Abstract—For intelligent reflecting surface (IRS)-aided wireless communications, channel estimation is essential and usually requires excessive channel training overhead when the number of IRS reflecting elements is large. The acquisition of accurate channel state information (CSI) becomes more challenging when the channel is not quasi-static due to the mobility of the transmitter and/or receiver. In this work, we study an IRS-aided wireless communication system with a practical channel model that characterizes the time-varying propagation property and propose an innovative two-stage transmission protocol. In the first stage, we send pilot symbols and track the direct/ reflected channels based on the received signal, and then data signals are transmitted. In the second stage, instead of sending pilot symbols first, we directly predict the direct/ reflected channels and all the time slots are used for data transmission. Based on the proposed transmission protocol, we propose a two-stage channel tracking and prediction (2SCTP) scheme to obtain the direct and reflected channels with low channel training overhead, which is achieved by exploiting the temporal correlation of the time-varying channels. Specifically, we first consider a special case where the IRS-access point (AP) channel is assumed to be static, for which a Kalman filter (KF)-based algorithm and a long short-term memory (LSTM)-based neural network are proposed for channel tracking and prediction, respectively. Then, for the more general case where the IRS-AP, user-IRS and user-AP channels are all assumed to be time-varying, we present a generalized KF (GKF)-based channel tracking algorithm, where proper approximations are employed to handle the underlying non-Gaussian random variables. Numerical simulations are provided to verify the effectiveness of our proposed transmission protocol and channel tracking/prediction algorithms as compared to existing ones.

Index Terms—Intelligent reflecting surface (IRS), channel tracking, channel prediction, Kalman filter, deep learning.

I. INTRODUCTION

A promising cost-effective technology for enhancing the spectral and energy efficiency of wireless communication systems, intelligent reflecting surface (IRS), also known as reconfigurable intelligent surface (RIS), has drawn significant attention in both academy and industry [1]. The IRS is composed of a massive number of low-cost passive reflecting elements, which can be smartly controlled to dynamically configure the wireless communication environment for transmission enhancement and interference suppression. Due to its passive nature, IRS requires lower hardware cost and energy consumption as compared to the traditional active relays. As such, IRS can be densely deployed in wireless communication systems to flexibly reconfigure the propagation environment, achieving improved communication capacity and reliability [2].

Several current research activities focus on how to implement IRSs [3], [4], as well as the potentials and challenges of IRS-aided wireless communications [5], [6]. In most of the existing works, the acquisition of channel state information (CSI) is essential since the performance gain provided by IRS is heavily dependent on the channel estimation accuracy. However, since IRS is passive in general and can neither send nor receive pilot symbols, the IRS-user and access point (AP)-IRS channels cannot be estimated separately. Therefore, the channel estimation problem in IRS-aided wireless communication systems is much more challenging as compared to those in conventional systems without IRS. Fortunately, the knowledge of the cascaded user-IRS-AP channel, also known as the reflected channel, is sufficient for signal detection and beamforming design [7]. As such, most of the existing channel estimation related contributions in the IRS literature focused on the reflected channel estimation problem [8]–[16]. Specifically, in [8], the authors proposed to estimate the cascaded channel coefficients one-by-one by switching only one IRS reflecting element on at each time. In [9], a discrete Fourier transform (DFT) based IRS phase-shift matrix (also named as reflection pattern) was proposed, where all IRS reflecting elements are designed to be active and the estimation variance is reduced as compared to the scheme in [8]. Moreover, the works [10] and [11] focused on the channel estimation problem in IRS-aided multi-user MISO systems. Both of them exploited the correlation among the reflected channels to reduce the channel training overhead, and the latter further improved the scheme in [10] by alleviating the negative
effects designed channel estimation algorithms based on some special properties of the channels [12]–[14]. In particular, the works [12] and [13] formulated the channel estimation problem as a sparse channel matrix recovery problem using the compressive sensing (CS) technique. The work [14] exploited the low-rank structure of the channels in massive multi-input multi-output (MIMO) systems and formulated the reflected channel estimation problem as a combined sparse matrix factorization problem. Furthermore, when the full instantaneous CSI is not available, the statistical CSI can be utilized to design advanced active and passive beamforming algorithms, which usually incurs much less channel estimation overhead [15], [16].

In most of the aforementioned works, the quasi-static channel model is assumed, i.e., the channels are assumed to be approximately constant over a relatively long coherence time, such that accurate estimation of the instantaneous CSI is possible. However, in practice, when the users are with mobility, the channel coefficients are more likely to vary and be temporally correlated, which can be utilized to further reduce the channel training overhead. For such time-varying channels, the existing channel estimation algorithms are not efficient in general and may even be inapplicable. Instead, efficient channel tracking methods are usually required to be designed to track the time-varying CSI. Channel tracking has been widely studied in the literature for traditional wireless communication systems without IRS [17]–[19]. Specifically, the work [17] studied the channel tracking and equalization problem for time-varying frequency-selective MIMO channels by approximating the MIMO channel variation using a low-order autoregressive model and tracking the approximated channel via a Kalman filter (KF). In orthogonal frequency division multiplexing (OFDM) systems, when time-varying frequency selective channels are considered, the work [18] proposed to first successively track the delay-subspace by KF and then track the channel impulse response. The authors in [19] considered an frequency division duplexing (FDD) massive MIMO system with limited scattering around the base station (BS) and proposed a two-dimensional (2D) Markov model to capture the 2D dynamic sparsity of massive MIMO channels. An efficient message passing algorithm was derived to recursively track the dynamic channel. In millimeter wave (mmWave) communication systems, the work [20] combined conventional channel estimation methods and deep learning techniques to address the beam tracking problem. However, since the distributions of the reflected channels are very complicated, the existing deep learning based beam tracking method [20] cannot be directly applied to IRS-aided wireless communication systems. For IRS-aided FDD communication systems, the work [21] considered a time-varying channel model and proposed to track the downlink direct and reflected channel using two independent KFs. It was assumed in [21] that the reflected channel, i.e., the cascaded AP-IRS-user channel, follows the Gauss-Markov model. However, this assumption may not hold in practice. Although the individual AP-user, AP-IRS and IRS-user channels can be well approximated as Gauss-Markov models as reported in many related works, the cascaded user-IRS-AP channel is no longer Gaussian distributed.1

In this work, we advance the aforementioned works by studying the channel tracking and prediction (CTP) problem in an IRS-aided wireless communication system under a more general time-varying channel model. Specifically, the user-IRS, IRS-AP and user-AP channels are assumed to be independent with each other and all follow stationary steady-state Gauss-Markov processes. In order to obtain the time-varying channels under this more realistic assumption, a two-stage transmission protocol is proposed, which can reduce the channel training overhead by exploiting the temporal correlation of the channels. Based on the proposed transmission protocol, we design an innovative two-stage CTP (2SCTP) scheme, where a KF-based channel tracking algorithm and a deep learning (DL)-enabled channel prediction algorithm are integrated for performance enhancement. Note that as compared to the existing channel/beam tracking method [20] where the channels are first estimated and then tracked, the proposed 2SCTP scheme first tracks the IRS-related channels using a limited number of pilot symbols, and then predicts the channels without any pilot symbols, which can further reduce the required channel training overhead. The main contributions of this work are summarized as follows:

- First, we propose a two-stage transmission protocol and a 2SCTP scheme. Specifically, the first stage contains two phases, in the channel training phase, the user sends pilot symbols and the AP tracks the channel based on the received pilot signals, while in the data transmitting phase, data signals are sent at the AP and the user aims to recover the data based on the tracked channel obtained in the channel training phase. In the second stage, the AP obtains the channel coefficients via prediction and all time slots in this stage are allocated for data transmission. Adopting the proposed two-stage transmission protocol is able to reduce the channel training overhead significantly, especially in the second stage.
- Second, for the special case where the IRS-AP channel is assumed to be approximately constant, we propose a KF-based algorithm for channel tracking and a long short-term memory (LSTM)-based neural network for channel prediction.2 Specifically, the KF-based channel tracking algorithm is derived by modeling the channel varying model as a linear state transition equation and using historical channel information (i.e., the covariance matrix of the previously estimated channels). It is able to recover the high-dimensional channel vector from received pilot signals (i.e., observations), whose dimension is much lower. In the second stage, the LSTM-based neural network, namely the observation (OB)-LSTM,

1Note that even though this work and [21] both address the channel tracking problem in IRS-aided communication systems, but the assumed channel temporal correlation models and the proposed methods are totally different.

2Considering this special case is meaningful since the locations of the IRS and AP are usually fixed in practice and thus the IRS-AP channel can remain constant for a relatively long time interval.
is designed to predict some virtual observations, instead of directly predicting the channels, and then the KF-based algorithm is employed again to acquire the channel coefficients based on these virtual observations. Note that predicting the virtual observations and then employing the KF-based algorithm is more efficient than directly predicting the channels using LSTM since the dimensions of the channel vectors are usually much larger than the observations, especially when the number of reflecting elements is large, which makes directly predicting the channel vectors very difficult.

- Third, we study the general case where the user-IRS, IRS-AP and user-AP channels are all assumed to be time-varying. In this case, the KF-based channel tracking algorithm cannot be directly applied, since not all the random variables (i.e., the channel coefficients) involved in the state transition equation are complex Gaussian distributed. To tackle this difficulty, we propose a novel approximation method to approximate the underlying non-Gaussian variables with Gaussian ones, based on which the state transition equation is transformed into a linear and Gaussian system. Then, a generalized KF (GKF)-based channel tracking algorithm is proposed, and by combining it with the LSTM-based channel prediction algorithm, we show that the 2SCTP scheme is also effective in the considered more general case.

- Finally, extensive numerical simulations are presented to demonstrate the effectiveness of the proposed 2SCTP scheme. We show that by exploiting the temporal correlation of the channel, much lower channel training overhead can be achieved by the proposed 2SCTP scheme as compared to the existing channel estimation methods.

The rest of the paper is organized as follows. Section II presents the system model and the framework of the proposed 2SCTP scheme. Then, the proposed CTP algorithms for the special and general cases are respectively given in Section III and Section IV. Numerical results are presented in Section V, and finally Section VI concludes the paper.

Notations: Scalars, vectors and matrices are respectively denoted by lower (upper) case, boldface lower case and boldface upper case letters. For a matrix \( X \) of arbitrary size, \( X^T \), \( X^* \), \( X^H \) denote the transpose, conjugate and conjugate transpose of \( X \), respectively. \( |x| \) represents the \( n \)-th element of the vector \( x \), and \( X_n \) denotes the \( n \)-th column of the matrix \( X \). The symbol \( || \cdot || \) denotes the Euclidean norm of a complex vector, and \( \cdot^T \) is the absolute value of a complex scalar. \( \text{diag}(x_1, \cdots, x_N) \) denotes a diagonal matrix whose diagonal elements are set as \( x_1, \cdots, x_N, \mathbb{C}^{m \times n} \) denotes the space of \( m \times n \) complex matrices. The notations \( \mathbb{E}[\cdot] \) and \( \text{Var}[\cdot] \) represent the expectation and variance of a random variable, and \( \text{Cov}[\cdot, \cdot] \) denote the covariance of two random variables. The remainder operation is denoted by \( \% \), i.e., \( x_1 \% x_2 = x_1 - x_2 \lfloor \frac{x_1}{x_2} \rfloor \) with \( \lfloor x \rfloor \) denoting the maximum integer that is smaller than \( x \). The symbol \( j \) is used to represent \( \sqrt{-1}. \) Finally, we define the complex Gaussian distribution with mean \( \mu \) and variance \( \sigma^2 \) as \( \mathcal{CN}(\mu, \sigma^2) \).

II. SYSTEM MODEL AND TRANSMISSION PROTOCOL

A. System Model

As shown Fig. 1, we consider an IRS-aided time-division duplexing (TDD) system where a single-antenna AP communicates with a single-antenna user via an IRS with \( N \) reflecting elements. The CSI is obtained via uplink pilot transmission in the considered system with the assumption of channel reciprocity. Let \( g \in \mathbb{C}^{N \times 1} \), \( h_r \in \mathbb{C}^{N \times 1} \) and \( h_d \) denote the baseband equivalent IRS-AP, user-IRS and user-AP channels, respectively. Assume that the channel statistics will stay unchanged for a long period of time called a super frame, which includes a number of frames. Each frame is divided into \( T \) time intervals and each time interval further consists of \( \tau \) time slots, as illustrated in Fig. 2. It is assumed that all the channels remain approximately constant in each time interval and vary from the current time interval to the next. Due to the insufficient angular spread of the scattering environment and closely spaced antennas/reflecting elements [22], [23], both line-of-sight (LoS) and non-LoS (NLoS) components may exist in practical channels. During one super frame, the LoS components are assumed to stay unchanged [24], [25], such that the IRS-AP, user-IRS and user-AP channels in the \( t \)-th time interval can be respectively modeled as [8], [10], [11], [16], i.e.,

\[
g(t) = \sqrt{\frac{l_{IA}/\beta}{1 + \beta}} g_{\text{LoS}} + \sqrt{\frac{l_{IA}}{1 + \beta}} g_{\text{NLoS}}(t),
\]

\[
h_r(t) = \sqrt{\frac{l_{UI}/\beta_{UI}}{1 + \beta_{UI}}} h_{r\text{LoS}} + \sqrt{\frac{l_{UI}}{1 + \beta_{UI}}} h_{r\text{NLoS}}(t),
\]

\[
h_d(t) = \sqrt{\frac{l_{UA}/\beta_{UA}}{1 + \beta_{UA}}} h_{d\text{LoS}} + \sqrt{\frac{l_{UA}}{1 + \beta_{UA}}} h_{d\text{NLoS}}(t),
\]

where \( g_{\text{LoS}}, h_{r\text{LoS}} \) and \( h_{d\text{LoS}} \) represent the LoS components; \( g_{\text{NLoS}}(t), h_{r\text{NLoS}}(t) \) and \( h_{d\text{NLoS}}(t) \) denote the NLoS components; \( \beta_{IA}, \beta_{UI} \) and \( \beta_{UA} \) are the Rician factors of the IRS-AP, user-IRS and user-AP channels, respectively. \( l_{IA}, l_{UI} \) and \( l_{UA} \) are the corresponding path losses, which are given by \( l_{IA} = l_0 (d_{IA}/d_0)^{-\gamma_{IA}}, l_{UI} = l_0 (d_{UI}/d_0)^{-\gamma_{UI}} \) and \( l_{UA} = l_0 (d_{UA}/d_0)^{-\gamma_{UA}} \), respectively, where \( d_0 \) represents the reference distance and \( l_0 \) is the path loss at the reference distance, \( d_{IA}, d_{UI} \) and \( d_{UA} \) denote the link distances from the IRS to the AP, from the user to the IRS, and from the user to the AP, respectively; \( \gamma_{IA}, \gamma_{UI} \) and \( \gamma_{UA} \) denote the path-loss
exponents. Equivalently, we have $g(t) \sim \mathcal{C}\mathcal{N}(\bar{g}, C_{IA})$, $h_r(t) \sim \mathcal{C}\mathcal{N}(\bar{h}_r, C_{UI})$ and $h_d(t) \sim \mathcal{C}\mathcal{N}(\bar{h}_d, C_{UA})$, where $\bar{g} = \mathbb{E}[g(t)], \bar{h}_r = \mathbb{E}[h_r(t)], \bar{h}_d = \mathbb{E}[h_d(t)]$, $C_{IA} = \frac{\bar{h}_{d1}}{1+\alpha_{ua}} \mathbb{E}[g_{\text{NLoS}}(g_{\text{NLoS}})^H]$, $C_{UI} = \frac{\bar{h}_{d1}}{1+\alpha_{ui}} \mathbb{E}[h_{\text{NLoS}}(h_{\text{NLoS}})^H]$ and $C_{UA} = \frac{\bar{h}_{d1}}{1+\alpha_{ua}} \mathbb{E}[h_{\text{NLoS}} h_{\text{NLoS}}^H]$. Due to the mobility of the AP and/or user, the channels are usually time varying and exhibit a high degree of temporal correlation [26], [27]. Therefore, we employ the independent stationary steady-state Gauss-Markov process [19], [28] to model the temporal evolution of the channel parameters, which are given as follows:

\begin{align}
    g(t) &= \sqrt{1-\alpha_{ua}}(g(t-1) - \bar{g}) + \sqrt{\alpha_{ua}} u_{IA}(t) + \bar{g}, \\
    h_r(t) &= \sqrt{1-\alpha_{ui}}(h_r(t-1) - \bar{h}_r) + \sqrt{\alpha_{ui}} u_{UI}(t) + \bar{h}_r, \\
    h_d(t) &= \sqrt{1-\alpha_{ua}}(h_d(t-1) - \bar{h}_d) + \sqrt{\alpha_{ua}} u_{UA}(t) + \bar{h}_d,
\end{align}

where $u_{IA}(t) \sim \mathcal{C}\mathcal{N}(0, \sigma_{IA}^2 I)$, $u_{UI}(t) \sim \mathcal{C}\mathcal{N}(0, \sigma_{UI}^2 I)$ and $u_{UA}(t) \sim \mathcal{C}\mathcal{N}(0, \sigma_{UA}^2 I)$ are the perturbation terms in the IRS-AP, user-IRS and user-AP channels, respectively; $\alpha_{IA}$, $\alpha_{UI}$ and $\alpha_{UA}$ represent the temporal correlation coefficients of the IRS-AP, user-IRS and user-AP channels, respectively. Note that the channel realizations $g(t-1)$, $h_r(t-1)$ and $h_d(t-1)$ are statistically independent of $u_{IA}(t)$, $u_{UI}(t)$ and $u_{UA}(t)$. In the $t$-th time interval, the received signal at the AP can be expressed as

$$y(t) = \sqrt{p_h} h_r^H(t) \Theta(t) g(t) + h_d(t) s(t) + z(t),$$

where $s(t)$ denotes the transmit symbol, $p$ represents the transmit power, $z(t)$ denotes the additive white Gaussian noise (AWGN) with zero-mean and variance $\sigma^2$, $\Theta(t)$ represents the reflection pattern at the IRS. Define the user-AP equivalent channel as

$$h(t) \triangleq [h_d(t); \text{diag}(h_r^H(t))] g(t).$$

Then, the received signal in (5) can be equivalently rewritten as

$$y(t) = \sqrt{p} v^H(t) h(t) s(t) + z(t),$$

where $v(t) \triangleq [1, \theta_1(t), \cdots, \theta_N(t)]^H$ with $\theta_n(t)$ denoting the $n$-th diagonal element of the reflection pattern $\Theta(t)$. Notice that when the number of IRS reflecting elements is large, the acquisition of the instantaneous CSI may cause considerable channel training/estimation overhead, which leads to reduced user transmission rate due to the limited time left for data transmission. To address this issue, we propose to exploit the temporal correlation of the channel in this work and present a 2SCTP scheme, where a KF-based channel tracking algorithm and a DL-enabled channel prediction algorithm are integrated to reduce the channel training overhead and thus improve the transmission rate.

### B. Transmission Protocol

The proposed two-stage transmission protocol is shown in Fig. 2. As can be seen, in the first stage that contains $T_1$ time intervals, each time interval is divided into two phases, i.e., the channel training phase (consists of $\tau_1$ time slots) and the data transmission phase (consists of $\tau_2 = T - T_1$ time slots). More specifically, in the first $\tau_1$ time slots of the first stage, the user sends pilot symbols that are known at the AP such that the AP can track the channel according to the received pilot signals, then in the remaining $T - \tau_1$ time slots of each time interval, the user sends data signals, which can be detected based on the tracked channels in the channel training phase. In the second stage which includes $T_2 = T - T_1$ time intervals, there is no channel training phase and the channels are predicted such that all the time slots in these time intervals are utilized for data transmission.

As compared with the existing channel estimation methods, the proposed 2SCTP scheme is able to reduce the channel training overhead in the following two ways:

- In the first stage, we exploit the temporal correlation of the channels to reduce the channel training overhead in each time interval, thus $\tau_1$ can be much smaller than $N + 1$.
- In the second stage, we predict the channels based on the observations and statistical channel information extracted.
from CSI (i.e., the prediction covariance matrix) collected in the first stage, hence the channel training overhead can be completely removed.

It is noteworthy that the proposed 2SCTP scheme can be regarded as a data transmission scheme with uniformly distributed sparse training symbols, where low-dimensional pilot vectors are sent in the channel tracking stages and no pilot is required in the channel prediction stages. In other words, the training symbols are arranged in a comb structure.

Based on the proposed transmission protocol, we propose the 2SCTP scheme, whose details are given in the following two sections. It is important to mention that at the beginning of each super frame, i.e., when the channel statistics change, we need to rerun the proposed 2SCTP scheme to track the channel again. In the rest of this paper, we focus on the CTP algorithm design within one super frame. Besides, the temporal correlation coefficients $\alpha_{UA}$, $\alpha_{UI}$ and $\alpha_{IA}$, the variances of the perturbation terms $\sigma_{UA}^2$, $\sigma_{UI}^2$ and $\sigma_{IA}^2$, and the channel means $\bar{g}$, $\bar{h}$, and $\tilde{h}$ are regarded as prior knowledge of the existing works, e.g., [29] and [30]. Note that at the beginning of each super frame, we should allocate a certain number of time intervals to measure the new channel distribution and obtain these key parameters by efficient parameter estimation methods. For example, the expectation-maximization (EM) algorithm is a general approach for performing maximum likelihood estimation in the presence of latent variables, which can effectively approximate probability distributions and their parameters [31], [32]. In the estimation step, the Bayesian estimation (i.e., posterior distribution of channel) is calculated, and in the maximization step, the channel statistical parameters are optimized. By alternating between these two steps, the initial channel statistical parameters can be obtained in advance.

III. 2SCTP Scheme for the Special Case

In this section, we focus on the special case where the IRS-AP channel is assumed to change much slower than the user-IRS and user-AP channels, i.e., $\alpha_{IA} \ll \alpha_{UI}$ and $\alpha_{IA} \ll \alpha_{UA}$. This assumption is based on the scenarios where the IRS/AP are deployed on immovable walls or ceilings, and users are usually with high mobility. Furthermore, we assume that the user-AP, IRS-AP and user-IRS channels follow the Rayleigh channel model, extension to the more general Rician channel model will be discussed later. According to (4) and the definition of the user-AP equivalent channel in (6), $h(t)$ can be written as

$$
h(t) = [\sqrt{1 - \alpha_{UA}} \bar{h}_d(t - 1) + \sqrt{\alpha_{UA}} u_{UA}(t), \bar{h}_1(t), \cdots, \bar{h}_N(t)]^T,
$$

where $\tilde{h}_n(t), n \in [1, N]$ consists of four random variables and is given by

$$
\tilde{h}_n(t) = \sqrt{1 - \alpha_{IA}(1 - \alpha_{UI})}\left[\begin{array}{c} g(t - 1) \end{array}\right]_n h_r(t - 1), n + \sqrt{\alpha_{IA}(1 - \alpha_{UI})}\left[\begin{array}{c} u_{IA}(t) \end{array}\right]_n h_r(t - 1) + \sqrt{\alpha_{UI}(1 - \alpha_{IA}) g(t - 1), n u_{UI}(t), n + \sqrt{\alpha_{UA}(1 - \alpha_{IA}) u_{UA}(t), n u_{UI}(t), n}.
$$

Moreover, due to the fact that $\alpha_{IA} \ll \alpha_{UI}$ and $\alpha_{IA} \ll \alpha_{UA}$, there is $\tilde{h}_n(t) \approx \sqrt{1 - \alpha_{UA}}[g_r(t - 1)]_n h_r(t - 1) + \sqrt{\alpha_{UA}}[g_r(t - 1)]_n u_{UI}(t)$. Therefore, the user-AP equivalent channel $h(t)$ can be modeled as

$$
h(t) = A_h h(t - 1) + B_h u(t),
$$

where

$$
A_h = \left[ \begin{array}{c} \sqrt{1 - \alpha_{UA}} \quad 0_{1 \times N} \end{array} \right],
B_h = \left[ \begin{array}{c} \sqrt{\alpha_{UA}} \quad 0_{1 \times N} \end{array} \right],
$$

$u(t) \sim \mathcal{CN}(0, C_h)$ and $C_h$ is given by

$$
C_h = \left[ \begin{array}{cc} \sigma_{UA}^2 & \sigma_{UA}^2 \sigma_{UI} \text{diag} \left[ gg^H \right] \end{array} \right].
$$

In this special case, although the reflected channel $g^H(t)\text{diag} \left[ h_r(t) \right]$ is related with the IRS-AP and user-IRS channels, it can be assumed to change according to one temporal correlation coefficient $\alpha_{UI}$ since $\alpha_{IA} \ll \alpha_{UI}$. At one extreme, if $\alpha_{UA} = \alpha_{UI} = 0$, the reflected channel parameters are totally correlated, (i.e., $h(t) = h(t - 1)$), while at the other extreme when $\alpha_{UA} = \alpha_{UI} = 1$, the reflected channel coefficients evolve according to an uncorrelated Gaussian random process.

A. KF-Based Channel Tracking in Both Stages

As a well-known and powerful variable estimation method, the KF method has been widely applied in time series analysis, such as signal processing and econometrics [33], [34]. The KF method keeps track of the estimated state of the system and the uncertainty of the estimate, and only the estimated state from the previous time step and the current measurement are required to obtain the current state estimate and the corresponding uncertainty. Inspired by the superiority of the KF method in time series processing, we propose a KF-based channel tracking algorithm to estimate the time-varying channels for the considered special case with limited channel training overhead. Notice that the proposed channel tracking algorithm is employed in both stages.

First of all, let us introduce the problem setting for a general KF method. The KF model assumes that the state of a system at time step $t$ evolved from the prior state at time step $t - 1$ according to the linear state transition equation $x_t = F_t x_{t-1} + w_t$, where $F_t$ is the state transition model, the state variable $x_t$ and state noise variable $w_t$ follow independent complex Gaussian distributions. Then, the observation at time step $t$ is obtained according to the linear observation equation $z_t = H_t x_{t-1} + r_t$, where $H_t$ is the measurement matrix, the observation variable $z_t$ and observation noise variable $r_t$ also follow complex Gaussian distributions. The KF method aims to predict the state variable by utilizing a series of observations.

To apply the KF method to the considered channel tracking problem, we can regard (10) as the state transition equation. By assuming that the pilot symbol is set as $s = 1$ without loss
of generality, the observation functions in the first and second stages are respectively given by
\begin{equation}
    y(t) = \sqrt{V(t)}h(t) + z(t), \quad t = 1, \cdots, T_1, \tag{13}
\end{equation}
\begin{equation}
    y_i(t) = \sqrt{V(t)}\tilde{h}_i(t) + z(t), \quad t = T_1 + 1, \cdots, T, \tag{14}
\end{equation}
where the received signal \( y(t) \) and \( y_i(t) \) are vectors with \( \mathbb{C}^{\tau_i \times 1} \) and \( \mathbb{C}^{\tau_i \times 1} \), respectively. The measurement matrix \( V(t) \) is defined as the received signal if we imagine the virtual observation \( y_i(t) \) (related to the second stage or \( \tau_i \) time slot of the \( t \)-th time interval). Note that the virtual observation is used when we imagine the received signal if we imagine the virtual observation \( y_i(t) \). The measurement matrix \( V(t) \) is defined as the received signal if we imagine the virtual observation \( y_i(t) \). The virtual observation \( y_i(t) \) serves as the measured and virtual observations, respectively, \( V(t) \) in the first stage or \( V(t) \) in the second stage. Furthermore, the observation functions in the first and second stages are both regarded as a time series processing task. Recently, DL techniques have been used to solve such problems. However, in this paper, we focus on the channel tracking problem.

In practice, the tracking algorithm and the required channel training overhead.

In the prediction step, the estimate of the current state \( \hat{h}(t) \) is obtained by directly applying Algorithm 1, after which we can acquire the refined state estimate in the \( t \)-th time interval (also named as correction), which will be introduced later, and the estimation covariance matrix \( M(t) = E[(h(t) - h_{KF}(t - 1))(h(t) - h_{KF}(t - 1))^H] \) denotes the estimation covariance matrix.

In the update step, we first update the Kalman gain \( G_{KF}(t) \) as
\begin{equation}
    G_{KF}(t) = M(t)V^H(t)(V(t)M(t)V^H(t) + \sigma^2 I)^{-1}. \tag{20}
\end{equation}
Then, multiplying the innovation \( \tilde{y}(t) \) by the Kalman gain \( G_{KF}(t) \) and combining \( \tilde{y}(t) \) with the state estimate \( \hat{h}(t) \), the correction \( h_{KF}(t) \) can be obtained via
\begin{equation}
    h_{KF}(t) = \mathbb{E}[h(t)|y_{t-1}] + \mathbb{E}[h(t)|\tilde{y}(t)]
    = \hat{h}(t) + G_{KF}(t)\tilde{y}(t). \tag{21}
\end{equation}
Finally, the estimation covariance matrix \( M(t) \) required to calculate the prediction covariance matrix \( M(t) \) in (20), is updated by
\begin{equation}
    M(t) = \mathbb{E}[(h(t) - h_{KF}(t - 1))(h(t) - h_{KF}(t - 1))^H] = M(t) - G_{KF}(t)V^H(t)M(t). \tag{22}
\end{equation}
To summarize, the proposed KF-based channel tracking algorithm is shown in Algorithm 1, where we set \( h_{KF}(0) = 0 \) and \( M_{KF} = \mathbb{E}[h(t)h^H(t)] \) to initialize the KF iterations. Besides, Algorithm 1 can be readily extended to the Rician channel model, which has LoS component with mean \( \bar{h} \). By substituting \( h(t) = h(t) + \bar{h} \) into (7), we obtain
\begin{equation}
    \hat{h}(t) = A_h\bar{h} + B_hu(t), \tag{23}
\end{equation}
where \( \bar{h}(t) \) follows the Rayleigh channel model. Regarding (23) as the state transition equation, the estimation of the state variable in the \( t \)-th time interval, i.e., \( h_{KF}(t) \), can be obtained by directly applying Algorithm 1, after which we can acquire the refined state estimate in the \( t \)-th time interval as \( h_{KF}(t) = h_{KF}(t) + \bar{h} \). This is the main difference in the proposed 2SCPT scheme when considering Rayleigh and Rician channel models. Furthermore, it is noteworthy that under more practical 3GPP channel data or measured channel data, the proposed 2SCPT scheme may still be applicable if these channel data can be approximately formulated as (10). Further investigation into more general time-varying channel models is left for future work.

### B. DL-Enabled Channel Prediction in the Second Stage
Solving the channel prediction problem can be interpreted as a time series processing task. Recently, DL techniques have been used to solve such problems. However, in this paper, we focus on the channel tracking problem.

3Note that the user-AP and user-IRS-AP channels can also be tracked independently using two KFs, where we switch off the reflection elements first to obtain the observations of the direct channels, and then switch on the reflection elements and cancel the interference caused by the direct channels to obtain the observations of the reflected channels. However, the performance of such a benchmark is inferior to that of the proposed algorithm due to the interference of the estimated direct channels. Moreover, the proposed algorithm is easier to implement because we do not need to consider the time resource allocation problem for tracking the direct and reflected channels. Furthermore, the observations acquisition process of the proposed algorithm is more convenient and its formulation is more concise.
Algorithm 1 KF-Based Channel Tracking Algorithm

**Input:** covariance matrix $C_h = \mathbb{E}[hh^H]$ and observations $\mathbf{y}(t)$

**Output:** Estimated channel vectors $\mathbf{h}_\text{KF}(t)$

1. **Initialization:** $t = 0$, $\mathbf{h}_\text{KF}(-1) = 0$, $\mathbf{M}_\text{MF} = \mathbb{E}[hh^H]
2. while $t > 0$ do
3. 3. **Prediction step:**
   3.1. Calculate the prediction $\hat{\mathbf{h}}(t)$ according to (18).
   3.2. Calculate the prediction covariance matrix $\mathbf{M}(t)$ according to (19).
4. 4. **Update step:**
   4.1. Update the Kalman gain vector $\mathbf{G}_\text{KF}(t)$ according to (20).
   4.2. Calculate the correction $\mathbf{h}_\text{KF}(t)$ according to (21).
   4.3. Calculate the estimation covariance matrix $\mathbf{M}_\text{KF}(t)$ according to (22).
5. $t = t + 1,$
6. **end while**

Different from classical deep learning applications, e.g., computer vision and natural language processing, complex data format is usually considered in wireless communications. To handle the complex input data, the proposed OB-LSTM network is designed to contain two sub-networks with the same structure, which are called the R sub-network and the I sub-network, respectively. In the following, we only focus on the structure of the R sub-network since the I sub-network can be similarly designed. Specifically, each sub-network contains one input fully-connected layer, $K$ LSTM layers and one output fully-connected layer. The input fully-connected layer is designed to augment the input dimension of the LSTM units, which can help to extract the high-dimensional features of the input data and improve the prediction performance. Let $\epsilon$ represent a scaling factor, then the output of the input fully-connected layer, denoted by $x_h(t) \in \mathbb{R}^{D_1 \times 1}$, can be expressed as

$$x_h(t) = \text{ReLU}(\mathbf{W}_\epsilon x(t) + \mathbf{b}_\epsilon),$$

where $\mathbf{W}_\epsilon \in \mathbb{C}^{D_1 \times D}$ and $\mathbf{b}_\epsilon \in \mathbb{C}^{D_1 \times 1}$ denote the weight and bias (i.e., the learnable network parameters), respectively, and ReLU($\cdot$) is employed as the activation function. Note that the scaling factor $\epsilon$ is an important hyper-parameter that balances the prediction performance and network complexity, and it should be properly chosen according to the system requirement, and it should be properly chosen according to the system requirement. We will investigate the impact of $\epsilon$ in Section V. The LSTM layers are the core of the proposed network, since they are developed to deal with the vanishing gradient problem that might be encountered when training traditional RNNs and are applicable to tasks such as classifying, processing and predicting based on time series data [39]. In this work, one LSTM layer consists of $L_1$ cascading LSTM units. Each LSTM unit is composed of a cell, an input gate, an output gate and a forget gate, where the cell remembers information over arbitrary time interval and the three gates

$$\{\mathbf{y}(t - L_1 + 1), \cdots, \mathbf{y}(t)\}$$

into the OB-LSTM network and train it to predict the next $L_P$ virtual observations, i.e., $\{\mathbf{y}_I(t + 1), \cdots, \mathbf{y}_I(t + L_P)\}$. Since the observations are related to both the unknown channels and the known reflection matrix $\mathbf{V}(t)$, we also treat $\mathbf{V}(t)$ as the part of the network input. Besides, the historical observations are pre-processed by normalization to improve the performance of the multilayer perceptron models in the proposed OB-LSTM network. After data normalization, the overall input data sequence is given by $\{\mathbf{x}(1), \cdots, \mathbf{x}(t), \cdots, \mathbf{x}(L_1)\}$, where $\mathbf{x}(t) = [\mathbf{V}_I(\mathbf{y}_I(t), \mathbf{V}_I(\mathbf{y}_I(t), \cdots, \mathbf{V}_I(\mathbf{y}_I(t))))]^{T} \in \mathbb{C}^{D_1 \times 1}$ with $D_1 \equiv \tau_1(N + 1)$ and $\mathbf{V}_I(\cdot)$ denoting the dimension of the input and the normalization function, respectively. Accordingly, we collect a set of measured observations and construct the training data-label sample set as $\{\{\mathbf{x}(j + 1), \cdots, \mathbf{x}(j + L_1)\}, \{\mathbf{y}(j + L_1 + 1), \cdots, \mathbf{y}(j + L_1 + L_P)\}\}_{j=0}^{J-1}$.

The detailed structure of the proposed OB-LSTM network is provided as follows. First, let $L_1$ and $L_P$ denote the lengths of the input observations and virtual observations, respectively, where $L_1 \leq T_1$ and $L_P \leq T_2$. Then, in the $t$-th time interval for $t = T_1, \cdots, T$, we feed $L_1$ observations

4Alternatively, similar to [38], we can combine the real and imaginary parts of the input observations into one vector as the network input, such that no sub-networks are needed. However, as the real and imaginary parts of the channels are generated independently (see (1), (2) and (3)), the proposed sub-network structure can achieve similar performance with less network parameters, thus is more favorable here.
The overall procedure of the proposed channel prediction method is depicted in Fig. 4, where the structure of the proposed OB-LSTM network is also illustrated.

In this work, we adopt the supervised learning strategy and construct the following loss function which is based on the $\ell_2$-norm of the difference between the future observations and their estimates

$$
L_2(\Theta_R, \Theta_I) = \frac{1}{T} \sum_{j=0}^{T-1} \sum_{k=1}^{P} \| \hat{y}(j+L_1+k) - y(j+L_1+k) \|^2_2.
$$

(28)

All the network parameters in the proposed OB-LSTM network are updated by minimizing this loss function using the Adam optimizer [40]. It is noteworthy that the parameters of the OB-LSTM network are learned through offline training. When the channel statistics change, the user should send pilot symbols in the first several frames of a super frame, as such a set of measured observations can be obtained, based on which we can further finetune the network parameters to alleviate the performance loss caused by model mismatch.

When the training is finished, the OB-LSTM network can be flexibly integrated into the proposed 2SCPT scheme. We take the case of $T_1 = 6$ and $T_2 = 6$ as a toy example to elaborate this. The most simple strategy (also called Strategy A) is to let $L_1 = T_1 = 6$ and $L_P = T_2 = 6$, then 6 measured observations, i.e., $y(1), \cdots, y(6)$, collected in the first stage can be directly input into the OB-LSTM network to obtain 6 virtual observations, i.e., $\hat{y}(1), \cdots, \hat{y}(12)$. However, in this strategy, if the value of $T_1$ or $T_2$ is large, the computational complexity of the OB-LSTM network will be high since the input/output dimensions and the number of neurons/layers are large. Thus, it would be difficult to train such a network with good prediction performance. Another alternative strategy (also called Strategy B) is to set $L_P < L_I \leq T_1$, e.g., $L_I = 6$ and $L_P = 1$, such that the virtual observations can be reused as the network input. Specifically, in the first time interval of the second stage, the OB-LSTM network predicts the next virtual observation $\hat{y}(7)$ based on the last 6 measured observations $y(1), \cdots, y(6)$ collected in the first stage, while in the subsequent time intervals, we can construct the input data using 1 virtual observation and the last 5 measured observations, i.e., $y(2), \cdots, y(6), \hat{y}(7)$, to obtain $y(8)$. Note that the hyper-parameters of the OB-LSTM network, e.g., $L_I, L_P$ and $c$, and the employed strategies should be carefully chosen to balance the prediction performance, the computational complexity of the network and the required channel training overhead. Investigation into the impacts of different hyper-parameters and strategies will be shown in Section V. Furthermore, in practice, we can employ other promising deep learning techniques, such as transfer learning and meta learning, to reduce the complexity of offline training.

Remark 1: When we consider the case where the AP is deployed with multiple antennas, the user-AP equivalent channel matrices, i.e., the reflected channel matrices and the direct channel vectors, are required to be tracked and predicted based on the observations. By vectorizing the channel matrices and constructing the observation functions as in (13) and (14),

![Fig. 3. The structure of one LSTM unit.](image-url)
the proposed 2SCTP scheme can be readily extended to this more general case.

IV. 2SCTP SCHEME FOR THE GENERAL CASE

In this section, we extend the proposed 2SCTP scheme to the more general case where the IRS-AP, user-IRS and user-AP channels are all assumed to be time-varying. In this case, we can simplify the state transition equation (8) as

$$\mathbf{h}(t) = \mathbf{A}_g \mathbf{h}(t - 1) + \mathbf{u}_g(t),$$

(29)

where

$$\mathbf{A}_g = \begin{bmatrix} \sqrt{1 - \alpha_{UA}} & 0_{1\times N} \\ N^{-1} \{1 - \alpha_{UA}\} \left[1 - \alpha_{UI}\right] & \sqrt{1 - \alpha_{UI}} \right],$$

(30)

$$\mathbf{u}_g(t)$$ represents the composite noise variable that satisfies $\mathbf{u}_g(t) \sim \mathbf{u}_{UA}(t)$ and $\mathbf{u}_g(t) \sim \mathbf{u}_{UI}(t)$, where $\mathbf{u}_{UA}(t)$ and $\mathbf{u}_{UI}(t)$ are non-Gaussian vectors. This is quite different from the state transition equation (10) discussed in the special case, and the KF-based channel tracking algorithm cannot be directly applied to this general case. Note that for the non-Gaussian scenario, the particle filter (PF) method is widely-used [41], such as in target tracking, signal processing and automatic control, etc. The PF method uses a set of particles, i.e., observations, to represent the posterior distribution of the state variable which follows an arbitrary distribution. However, collecting these particles can be quite time-consuming, thus applying the PF method for the general case can be very inefficient.

To tackle this difficulty, we propose in this paper a GKF-based channel tracking algorithm for the general case. Our idea is inspired by the extended KF (EKF) and unscented KF (UKF) methods [42], both of which are designed for non-linear systems. Their difference is that the EKF method employs multivariate Taylor series expansions to linearize the state transition equation at the working point, while the UKF method tries to approximate the probability density function of the output of the non-linear function included in the system by a number of deterministic sampling points which represent the underlying distribution as a Gaussian distribution. Then, the original KF method can work on these modified or linearized systems. In the considered problem, our state transition equation and observation function are both linear, but the composite noise variable $\mathbf{u}_g(t)$ in the state transition equation is not Gaussian. Therefore, some necessary modifications should be made on $\mathbf{u}_g(t)$ such that the resulting problem can be solved by the KF method. Specifically, to tackle the difficulty caused by the non-Gaussian noise variable $\mathbf{u}_g(t)$, we propose to use a complex Gaussian distribution to approximate the distribution of $\mathbf{u}_g(t)$ in each time interval. Then, the predicted state variable $\mathbf{h}(t)$, obtained from the linear state transition equation (29), also follows a complex Gaussian distribution. With the help of such an approximation, we can apply the KF method to address the resulting channel tracking problem. In the following, we will first present the proposed approximation method and then introduce the GKF-based channel tracking algorithm.

A. Channel Distribution Analysis

In this subsection, we aim to analyse the distribution of the noise variable $\mathbf{u}_g(t)$ included in the state function (29).

The proposed GKF-based channel tracking algorithm can be extended to handle other channel models as well as long as the state equation is linear.
First, it is readily seen that $[u_g(t)]_n$ follows the complex Gaussian distribution, i.e., $[u_g(t)]_1 \sim \mathcal{C}(0, \alpha_{UA}\beta_{UA})$. The other elements in $u_g(t)$ can be expressed as the sum of three random variables, i.e.,

$$[u_g(t)]_n = [u_{g,1}(t)]_n + [u_{g,2}(t)]_n + [u_{g,3}(t)]_n,$$

(31)

where $[u_{g,1}(t)]_n \triangleq \sqrt{\alpha_{UA}(1 - \alpha_{UI})}|h_r(t - 1)(t)|_n[u_A(t)]_n$, $[u_{g,2}(t)]_n \triangleq \sqrt{\alpha_{UI}(1 - \alpha_{UA})}|g(t - 1)|_n[u_A(t)]_n$, and $[u_{g,3}(t)]_n \triangleq \sqrt{\alpha_{IA} \alpha_{UA} |u_A(t)|_n [u_g(t)]_n}$. We can see that $[u_{g,1}(t)]_n$ and $[u_{g,2}(t)]_n$ both follow complex Gaussian distributions, i.e., $[u_{g,1}(t)]_n \sim \mathcal{C}(0, \alpha_{UA}(1 - \alpha_{UI})|h_r|^{2} |[h_r(t - 1)]_n|^{2}), [u_{g,2}(t)]_n \sim \mathcal{C}(0, \alpha_{UI}(1 - \alpha_{UA})|g_r|^{2} |g(t - 1)|_n|^{2} |\sigma_{IA}^{2})), while the third random variable $[u_{g,3}(t)]_n$ is the product of two independent complex Gaussian random variables, i.e., $\sqrt{\alpha_{IA} \alpha_{UA} |u_A(t)|_n [u_g(t)]_n}$ and $\sqrt{\alpha_{UI} |u_A(t)|_n [u_g(t)]_n}$.

Then, we focus on the analysis of $[u_{g,3}(t)]_n$. For clarity, we refer to the distribution of this type of random variables as the product complex Gaussian distribution in the following. Note that the product of two real Gaussian distributions, named as product real Gaussian distribution, has been studied before in [43], [44], where the authors proved that it can be expressed in terms of Meijer G-functions. Consider a complex random variable $X$ that satisfies $X = X_1 X_2$, where $X_1, X_2 \sim \mathcal{C}(0, \frac{1}{2})$. Since it is difficult to directly analyse the distribution of $X$, we first investigate it via Monte-Carlo simulation using $10^5$ samples. Fig 5 demonstrates the marginal probability distributions of the real and imaginary parts of $X$, $X_1$, and $X_2$. It is observed that the joint (or marginal) distribution resembles a shaper and slimmer complex Gaussian (or Gaussian) distribution, which inspires us to approximate the product complex Gaussian distribution by a simple complex Gaussian distribution with the same mean and variance. However, since $[u_{g,3}(t)]_n$ is in essence a component of the system noise variable $[u_g(t)]_n$, we cannot sample this random variable independently and calculate its mean and variance. Hence, the ideas employed in the PF and UKF methods, i.e., sampling a set of particles to represent the posterior distribution of the state variable, cannot be applied here to approximate the distribution of $[u_{g,3}(t)]_n$. To overcome this challenge, we resort to the following result about the mean and variance of the random variable $X$, i.e., $\mu_P$ and $\sigma_P^2$,

$$\mu_P = \mathbb{E}[X_1] \mathbb{E}[X_2] = 0,$n

$$\sigma_P^2 = \mathbb{E}[XX^*] = \mathbb{E}[X_1 X_2 X_2^* X_1^*] = \mathbb{E}[X_1 X_1^* X_2 X_2^*] \approx \mathbb{E}[X_1 X_1^*] \mathbb{E}[X_2 X_2^*] = \sigma_1^2 \sigma_2^2,$$

(32)

As such, the mean and variance of $[u_{g,3}(t)]_n$ can be directly obtained from those of $\sqrt{\alpha_{IA} |u_A(t)|_n} \sqrt{\alpha_{UI} |u_A(t)|_n}$. Then, based on (32) and the numerical results shown in Fig. 5, a random product complex Gaussian variable $X$ with mean $\mu_P$ and variance $\sigma_P^2$ can be approximated by a complex Gaussian variable $\tilde{X} \sim \mathcal{C}(\mu_P, \sigma_P^2)$. As can be seen, although the distribution of $\tilde{X}$ is similar to that of $X$, it is still not sharp enough as compared to the true distribution. However, since only one component in $[u_g(t)]_n$, i.e., $[u_{g,3}(t)]_n$, requires to be approximated and the scaling coefficient of $[u_{g,3}(t)]_n$, i.e., $\sqrt{\alpha_{IA} \alpha_{UA}}$, is smaller than those of $[u_{g,1}(t)]_n$ and $[u_{g,2}(t)]_n$, i.e., $\sqrt{\alpha_{IA}(1 - \alpha_{UI})}$ and $\sqrt{(1 - \alpha_{IA}) \alpha_{UA}}$, we can infer that the negative effect caused by the approximation error should be limited. In Section V, we will show that the proposed GKF-based channel tracking algorithm can achieve good performance although under the employed approximation.

### B. Complex Gaussian Approximation

Next, we focus on approximating the distribution of $[u_{g}(t)]_n$ based on the result in Section IV-A. According to the analysis in the previous subsection, $[u_{g,3}(t)]_n$ can be approximated by $[u_{g,3}(t)]_n$ which satisfies $[u_{g,3}(t)]_n \sim \mathcal{C}(0, \alpha_{IA} \alpha_{UI} \sigma_{IA}^2 \sigma_{UI}^2)$, thus $[u_g(t)]_n$ can be approximated by $[u_g(t)]_n \triangleq [u_{g,1}(t)]_n + [u_{g,2}(t)]_n + [u_{g,3}(t)]_n \in [2, N + 1]$. It is noteworthy that the temporal correlation coefficients, i.e., $c_{IA}$, $c_{UI}$, $c_{UA}$, and the variances of $|u_A|_n$, $|u_U|_n$, and $|u_{UA}|_n$, i.e., $\sigma_{IA}^2$, $\sigma_{UI}^2$, $\sigma_{UA}^2$, are prior knowledge that are known. Since the sum of Gaussian distributions is still Gaussian, the covariance matrix of $u_g(t)$, i.e., $C_g(t)$, can be approximated by

$$\hat{C}_{g,1}(t) = \text{diag}(\alpha_{UA} \sigma_{UA}^2, \sigma_{g,1}(t), \cdots, \sigma_{g,N}(t)),$$

(33)

where

$$\sigma_{g,n}^2 = \alpha_{IA}(1 - \alpha_{UI})|h_r(t - 1)|_n|^{2} \sigma_{IA}^2 + \alpha_{UI}(1 - \alpha_{IA})|g(t - 1)|_n|^{2} \sigma_{UI}^2 + \alpha_{IA} \alpha_{UA} \sigma_{IA}^2 \sigma_{UI}^2,$$

(34)
This is referred to as type one complex Gaussian approximation (CGA-I) in the following. It can be seen that the amplitudes of $[h(t-1)_n]$ and $[g(t-1)_n]$ are required in order to calculate $\sigma_{g,n}^2(t)$, which is however difficult to obtain in the tracking process due to the passive nature of IRS, i.e., the CSI of the IRS-AP and user-IRS channels are difficult to obtain. To address this issue, we propose to approximate $\alpha_{IA}(1 - \alpha_{UA})|b_n(t-1)|^2\sigma_{g,n}^2(t) + \alpha_{UA}(1 - \alpha_{IA})|g(t-1)|^2\sigma_{U,n}^2$ using its lower bound, which is based on the following theorem.

**Theorem 1:** Assuming that $a = a_r + ja_i$ and $b = b_r + jb_i$ are two arbitrary complex scalars, then we have

$$|a|^2 + |b|^2 \geq |R(ab)| + |\Im(ab)|,$$

(35)

**Proof:** Please relegate to Appendix. 

Based on Theorem 1, we employ a random variable $[u_n(t)]_n \sim \mathcal{CN}(0, \sigma_{g,n}^2(t))$ to approximate $[u_n(t)]_n$, where $\sigma_{g,n}^2(t) \leq \text{Var}([u_n(t)]_n)$. Specifically, the value of $\sigma_{g,n}^2(t)$ is given by

$$\sigma_{g,n}^2(t) = \delta_1 \left( |R([h(t-1)]_n)| + |\Im([h(t-1)]_n)| \right) + \delta_2,$$

(36)

where $\delta_1 \triangleq \sqrt{\alpha_{IA}(1 - \alpha_{UA})|b(t-1)|^2\sigma_{g,n}^2(t) + \alpha_{UA}(1 - \alpha_{IA})|g(t-1)|^2\sigma_{U,n}^2}$ and $\delta_2 \triangleq \alpha_{IA}\sigma_{UA}^2/\alpha_{UA}\sigma_{U,n}^2$. Then, the covariance matrix $C_{g}(t)$ can be approximated by

$$C_{g,t}(t) = \begin{bmatrix} \alpha_{UA}\sigma_{UA}^2 & 0_{1\times N} \\ 0_{N\times 1} & \Sigma_{II} \end{bmatrix},$$

(37)

where $\Sigma_{II} = \delta_1 \text{diag}\left( |R([h(t-1)]_n)|, |\Im([h(t-1)]_n)| \right) + \delta_2 I_N$. Then, the estimation covariance matrix $C_{gl}(t)$ can be obtained by

$$\tilde{C}_g(t) = \begin{bmatrix} \alpha_{UA}\sigma_{UA}^2 & 0_{1\times N} \\ 0_{N\times 1} & \Sigma_{gl} \end{bmatrix},$$

(38)

where $\Sigma_{gl} = \delta_1 \text{diag}\left( |R([h(t-1)]_n)|, |\Im([h(t-1)]_n)| \right) + \delta_2 I_N$.

In the second stage of the proposed transmission protocol, virtual observations can be similarly obtained by the OB-LSTM network as in Section III, which serve as the input of Algorithm 2, then the channels in this stage can be estimated and no channel training overhead is required.

**Algorithm 2 GKF-Based Channel Tracking Algorithm**

**Input:** covariance matrix $C_{h} = \mathbb{E}[hh^H]$ and observations $\hat{y}(t)$

**Output:** Estimated channel vector $\hat{h}_{KF}(t)$

1: **Initialization:** $t = 0$, $\hat{h}_{KF}(-1) = 0$, $M_{MF} = \mathbb{E}[hh^H]$, $C_{gl}(0) = \mathbb{E}[hh^H]$

2: while $t \geq 1$

3: **Prediction step:**

4: Calculate the prediction according to (18),

5: Calculate the prediction covariance matrix according to $M(t) = \mathbb{E}\left[ (\hat{h}(t) - \hat{h}(t)) (\hat{h}(t) - \hat{h}(t))^H \right] = A_h M_{KF}(t-1) A_h^T + B_h C_{gl}(t) B_h^T$,

6: **Update step:**

7: Update the Kalman gain vector according to (20),

8: Update the correction according to (21),

9: Update the estimation covariance matrix according to (22),

10: Update the approximated covariance matrix of $u_g$, $C_{gl}(t+1)$, according to (38),

11: $t = t + 1$,

12: end while

**V. Simulation Results**

In this section, we present numerical results to validate the effectiveness of the proposed two-stage transmission protocol and the 2SCTP scheme. We consider an IRS-aided wireless communication system as shown in Fig. 6, where a three-dimensional coordinate system is assumed and the AP and IRS are located on the $x$-axis and $y$– $z$ plane, respectively. The reference antenna/element at the AP/IRS are located at $(3\text{ m}, 0, 0)$ and $(0, 50\text{ m}, 2\text{ m})$, and the user is located at $(2\text{ m}, 50\text{ m}, 0)$. In the simulations, the IRS is equipped with a $5 \times 7$ uniform rectangular array. The reference distance and the pass loss at the reference distance are set as $d_0 = 1\text{ m}$ and $l_0 = -30\text{ dB}$. According to [2], the IRS-AP, user-IRS and user-AP path-loss exponents are fixed to $\gamma^{UA} = 2.2$, $\gamma^{UI} = 2.2$ and $\gamma^{UA} = 3.6$, respectively, to model the scenario that the users suffer from severe signal attenuation in the direct link. The transmit power and noise variance are set to $p = 26\text{ dBm}$ and $\sigma^2 = -80\text{ dBm}$. Furthermore, the variances of the elements of $u_h, u_{UI}$ and $u_{UA}$ are respectively set to $\sigma^2_I = l_{II}, \sigma^2_{UA} = l_{UA}$ and $\sigma^2_{U,n} = l_{U,n}$. In all our simulations, the definitions of the normalized mean square error (NMSE) and average NMSE (ANMSE) in the $t$-th time interval.


dNote that in the KF method, the covariance matrix of the noise variable, i.e., $u_{h}(t)$, is used to measure the estimation error of the state variable, i.e., $h(t)$. Therefore, approximating $\text{Var}([u_h(t)]_n)$ using the lower bound $\sigma_{g,n}^2(t)$ will make the estimated variable more stable.

7If the number of IRS reflecting elements is large, the channel training overhead required to obtain the observations and the number of network parameters included in the OB-LSTM network will also be increased. In this case, we can employ the grouping and partition method [45] to achieve a good trade-off between channel tracking performance and channel training overhead/network complexity.
are given by $\text{NMSE}(t) = \frac{\|h_{KF}(t) - h(t)\|^2}{\|h(t)\|^2}$, $\text{ANMSE}(t) = \frac{1}{T} \sum_{i=1}^{T} \frac{\|h_{KF}(i) - h(i)\|^2}{\|h(i)\|^2}$.

### A. Special Case

In this subsection, we consider the special case where the IRS/AP are deployed on immovable walls/ceilings, i.e., the IRS-AP channel stays unchanged, and the users are with high mobility.

First, we investigate in Fig. 7 the effect of the temporal correlation coefficients on the NMSE performance of the KF-based channel tracking algorithm (i.e., Algorithm 1), where we fix $\alpha_{\text{IA}} = 0$ and set $\alpha_{\text{UI}} = \alpha_{\text{UA}} = \alpha \in [0.001, 0.01, 0.05, 0.1]$. The number of time slots in each time interval for transmitting pilot symbols is fixed to $\tau_1 = 6$, and we employ the random periodic reflection patterns whose elements are independently generated from a complex Gaussian distribution $CN(0, 1)$. As can be seen in Fig. 7, the performance of Algorithm 1 improves with the decreasing of $\alpha$. This is mainly due to the fact that when the temporal correlation is stronger, more information can be extracted from the received pilot symbols in previous time intervals to track the channels in the current time interval. In the following simulations, we fix $\alpha_{\text{UI}} = 0.01$ and $\alpha_{\text{UA}} = 0.01$ for better illustration [28].

Second, we investigate the channel tracking performance in the first stage. Fig. 8 shows the NMSE performance of Algorithm 1 with different numbers of time intervals. The number of time slots in each time interval for transmitting pilot symbols is assumed to be $\tau_1 \in \{2, 6, 10\}$. For comparison, we also provide the performance achieved by the channel estimation (CE) scheme proposed in [9] as the benchmark, i.e., in each time interval, $N + 1$ time slots are allocated to transmit pilot symbols and the channels are estimated by using the DFT reflection pattern and the MMSE channel estimation algorithm. As can be seen, the performance of Algorithm 1 improves with the increasing of $\tau_1$ and in the meantime, higher convergence speed can be achieved. For example, when $\tau_1 = 10$, the number of time intervals required by Algorithm 1 to achieve convergence is less than 20, while when $\tau_1 = 2$, this number increases to 20. Moreover, with larger $\tau_1$, Algorithm 1 shows more stable NMSE performance after convergence, e.g., the fluctuation of the NMSE curve when $\tau_1 = 10$ is smaller than that when $\tau_1 = 2$. Furthermore, as compared to the benchmark, Algorithm 1 can achieve much lower (150 time slots versus 525 time slots if 15 time intervals and $\tau_1 = 10$ are considered) channel training overhead with only minor NMSE performance loss.

In Fig. 9, we investigate the ANMSE performance of Algorithm 1 in the first stage when two different measurement matrices $V(t)$ are employed, i.e., the random measurement matrix and the DFT measurement matrix. In the random measurement matrix, each element of the reference matrix $Q$
in (15) is generated from a complex Gaussian distribution \( CN(0,1) \), while in the DFT measurement matrix, \( Q \) is an \((N+1)-\text{DFT matrix} \). \( \tau_1 \) is fixed to 6. One can see that as compared to using random measurement matrix, Algorithm 1 with the DFT measurement matrix exhibits better convergence. This is due to the fact that all the columns of the DFT measurement matrix are independent with each other, which forms a better sensing matrix.

Then, we investigate the channel prediction performance in the second stage. In our simulations, the number of LSTM layers in the OB-LSTM network is set to \( K = 4 \). The DFT measurement matrix is used due to its superiority demonstrated in Fig. 9. The number of training samples is set to \( 10^4 \), and the batchsize and learning rate are fixed to 10 and 0.0001, respectively. The training process is conducted offline on a Windows server with Intel Xeon Gold 6230 CPU and an Nvidia 2080Ti GPU, and the proposed networks are implemented in Python using the TensorFlow library with the Adam optimizer.

We present in Fig. 10 the convergence of the proposed OB-LSTM network during training, where the \( L_2 \) loss in (28) when feeding the validation data set into the OB-LSTM network is regarded as the performance metric. We consider four different configurations of the network structure, i.e., the scaling factor of the LSTM units \( \epsilon \) is set to be varying in \([1, 3, 5, 10] \). Besides, two different configurations of the lengths of the input observations and virtual observations are considered: (a) \( L_I = 6, L_P = 1 \), and (b) \( L_I = 6, L_P = 6 \).

From Fig. 10, we can observe that the \( L_2 \) loss achieved by the OB-LSTM network is able to converge in about 20000 iterations. Besides, the performance of the OB-LSTM network improves with the increasing of the scaling factor \( \epsilon \). However, larger \( \epsilon \) also means higher computational complexity. Therefore, the hyper-parameter \( \epsilon \) should be carefully chosen before training to strike a good balance between performance and computational complexity. Furthermore, by comparing Fig. 10 (a) and (b), we can see that with the same value of \( \epsilon \), the OB-LSTM network in case (a) outperforms that in case (b). This implies that the prediction performance of the OB-LSTM network deteriorates as the length of the virtual observations \( L_P \) increases, and there is a tradeoff between channel training overhead (i.e., prediction length) and prediction performance.

Next, we exhibit in TABLE I the NMSE of the virtual observations (OB-NMSE) achieved by the proposed OB-LSTM network, when the number of time intervals allocated for the two stages are fixed to \( T_1 = 6 \) and \( T_2 = 6 \), respectively. We consider Strategy A and Strategy B discussed in Section III-B which meet the requirement of \( T_1 = 6 \) and \( T_2 = 6 \). As shown in TABLE I, the OB-NMSE achieved by Strategy A is quite stable over all the considered time intervals, while that by Strategy B gradually deteriorates over time. Besides, Strategy B outperforms Strategy A in the first three time intervals, while Strategy A shows better performance in the other time intervals. The former is due to the non-ideal processing introduced by the OB-LSTM network when the output dimension \( L_P \) is larger (\( L_P = 6 \)) in Strategy A. The latter is because the OB-LSTM network in Strategy B outputs the following virtual observations one by one based on both measured and virtual observations, as such the prediction errors in the previous time intervals will deteriorate the prediction performance in the subsequent time intervals and thus Strategy B becomes worse than Strategy A in the later time intervals.

Then, in Fig. 11, we investigate the overall NMSE performance of the proposed 2SCTP scheme in two different scenarios.
scenarios, i.e., Scenario A and Scenario B. Specifically, in Scenario A, we set $T_1 = 6$, $T_2 = 6$, $L_I = 6$, $L_P = 6$ and Strategy $A$ is employed to generate the virtual observations; while in Scenario B, we set $T_1 = 6$, $T_2 = 3$, $L_I = 6$, $L_P = 1$ and Strategy $B$ is employed. For comparison, we consider a pure channel tracking (CT) scheme, where there is no channel prediction, i.e., all the time intervals are allocated to the first stage. Note that the ratio $T_2/T_1$ of Scenario A is larger than that of Scenario B, thus lower channel training overhead can be achieved by Scenario A. As shown in Fig. 11, in the first stage, the NMSE performance of the 2SCTP scheme in Scenario B is the same to that in Scenario A, however in the second stage, the performance in Scenario B is better than that in Scenario A and it is quite close to that of the benchmark. Therefore, for the proposed 2SCTP scheme, adopting the parameters in Scenario B can effectively reduce the channel training overhead with almost no performance loss.

At last, Table II compares the channel training overhead required by the 2SCTP scheme in Scenarios A and B when the number of time intervals is fixed to $T = 3600$. The amount of channel training overhead is measured by the number of time slots allocated for pilot transmission. The overheads required by the CE and CT schemes are regarded as benchmarks. It can be observed from Table II that in Scenario B, the 2SCTP scheme requires less channel training overhead than that in Scenario A, however the NMSE performance is worse in Scenario B (see Fig. 11). Moreover, the proposed 2SCTP scheme is able to reduce the channel training overhead significantly as compared to the CE and CT schemes. In particular, the overhead required in Scenario B is only 8.3% and 50.0% of those by the CE and CT schemes, respectively.

### B. General Case

In this subsection, we consider the general case and set the temporal correlation coefficients as $\alpha^{IA} = \alpha^{UI} = \alpha^{UA} = 0.01$.

First, we show in Fig. 12 the real distribution of $u_g$ and its approximations using CGA-I in (33) and CGA-II in (37). For simplicity, we take the marginal distribution of the real and imaginary parts of $|u_g|^2$ as an example. Besides, since the marginal distributions of $\Re(|u_g|^2)$ and $\Im(|u_g|^2)$ are the same, we only plot the distribution of $\Re(|u_g|^2)$ and its approximation in Fig. 12. It is observed that the distribution achieved by CGA-I is almost the same with the real distribution, while the distribution achieved by CGA-II has a smaller variance than the real distribution.

Then, Fig. 13 shows the performance achieved by the proposed GKF-based channel tracking algorithm (i.e., Algorithm 2), when two different approximated
covariance matrices of $u_g$ are used, i.e., $C_{g,1}$ given in (33) and $C_{gl}$ given in (38), which are obtained via CGA-I and CGA-II, respectively. It is noteworthy that $h_r(t-1)$ and $g(t-1)$ are needed to calculate $C_{gl}(t)$, while only $h_{KF}(t-1)$ is required to obtain $C_{g,1}(t)$. To employ $C_{g,1}$ in Algorithm 2, we replace (38) with (33) in step 8 to update the approximated covariance matrix of $u_g$, and $h_r(t-1)$ and $g(t-1)$ are assumed to be known.\footnote{Due to the passive nature of IRS, the IRS-AP channel $g$ and the user-IRS channel $h_r$ are difficult to be estimated. Here, to show the performance of the proposed CGA-I method, we assume that $g$ and $h_r$ are known.} The number of time slots allocated for the channel training phase is set to $\tau_1 = 6$. As shown in Fig. 13, by using $C_{gl}$ and $C_{g,1}$, Algorithm 2 can achieve similar NMSE performance, and the converged NMSE can reach 0.05 after about 10 time intervals.

Next, in Fig. 14, we investigate the NMSE performance achieved by Algorithm 2 with different values of $\tau_1$, i.e., $\tau_1 \in [2, 4, 6, 8]$. Similar to Fig. 8, we regard the NMSE performance of the CE scheme as the benchmark. It can be observed that when $\tau_1$ is small, i.e., $\tau_1 = 2$, Algorithm 2 almost fails to track the varying channel. When $\tau_1 = 4, 6, 8$, the NMSE performance achieved by Algorithm 2 can achieve convergence within 10 time intervals. Besides, as $\tau_1$ increases, the convergence will become faster and more stable. However, there is a performance gap (though not large) between Algorithm 2 and the benchmark, which is because the number of pilots required by Algorithm 2 is much less than that of the CE scheme.

Finally, Fig. 15 presents the NMSE performance of the proposed 2SCTP scheme in the general case by combining Algorithm 2 and the OB-LSTM network. Different from that in Fig. 11, we consider two scenarios with the same amount of channel training overhead, i.e., in Scenario A, we set $T_1 = 6$, $T_2 = 3$, $L_I = 6$, $L_P = 3$ and employ Strategy A to generate the virtual observations; while in Scenario B, we set $T_1 = 6$, $T_2 = 3$, $L_I = 6$, $L_P = 1$ and use Strategy B. It can be seen that in the general case, the proposed 2SCTP scheme can also achieve very similar NMSE performance with the CT scheme, yet with much lower channel training overhead.

**VI. CONCLUSION**

In this paper, we investigated the CTP problem in an IRS-aided wireless communication system with time-varying channel and designed an innovative two-stage transmission protocol. Based on the proposed transmission protocol, we proposed a novel 2SCTP scheme to track and predict the channels with low channel training overhead. By exploiting the temporal correlation of the channels, we proposed a KF-based channel tracking algorithm for the special case when the IRS-AP channel is static, while for the general case, we developed a GKF-based channel tracking algorithm by devising a simple Gaussian approximation method. Furthermore, we presented an LSTM-based neural network, namely the OB-LSTM network, to predict the virtual observations based on which the channels can be estimated by applying the KF/GKF-based channel tracking algorithms. Numerical results showed that the proposed 2SCTP scheme is able to outperform the existing CE and CT schemes significantly in terms of channel training overhead.

Note that this work can be viewed as an initial attempt to introduce the deep learning technologies to address the CTP problem in IRS-aided communication systems. There are many interesting directions to pursue based on the proposed idea, such as using more advanced network structure for better prediction performance, considering more general time-varying channel models, and the joint design of channel estimation and data detection, etc.

**APPENDIX**

*Proof:* First, the quadratic sum of $a$ and $b$, i.e., $|a|^2 + |b|^2$, can be rewritten as

$$|a|^2 + |b|^2 = \frac{1}{2}(|a_r^2 + b_r^2| + (a_t^2 + b_t^2)| + \frac{1}{2}|(a_r^2 + b_r^2) + (a_t^2 + b_t^2)|. \quad (39)$$

According to the arithmetic and geometric (AM-GM) inequality [46], $x^2 + y^2 \geq 2xy$ and $x^2 + y^2 \geq -2xy$ hold for all
x, y ∈ ℝ. As such, we have

\[ (a^2 + b^2) + (a'^2 + b'^2) \geq 2a_b - a'a_b \]
\[ (a^2 + b^2) + (a'2 + b'^2) \geq -2a_b + 2a'a_b \]
\[ \Rightarrow (a^2 + b^2) + (a'2 + b'^2) \geq 2|a_b - a'b|, \]
\[ (a^2 + b^2) + (a'2 + b'^2) \geq 2a_b + 2a'b \]
\[ (a^2 + b^2) + (a'2 + b'^2) \geq -2a_b - 2a'b \]
\[ \Rightarrow |(a^2 + b^2) + (a'2 + b'^2)| \geq 2|a_b + a'b|. \]  

(40)

Then, it is easy to see that \(|a|^2 + |b|^2|\) can be lower bounded by

\[ |a|^2 + |b|^2 = \frac{1}{2}[(a^2 + b^2) + (a'2 + b'^2)] \]
\[ + \frac{1}{2}[|a^2 + b^2| + (a'^2 + b'^2)] \]
\[ \geq |a_b - a'b| + |a + a_b|. \]  

(41)

Besides, since \(ab = (a_a - b_b + j(a+b + a'b))\) holds, (41) can be written as \(|a|^2 + |b|^2 \geq |\Re(ab)| + |\Im(ab)|\). This thus completes the proof. □

REFERENCES

[1] Q. Wu, S. Zhang, B. Zheng, C. You, and R. Zhang, “Intelligent reflecting surface-aided wireless communications: A tutorial,” IEEE Trans. Commun., vol. 69, no. 5, pp. 3313–3351, May 2021.
[2] B. Zheng et al., “Intelligent reflecting surface assisted multi-user OFDMA: Channel estimation and training design,” IEEE Trans. Wireless Commun., vol. 19, no. 12, pp. 8315–8329, Dec. 2020.
[3] X. Tan, Z. Sun, D. Koutsonikolas, and J. M. Jornet, “Enabling indoor mobile millimeter-wave networks based on smart reflect-arrays,” in Proc. IEEE Conf. Comput. Commun. (INFOCOM), Apr. 2018, pp. 270–278.
[4] L. Subrt and P. Pechac, “Controlling propagation environments using intelligent walls,” in Proc. 6th Eur. Conf. Antennas Propag. (EUCAP), Mar. 2012, pp. 1–5.
[5] M. M. Zhao, Q. Wu, M. J. Zhao, and R. Zhang, “Exploiting amplitude control in intelligent reflecting surface aided wireless communication with imperfect CSI,” IEEE Trans. Commun., vol. 69, no. 6, pp. 4216–4231, Jun. 2021.
[6] S. Gong et al., “Toward smart wireless communications via intelligent reflecting surfaces: A contemporary survey,” IEEE Commun. Surveys Tut., vol. 22, no. 4, pp. 2283–2314, 4th Quart., 2020.
[7] S. Abeywickrama, R. Zhang, Q. Wu, and C. Yuen, “Intelligent reflecting surface: Practical phase shift model and beamforming optimization,” IEEE Trans. Commun., vol. 68, no. 9, pp. 5849–5863, Sep. 2020.
[8] D. Mishra and H. Johansson, “Channel estimation and low-complexity beamforming design for passive intelligent surface assisted MISO wireless energy transfer,” in Proc. ICASSP - IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2019, pp. 4659–4663.
[9] T. L. Jensen and E. De Carvalho, “An optimal channel estimation scheme for intelligent reflecting surfaces based on a minimum variance unbiased estimator,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2020, pp. 5000–5004.
[10] Z. Wang, L. Liu, and S. Cui, “Channel estimation for intelligent reflecting surface assisted multi-user communications: Framework, algorithms, and analysis,” 2019, arXiv:1912.11783.
[11] Y. Wei, M. M. Zhao, M. -J. Zhao, and Y. Cai, “Channel estimation for IRS-aided multiuser communications with reduced error propagation,” IEEE Trans. Wireless Commun., vol. 21, no. 4, pp. 2725–2741, Apr. 2022.
[12] J. Chen, Y-C. Liang, H. Victor Cheng, and W. Yu, “Channel estimation for reconfigurable intelligent surface aided multi-user MIMO systems,” 2019, arXiv:1912.03619.
[13] A. Taha, M. Alrabeiah, and A. Alkhateeb, “Enabling large intelligent surfaces with compressive sensing and deep learning,” IEEE Access, vol. 9, pp. 44304–44321, 2021.
[14] Z.-Q. He and X. Yuan, “Cascaded channel estimation for large intelligent metasurface assisted massive MIMO,” IEEE Wireless Commun. Lett., vol. 9, no. 2, pp. 210–214, Feb. 2020.
[40] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. Int. Conf. Learn. Represent., 2015, pp. 1–15.
[41] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” IEEE Trans. Signal Process., vol. 50, no. 2, pp. 174–188, Aug. 2002.
[42] S. J. Julier and J. K. Uhlmann, “Unscented filtering and nonlinear estimation,” Proc. IEEE, vol. 92, no. 3, pp. 401–422, Mar. 2004.
[43] M. D. Springer and W. E. Thompson, “The distribution of products of beta, gamma and Gaussian random variables,” SIAM J. Appl. Math., vol. 18, no. 4, pp. 721–737, 1970.
[44] Ž. Stojanac, D. Suess, and M. Kliesch, “On products of Gaussian random variables,” 2017, arXiv:1711.10516.
[45] C. You, B. Zheng, and R. Zhang, “Channel estimation and passive beamforming for intelligent reflecting surface: Discrete phase shift and progressive refinement,” IEEE J. Sel. Areas Commun., vol. 38, no. 11, pp. 2604–2620, Nov. 2020.
[46] J. M. Steele, The Cauchy-Schwarz Master Class: An Introduction to the Art of Mathematical Inequalities. Cambridge, U.K.: Cambridge Univ. Press, 2004.

Yi Wei (Student Member, IEEE) received the B.Eng. and Ph.D. degrees in information and communication engineering from Zhejiang University in 2017 and 2022, respectively. From May 2021 to April 2022, she was a Visiting Scholar with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. She is currently working as a Research Engineer with China Aerospace Science and Technology Corporation. Her research interests include signal processing for wireless communications, algorithm design for advanced MIMO, and deep learning for wireless communications.

Ming-Min Zhao (Member, IEEE) received the B.Eng. and Ph.D. degrees in information and communication engineering from Zhejiang University in 2012 and 2017, respectively. From December 2015 to August 2016, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, Iowa State University, Ames, USA. From July 2017 to July 2018, he worked as a Research Engineer with Huawei Technologies Company Ltd. From May 2019 to June 2020, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, National University of Singapore. Since August 2018, he has been working with Zhejiang University, where he is currently an Associate Professor with the College of Information Science and Electronic Engineering. His research interests include channel coding, algorithm design and analysis for advanced MIMO, cooperative communication, and machine learning for wireless communications.

Min-Jian Zhao (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in communication and information systems from Zhejiang University, Hangzhou, China, in 2000 and 2003, respectively. He was a Visiting Scholar with the University of York, York, U.K., in 2010. He is currently a Professor and the Deputy Director with the College of Information Science and Electronic Engineering, Zhejiang University. His research interests include modulation theory, channel estimation and equalization, MIMO, signal processing for wireless communications, anti-jamming technology for wireless transmission and networking, and communication SOC chip design.

An Liu (Senior Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from Peking University, China, in 2004 and 2011, respectively. From 2008 to 2010, he was a Visiting Scholar with the Department of ECEE, University of Colorado Boulder. He has been a Post-Doctoral Research Fellow from 2011 to 2013, a Visiting Assistant Professor in 2014, and a Research Assistant Professor from 2015 to 2017 with the Department of ECE, HKUST. He is currently a Distinguished Research Fellow with the College of Information Science and Electronic Engineering, Zhejiang University. His research interests include wireless communications, stochastic optimization, compressive sensing, and machine/deep learning for communications. He is serving as an Editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and IEEE WIRELESS COMMUNICATIONS LETTERS.