alpha\nabla_\theta L\big(\mathbf{S},\theta\big) \tag{14}$$

where $\alpha > 0$ is the learning rate, which determines the step size of the update. On the other hand, the weights of the encoder (i.e., pilot signal $\mathbf{S}$) are constrained by the transmit power threshold, i.e., $\|\mathbf{S}\|_F^2 = \mathrm{Tr}(\mathbf{S}\mathbf{S}^H) \le P$. Unfortunately, the naive gradient decent method of (14) cannot be used to optimize the weights of the encoder because the updated weights do not generally satisfy the power constraint. To address this issue and to satisfy the power constraint, we update the weights of the encoder using the projected gradient descent method as follows [21], [29]:

$$\mathbf{S} \leftarrow \Pi_{\mathcal{P}}\Big[\mathbf{S}-\alpha\nabla_{\mathbf{S}}L\big(\mathbf{S},\theta\big)\Big] \tag{15}$$

where $\mathcal{P} = \big\{\mathbf{X}\in\mathbb{C}^{M_t\times N_p} : \mathrm{Tr}(\mathbf{X}\mathbf{X}^H) \le P\big\}$ denotes the set of matrices satisfying the power constraint. Also, $\Pi_{\mathcal{P}}[\mathbf{A}] = \arg\min_{\mathbf{X}\in\mathcal{P}}\|\mathbf{X}-\mathbf{A}\|_F^2$ denotes the projection of an $M_t\times N_p$ matrix $\mathbf{A}$ onto the set $\mathcal{P}$. To perform the projected gradient descent step of (15), a very important (but, nontrivial) question is: what is the analytical expression of the projection operator? This question is answered in the following.

Lemma 1: The projection of $\mathbf{A}$ onto the set $\mathcal{P}$ is given by[7]

$$\Pi_{\mathcal{P}}[\mathbf{A}] = \begin{cases} \mathbf{A}, & \text{if } \mathrm{Tr}(\mathbf{A}\mathbf{A}^H) \le P \\ \sqrt{\frac{P}{\mathrm{Tr}(\mathbf{A}\mathbf{A}^H)}}\mathbf{A}, & \text{if } \mathrm{Tr}(\mathbf{A}\mathbf{A}^H) > P \end{cases}. \tag{16}$$

Proof: First, when $\mathrm{Tr}(\mathbf{A}\mathbf{A}^H) \le P$, it directly follows that $\Pi_{\mathcal{P}}[\mathbf{A}] = \mathbf{A}$ since $\mathbf{A} = \arg\min_{\mathbf{X}\in\mathcal{P}}\|\mathbf{X}-\mathbf{A}\|_F^2$. On the other hand, when $\mathrm{Tr}(\mathbf{A}\mathbf{A}^H) > P$, the Lagrangian function for the problem $\min_{\mathbf{X}\in\mathcal{P}}\|\mathbf{X}-\mathbf{A}\|_F^2$ is given by $\|\mathbf{X}-\mathbf{A}\|_F^2 + \lambda(\|\mathbf{X}\|_F^2 - P)$ [33], where λ is the Lagrange multiplier. Differentiating it with respect to $\mathbf{X}$ and setting the derivative to zero, we have $\frac{\mathbf{A}}{1+\lambda} = \arg\min_{\mathbf{X}\in\mathcal{P}}\|\mathbf{X}-\mathbf{A}\|_F^2$. From this and $\mathrm{Tr}(\mathbf{A}\mathbf{A}^H) = P$,

[7] When the PAPR constraint is considered (i.e., $\max_{\forall i,j}\big|[\mathbf{S}]_{i,j}\big|^2/P \le \beta$ where β denotes a threshold for the PAPR), the projection operator is given by $\big[\Pi_{\mathcal{P}}[\mathbf{A}]\big]_{i,j} = [\mathbf{A}]_{i,j}$ if $[\mathbf{A}]_{i,j} \le \sqrt{\beta P}$ and $\big[\Pi_{\mathcal{P}}[\mathbf{A}]\big]_{i,j} = \sqrt{\beta P}$ otherwise, where $[\mathbf{A}]_{i,j}$ denotes the (i,j)th element of the matrix $\mathbf{A}$. By properly choosing the projection operator, it is generally possible to satisfy any (power) constraint, e.g., the sum power constraint on the pilot signal of each transmit antenna.

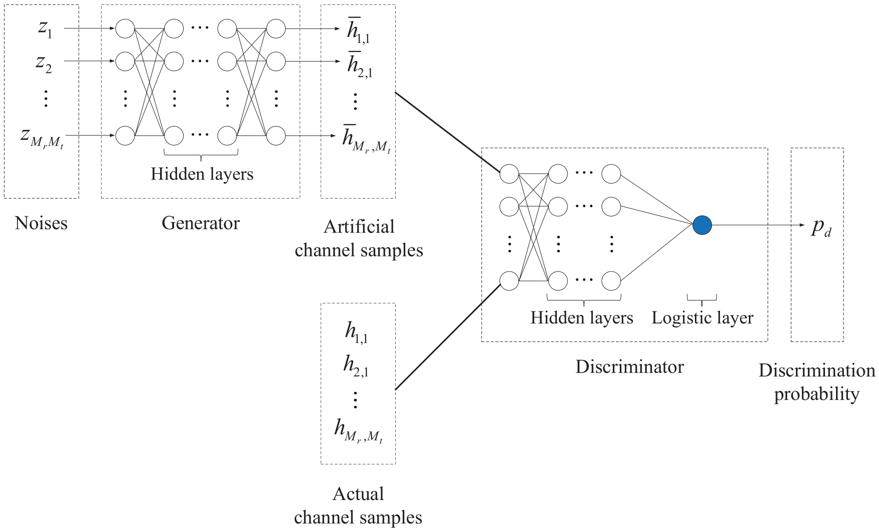


FIGURE 3. **Structure of the constructed GAN to generate the artificial channel samples in the quasi-static block fading scenario.**

it follows that $\lambda = \sqrt{\frac{\text{Tr}(\mathbf{A}\mathbf{A}^H)}{P}} - 1$, and thus, we have $\Pi_{\mathcal{P}}[\mathbf{A}] = \sqrt{\frac{P}{\text{Tr}(\mathbf{A}\mathbf{A}^H)}}\mathbf{A}$. ■

From the result of Lemma 1, the projected gradient descent method of (15) can be interpreted as follows. First, the naive gradient descent step of (14) is performed on the pilot signal $\mathbf{S}$. If the updated pilot signal does not satisfy the power constraint, it is scaled such that the power constraint is satisfied with equality; otherwise, the projection operation is not performed. In the latter case, therefore, the projected gradient descent method reduces to the naive gradient descent method.

Note that the above training procedure can be done *offline* only at the transmitter. Thus, the complicated online training procedure (which requires a large number of pilot/feedback transmissions) does not need to be carried out in practice. Specifically, using the training samples of the MIMO channel coefficients $\{h_{k,j}\}$ acquired by the field measurement or generated according to the fading distribution, the gradient steps of (14) and (15) (and thus, the training) can be performed at the transmitter alone prior to the actual system operation or deployment.

Once the proposed scheme in Fig. 2 is trained, the optimized weights of the encoder can be used as the pilot signal at the actual transmission stage. It is worth noting that the pilot signal designed by the proposed encoder is generally non-orthogonal since the projected gradient descent method of (15) does not generally guarantee the orthogonality. This is in sharp contrast to the existing result such as reported in [31]. Also, using the proposed decoder, the MIMO channel coefficients can be estimated very efficiently without any iteration, which is in sharp contrast to the GS algorithm of (6).

However, the offline training procedure above generally requires many actual channel samples (enough to guarantee the convergence of $L(\mathbf{S}, \theta) \rightarrow C(\mathbf{S}, \theta)$ and satisfactory learning) or to exactly know the fading distribution, which might be very difficult to carry out in practice because it takes very long time. To address this critical issue, in the next subsection, we propose a novel and effective way to train the proposed scheme with a deep generative model as follows: first, given a limited number of actual channel samples, additional artificial channel samples are generated using the GAN, and then, the proposed scheme in Fig. 2 is trained using the artificially generated channel samples.

C. TRAINING WITH GENERATIVE ADVERSARIAL NETWORK (GAN)

The GAN is an advanced and novel generative model, which is able to produce the artificial data (e.g., artificial images for human faces) that looks very similar to the actual data by effectively learning the probability distribution of the actual data [34]. The GAN is composed of two parts: 1) the generator that synthesize the artificial data and 2) the discriminator that determines whether the data is drawn from the actual distribution or not. In the GAN, the generator and the discriminator are trained in an adversarial manner. Through the adversarial training, the generator learns how to generate the data that looks like the actual one in order to fake or pool the discriminator, whereas the discriminator learns how to distinguish between the actual and artificial data as correctly as possible.

In this paper, to generate the artificial channel samples in the quasi-static block fading scenario, we construct a GAN, of which structure is shown in Fig 3. In this scheme, we construct both the generator and the discriminator using the deep FNN composed of the input layer, multiple hidden layers, and the output layer, where the operation performed at each node in the hidden and output layers is the same as that of the standard artificial neuron as in (11). The generator takes

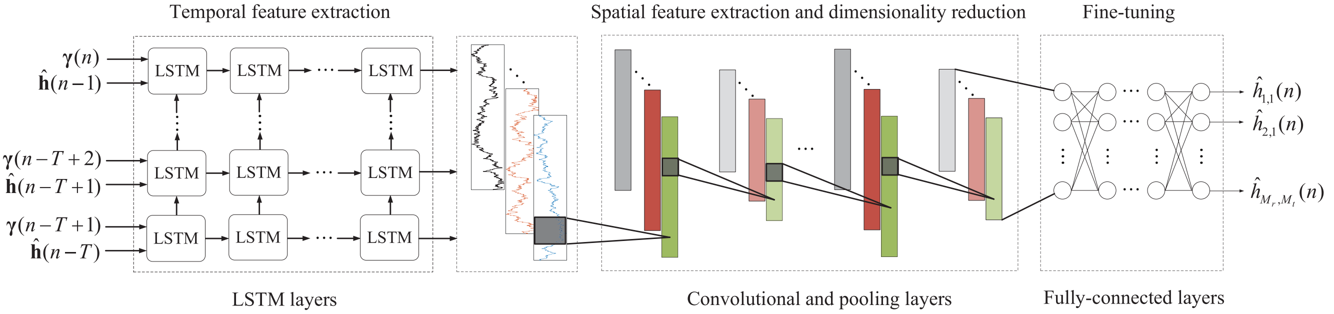


FIGURE 4. Network architecture of the proposed scheme for the channel estimation in the time-varying fading scenario.

random noises $\{z_{k,j}\}$ as its input, which are drawn from a priori known distribution $p_z(\mathbf{z})$, where $\mathbf{z} = [z_1, z_2, \cdots, z_{M_tM_r}]^T$. It then generates the artificial MIMO channel coefficients (which are denoted by $\{\bar{h}_{k,j}\}$) as its output. On the other hand, the discriminator takes both the actual and artificial channel samples as its input, and then, it outputs a scalar, namely, the discrimination probability p_d, which denotes the possibility of the input coming from the actual channel samples. To predict the value of $p_d \in [0, 1]$, the output layer of the discriminator is set to be the logistic layer, i.e., the activation function of each output node is set to be the logistic (or sigmoid) function $\sigma(x) = \frac{1}{1+e^{-x}}$.

In the constructed GAN in Fig. 3, the generator and the discriminator are trained according to the following two-player mini-max game [34]:

$$\min_{\theta_G} \max_{\theta_D} V\left(\theta_G, \theta_D\right) \tag{17}$$

where θ_G and θ_D denote the sets of the parameters of the generator and the discriminator, respectively. The loss function $V(\cdot)$ is given by

$$\begin{aligned} V\left(\theta_G, \theta_D\right) &= \mathbb{E}_{\mathbf{h}\sim p_h(\mathbf{h})}\Big[\log D(\mathbf{h}; \theta_D)\Big] \\ &\quad + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}\Big[\log\left(1 - D(G(\mathbf{z}; \theta_G); \theta_D)\right)\Big] \end{aligned} \tag{18}$$

where $p_h(\mathbf{h})$ denotes the distribution of the actual channel samples. Also, $G(\cdot; \theta_G)$ and $D(\cdot; \theta_D)$ denote the overall mathematical functions of the generator and the discriminator, respectively. The generator and the discriminator of the GAN in Fig. 3 can be trained by iteratively optimizing the parameters θ_G and θ_D according to the gradient descent and ascent methods, respectively, as follows:

$$\theta_G \leftarrow \theta_G - \alpha \nabla_{\theta_G} V\left(\theta_G, \theta_D\right), \tag{19}$$

$$\theta_D \leftarrow \theta_D + \alpha \nabla_{\theta_D} V\left(\theta_G, \theta_D\right). \tag{20}$$

Given the limited number of actual channel samples, the proposed scheme in Fig. 2 can be trained with the aid of the GAN in Fig. 3 as follows:

- *(Step 1)* Using the actual channel samples, train the GAN in Fig. 3 by iteratively performing the gradient descent and ascent steps of (19) and (20), respectively.
- *(Step 2)* Using the trained generator, generate the artificial channel samples.
- *(Step 3)* Using the actual and artificial channel samples altogether, train the proposed scheme in Fig. 2 according to the training procedure presented in Section III-B.

IV. DEEP LEARNING BASED CHANNEL ESTIMATION IN TIME-VARYING FADING

A. NETWORK ARCHITECTURE

The network architecture of the proposed scheme for the channel estimation in the time-varying fading scenario is shown in Fig. 4. It is composed of three parts: 1) the LSTM layers to first extract the temporal features of the time-varying MIMO channels by learning their time dependencies; 2) the convolutional and pooling layers to extract the spatial (and further useful high-level) features of the MIMO channels, and 3) the fully-connected layers to fine-tune the extracted features for accurately estimating the channel coefficients. In the following, each part is explained in detail.

1) LSTM LAYERS

In the initial stage of the proposed scheme in Fig. 4, we construct an RNN architecture by stacking several LSTMs vertically (for the temporal processing) and horizontally (for the deep architecture) to form the LSTM layers. To carry out the time-varying channel estimation (i.e., the optimization of (P2)), we construct the LSTM layers such that the first vertically stacked LSTMs to accommodate the input sequence of length T composed of the previous channel estimates $\{\hat{\mathbf{h}}(l)\}_{l=n-T}^{n-1}$ and the received SNR feedback information $\{\boldsymbol{\gamma}(l)\}_{l=n-T+1}^{n}$.

2) CONVOLUTIONAL AND POOLING LAYERS

In the proposed scheme in Fig. 4, we stack the convolutional and pooling layers after the LSTM layers, in which a convolutional layer is stacked, followed by a pooling layer. We construct the first convolutional layer to accommodate the output sequence of the last vertically stacked LSTMs. Note that the output sequence of the last vertically stacked LSTMs is a one-dimensional (1D) sequence. Thus, we employ 1D convolution and pooling filters in the convolutional and pooling layers.

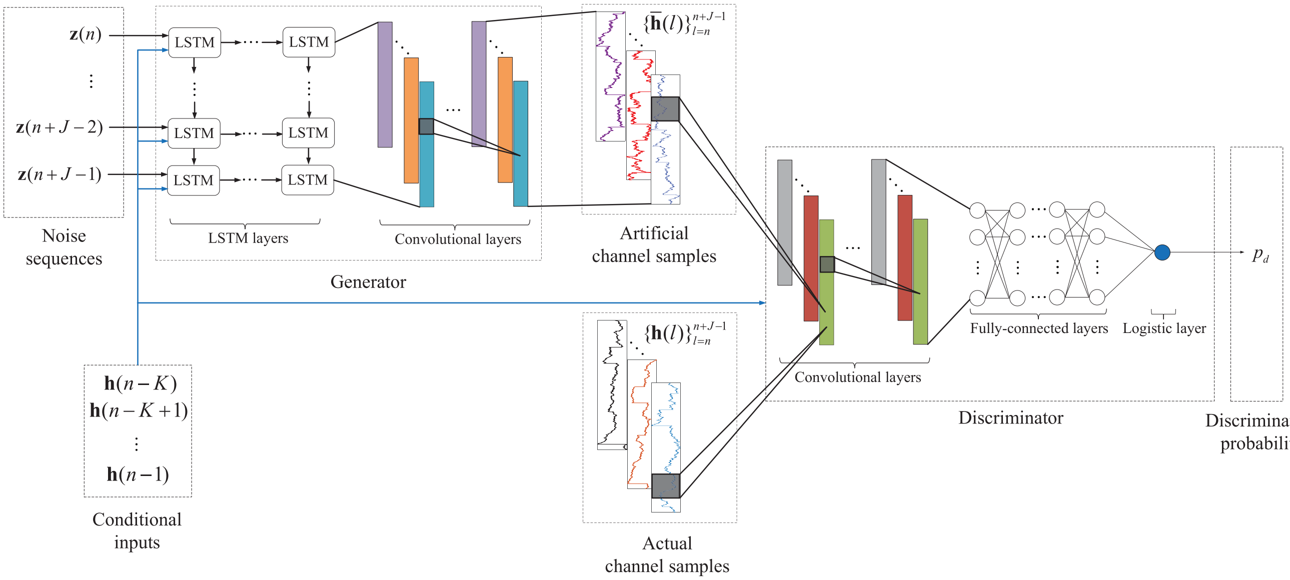


FIGURE 5. **Structure of the constructed CGAN to generate the artificial channel samples in the time-varying fading scenario.**

3) FULLY-CONNECTED LAYERS

To fine-tune the extracted features, subsequent to the convolutional and pooling layers, we stack several FNNs to form the fully-connected layers.

B. TRAINING PROCEDURE

In order to learn (or estimate) the time-varying MIMO channels $\{\mathbf{h}(n)\}$ as accurately as possible and to perform the optimization of (P2), we select the cost function of (P2) (i.e., MSE) as the loss function for the training of the proposed scheme in Fig. 4 as follows:

$$\mathcal{L}(\vartheta)=\sum_{\mathbf{h}(n)\in\mathcal{T}}\frac{\left\|\mathbf{h}(n)-\mathcal{G}\left(\{\boldsymbol{\gamma}(l)\}_{l=n-T+1}^{n},\{\hat{\mathbf{h}}(l)\}_{l=n-T}^{n-1};\vartheta\right)\right\|^2}{|\mathcal{T}|} \tag{21}$$

where $\mathcal{G}(\cdot;\vartheta)$ denote the overall mathematical function of the proposed scheme in Fig. 4. Also, ϑ denotes the set of all the parameters in the LSTM layers, the convolutional/pooling layers, and the fully-connected layers, which can be trained by minimizing the loss function $\mathcal{L}(\cdot)$ of (21) according to the gradient descent method as

$$\vartheta\leftarrow\vartheta-\alpha\nabla_{\vartheta}\mathcal{L}(\vartheta). \tag{22}$$

In the next subsection, we develop a novel scheme to artificially generate the time-varying MIMO channels based on the CGAN and we propose to train the proposed scheme in Fig. 4 using the artificially generated channel samples.

C. TRAINING WITH CONDITIONAL GENERATIVE ADVERSARIAL NETWORK (CGAN)

In this paper, to generate the artificial channel samples in the time-varying fading scenario, we construct a CGAN, of which structure is shown in Fig. 5. In the CGAN in Fig. 5, the generator is composed of two parts: 1) the LSTM layers, followed by 2) the convolutional layers. Also, the discriminator is composed of three parts: 1) the convolutional layers, followed by 2) the fully connected layers, and 3) the logistic layer.

In the constructed CGAN in Fig. 5, the conditional inputs that are fed into both the generator and the discriminator are selected as $\{\mathbf{h}(l)\}_{l=n-K}^{n-1}$. In addition to these conditional inputs, the generator is constructed to additionally accommodate a noise sequence $\{\mathbf{z}(l)\}_{l=n}^{n+J-1}$, which is drawn from a priori known distribution $p_z(\{\mathbf{z}(n)\})$. Also, it is constructed to output (or generate) the artificial samples of the time-varying MIMO channels (which are denoted by $\{\bar{\mathbf{h}}(l)\}_{l=n}^{n+J-1}$). On the other hand, the discriminator is constructed to take both the actual and artificial channel samples as its input and to output the discrimination probability $p_d\in[0,1]$. In the CGAN in Fig. 5, the generator and the discriminator can be trained according to the following two-player mini-max game [36]:

$$\min_{\vartheta_G}\max_{\vartheta_D}\mathcal{V}\left(\vartheta_G,\vartheta_D\right) \tag{23}$$

where ϑ_G and ϑ_D denote the sets of the parameters of the generator and the discriminator, respectively. Also, the loss function $\mathcal{V}(\cdot)$ is given by (24) (as shown at the bottom of the next page), where $p_h(\{\mathbf{h}(n)\})$ denotes the distribution of the actual channel samples. Also, $\mathcal{G}(\cdot;\vartheta_G)$ and $\mathcal{D}(\cdot;\vartheta_D)$ denote the overall mathematical describtions of the generator and the discriminator, respectively. The parameters ϑ_G and ϑ_D can be optimized according to the gradient descent and ascent methods, respectively, similarly as in (19) and (20).

Given the limited number of actual channel samples, the proposed scheme in Fig. 4 can be trained with the constructed CGAN in Fig. 5 in a similar manner as discussed in Section III-C, and hence, the details are omitted.

V. NUMERICAL RESULTS

In this section, we evaluate the performance of the proposed schemes and compare it to that of the other schemes through the numerical results.[8] We first describe the parameter setting[9] and the simulation setup. Then the numerical results for both fading scenarios are presented.

A. PARAMETER SETTING

In the proposed scheme in Fig. 2, we employ 3 convolutional layers, 3 pooling layers, and one fully-connected layer. The fully-connected layer involves 4 nodes, where the activation function at each node is set to the linear function (i.e., $\varphi(x) = x$). In each convolutional layer, the input is zero padded such that the output size is the same as the input size after the filtering. Also, in the kth convolutional layer, we employ $2\,kM_r$ filters, $k = 1, 2, 3$, where the activation function of each filter is set to the exponential linear unit (ELU) (i.e., $\varphi(x) = e^x - 1$ for $x \leq 0$ and $\varphi(x) = x$ for $x > 0$). The size and the stride rate of each filter are set to 2 and 1, respectively. In the pooling layers, we use the max pooling (i.e., $\psi(\cdot) = \max(\cdot)$), where the size and the stride rate of the pooling filter are both set to 2.

In the GAN in Fig. 3, we employ 3 hidden layers at both the generator and the discriminator, where each hidden layer involves 10 nodes and the activation function at each hidden node is set to the leaky rectified linear unit (ReLU) (i.e., $\varphi(x) = 0.2\,x$ for $x \leq 0$ and $\varphi(x) = x$ for $x > 0$) as the activation function. The activation function at each output node of the generator is set to the linear function.

In the proposed scheme in Fig. 4, we set $T = 16$. Also, we employ one LSTM layer, 4 convolutional layers, 4 pooling layers, and one fully-connected layer. The LSTM layer involves 20 nodes. The fully-connected layer involves 4 nodes, where the activation function at each node is set to the linear function. In each convolutional layer, the input is zero padded such that the output size is the same as the input size after the filtering. Also, in the kth convolutional layer, we employ $2\,k(M_t + N_p)M_r$ filters, $k = 1, \cdots, 4$, where the activation function, size, and stride rate of each filter are set to the ELU, 2, and 1, respectively. In the pooling layers, we use the max pooling, where the size and stride rate of the pooling filter are both set to 2.

In the CGAN in Fig. 5, we set $K = J = 16$. Also, we employ 2 LSTM layers and one convolutional layer at the generator, where each LSTM layer involves 20 nodes and the convolutional layer involves 4 filters. We employ 4 convolutional layers and one fully-connected layer at the discriminator, where the kth convolutional layer involves $2\,kM_tM_r$ filters, $k = 1, \cdots, 4$. In both the generator and discriminator, the activation function at each node in the convolutional layers is set to the leaky ReLU and the activation function at each node in the fully-connected layer is set to the linear function.

In all the schemes in Figs. 2–5, the learning rate is set to $\alpha = 0.001$. Also, in the proposed schemes in Figs. 2 and 4, we apply the dropout with the keep probability of 0.5 to prevent overfitting. In the constructed GAN in Fig. 3, the elements of the noises $\{z_{k,j}\}$ are independently drawn from the Gaussian distribution with zero mean and unit variance. In the constructed CGAN in Fig. 5, the elements of the noise sequence $\{\mathbf{z}(n)\}$ are independently drawn from the uniform distribution over $[-1, 1]$.

B. SIMULATION SETUP

The key parameters used in the simulations are summarized in Table 1.[10] In the simulations, unless stated otherwise, we consider a MIMO system with $M_t = M_r = 2$ and $N_p = 8$. Also, we set $\xi^2(n) = 1, \forall n$. The elements of the feedback noises $\{\boldsymbol{\eta}(n)\}$ are independently drawn from the Gaussian distribution with zero mean and variance δ^2, and thus, the transmit SNR is defined as $\frac{P}{\delta^2}$. We fix the value of P while varying the value of δ^2 such that the effects of the feedback noise can be easily observed from those of the transmit SNR. In the proposed scheme in Fig. 4 and in the GS algorithm of (6), the orthogonal pilot signal

$$\mathcal{V}\big(\vartheta_G, \vartheta_D\big) = \mathbb{E}_{\{\mathbf{h}(l)\}\sim p_h(\{\mathbf{h}(l)\})}\Big[\log \mathcal{D}\big(\{\mathbf{h}(l)\}_{l=n}^{n+J-1} | \{\mathbf{h}(l)\}_{l=n-K}^{n-1}; \theta_D\big)\Big] \\ + \mathbb{E}_{\{\mathbf{z}(l)\}\sim p_z(\{\mathbf{z}(l)\}), \{\mathbf{h}(l)\}\sim p_h(\{\mathbf{h}(l)\})}\Big[\log\big(1 - \mathcal{D}\big(\mathcal{G}\big(\{\mathbf{z}(l)\}_{l=n}^{n+J-1} | \{\mathbf{h}(l)\}_{l=n-K}^{n-1}; \vartheta_G\big) | \{\mathbf{h}(l)\}_{l=n-K}^{n-1}; \vartheta_D\big)\big)\Big] \tag{24}$$

[8] In this paper, we do not adopt the Bayesian, such as the minimum MSE (MMSE), the maximum a posterior (MAP), or the maximum likelihood (ML), channel estimator as the performance benchmark (or reference) for the proposed scheme because those require very high complexity and take extremely long time to compute the channel estimate (particularly, for the time-varying fading scenario). Due to the high computational burden, it is indeed very hard (in fact, appears to be infeasible particularly in the time-varying fading scenario) to implement (or realize) the MMSE, MAP, or ML estimator even numerically.

[9] The hyper-parameters for the proposed schemes are set as in Section V-A because in these settings, the proposed schemes perform the best. We confirmed this exhaustively. Thus, in the settings presented in Section V-A, the hyper-parameters of the proposed schemes can be considered to be optimized.

[10] The values of M_t and N_p used in the simulations are widely adopted in many standards. Specifically, in the existing wireless standards such as IEEE 802.11 family or 4G LTE, a set of small number of transmit antennas such as $M_t \in \{2, 4, 8\}$ has been adopted. On the other hand, in the emerging wireless standards such as 5G or beyond 5G, another set of large number of transmit antennas such as $M_t \in \{16, 32, 64, 128\}$ will be adopted. In these standards, it is also common to use the pilot signal with an appropriate length, e.g., $N_p = 8$. Furthermore, we would like to note that in practical scenarios of the downlink MIMO, the receiver is typically a mobile device, which is equipped with a (very) small number of antennas due to its limited hardware size. Considering this, in the simulations, we set $M_r = 2$.

TABLE 1. Key simulation parameters.

	Figs. 6–9, 11, and 12	Fig. 10	Figs. 13–17	Fig. 18
Type of fading	Quasi-static block fading	Quasi-static block fading	Time-varying fading	Time-varying fading
Number of transmit antennas, M_t	2	$\{4, 8, 16, 32, 64, 128\}$	2	$\{4, 8, 16, 32, 64, 128\}$
Number of receive antennas, M_r	2	2	2	2
Number of pilot signals (or pilot length), N_p	8	8	8	8
Channel model	Rician channel model	Rician channel model	Gauss-Markov channel model	Gauss-Markov channel model

(i.e., $\mathbf{S}(n)\mathbf{S}^H(n) = \frac{P}{M_t}\mathbf{I}_{M_t}$) is used for learning. We use the channel samples drawn from a particular fading distribution as the actual channel samples.[11]

In the quasi-static block fading scenario, we draw the actual channel samples according to the Rician fading model as follows:[12] $\mathbf{h} = \sqrt{\frac{\kappa}{\kappa+1}}\mathbf{h}_{LOS} + \sqrt{\frac{1}{\kappa+1}}\mathbf{h}_{scatter}$ [2], where κ is the Rician factor, $\mathbf{h}_{LOS}$ is the vector of line-of-sight (LOS) components, and $\mathbf{h}_{scatter}$ is the vector of scattering components. We set $\kappa = 2$. Also, we set $\mathbf{h}_{LOS} = \bar{h}\mathbf{1}$, where $\mathbf{1}$ denotes the vector whose elements are all equal to one. The values of $\bar{h}$ are drawn from the Nakagami distribution with the shape parameter of 0.5 and the scale parameter of 5. The elements of $\mathbf{h}_{scatter}$ are drawn from the Gaussian distribution with zero mean and covariance matrix Λ. Adopting the Kronecker spatial correlation model, we set $\Lambda = \Lambda_t \otimes \Lambda_r$ [1], where Λ_t and Λ_r are the spatial correlation matrices at the transmitter and the receiver, respectively. The (k, j)th elements of Λ_t and Λ_r are set to $\rho_t^{|k-j|}$ and $\rho_r^{|k-j|}$, respectively, where $\rho_t \in [0, 1]$ and $\rho_r \in [0, 1]$ are the spatial correlation coefficients at the transmitter and the receiver, respectively. We set $\rho_t = 0.6$ and $\rho_r = 0.3$, which mean the strong and weak spatial correlations at the transmitter and the receiver, respectively.

In the time-varying fading scenario, we draw the actual channel samples according to the Gauss-Markov channel model as follows:[13] $\mathbf{h}(n) = \boldsymbol{\mu}$ if $n\% n_\mu = 0$; and $\mathbf{h}(n) = a\mathbf{h}(n-1) + \boldsymbol{\nu}(n)$ otherwise, where $a \in [0, 1]$ is the temporal fading correlation coefficient and $\boldsymbol{\nu}(n)$ is the vector of process noises. We set $n_\mu = 256$. Also, we set $a = 0.9978$, which corresponds to the Jake's model with the carrier frequency of 2.5 GHz and the receiver mobility of 64 km/h [2]. The elements of $\{\boldsymbol{\nu}(n)\}$ are independently drawn from the Gaussian distribution with zero mean and variance $(1 - a^2)$. The values of $\boldsymbol{\mu}$ are drawn according to the Rician fading model similarly as in the quasi-static block fading scenario.

We train the proposed scheme in Fig. 2 (resp. Fig. 4) in two different ways. First, we train the proposed scheme in Fig. 2 (resp. Fig. 4) using 10^5 (resp. 512×10^2) actual channel samples directly drawn from the above procedure. This scheme is considered as the performance benchmark for the case of ideal training with the enough number of actual channel samples. Second, we train the proposed scheme in Fig. 2 (resp. Fig. 4) using both 5×10^4 (resp. 256×10^2) actual channel samples and 5×10^4 (resp. 256×10^2) artificial channel samples generated by the GAN in Fig. 3 (resp. CGAN in Fig. 5). This scheme correspond to the case of practical training given the limited number of actual channel samples. For the case of this practical training, we first train the constructed GAN in Fig. 3 (resp. CGAN in Fig. 5) using 5×10^4 (resp. 256×10^2) actual channel samples, and then, we generate 5×10^4 (resp. 256×10^2) artificial channel samples using the trained generator. After the training, we evaluate the performance of the proposed scheme in Fig. 2 (or Fig. 4) using 3×10^4 (or 256×10^2) actual channel samples that are different from those used for the training.

C. QUASI-STATIC BLOCK FADING SCENARIO

In Fig. 6, for the quasi-static block fading scenario, we plot the fading distribution of the actual channel samples and that of the artificial channel samples generated by the proposed GAN in Fig. 3. In this figure, only the distributions of the real parts of the actual and artificial channel samples are depicted for the illustration purpose. From Fig. 6, it can be seen that the constructed GAN generates the channel samples very well, of which distribution is almost the same as that of the actual channel samples. In Fig. 7, we plot the scatter diagrams of the actual channel samples and those of the artificial channel samples generated by the GAN in Fig. 3 for the quasi-static fading scenario. From Fig. 7, it can be clearly seen that the proposed GAN generates the artificial channel samples that have the very similar (spatial) correlations as those of the actual channel samples. Therefore, using the artificial channel samples generated by the proposed GAN (or CGAN)

[11] That is, in the simulations, we use the data generated from the realistic, experimentally validated, and widely adopted channel models (i.e., Rician channel model and Gauss-Markov channel model), because suitable real-world data for received SNR feedback was not available to the authors. Note that the numerical simulations with the synthetic (or generated) data performed in our paper are the key first step to conduct the experiments with the real measurement data. Also, the numerical results presented in our paper for the synthetic data provide a useful guidance or performance benchmark for the experiments with the real measurement data.

[12] The Rician channel model is a realistic, general, and widely adopted channel model for the quasi-static block fading [1], [2]. For this reason, in the numerical simulations, we adopt the Rician channel model for the quasi-static block fading scenario.

[13] The Gauss-Markov channel model is a practical and widely adopted channel model for the time-varying fading [1], [2], in which the time-variation of the CSI is modeled based on a first-order autoregressive (AR) process. Thus, in the numerical simulations, we adopt the Gauss-Markov channel model for the time-varying fading scenario.

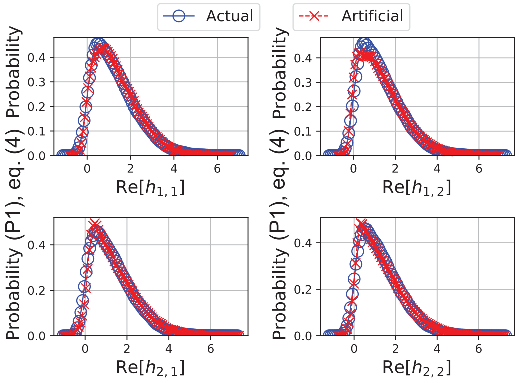


FIGURE 6. Fading distributions of the real parts of the actual channel samples and the artificial channel samples generated by the constructed GAN in Fig. 3. The results are shown for the quasi-static block fading scenario.

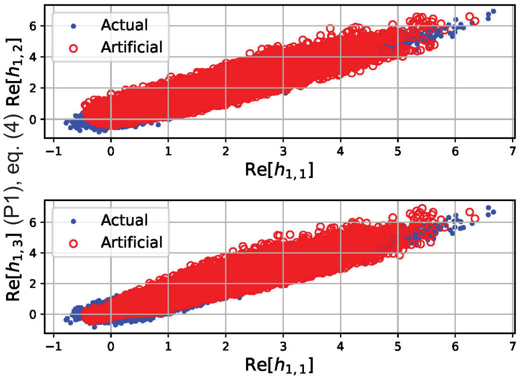


FIGURE 7. Scatter diagrams of the real parts of the actual channel samples and the artificial channel samples generated by the proposed GAN in Fig. 3. The results are shown for the quasi-static block fading scenario.

along with the actual channel samples, it is possible to extract (or estimate) the spatial (or temporal) correlation information.

In Figs. 8 and 9, for the quasi-static block fading scenario, the performance of the proposed scheme in Fig. 2 is compared to the following baseline schemes: (i) the GS algorithm of (6), (ii) linear MMSE (LMMSE) estimator (refer to Appendix B for details), and (iii) the proposed scheme in Fig. 2 without (w/o) the CNN (i.e., the convolutional and pooling layers). In Fig. 8, the channel estimation MSE is shown versus the transmit SNR. In Fig. 9, the lower bound of mutual information (LBMI) is shown versus the transmit SNR, where the LBMI is calculated according to [37], which mainly depends on the channel estimation MSE.

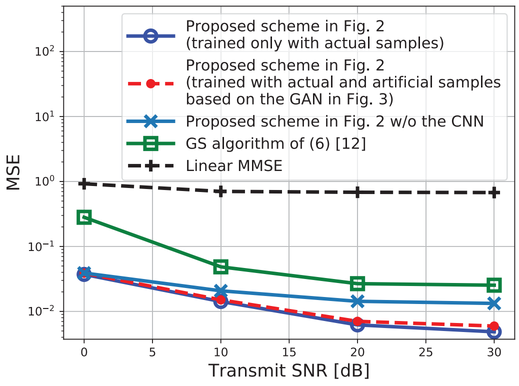


FIGURE 8. MSE versus the transmit SNR for the quasi-static block fading scenario.

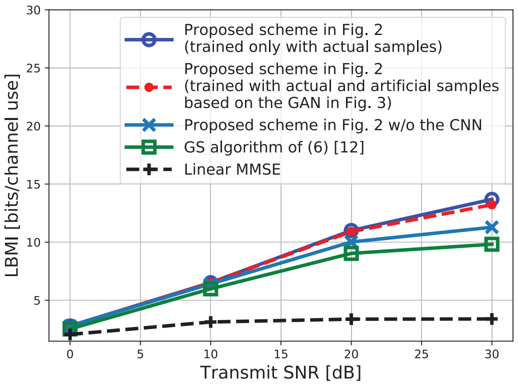


FIGURE 9. LBMI versus the transmit SNR for the quasi-static block fading scenario.

From Fig. 8, it can be observed that the proposed scheme in Fig. 2 achieves much lower MSE than the baseline schemes. This clearly shows that the proposed scheme successfully and effectively learns the MIMO channel coefficients from the received SNR feedback information by addressing the global phase ambiguity issue. It also clearly shows the effectiveness of utilizing the CNN (or exploiting the spatial features of the MIMO channels) for the joint channel estimation and pilot signal design in the MIMO system. Thanks to the lower MSE, the LBMI of the proposed scheme in Fig. 2 is much larger than that of the other schemes, indicating that a substantially higher capacity can be achieved in practice by the proposed scheme. Meanwhile, the LMMSE estimator performs worst among the schemes. This is mainly because the LMMSE estimate of the channel Gram matrix is not always guaranteed to be rank-one, and thus, the obtained channel estimate suffers significantly from the approximation error.

Also, it is important to note that the proposed scheme in Fig. 2 trained using the artificial channel samples generated by the GAN in Fig. 3 (i.e., for the case of practical training) works very well: its performance is essentially the same as that trained only with the actual channel samples (i.e., for the case of ideal training). Thus, the proposed training approach using the GAN is very useful and effective for the practical applications.

Fig. 10 shows the MSE versus the transmit SNR for the quasi-static block fading scenario when the proposed scheme in Fig. 2 is trained with different values of the transmit SNR, where the SNR values in the square boxes mean the transmit

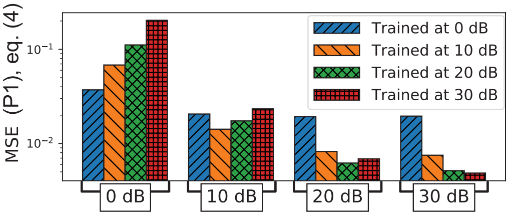


FIGURE 10. MSE versus the transmit SNR for the quasi-static block fading scenario when the proposed scheme in Fig. 2 is trained with different values of transmit SNR. The SNR values in the square boxes denote the transmit SNR values for the test and those in the legend denote the transmit SNR values used for the training.

SNR values used for the test (i.e., performance evaluation), whereas those in the legend mean the transmit SNR values used for the training. It can be seen that if the proposed scheme is trained at the training SNR values of 0 dB, 10 dB, 20 dB, and 30 dB, then it yields the lowest MSE at the test SNR values of 0 dB, 10 dB, 20 dB, and 30 dB, respectively. This means that to achieve the best performance of the proposed scheme in practice, the transmit SNR value used for the training should be judiciously and carefully selected such that it matches very close to that used for the actual system operation.

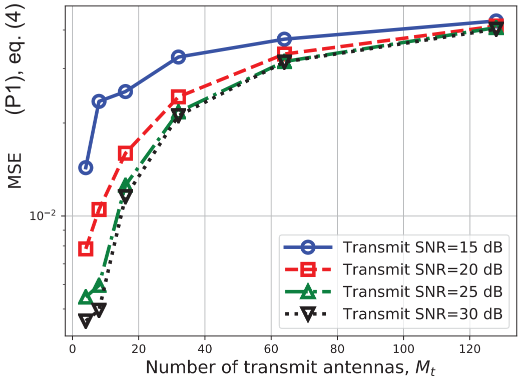


FIGURE 11. MSE of the proposed scheme in Fig. 2 versus the number M_t of transmit antennas for the quasi-static fading scenario.

In Fig. 11, we plot the channel estimation MSE of the proposed scheme in Fig. 2 versus the number M_t of transmit antennas for the quasi-static fading scenario when it is trained at the transmit SNR value of 30 dB. From Fig. 11, it can be seen that the proposed scheme in Fig. 2 scales and works well over the wide range of M_t including the massive regime of M_t (e.g., $M_t \in \{16, 32, 64, 128\}$). When M_t increases, the channel estimation MSE of the proposed scheme increases because the number of channel coefficients to be estimated increases.

In Figs. 12 and 13, we plot the (real parts of) pilot signals of the (two) transmit antennas learned by the proposed scheme in Fig. 2 when the transmit SNR values are 0 dB and 20 dB, respectively. From Fig. 12, it can be observed that when the transmit SNR is low (i.e., 0 dB), the pilot signals of the transmit antennas are perfectly correlated (i.e., the same), and thus, they will probe the channels of all the transmit antennas in exactly the same manner. This is because in our simulation setting, the MIMO channel coefficients are spatially correlated, and thus, the learned pilot signals are generally correlated, meaning that the pilot signals generally probe the channels of different transmit antennas in a correlated manner. On the other hand, from Fig. 13, it appears that when the transmit SNR is high (i.e., 20 dB), the pilot signals of the transmit antennas are less correlated (i.e., look dissimilar), and thus, they will probe (or estimate) the channel of each transmit antenna in a less correlated manner. This is because when the SNR is sufficiently high, the correlations among the pilot signals tend to be sufficiently weak [28], [38], and thus, it becomes optimal to estimate the channel of each transmit antenna independently.

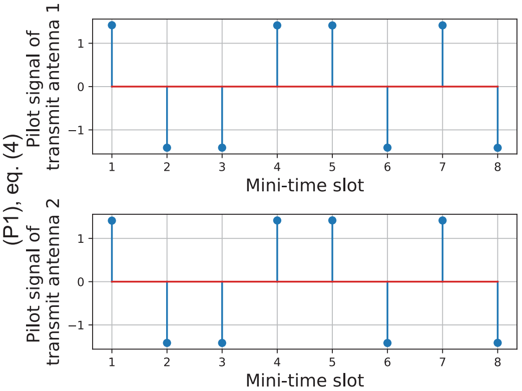


FIGURE 12. Pilot signals of the transmit antennas learned by the proposed scheme in Fig. 2 when the transmit SNR is 0 dB. The results are shown for the quasi-static block fading scenario.

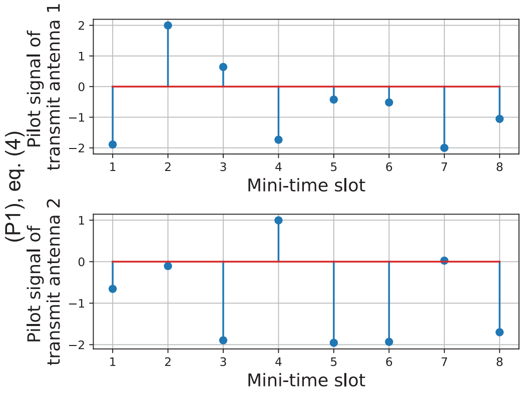


FIGURE 13. Pilot signals of the transmit antennas learned by the proposed scheme in Fig. 2 when the transmit SNR is 20 dB. The results are shown for the quasi-static block fading scenario.

D. TIME-VARYING FADING SCENARIO

In Fig. 14, for the time-varying scenario, we plot the real parts of the actual channel samples and the artificial channel samples generated by the CGAN in Fig. 5, where the results are

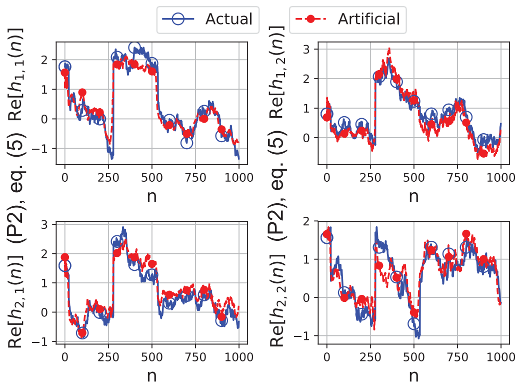


FIGURE 14. Real parts of the actual channel samples and the artificial channel samples generated by the proposed scheme in Fig. 5. The results are shown for the time-varying fading scenario.

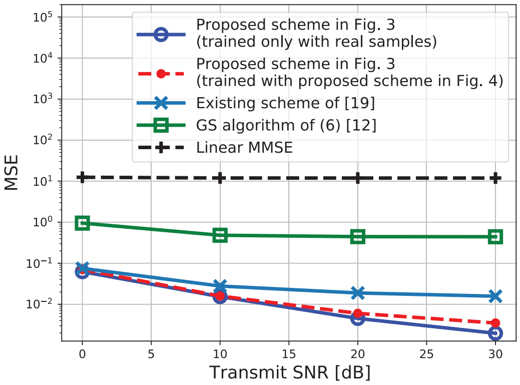


FIGURE 15. MSE versus the transmit SNR for the time-varying fading scenario.

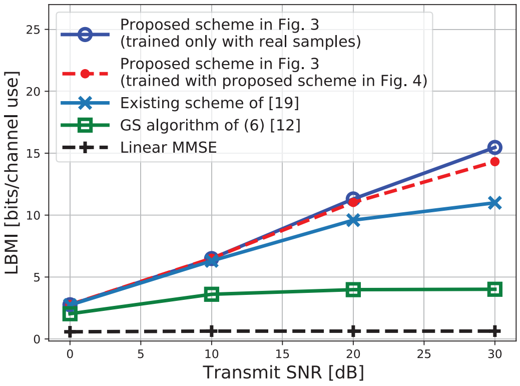
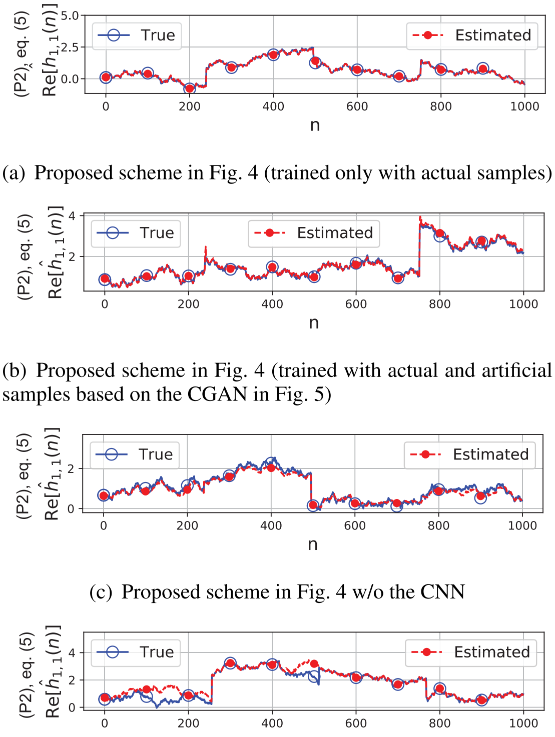


FIGURE 16. LBMI versus the transmit SNR for the time-varying fading scenario.

shown when the loss function $\mathcal{V}(\vartheta_G, \vartheta_D)$ of (24) converges with respect to both the parameters ϑ_G and ϑ_D. In this setting, the constructed CGAN is observed to generate the artificial channel samples that look very similar to the actual ones.

In Figs. 15 and 16, the performance of the proposed scheme in Fig. 4 is compared to the following baseline schemes: (i) the GS algorithm of (6), (ii) linear MMSE (LMMSE) estimator, and (iii) the proposed scheme in Fig. 4 without (w/o) the CNN (i.e., the convolutional and pooling layers). From Figs. 15 and 16, one can make the similar observations as in Figs. 8 and 9, respectively. Particularly, the proposed scheme in Fig. 4 performs much better than the other schemes, which clearly shows the effectiveness and benefit of exploiting the spatial features and the other useful features for the time-varying MIMO channel estimation along with the temporal features. We also note that the proposed scheme in Fig. 4 trained using the artificial channel samples generated by the CGAN in Fig. 5 performs very close to that trained using only the actual channel samples.



(a) Proposed scheme in Fig. 4 (trained only with actual samples)

(b) Proposed scheme in Fig. 4 (trained with actual and artificial samples based on the CGAN in Fig. 5)

(c) Proposed scheme in Fig. 4 w/o the CNN

(d) GS algorithm of (6) [12]

FIGURE 17. Real parts of the first elements (i.e., $\{Re[\hat{h}_{1,1}(n)]\}$) of the time-varying MIMO channels estimated by the various schemes.

In Fig. 17, to see the channel estimation results for the time-varying fading scenario, the real parts of the first elements of the time-varying MIMO channels (i.e., $\{Re[\hat{h}_{1,1}(n)]\}$) estimated by the various schemes are plotted versus the symbol time n when the transmit SNR is 30 dB, where the true channel values mean the actual or artificial channel values to be estimated. The channel estimates obtained by the proposed scheme in Fig. 4 are closest to the true channel values, meaning that it learns how to accurately estimate the time-varying MIMO channels very well.

In Fig. 18, we plot the channel estimation MSE of the proposed scheme in Fig. 4 versus the number M_t of transmit antennas for the time-varying fading scenario when it is

TABLE 2. Computational complexities of the proposed schemes and the GS Algorithm.

GS algorithm	Proposed scheme in Fig. 2	Proposed scheme in Fig. 4
$\mathcal{O}\big(M_t^3 M_r^3\big)$	$\mathcal{O}\big(\mathcal{N}_{conv}^2\mathcal{L}_{conv}\mathcal{M}_{conv}+\mathcal{N}_{fully}^2\mathcal{L}_{fully}\big)$	$\mathcal{O}\big(N_{LSTM}^2 L_{LSTM}+N_{conv}^2 L_{conv}M_{conv}+N_{fully}^2 L_{fully}\big)$

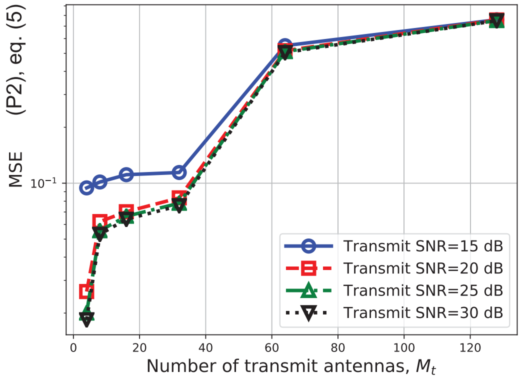


FIGURE 18. MSE of the proposed scheme in Fig. 4 versus the number M_t of transmit antennas for the time-varying fading scenario.

trained at the training SNR value of 30 dB. From Fig. 18, one can make the similar conclusions as in Fig. 11.

E. EFFECTS OF THE FEEDBACK NOISE

In this subsection, we investigate the effects of the feedback noise, which can be observed from the numerical results presented in the previous subsections. Specifically, from Figs. 8–11, 15, 16, and 18, it can be clearly seen that as the transmit SNR increases, the performance of all the schemes improves, because the impact of the feedback noise becomes reduced. Therefore, it can be concluded that the feedback noise adversely affects the channel estimation with received SNR feedback, as expected. In particular, as can be clearly seen from Figs. 8, 9, 15, and 16, the performance gap between the proposed and the other schemes gets larger as the transmit SNR increases, meaning that in the high SNR region, the impact of the feedback noise becomes more pronounced. Also, from Figs. 11 and 18, it turns out that when the number M_t of transmit antennas is small, the performance variation of the proposed scheme over the transmit SNR values is large. This clearly indicates that the feedback noise has more significant impact on the proposed scheme for the small values of M_t.

F. COMPUTATIONAL COMPLEXITY

In this subsection, we compare the computational complexities of the proposed schemes and the GS algorithm. For the proposed scheme in Fig. 2, let $\mathcal{N}_{conv}$ and $\mathcal{N}_{fully}$ (resp. $\mathcal{L}_{conv}$ and $\mathcal{L}_{fully}$) denote the largest numbers of nodes (resp. the number of layers) in the convolutional/pooling and fully-connected layers, respectively. Also, for the proposed scheme in Fig. 4, let L_{conv}, N_{LSTM}, and N_{fully} (resp. L_{conv}, L_{LSTM}, and L_{fully}) denote the largest numbers of nodes (resp. the number of layers) in the LSTM, convolutional/pooling, and fully-connected layers, respectively. Then the overall (worst-case) computational complexities of these schemes can be estimated as $\mathcal{O}\big(\mathcal{N}_{conv}^2\mathcal{L}_{conv}\mathcal{M}_{conv}+\mathcal{N}_{fully}^2\mathcal{L}_{fully}\big)$ and $\mathcal{O}\big(N_{LSTM}^2 L_{LSTM}+N_{conv}^2 L_{conv}M_{conv}+N_{fully}^2 L_{fully}\big)$, respectively [9]. On the other hand, the computational complexity of the GS algorithm is given by $\mathcal{O}\big(M_t^3 M_r^3\big)$, which is mainly dominated by computing the inversion of the matrix of the size $M_tM_r \times M_tM_r$ (i.e., $\big(\mathbf{S}^*(n)\mathbf{S}^T(n)\big)^{-1}$ in (6)).

In Table 2, we summarize the computational complexities of the proposed schemes and the GS algorithm. If the sizes of the constructed networks are properly chosen, then the computational complexities of the proposed schemes can be lower than that of the GS algorithm. For example, if we select the parameters as $\mathcal{N}_{conv} = \mathcal{L}_{conv} = \mathcal{M}_{conv} = \mathcal{N}_{fully} = \mathcal{L}_{fully} \leq (M_tM_r)^{3/4}$, the computational complexity of the proposed scheme in Fig. 2 is lower than that of the GS algorithm; but, even in this simple setting, it considerably outperforms the GS algorithm by effectively and intelligently resolving the global phase ambiguity issue, as demonstrated by the numerical results. Therefore, it is very useful and effective for the practical applications.

VI. CONCLUSION

In this paper, we studied the deep learning based channel estimation for the MIMO system with received SNR feedback in the quasi-static block-fading and time-varying fading scenarios. In the quasi-static block fading scenario, we developed the novel technique for the joint MIMO channel estimation and pilot signal design by constructing the deep autoencoder with the CNN. In the time-varying scenario, the new channel estimation technique was proposed by combining the RNN and CNN. For the two fading scenarios, we construct the effective GAN and CGAN, respectively, to generate the artificial channel samples, and the training procedures using the artificial channel samples were also presented. The effectiveness and the superior performance of the proposed schemes were demonstrated through the numerical results.

A very important and interesting further work is to design the feedback signaling. Also, it would be very persuasive and meaningful to conduct the experiments/simulations with the real measurement data.

APPENDIX A
DETAILS OF GERCHEBERG-SAXTON ALGORITHM

This section presents the details of the GS algorithm. Let us consider the general form of the phase retrieval problem, of which goal is to retrieve a complex-valued signal $\mathbf{x}_o \in \mathbb{C}^N$

from the magnitude of several (noisy) linear measurements:

$$y_m = \left|\mathbf{a}_m^H \mathbf{x}_o\right| + n_m \in \mathbb{R}, \quad \forall m = 1, \ldots, M, \tag{25}$$

where the measuring vectors $\{\mathbf{a}_m \in \mathbb{C}^N\}_{m=1}^M$ are given and $\{n_m\}_{m=1}^M$ denote the additive noises. The GS algorithm aims at retrieving the unknown signal $\mathbf{x}_o$ up to a global phase ambiguity: $\mathbf{x}_o e^{j\phi}$ for any ϕ yields the same magnitude information $\{y_m\}_{m=1}^M$. In the GS algorithm, this global phase ambiguity cannot be removed, which in turn severely limits the performance of the channel estimation with received SNR feedback.

In [12], Gerchberg and Saxton suggested to find an estimate of the unknown variable $\mathbf{x}_o \in \mathbb{C}^N$ as the solution of the following optimization problem:

$$\underset{\mathbf{x} \in \mathbb{C}^N}{\text{minimize}} \quad \sum_{m=1}^{M} \left(y_m - \left|\mathbf{a}_m^H \mathbf{x}\right| \right)^2. \tag{26}$$

This problem is not convex. Introducing new variables $\{\phi_m\}_{m=1}^M$ to represent the missing phase information, (26) is equivalent to

$$\underset{\mathbf{x} \in \mathbb{C}^N, \{\phi_m\}_{m=1}^M}{\text{minimize}} \quad \sum_{m=1}^{M} \left| y_m e^{j\phi_m} - \mathbf{a}_m^H \mathbf{x} \right|^2. \tag{27}$$

Even though (27) is not jointly convex in $\mathbf{x}$ and $\{\phi_m\}_{m=1}^M$, it is quadratic and convex in $\mathbf{x}$ when the phase completion variables $\{\phi_m\}_{m=1}^M$ are fixed. Therefore, the reformulated problem in (27) can be handled via alternating optimization. The solution to the subproblem of finding $\mathbf{x}$ given $\{\phi_m\}_{m=1}^M$ can be obtained by setting the derivative of the objective function of (27) to zero as follows:

$$\hat{\mathbf{x}} = \left(\sum_{m=1}^{M} \mathbf{a}_m \mathbf{a}_m^H \right)^{-1} \cdot \sum_{m=1}^{M} \mathbf{a}_m y_m e^{j\phi_m}. \tag{28}$$

On the other hand, from the Cauchy-Schwarz inequality, the solution to the subproblem of finding $\{\phi_m\}_{m=1}^M$ given $\mathbf{x}$ can be obtained as

$$\hat{\phi}_m = \arg\left(\mathbf{a}_m^H \mathbf{x}\right) - \pi \cdot 1_{y_m < 0}, \quad \forall m = 1, \ldots, M, \tag{29}$$

where $1_{y_m<0}$ is equal to 1 if $y_m < 0$ and 0 otherwise. Note that the magnitude information y_m can be negative due to the additive noise n_m according to (25). Substituting (29) into (28) yields the following simple update of the complex-valued variable $\mathbf{x}$ in the problem of (26):

$$\begin{aligned} \mathbf{x}^{(t+1)} &= \left(\sum_{m=1}^{M} \mathbf{a}_m \mathbf{a}_m^H \right)^{-1} \cdot \sum_{m=1}^{M} \mathbf{a}_m \left|y_m\right| e^{j \arg\left(\mathbf{a}_m^H \mathbf{x}^{(t)}\right)} \\ &= \left(\mathbf{A}\mathbf{A}^H\right)^{-1} \mathbf{A} \left|\mathbf{y}\right| \exp\left(\jmath \angle \left\{ \mathbf{A}^H \mathbf{x}^{(t)} \right\} \right) \end{aligned} \tag{30}$$

where $\mathbf{x}^{(t)}$ denotes the value of $\mathbf{x}$ at the tth iteration. Also, $\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_M]$ and $\mathbf{y} = [y_1, \cdots, y_M]^T$. The algorithm presented in (30) is referred to as the GS algorithm.

Replacing the terms $\mathbf{A}$, $\mathbf{y}$, and $\mathbf{x}$ by $\left(\mathbf{S}^* \otimes \mathbf{I}_{M_r}\right)$, $\boldsymbol{\gamma}(n)$, and $\mathbf{h}(n)$ respectively, the GS channel estimate based on the received SNR feedback information can be obtained as in (6).

APPENDIX B
DETAILS OF LMMSE ESTIMATOR

This section presents the details of the LMMSE estimator. Let $\mathbf{x}_i^T(n) \in \mathbb{C}^{1 \times M_t M_r}$ denote the ith row of the matrix $g(n)\left(\mathbf{S}^T(n) \otimes \mathbf{I}_{M_r}\right)$, $i = 1, \cdots, N_p M_r$. Also, let $\gamma_i(n)$ and $\eta_i(n)$ denote the ith elements of the vectors $\boldsymbol{\gamma}(n)$ and $\boldsymbol{\eta}(n)$, respectively, $i = 1, \cdots, N_p M_r$. Then the received SNR feedback information in (3) can be equivalently written as

$$\begin{aligned} \gamma_i(n) &= \left|\mathbf{x}_i^T(n)\mathbf{h}(n)\right| + \eta_i(n) \\ &= \mathrm{Tr}\left(\mathbf{X}_i^*(n)\mathbf{H}(n)\right) + \eta_i(n), \quad i = 1, \cdots, N_p M_r \end{aligned} \tag{31}$$

where $\mathbf{H}(n) \triangleq \mathbf{h}(n)\mathbf{h}^H(n) \triangleq$ denotes the Gram matrix of the MIMO channel vector $\mathbf{h}(n)$. Also, $\mathbf{X}_i(n) \triangleq \mathbf{x}_i(n)\mathbf{x}_i^H(n)$. For any $N \times N$ Hermitian matrices $\mathbf{A}$ and $\mathbf{B}$, it follows that [39], [40]

$$\mathrm{Tr}\left(\mathbf{A}\mathbf{B}\right) = cvec^T(\mathbf{A}) cvec(\mathbf{B}) \tag{32}$$

where $cvec(\mathbf{A})$ denotes the complex vectorization of the matrix $\mathbf{A}$, which is defined as

$$\begin{aligned} \mathrm{cvec}(\mathbf{A}) = [&\boldsymbol{\alpha}(\mathbf{A}); \boldsymbol{\alpha}_1^{\mathrm{re}}(\mathbf{A}); \boldsymbol{\alpha}_1^{\mathrm{im}}(\mathbf{A}); \boldsymbol{\alpha}_2^{\mathrm{re}}(\mathbf{A}); \boldsymbol{\alpha}_2^{\mathrm{im}}(\mathbf{A}); \\ &\ldots; \boldsymbol{\alpha}_{N-1}^{\mathrm{re}}(\mathbf{A}); \boldsymbol{\alpha}_{N-1}^{\mathrm{im}}(\mathbf{A})]. \end{aligned} \tag{33}$$

In the above equation, $[\mathbf{a}; \mathbf{b}]$ stands for the vertical concatenation of the column vectors $\mathbf{a}$ and $\mathbf{b}$. Also,

$$\boldsymbol{\alpha}(\mathbf{A}) = [a_{1,1}, \cdots, a_{N,N}]^T, \tag{34}$$

$$\begin{aligned} \boldsymbol{\alpha}_n^{\mathrm{re}}(\mathbf{A}) = [&\sqrt{2}\,\mathrm{Re}(a_{n,n+1}), \sqrt{2}\,\mathrm{Re}(a_{n,n+2}), \\ &\ldots, \sqrt{2}\,\mathrm{Re}(a_{n,N-1}), \sqrt{2}\,\mathrm{Re}(a_{n,N})]^T, \end{aligned} \tag{35}$$

$$\begin{aligned} \boldsymbol{\alpha}_n^{\mathrm{im}}(\mathbf{A}) = [&\sqrt{2}\,\mathrm{Im}(a_{n,n+1}), \sqrt{2}\,\mathrm{Im}(a_{n,n+2}), \\ &\ldots, \sqrt{2}\,\mathrm{Im}(a_{n,N-1}), \sqrt{2}\,\mathrm{Im}(a_{n,N})]^T. \end{aligned} \tag{36}$$

In (34)–(36), $\boldsymbol{\alpha}(\mathbf{A})$ denotes the vector of the diagonal elements of $\mathbf{A}$ (which are real-valued), $\boldsymbol{\alpha}_n^{\mathrm{re}}(\mathbf{A})$ denotes the (scaled) real parts of the upper (or lower) off-diagonal elements of $\mathbf{A}$, and $\boldsymbol{\alpha}_n^{\mathrm{im}}(\mathbf{A})$ denotes the (scaled) imaginary parts of the upper (or lower) off-diagonal elements of $\mathbf{A}$. Applying (32) to (31) and denoting $\mathbf{a}_i(n) \triangleq cvec(\mathbf{X}_i^*(n))$ and $\mathbf{b}(n) \triangleq cvec(\mathbf{H}(n))$, we have

$$\gamma_i(n) = \mathbf{a}_i^T(n)\mathbf{b}(n) + \eta_i(n), \quad i = 1, \cdots, N_p M_r. \tag{37}$$

Let us define

$$\mathbf{A}(n) \triangleq [\mathbf{a}_1^T(n), \mathbf{a}_2^T(n), \cdots, \mathbf{a}_{N_p M_r}^T(n)]^T. \tag{38}$$

Then $\{\gamma_i(n)\}$ in (37) (i.e., the received SNR feedback information in (3)) can be written in terms of a linear function of $\mathbf{b}(n)$ (i.e., the Gram matrix of the MIMO channel $\mathbf{h}(n)$) as follows:

$$\boldsymbol{\gamma}(n) = \mathbf{A}(n)\mathbf{b}(n) + \boldsymbol{\eta}(n) \tag{39}$$

for $\forall n$.

Since the received SNR feedback information $\boldsymbol{\gamma}(n)$ (i.e., the observation) is a linear function of $\mathbf{b}(n)$ (i.e., the variable to be estimated), the LMMSE estimator can be applied to estimating the value of $\mathbf{b}(n)$ from $\boldsymbol{\gamma}(n)$. Let $\mathbf{C}(n) \triangleq \mathbb{E}[\mathbf{b}(n)\mathbf{b}^T(n)]$

denote the covariance matrix of $\mathbf{b}(n)$. Then the LMMSE channel estimate of $\mathbf{b}(n)$ is given by [41]

$$\hat{\mathbf{b}}(n) = \mathbf{C}(n)\mathbf{A}^T(n)\big(\mathbf{A}(n)\mathbf{C}(n)\mathbf{A}^T(n) + \delta^2\mathbf{I}\big)^{-1}\boldsymbol{\gamma}(n) \tag{40}$$

where δ^2 denotes the variance of $\gamma_i(n)$. Using the LMMSE estimate of $\mathbf{b}(n)$ above, the estimate of the Gram matrix of the MIMO channel can be obtained as

$$\hat{\mathbf{H}}(n) = cvec^{-1}(\hat{\mathbf{b}}(n)) \tag{41}$$

where $cvec^{-1}(\cdot)$ denotes the inverse operator of $cvec(\cdot)$. Let us denote the eigenvalue decomposition (EVD) of $\hat{\mathbf{H}}(n)$ by

$$\hat{\mathbf{H}}(n) = \sum_{i=1}^{rank(\hat{\mathbf{H}}(n))} \lambda_i(n)\mathbf{u}_i(n)\mathbf{u}_i^H(n) \tag{42}$$

where $\lambda_i(n)$ and $\mathbf{u}_i(n)$ are the ith (non-zero) eigenvalue and the corresponding eigenvector of $\hat{\mathbf{H}}(n)$, respectively. Suppose that the eigenvalues of $\hat{\mathbf{H}}(n)$ are arranged in descending order (i.e., $\lambda_1(n) \geq \cdots \geq \lambda_{rank(\hat{\mathbf{H}}(n))}$). Also, $rank\big(\hat{\mathbf{H}}(n)\big)$ is the rank of $\hat{\mathbf{H}}(n)$. If $\hat{\mathbf{H}}(n)$ is rank-one (i.e., $rank\big(\hat{\mathbf{H}}(n)\big) = 1$), then the estimate of the MIMO channel $\mathbf{h}(n)$ can be directly obtained from the EVD of $\hat{\mathbf{H}}(n)$ as

$$\hat{\mathbf{h}}(n) = \lambda_1\mathbf{u}_1\mathbf{u}_1^H. \tag{43}$$

On the other hand, if $\hat{\mathbf{H}}(n)$ is not rank-one (i.e., $rank\big(\hat{\mathbf{H}}(n)\big) > 1$), then the estimate of the MIMO channel $\mathbf{h}(n)$ can be obtained from the rank-one approximation of $\hat{\mathbf{H}}(n)$ as follows:

$$\hat{\mathbf{h}}(n) \approx \lambda_1\mathbf{u}_1\mathbf{u}_1^H. \tag{44}$$

REFERENCES

[1] E. Biglieri, R. Calderbank, A. Constantinides, A. Goldsmith, A. Paulraj, and V. H. Poor, *MIMO Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2007.
[2] D. N. C. Tse and P. Viswanath, *Fundamentals of Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2005.
[3] G. Caire, N. Jindal, M. Kobayashi, and N. Ravindran, "Multiuser MIMO achievable rates with downlink training and channel state feedback," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2845–2866, Jun. 2010.
[4] 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.
[5] Y. Zeng and R. Zhang, "Optimized training design for wireless energy transfer," *IEEE Trans. Commun.*, vol. 63, no. 2, pp. 536–550, Feb. 2015.
[6] J. Jose, A. Ashikhmin, T. L. Marzetta, and S. Vishwanath, "Pilot contamination and precoding in multi-cell TDD systems," *IEEE Trans. Wireless Commun.*, vol. 10, no. 8, pp. 2640–2651, Aug. 2011.
[7] S. Abeywickrama, T. Samarasinghe, C. K. Ho, and C. Yuen, "Wireless energy beamforming using received signal strength indicator feedback," *IEEE Trans. Signal Process.*, vol. 66, no. 1, pp. 224–235, Jan. 2018.
[8] T. Qiu, X. Fu, N. D. Sidiropoulos, and D. P. Palomar, "MISO channel estimation and tracking from received signal strength feedback," *IEEE Trans. Signal Process.*, vol. 66, no. 7, pp. 1691–1704, Apr. 2018.
[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
[11] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 3, pp. 794–816, Feb. 2017.
[12] R. W. Gerchberg and A. W. O. Saxton, "A practical algorithm for the determination of phase from image and diffraction plane pictures," *Optik*, vol. 35, no. 2, pp. 237–250, 1971.
[13] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cognit. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.
[14] M. Kim, W. Lee, and D.-H. Cho, "A novel PAPR reduction scheme for OFDM system based on deep learning," *IEEE Commun. Lett.*, vol. 22, no. 3, pp. 510–513, Mar. 2018.
[15] M. Kim, N.-I. Kim, W. Lee, and D.-H. Cho, "Deep learning-aided SCMA," *IEEE Commun. Lett.*, vol. 22, no. 4, pp. 720–723, Apr. 2018.
[16] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based channel estimation for beamspace mmWave massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 852–855, Oct. 2018.
[17] P. Dong, H. Zhang, G. Y. Li, N. NaderiAlizadeh, and I. S. Gaspar, "Deep CNN based channel estimation for mmWave massive MIMO systems," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Proc. (ICASSP)*, May 2019, pp. 12–17.
[18] Y. Jin, J. Zhang, S. Jin, and B. Ai, "Channel estimation for cell-free mmWave massive MIMO through deep learning," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10325–10329, Oct. 2019.
[19] Y. Yang, F. Gao, G. Y. Li, and M. Jian, "Deep learning-based downlink channel prediction for FDD massive MIMO system," *IEEE Commun. Lett.*, vol. 23, no. 11, pp. 1994–1998, Nov. 2019.
[20] C.-J. Chun, J.-M. Kang, and I.-M. Kim, "Deep learning-based channel estimation for massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1228–1231, Aug. 2019.
[21] C.-J. Chun, J.-M. Kang, and I.-M. Kim, "Deep learning-based joint pilot design and channel estimation for multiuser MIMO channels," *IEEE Commun. Lett.*, vol. 23, no. 11, pp. 1999–2003, Nov. 2019.
[22] H. Jiang, Z. Zhang, L. Wu, J. Dang, and G. Gui, "A 3-D non-stationary wideband geometry-based channel model for MIMO Vehicle-to-Vehicle communications in tunnel environments," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6257–6271, Jul. 2019.
[23] H. Jiang, Z. Zhang, J. Dang, and L. Wu, "A novel 3-D massive MIMO channel model for vehicle-to-vehicle communication environments," *IEEE Trans. Commun.*, vol. 66, no. 1, pp. 79–90, Jan. 2018.
[24] C.-X. Wang, J. Bian, J. Sun, W. Zhang, and M. Zhang, "A survey of 5G channel measurements and models," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3142–3168, 4th Quart., 2018.
[25] H. Huang, Y. Song, J. Yang, G. Gui, and F. Adachi, "Deep-learning-based millimeter-wave massive MIMO for hybrid precoding," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3027–3032, Mar. 2019.
[26] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, "Deep learning for super-resolution channel estimation and DOA estimation based massive MIMO system," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8549–8560, Sep. 2018.
[27] H. Lutkepohl, *Handbook Matrices*. New York, NY, USA: Wiley, 1996.
[28] E. Bjornson and B. Ottersten, "A framework for training-based estimation in arbitrarily correlated Rician MIMO channels with Rician disturbance," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1807–1820, Mar. 2010.
[29] J.-M. Kang, C.-J. Chun, and I.-M. Kim, "Deep-learning-based channel estimation for wireless energy transfer," *IEEE Commun. Lett.*, vol. 22, no. 11, pp. 2310–2313, Nov. 2018.
[30] Y. Shechtman, Y. C. Eldar, O. Cohen, H. N. Chapman, J. Miao, and M. Segev, "Phase retrieval with application to optical imaging: A contemporary overview," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 87–109, May 2015.
[31] B. Hassibi and B. M. Hochwald, "How much training is needed in multiple-antenna wireless links?" *IEEE Trans. Inf. Theory*, vol. 49, no. 4, pp. 951–963, Apr. 2003.
[32] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.
[33] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[34] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[35] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.
[36] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*. [Online]. Available: http://arxiv.org/abs/1411.1784

[37] T. Yoo and A. Goldsmith, "Capacity and power allocation for fading MIMO channels with channel estimation error," *IEEE Trans. Inf. Theory*, vol. 52, no. 5, pp. 2203–2214, May 2006.

[38] M. Biguesh and A. B. Gershman, "Training-based MIMO channel estimation: A study of estimator tradeoffs and optimal training signals," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 884–893, Mar. 2006.

[39] J. Xu and R. Zhang, "Energy beamforming with one-bit feedback," *IEEE Trans. Signal Process.*, vol. 62, no. 20, pp. 5370–5381, Oct. 2014.

[40] K. Choi, D. Kim, and M. Y. Chung, "Received power-based channel estimation for energy beamforming in multiple-antenna RF energy transfer system," *IEEE Trans. Signal Process.*, vol. 65, no. 6, pp. 1461–1476, Mar. 2017.

[41] S. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.

JAE-MO KANG (Member, IEEE) received the Ph.D. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2017. He was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, Queen's University, Kingston, ON, Canada. He was an Assistant Professor with the School of Intelligent Mechatronics Engineering, Sejong University, Seoul, South Korea. He is currently an Assistant Professor with the Department of Artificial Intelligence, Kyungpook National University, Daegu, South Korea. His research interests include deep learning, reinforcement learning, federated learning, the Internet-of-Things (IoT), and emerging information and communication technology (ICT).

CHANG-JAE CHUN (Member, IEEE) received the Ph.D. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2018. He was a Postdoctoral Researcher with the Department of Electrical and Computer Engineering, Queen's University, Kingston, ON, Canada. He is currently a Researcher with the Industry Applications Research Division, Korea Electrotechnology Research Institute, Changwon, South Korea. His current research interests include massive MIMO, relay, energy harvesting, hybrid beamforming, deep learning, and reinforcement learning.

IL-MIN KIM (Senior Member, IEEE) received the B.Sc. degree in electronics engineering from Yonsei University, Seoul, South Korea, in 1996, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 1998 and 2001, respectively. From October 2001 to August 2002, he was with the Department of Electrical Engineering and Computer Sciences, MIT, Cambridge, MA, USA. From September 2002 to June 2003, he was a Postdoctoral Research Fellow with the Department of Electrical Engineering, Harvard University, Cambridgeh, MA, USA. In July 2003, he joined the Department of Electrical and Computer Engineering, Queen's University, Kingston, ON, Canada, where he is currently a Professor. His current research interests include deep learning, reinforcement learning, future wireless systems (e.g., 6G), wireless artificial intelligence, on-device AI, federated learning, neural networks for IoT, distributed edge intelligence, convergence of computing, learning, and communications, etc. He has been an Editor of the IEEE Transactions on Wireless Communications, the IEEE Wireless Communications Letters, and the *Journal of Communications and Networks* (JCN).