Deep Learning-Based End-to-End Wireless Communication Systems With Conditional GANs as Unknown Channels

Hao Ye, *Student Member, IEEE*, Le Liang, *Member, IEEE*, Geoffrey Ye Li, *Fellow, IEEE*, and Biing-Hwang Juang, *Life Fellow, IEEE*

***Abstract*— In this article, we develop an end-to-end wireless communication system using deep neural networks (DNNs), where DNNs are employed to perform several key functions, including encoding, decoding, modulation, and demodulation. However, an accurate estimation of instantaneous channel transfer function, *i.e.*, channel state information (CSI), is needed in order for the transmitter DNN to learn to optimize the receiver gain in decoding. This is very much a challenge since CSI varies with time and location in wireless communications and is hard to obtain when designing transceivers. We propose to use a conditional generative adversarial net (GAN) to represent channel effects and to bridge the transmitter DNN and the receiver DNN so that the gradient of the transmitter DNN can be back-propagated from the receiver DNN. In particular, a conditional GAN is employed to model the channel effects in a data-driven way, where the received signal corresponding to the pilot symbols is added as a part of the conditioning information of the GAN. To address the curse of dimensionality when the transmit symbol sequence is long, convolutional layers are utilized. From the simulation results, the proposed method is effective on additive white Gaussian noise (AWGN) channels, Rayleigh fading channels, and frequency-selective channels, which opens a new door for building data-driven DNNs for end-to-end communication systems.**

***Index Terms*— Channel GAN, CNN, end-to-end communication system, channel coding.**

I. Introduction

In a traditional wireless communication system shown in Fig. 1(a), the data transmission entails multiple signal processing blocks in the transmitter and the receiver. While the technologies in this system are quite mature, individual blocks therein are separately designed and optimized, often with different assumptions and objectives, making it difficult, if not impossible, to ascertain global optimality of the system. In addition, the channel propagation is expressed as an assumed mathematical model embedded in the design. The assumed model may not correctly or accurately reflect the actual transmission scenario, thereby compromising the system performance.

On the contrary, the learning based data-driven methods provide a new way for handling the imperfection of the assumed channel models [1]. Recently, deep learning has been applied to refine the traditional block-structure communication systems, including the multiple-input and multiple-output (MIMO) detection [2], [3], channel decoding [4]–[9], and channel estimation [10], [11]. In addition, deep learning based methods have also shown impressive improvement by jointly optimizing the conventional communication blocks, including joint channel estimation and detection [12], joint channel encoding and source encoding [13].

Besides enhancing the traditional communication blocks, deep learning provides a new paradigm for future communication systems. As a pure data-driven method, the features and the parameters of a deep learning model can be learned directly from the data, without handcraft or ad-hoc designs, by optimizing an end-to-end loss function. Inspired by this methodology, end-to-end learning based communication systems have been investigated in several prior works [14]–[18], where both the transmitter and the receiver are represented by deep neural networks (DNNs), as shown in Fig. 1(b). This framework can be interpreted as an auto-encoder system, which is widely used in learning representations of the data [19]–[21]. The transmitter and the receiver correspond to the auto-encoder and auto-decoder, respectively. From Fig. 1(b), the transmitter learns to encode the transmitted symbols into encoded data, $\mathbf{x}$, which is then sent to the channel while the receiver learns to recover the transmitted symbols based on the received signal, $\mathbf{y}$, from the channel. As a result, the traditional communication modules at the transmitter, such as the encoding and modulation, are replaced by a DNN while the modules at the receiver, such as the decoding and the demodulation, are replaced by another DNN. The transmitter and the receiver DNNs are trained offline using measurement or simulated data set in a supervised learning manner optimizing a loss function that reflects recovery accuracy. From Fig. 1(b), the loss function depends on the weight/parameter set of the transmit DNN,

Manuscript received March 4, 2019; revised July 8, 2019 and November 27, 2019; accepted January 20, 2020. Date of publication February 6, 2020; date of current version May 8, 2020. This work was supported in part by a research gift from Intel Corporation and the National Science Foundation under Grant 1815637 and Grant 1731017. The associate editor coordinating the review of this article and approving it for publication was E. De Carvalho. *(Corresponding author: Le Liang.)*

Hao Ye, Geoffrey Ye Li, and Biing-Hwang Juang are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: yehao@gatech.edu; liye@ece.gatech.edu; juang@ece.gatech.edu).

Le Liang was with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA. He is now with Intel Labs, Hillsboro, OR 97124 USA (e-mail: lliang@gatech.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2020.2970707

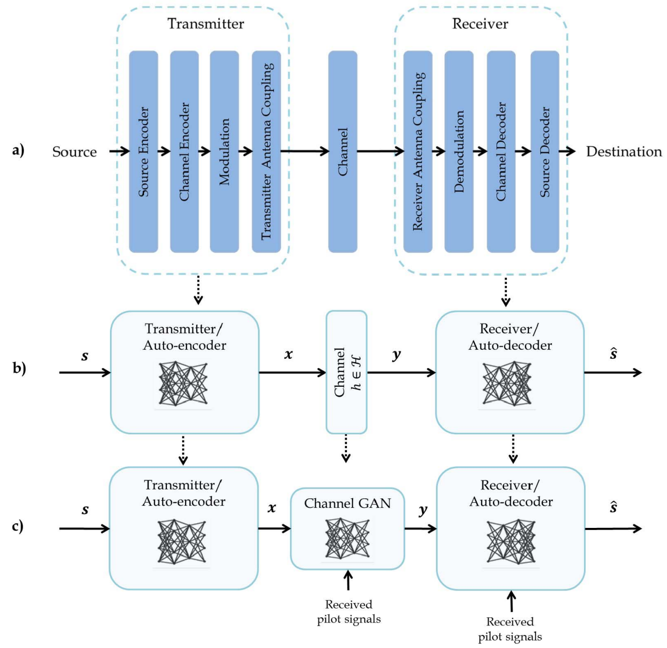

Fig. 1. Structures of a conventional wireless communication system and end-to-end learning based communication systems: (a) conventional wireless communication system; (b) end-to-end communication system based on auto-encoder, where both the transmitter and the receiver are represented by DNN; (c) proposed end-to-end communication system, where channel GAN is used to model the channel effects.

$\boldsymbol{\theta_T}$, the weight/parameter set of the receiver DNN, $\boldsymbol{\theta_R}$, and the channel realization, $\mathbf{h}$, and thus can be expressed as $\mathcal{L}(\boldsymbol{\theta_T}, \boldsymbol{\theta_R}, \mathbf{h})$. However, in the offline training phase, the channel realization, $\boldsymbol{h}$, is unknown. We only know that it is from a sample space/set, $\mathcal{H}$, with certain features or characteristics, such as indoor channels or typical urban channels. Therefore, the parameter sets of the transmitter and receiver DNNs are essentially optimized to minimize $\mathbb{E}_{(\mathbf{h}\in\mathcal{H})}\{\mathcal{L}(\boldsymbol{\theta_T}, \boldsymbol{\theta_R}, \mathbf{h})\}$.

From the above discussion, several critical challenges in the learning based end-to-end communication system need to be addressed in order to apply this framework to various wireless channels. As is well known, the weights of the DNN are usually updated using stochastic gradient descent (SGD) with the computed loss gradients propagated from the output layer back to the input layer. When the channel transfer function, $\mathbf{y} = f_{\mathbf{h}}(\mathbf{x})$, is not available, the back-propagation of the gradients from the receiver DNN to the transmitter DNN is blocked, preventing the overall learning of the end-to-end system. The channel transfer function may be assumed, but any such assumption would bias the learned weights, repeating the pitfalls caused by the likely discrepancy between the assumed model and the actual channel. In addition, in real communication systems, an accurate instantaneous CSI is hard to obtain in advance due to the various inherent uncertainties of wireless channels, such as channel noise and being time-varying. These uncertainties are often unknown or cannot be expressed analytically.

Another key challenge of the end-to-end paradigm is the curse of dimensionality during the training when the transmitted symbol sequence is long. The code block size in a communication system needs to be long enough to ensure a sufficient coding gain. However, as the size of possible codewords grows exponentially with the code block size, the portion of the unseen codewords during training will significantly increase accordingly. Previous works on the learning based decoding [7] show that the decoding performance of the DNN on the unseen codewords is still poor even if nearly 90% of the codewords have been included in the training of the DNN. Therefore, almost all the previous works on the end-to-end paradigm are concentrating on examples with a small block size, such as the Hamming codes (7,4) [14], [22]. As a result, it is desirable to develop a channel agnostic end-to-end communication system based on deep learning, where different types of channel effects can be automatically learned without knowing the specific channel transfer function and the block-length remains long enough to be practical.

In this article, we develop a channel agnostic end-to-end communication system to address the challenges, where the distributions of channel output are learned through a conditional generative adversarial net (GAN) [23], as shown in Fig. 1(c). The conditioning information for the GAN to generate samples is the encoded signals from the transmitter along with the received pilot information used for estimating the channel. By iteratively training the conditional GAN, the transmitter, and the receiver, the end-to-end loss can be optimized in a supervised way.

This channel agnostic end-to-end system provides a new way to optimize communication systems and is applicable to a wide range of wireless channels. In addition, the convolutional neural networks (CNNs) are used to overcome the curse of dimensionality and the block-length can be extended from several bits to a couple of hundred bits. Our main contributions in this article are threefold.

- We use the conditional GAN to model the conditional distribution, $p(\mathbf{y}|\mathbf{x})$, of wireless channels. By adding the received pilot symbol as a part of the conditioning information, the conditional GAN can generate more specific samples for the time-varying channels.
- Based on the conditional GAN modeling the channel conditional distribution, an end-to-end learning based communication system is developed, where the gradients of the end-to-end loss can be propagated to the transmitter DNN through the conditional GAN.
- CNNs are employed for alleviating curse of dimensionality. From the experimental results, the transmitter with convolutional layers can learn to encode the transmit data into a high dimensional embedding vector, which can be effectively decoded by the receiver.

Part of the work has been published in [22]. Compared with the previous work, we have made two significant improvements. First, we introduce convolutional layers so that this approach can be extended from several bits to a couple of hundred bits. Second, our framework is extended to more practical wireless channels, such as frequency-selective channels, where there exists inter-symbol interference (ISI).

The rest of the paper is organized as follows. The related works are discussed in Section II. In Section III, the conditional GAN based channel modeling approach is introduced. In

Section IV, the training for the end-to-end system is presented in detail. In Section V, the simulation results are presented and the conclusions are drawn in Section VI.

II. Related Works

Our proposed method is closely related to GANs, end-to-end learning based communication systems, and learning based decoders. In this section, previous works in the related topics are briefly reviewed.

A. GANs and Conditional GANs

GANs have been proposed in [25] as a generative framework, where a generator and a discriminator are competing with each other in the training stage. With the feedback of the discriminator, the generator improves its ability to generate samples that are similar to the real samples. GAN and its variants [26], [27] are most widely used in computer vision. Much of the recent GAN research is focusing on improving the quality of the generated images [28].

In order to generate samples with a specific property, a conditional GAN is proposed based on the GAN framework, where the context information is added to the generator and the discriminator. Originally, the condition added is the label information so that the generator can generate corresponding samples given a particular category. Nowadays, conditional GANs are widely used in changing the style and the content of the input [29], [33]. For instance, GANs have been utilized to generate high-resolution images from low-resolution images [29].

Apart from applications in computer vision, recently GANs have been exploited to model the channel effects of additive white Gaussian noise (AWGN) channels [24], similar to our work. However, our approach can be applied to more realistic fading channels by using conditional GAN, which employs the received pilot information as a part of the condition information when generating the channel outputs.

B. DNN Based End-to-End Communications

The objective of the paper is to build an end-to-end communication system for wireless channels, where the channel effects are modeled with conditional GAN. The idea of end-to-end learning communication systems has been proposed in [14] and has been shown to have a similar performance as the traditional approaches with block structures under the AWGN condition. In [15], the end-to-end method has been extended to handle various hardware imperfection. In [16], an end-to-end learning method is adopted within the orthogonal frequency-division multiplexing (OFDM) system. In [30], CNNs are employed for modulation and demodulation, where improved results have been shown for very high order modulation. In addition, source coding can also be considered as a part of the end-to-end communication system for transmitting text or image [13].

Training the end-to-end communication system without channel models has been investigated recently. A reinforcement learning based framework has been employed in [17] to optimize the end-to-end communication system without requiring the channel transfer function or CSI, where the channel and the receiver are considered as the environment when training the transmitter. The recovery performance at the receiver is considered as the reward, which guides the training of the transmitter. In [18], a model-free end-to-end learning method has been developed based on simultaneous perturbation methods. However, both works are conducted with a small block length. For example, blocks of eight information bits are used in [17]. How to extend the end-to-end framework to a large block size and how to model the unknown channel using a data-driven approach are still open problems, which are addressed in our proposed approach.

C. Learning Based Decoders

Our proposed method is also closely related to learning based encoding and decoding. Learning based approach has been utilized in improving decoding performance for a long time. It dates back to the 1990s when several attempts had been made to decode the codes with the recurrent neural network (RNN) [31].

With the widely used deep learning approaches, DNN has been utilized in decoding with a wide range of applications. In [7], a fully connected neural network is trained for decoding short Polar codes. The performance of decoding is similar to maximum likelihood decoding. But they find it difficult to break through the curse of dimensionality. In order to train long codewords, a partition method has been employed in [7]. Moreover, there have been several trials to incorporate some prior information in the decoding process. RNN is used in [5] for decoding the convolutional and turbo codes. In [4], the traditional belief-propagation decoding algorithm is extended as deep learning layers to decode linear codes. Recently, the curse of dimensionality is mitigated when the convolutional and circular convolutional layers are utilized to learn the encoding and decoding modules simultaneously rather than just learning to decode human-designed codewords [6].

III. Modeling Channel With Conditional GAN

An end-to-end communication system learns to optimize DNNs for the transmitter and the receiver. However, the back-propagation algorithm, which is used to train the weights of DNNs, is blocked by the unknown CSI, preventing the overall learning of the end-to-end system. To address the issue, we use a conditional GAN to learn the channel effects and to act as a bridge for the gradients to pass through. By the conditional GAN, the output distribution of the channel can be learned in a data-driven manner and therefore many complicated effects of the channel can be addressed. In this section, we introduce the conditional GAN and discuss how to use it to model the channel effects.

A. Conditional GAN

GAN [25] is a new class of generative methods for distribution learning, where the objective is to learn a model that can produce samples close to some target distribution, p_{data}.

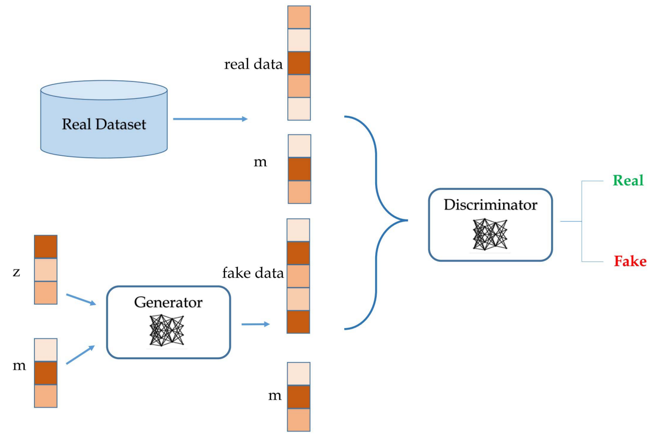

Fig. 2. Structure of conditional GAN.

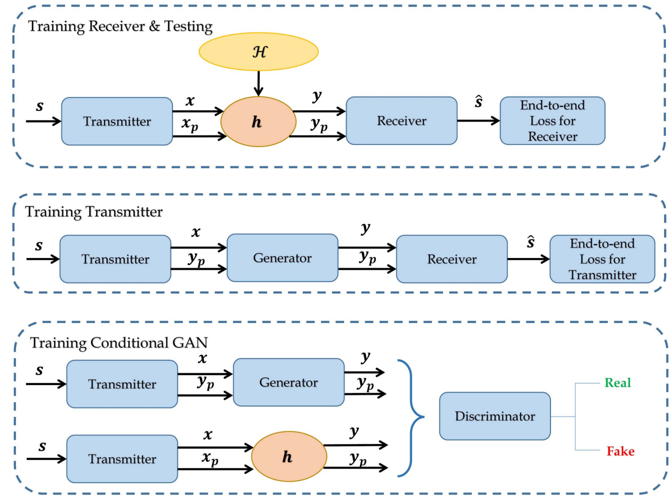

Fig. 3. Training and testing of the end-to-end system.

In our system, a GAN is applied to model the distribution of the channel output and the learned model is then used as a surrogate of the real channel when training the transmitter so that the gradients can pass through to the transmitter.

As shown in Fig. 2, the GAN system consists of two DNNs, *i.e.*, the generator, G, and the discriminator, D. The input to the generator, G, is a noise vector $\mathbf{z}$ sampled from a prior distribution p_z, *e.g.*, uniform distribution. The input $\mathbf{z}$ is transformed by the generator into a generated sample, $G(\mathbf{z})$. The input of the discriminator, D, is either a real sample from the target distribution, p_{data}, or a generated sample, $G(\mathbf{z})$, while the output of the discriminator is a real value representing the probability that the input is sampled from the target distribution, p_{data}.

During the training, the discriminator, D, learns to distinguish the data generated by the generator and the data from the target distribution while the generator, G, learns to generate samples to fool the discriminator into making mistakes. The samples from the real data and those generated from the generator are collected to train the discriminator to maximize the ability to distinguish between the two categories. If the discriminator can successfully classify the samples of the two sources, then its success will be used to generate feedback to the generator, so that the generator can learn to produce samples more similar to the real samples. The training procedure will end upon reaching an equilibrium, where the discriminator, D, can do no better than random guessing to distinguish the real samples and the generated fake samples.

Denote the parameter sets of the generator, G, and the discriminator, D, as $\boldsymbol{\theta_G}$ and $\boldsymbol{\theta_D}$, respectively, and the objective functions for training the generator and the discriminator can be expressed as

$$\mathcal{L}_G = \min_{\boldsymbol{\theta_G}} \mathbb{E}_{\mathbf{z}\sim p_z}[\log(1 - D(G(\mathbf{z})))], \tag{1}$$

$$\begin{aligned}\mathcal{L}_D = \max_{\boldsymbol{\theta_D}} \mathbb{E}_{\mathbf{x}\sim p_{data}}&[\log(D(\mathbf{x}))]\\ &+\mathbb{E}_{\mathbf{z}\sim p_z}[\log(1 - D(G(\mathbf{z})))],\end{aligned} \tag{2}$$

respectively. The objective of the discriminator, D, is to give a high value when the input belongs to the real dataset and a low one when the input is generated by the generator, G, while the objective of generator, G, is to maximize the output of the discriminator, D, given the generated samples, $G(\mathbf{z})$.

The GAN can be extended to a conditional model if both the generator, G, and the discriminator, D, are conditioned on some extra information, $\mathbf{m}$, as in Fig. 2. We only need to feed the conditioning information, $\mathbf{m}$, into both the generator and discriminator as the additional input. Therefore, the output of the G will be $G(\mathbf{x}, \mathbf{m})$ and the output of D will be $D(\mathbf{x}, \mathbf{m})$. The optimization objective functions for the generator and the discriminator become

$$\mathcal{L}_G = \min_{\boldsymbol{\theta_G}} \mathbb{E}_{\mathbf{z}\sim p_z}[\log(1 - D(G(\mathbf{z}, \mathbf{m}), \mathbf{m}))], \tag{3}$$

$$\begin{aligned}\mathcal{L}_D = \max_{\boldsymbol{\theta_D}} \mathbb{E}_{\mathbf{x}\sim p_{data}}&[\log(D(\mathbf{x}, \mathbf{m}))]\\ &+\mathbb{E}_{\mathbf{z}\sim p_z}[\log(1 - D(G(\mathbf{z}, \mathbf{m}), \mathbf{m}))],\end{aligned} \tag{4}$$

respectively. The conditional GAN is employed in our end-to-end communication system to model the channel output distribution with the given conditioning information on the encoded signal and the received pilot data.

B. Modeling Channels

Since the channel output, $\mathbf{y}$, for given input, $\mathbf{x}$, is determined by the conditional distribution, $p(\mathbf{y}|\mathbf{x})$, a conditional GAN can be employed for learning the output distribution of a channel by taking $\mathbf{x}$ as the condition information. The generator will try to produce the samples similar to the output of the real channel while the discriminator will try to distinguish data coming from the real channel and the data coming from the generator.

The instantaneous CSI, $\mathbf{h}$, is regarded as a sample from a large channel set $\mathcal{H}$ and is also vital for coherent detection of the transmit symbols at the receiver. In order to obtain the CSI, a common practice is to send some pilot information to the receiver so that the channel information is inferred based on the received signal corresponding to the pilot symbols, $\mathbf{y_p}$. In our proposed method, the received signal corresponding to the pilot symbols, $\mathbf{y_p}$, is added as a part of the conditioning information so that the output samples follow the distribution of $\mathbf{y}$ given the input $\mathbf{x}$ and the received pilot data, $\mathbf{y_p}$. If $\mathbf{y_p}$ is excluded from the condition information, the conditional GAN will learn to generate different channel conditions with different $\boldsymbol{z}$. With $\mathbf{y_p}$ as the conditional information, the output distribution of the conditional GAN will stick to the current channel instead of being a multi-modal distribution.

C. Convolutional Layers Based Channel GAN

The convolutional layers have been introduced to efficiently extract features for images based on their shared-weight architecture and translational invariance characteristics [34]. In a fully connected layer, each neuron is connected to all neurons in the previous layer. In contrast, in a convolutional layer, each neuron is only connected to a few nearby neurons in the previous layer, which is called the receptive field of this neuron, and the same set of weights is shared by all neurons in a layer. Inspired by the convolutional codes, where the encoding process can be represented by a convolutional transform, we use hierarchical one-dimensional convolutional layers in the channel GAN as well as the transmitter and the receiver.

Denote $u^{(i)}[n]$ as the output of the n-th neuron in the i-th layer of a DNN. For a fully connected layer, the output of the n-th neuron in the i-th layer is

$$u^{(i)}[n] = \sigma(\sum_k w_{nk}^{(i)} u^{i-1}[k]),$$

where $\sigma(\cdot)$ is an activation function and $w_{nk}^{(i)}$ is the weight connected the k-th neuron in the $(i-1)$-th layer and the n-th neuron in the i-th layer. $w_{nk}^{(i)}$ is different for different i, n, or k. Therefore, if there are N_i neurons in the i-th layer, there will be $N_i N_{i-1}$ weights in total to fully connect the $(i-1)$-th layer to the i-th layer.

On the other hand, for a convolutional layer, the output of the n-th neuron in the i-th layer will be

$$u^{(i)}[n] = \sigma(\sum_{k=1}^{L} w_k^{(i)} u^{(i-1)}[n-k]),$$

where $w_k^{(i)}$ is the coefficient for the convolution and is same for different n's in the i-th layer. There are L weights in total (L is usually much smaller then N_i or N_{i-1}) to implement $N_i L$ connections between the $(i-1)$-th and the i-th layers. In brief, compared with a fully connected DNN, the CNN has much fewer connections between adjacent layers and much fewer weights to train, which will reduce the complexity and significantly improve the convergence speed of training.

Apart from being easier to train, CNN has two additional merits in the end-to-end communication system. First, the curse of dimensionality can be alleviated by the usage of convolutional layers [6]. When both the transmitter and the receiver are represented by CNNs, the codes learned by a CNN are more easily recovered at the receiver than the conventional handcraft codes. Second, it is appropriate to employ convolutional layers to deal with the ISI channels since the effect of the channel can be expressed by the convolutional operation in the ISI channel.

IV. END-TO-END COMMUNICATION SYSTEM

As stated in the introduction, the end-to-end communication paradigm can be interpreted as a deep auto-encoder framework. With the conditional GAN, the gradients can be back-propagated to the transmitter even if channels are unknown. In this section, the proposed framework is first introduced and the training procedures for each module are presented in detail.

A. System Overview

As in Fig. 1(b), the auto-encoder learns to map N information bits, $\mathbf{s} \in \{0,1\}^N$, into a fixed length embedding of length K, $\mathbf{x} \in \mathbb{R}^K$, and sends the embedding to the channel while the auto-decoder learns to recover the original information according to the received signal $\mathbf{y}$ from the channel. The distance between the original information bits, $\mathbf{s}$, and the recovered information, $\hat{\mathbf{s}} \in [0,1]^N$, will be calculated. The binary cross-entropy loss is used to measure the distance, which can be expressed as

$$L = \sum_{n=1}^{N} (s_n \log(\hat{s}_n) + (1-s_n)\log(1-\hat{s}_n)), \tag{5}$$

where s_n and $\hat{s}_n$ represent the nth elements of $\mathbf{s}$ and $\hat{\mathbf{s}}$, respectively.

The training and testing of the proposed end-to-end communication system are shown in Fig. 3. To obtain training data set, the information bits, $\mathbf{s}$, are randomly generated and the instantaneous CSI is sampled randomly from the channel set. Due to different objectives in modules, the transmitter, the receiver, and the channel generator in the conditional GAN can be trained iteratively based on the training data. Each component is trained by fixing the parameters of the others. The objective is to minimize the end-to-end loss when training the receiver and the transmitter. The optimization objective functions (3) and (4) are used when training the conditional GAN for generating the channel. In the testing stage, the end-to-end reconstruction performance is evaluated on the learned transmitter and receiver with real channels.

B. Training Receiver

At the receiver, a DNN model is trained for recovering the transmitted signal $\mathbf{s}$, where the input is the received signals corresponding to the transmitted data, $\mathbf{y}$, while the output is the estimation $\hat{\mathbf{s}}$. By comparing the $\mathbf{s}$ and $\hat{\mathbf{s}}$, the loss function can be calculated based on (5). The receiver can be trained easily since the loss function is computed at the receiver and thus the gradients of the loss can be easily obtained. For the time-varying channels, by directly putting the received signal, $\mathbf{y}$, and the receive pilot data, $\mathbf{y_p}$, together as the input, the receiver can automatically infer the channel condition and perform the channel estimation and detection simultaneously without explicitly estimating the channel, as we have discussed in [12].

C. Training Transmitter

With the channel generator as a surrogate channel, the training of the transmitter will be similar to that of the receiver.

During the training, the transmitter, the generator, and the receiver can be viewed as a whole DNN. The output of the transmitter is the values of the last hidden layer in the transmitter DNN. The end-to-end cross-entropy loss is computed at the receiver as in (5), and the gradients are propagated back to the transmitter through the conditional GAN. The weights of the transmitter will be updated based on SGD while the weights of the conditional GAN and the receiver remain fixed. The transmitter can learn the constellation of the embedding, $\mathbf{x}$, so that the received signal can be easily detected at the receiver.

D. Training Channel GAN

The training procedure of channel GAN is illustrated in Algorithm 1 in detail. In each iteration, the generator and the discriminator are trained iteratively. The parameters of one model will be fixed while training the other. With the learned transmitter, the real data can be obtained with the encoded signal from the transmitter going through the real channel while the fake data is obtained from the encoded data going through the channel generator. The parameters of the generator and the discriminator are updated according to the loss function of equations (3) and (4), respectively.

Algorithm 1 Channel GAN Training Algorithm

for number of training iterations **do**
  % Updating the Generator
  Sample minibatch of transmit data $\{\mathbf{s}\}$ and channel $\{\mathbf{h}\}$.
  Get the minibatch of condition information $\{\mathbf{m}\}$ from the output of the transmitter DNN and the received pilot signal from the channel $\{\mathbf{h}\}$.
  Sample minibatch of samples $\{\mathbf{z}\}$.
  Update the generator by ascending the stochastic gradient of the loss function (3).
  % Updating the Discriminator
  Sample minibatch of transmit data $\{\mathbf{s}\}$ and channel $\{\mathbf{h}\}$.
  Get the minibatch of condition information $\{\mathbf{m}\}$ from the output of the transmitter DNN and the received pilot signal from the channel $\{\mathbf{h}\}$.
  Sample minibatch of examples real data by collecting the output of channel.
  Sample minibatch of samples $\{\mathbf{z}\}$.
  Update the discriminator by descending the stochastic gradient of the loss function (4).
end for

V. EXPERIMENTS

In this section, the implementation details of the end-to-end learning based approach are provided and the simulation results are presented. For several types of most commonly used channels, the channel GAN has shown the ability to model the channel effects in a data-driven way. In addition, the end-to-end communication system, which is built on the channel GAN, can achieve similar or better results even the channel information is unknown during training and optimizing the transmitter and the receiver.

TABLE I
MODEL PARAMETERS OF FCN

Parameters	Values
Transmitter: Neurons in each hidden layers	32, 32
Learning rate	0.001
Receiver: Neurons in each hidden layers	32, 32
Learning rate	0.001
Generator: Neurons in each hidden layers	128, 128, 128
Discriminator: Neurons in each hidden layers	32, 32, 32
Learning rate	0.0001

TABLE II
MODEL PARAMETERS OF CNN

Type of layer	Kernel size/Annotation	Output size
Transmitter		
Input	Input layer	$K \times 1$
Conv+Relu	5	$K \times 256$
Conv+Relu	3	$K \times 128$
Conv+Relu	3	$K \times 64$
Conv	3	$K \times 2$
Normalization	Power normalization	$K \times 2$
Receiver		
Conv+Relu	5	$K \times 256$
Conv+Relu	5	$K \times 128$
Conv+Relu	5	$K \times 128$
Conv+Relu	5	$K \times 128$
Conv+Relu	5	$K \times 64$
Conv+Relu	5	$K \times 64$
Conv+Relu	5	$K \times 64$
Conv+Sigmoid	3	$K \times 1$
Generator		
Conv+Relu	5	$K \times 256$
Conv+Relu	3	$K \times 128$
Conv+Relu	3	$K \times 64$
Conv	3	$K \times 2$
Discrimator		
Conv+Relu	5	$K \times 256$
Conv+Relu	3	$K \times 128$
Conv+Relu	3	$K \times 64$
Conv+Relu	3	$K \times 16$
FC +Relu	100	100
FC+Sigmoid	1	1

A. Experimental Settings

1) Implementation Details: Two types of DNN models are designed in our experiments. One is fully connected networks (FCN) and the other is the CNN. The FCN is used for a small block size and the CNN is used in the a large block size to avoid the curse of dimensionality. The parameters of the FCN and CNN are shown in Table I and Table II, respectively. The weights of both models are updated by Adam [32] and the batch size for training is 320.

2) Channel Types: Three types of channels are considered in our experiments, *i.e.*, AWGN channels, Rayleigh channels, and frequency-selective multipath channels. In an AWGN channel, the output of the channel, $\mathbf{y}$, is the summation of the input signal, $\mathbf{x}$, and Gaussian noise, $\mathbf{w}$, that is, $\mathbf{y} = \mathbf{x} + \mathbf{w}$. Rayleigh fading is a reasonable model for narrow-band wireless channels when many objects in the environment scatter the radio signal before arriving at the receiver. In a Rayleigh channel, the channel output is determined by $\mathbf{y} = h_n \cdot \mathbf{x} + \mathbf{w}$, where $h_n \sim \mathcal{CN}(0, 1)$. The channel coefficient h_n is time-varying and is unknown when designing transceivers. Therefore, channel estimation is required to get the instantaneous CSI, which is then used by the receiver to detect the transmit data. With frequency-selective channels, radio signal propagates via multiple paths, which differ in amplitudes, phases, and delay times, and cause undesired frequency-selective fading and time dispersion of the received signal. The baseband complex channel impulse response can be expressed as

$$h(t) = \sum_{k=0}^{K_p} b_k e^{j\theta_k} p(t - \tau_k),$$

where there are K_p paths in all, b_k, θ_k, and τ_k represent the path gain, the phase shift, and the time delay of the kth path, respectively, and $p(t)$ is the shaping pulse in the communication system. In our simulation, a three-tap channel with equal average power is considered, that is, $\mathbb{E}|b_k|^2 = 1$, and $\tau_k = 0, T, 2T$, with T as the symbol duration.

3) Baselines: The end-to-end learning based communication system is compared with the conventional communication system, which is composed of multiple signal processing modules and each module is designed based on the prior knowledge on the channel. The bit-error rate (BER) and block-error rate (BLER) are compared under each type of channel. In our baseline system, 4-QAM is used as the modulation and the Hamming code or convolutional codes are used. For the convolutional codes, the Viterbi algorithm [35] is used for maximum likelihood sequence decoding. A commonly used example of a convolutional code, rate-1/2 recursive systematic convolutional (RSC) code, is used for channel coding [5]. OFDM is utilized to deal with the ISI in the frequency-selective multipath channel. For simplicity, $\mathbf{x_p}$ is set as a delta signal in the following experiments. The received the signal $\mathbf{y_p}$ can be expressed as $\mathbf{y_p} = \mathbf{h} + \mathbf{n_p}$, where the $\mathbf{n_p}$ is the additive noise in the received pilot signal. In order to eliminate the effects of channel estimation error and get a fair comparison of the end-to-end approach with the traditional ones, including channel coding and modulation techniques, we assume the transmitter can increase the power of pilot signal so that the additive noise $\mathbf{n_p}$ is negligible. In general, DNN can work with various pilot signals other than the delta signal as indicated in our prior work [12]. In addition, for AWGN and Rayleigh fading channels, the proposed conditional GAN based end-to-end system is also compared with the end-to-end systems, which are trained with real channel models instead of using conditional GANs as surrogate channels. For the AWGN channel, the auto-encoder system is trained by injecting the Gaussian noise directly to the hidden layer [14]. For the Rayleigh fading channel, an additional equalization layer is employed [17]. The input of the auto-decoder is the equalized signal $\hat{\mathbf{y}} = \frac{\mathbf{y}}{\hat{h}}$, where $\hat{h}$ is obtained from the pilot information.

B. Modeling the Channel Effects

We use FCN to model the effects of Rayleigh fading channels. Since Rayleigh fading channels are time-varying, additional conditional information is added to the channel generator and the receiver. Besides the encoded signal, the received pilot data, $\mathbf{y_p}$, is used as the additional conditional information. We test the effectiveness of the conditional GAN in learning the distribution of the channel with standard 16 QAM as the encoded symbols. Fig. 4 shows generated samples of a Rayleigh fading channel with different encoded symbols and received pilot signal as the conditioning information. From the figure, the conditional GAN is able to produce the samples with different channel gains and phase rotations according to conditioning information.

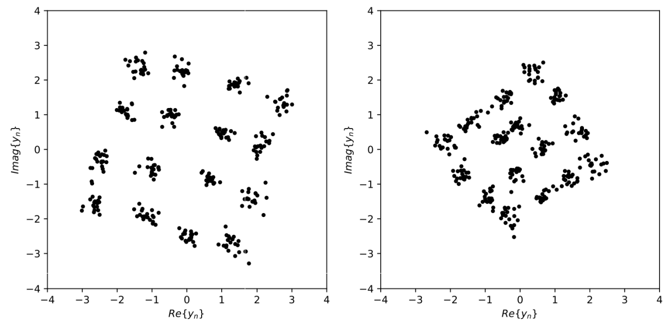

Fig. 4. Signal constellations at the output of a Rayleigh channel represented by a conditional GAN, where channel gains and phase rotations vary according to conditioning information $\mathbf{y_p}$.

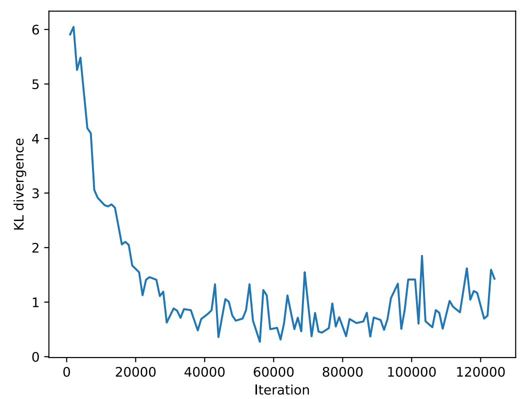

Fig. 5. KL divergence of the generated channel distribution and real channel distribution.

Fig. 5 shows the Kullback-Leibler (KL) divergence [36] of the generated distribution and the real distribution, $KL(G(z)||p_{data})$, which is estimated based on the k-nearest neighbor estimation approach in [37]. From the figure, the KL divergence deceases with the training iterations, indicating that the generated distribution converges to the target distribution p_{data}.

C. End-to-End Communication System

Based on the channel GAN, a channel agnostic end-to-end communication system is built on three types of channels, *i.e.*, the AWGN channel, the Rayleigh fading channel, and the frequency-selective multipath channel. We compare our channel agnostic end-to-end learning based approach with the traditional methods, which are designed based on the channel transfer functions.

1) AWGN Channel: We first use FCN for a small block size. The end-to-end recovering performance on the AWGN channel is shown in Fig. 6. At each time, four information

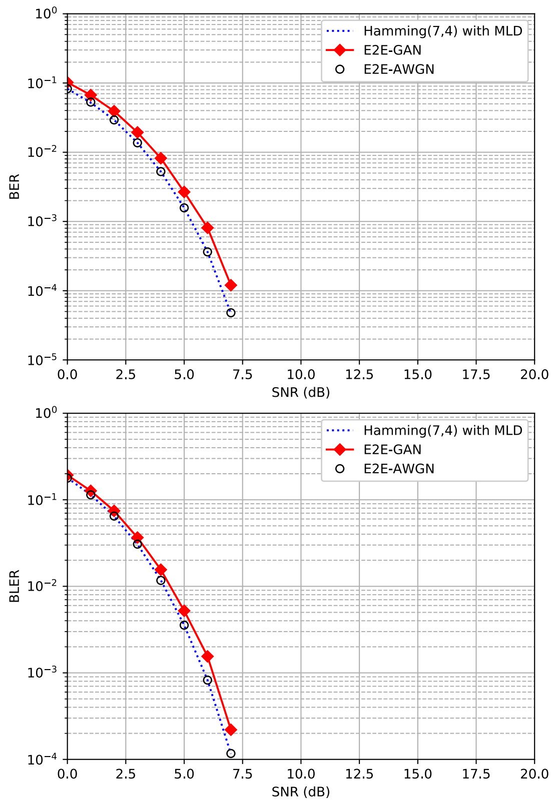

Fig. 6. BER and BLER with a block size of 4 bits under a AWGN channel.

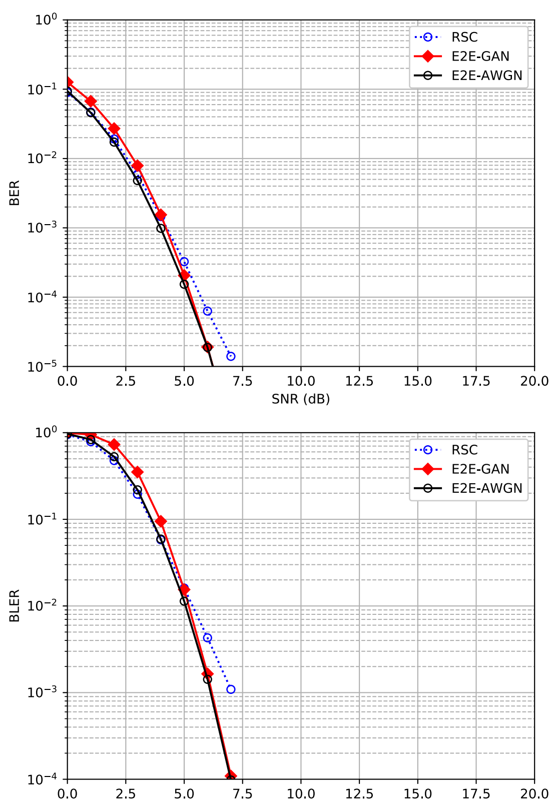

Fig. 7. BER and BLER with a block size of 128 bits under an AWGN channel.

bits are transmitted and the length of the transmitter output is set to be seven. From the figure, the BER and BLER of learning based approach are similar to Hamming (7,4) code with maximum-likelihood decoding (MLD) and with the model-aware end-to-end auto-encoder system trained under the AWGN channel, which proves the effectiveness of using the conditional GAN as a surrogate channel.

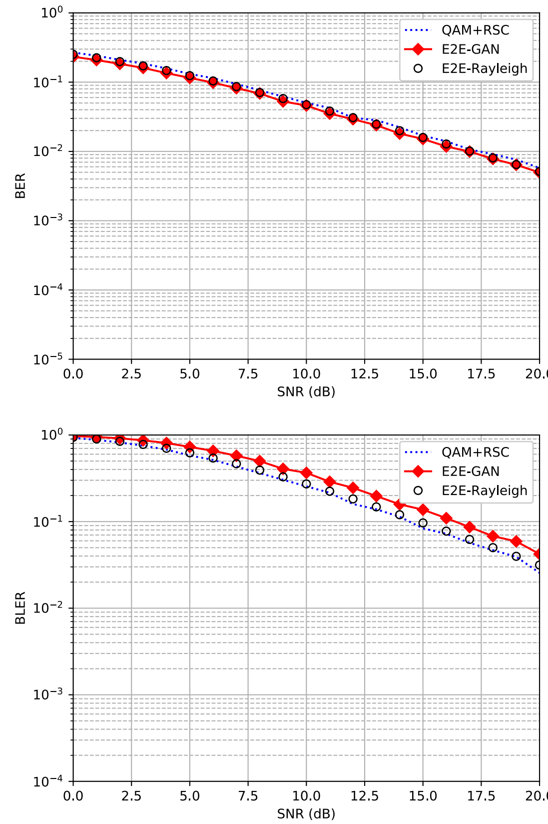

Fig. 8. BER and BLER under a Rayleigh channel.

In order to train models with a large block size, CNNs are then used to mitigate the curse of dimensionality. We first train the CNN under the AWGN channel where the noise is added to the hidden layer directly, as used in [14]. The network is trained at 3 dB fixed signal-to-noise ratio (SNR) and tested by different SNRs. Fig. 7 shows the BER and BLER curves of the proposed end-to-end method with the length of 128 bits, respectively. From the figure, the performance of the proposed method is similar to RSC in the low SNR area and significantly outperforms RSC in the high SNR area. Compared with the model-aware end-to-end auto-encoder system, the performance loss is negligible with the increased block size.

2) Rayleigh Fading Channel: CNNs are employed in transmission with a large block size and the channel encoding is included. We compare the end-to-end approach with a baseline method, where QAM is used as the modulation and the RSC of coderate $1/2$ are used as the coding. In each block, 64 information bits will be transmitted, thus the input size of the end-to-end approach is 64. From Fig. 8, the end-to-end approach shows similar performance to the traditional methods in terms of BER and BLER. The conditional GAN based end-to-end approach also shows similar performance to the model-aware end-to-end system trained with Rayleigh channels.

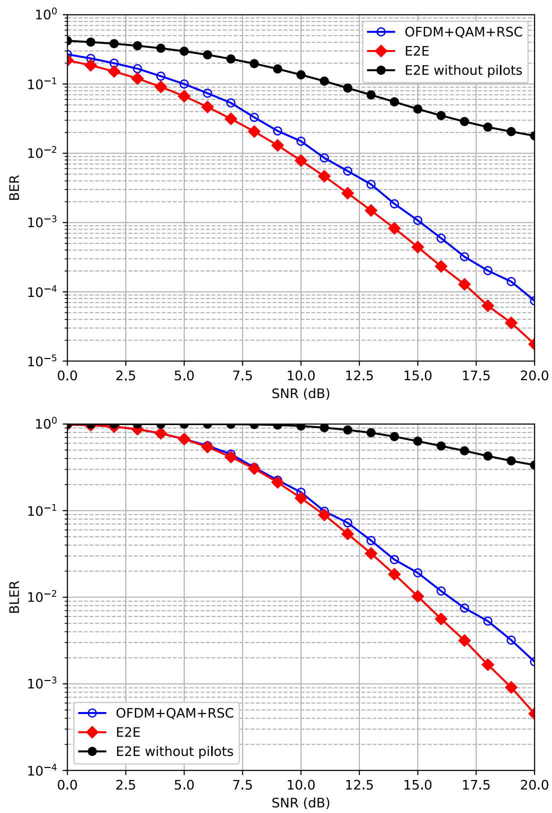

Fig. 9. BER and BLER under a frequency-selective multipath channel.

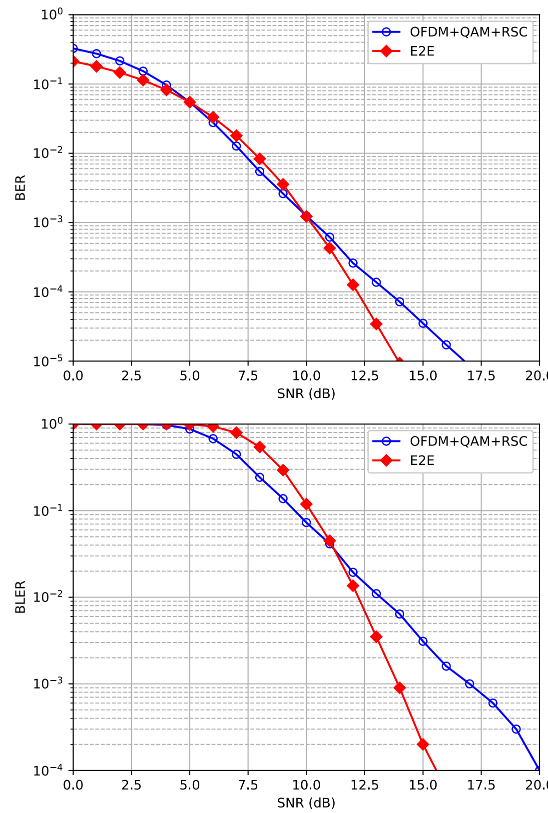

Fig. 10. BER and BLER under WINNER II channel.

3) Frequency-Selective Fading Channel: Under the frequency-selective channel, the end-to-end communication systems are developed with CNN and the OFDM system is used as the baseline. There are 64 subcarriers in the OFDM system and the length cycle-prefix is set as 16 and 4 QAM is used for the modulation. The RSC is utilized for channel coding. In order to have a fair comparison, we set the block size of the end-to-end system as 64 bits and pad 16 zeros between two consecutive blocks.

Fig. 9 shows the performance of the proposed end-to-end approach. From Fig. 9, the proposed end-to-end system outperforms the OFDM system by 2 dB in terms of BER. The BLER of the end-to-end system is also significantly lower in the high SNR area when compared with the OFDM system. The usage of the pilot information in our end-to-end system has two advantages. With the pilot information, the receiver can obtain information on a specific channel realization, which is very helpful for recovering the transmit data. Moreover, with the pilot data, the GAN based channel generator learns to generate the channel output distribution under the current channel realization. On the contrary, without the pilot information, the channel generator learns to produce the mixture channel output distributions of all the channel realizations. The importance of the pilot information is justified in Fig. 9, where the end-to-end performance degrades significantly without the pilot information.

Moreover, in order to show the performance in the real scenario, the WINNER II channels [38] are employed and the parameters for channel generation are the same with [12] except that the maximum delay 8 sampling period is used. The block size is 128 bits for both the OFDM system and the end-to-end approach. As shown in Fig. 10, the BER performance of the proposed end-to-end system is similar to the OFDM system when the SNR is below 10 dB and is significantly better when SNR is over 10 dB. The BLER performance of the proposed end-to-end system is slightly worse than the OFDM system in the low SNR area but outperforms the baseline system by about 4 dB in the high SNR area.

The computational complexities of the end-to-end system and the baselines are also compared. The computational complexity of the CNN is $\mathcal{O}(\sum_{l=1:L} Nk_l^2\ F_{l-1}F_l)$, where the N is the block size, L is the number of the layers, k_l is the kernel size of the lth layer, and F_{l-1} and F_l are the numbers of filters in the $(l-1)$th layer and the lth layer, respectively. For the OFDM system, the computational complexity of the fast Fourier transform (FFT) is $\mathcal{O}(N \log N)$. The complexity of the Viterbi decoding is $\mathcal{O}(N2^k)$, where k is the length of the memory. To complete our discussion of the computational complexity, we have measured the average running time of the proposed algorithm and the baseline approach on a Windows server with Intel i7 CPU and an Nvidia 1080Ti GPU. The average running time for the OFDM system is about 2.2×10^{-2} seconds while the average running time for the end-to-end system is only about 2.5×10^{-3} seconds.

VI. Conclusion and Discussion

In this article, we investigate the end-to-end learning of a communication system without prior information of the channel. We show that the conditional distribution of the channel can be modeled by a conditional GAN. In addition, by adding the pilot information into the condition information, the conditional GAN can generate data corresponding to the specific instantaneous channel.

The end-to-end pipeline consists of DNNs for the transmitter, the channel GAN, and the receiver. By iteratively training these networks, the end-to-end loss can be optimized in a supervised way. The simulation results on the AWGN channels, Rayleigh fading channels, and frequency-selective channels confirm the effectiveness of the proposed method, by showing similar or better performance compared with the traditional approaches, which are designed based on expert knowledge and channel models. Our research opens a new door for building the pure data-driven communication systems.

One of the future directions is to test the proposed method with real data. As we have indicated in the introduction, in real communication scenarios, many imperfections make the real channel difficult to be modeled analytically, but are very suitable to be modeled in a data-driven manner.

REFERENCES

[1] Z. Qin, H. Ye, G. Y. Li, and B.-H.-F. Juang, "Deep learning in physical layer communications," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 93–99, Apr. 2019.

[2] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO detection," in *Proc. IEEE 18th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2017, pp. 1–5.

[3] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "A model-driven deep learning network for MIMO detection," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Anaheim, CA, USA, Nov. 2018, pp. 1–5.

[4] E. Nachmani, Y. Be'ery, and D. Burshtein, "Learning to decode linear codes using deep learning," in *Proc. 54th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2016, pp. 341–346.

[5] H. Kim, Y. Jiang, R. Rana, S. Kannan, S. Oh, and P. Viswanath, "Communication algorithms via deep learning," 2018, *arXiv:1805.09317*. [Online]. Available: https://arxiv.org/abs/1805.09317

[6] H. Ye, L. Liang, and G. Y. Li, "Circular convolutional auto-encoder for channel coding," in *Proc. IEEE 20th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2019, pp. 1–5.

[7] T. Gruber, S. Cammerer, J. Hoydis, and S. ten Brink, "On deep learning-based channel decoding," in *Proc. 51st Annu. Conf. Inf. Sci. Syst. (CISS)*, Baltimore, MD, USA, Mar. 2017, pp. 1–6.

[8] S. Cammerer, T. Gruber, J. Hoydis, and S. ten Brink, "Scaling deep learning-based decoding of polar codes via partitioning," 2017, *arXiv:1702.06901*. [Online]. Available: https://arxiv.org/abs/1702.06901

[9] W. Xu, Z. Wu, Y.-L. Ueng, X. You, and C. Zhang, "Improved polar decoder based on deep learning," in *Proc. IEEE Int. Workshop Signal Process. Syst. (SiPS)*, Oct. 2017, pp. 1–6.

[10] 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.

[11] X. Gao, S. Jin, C.-K. Wen, and G. Y. Li, "ComNet: Combination of deep learning and expert knowledge in OFDM receivers," *IEEE Commun. Lett.*, vol. 22, no. 12, pp. 2627–2630, Dec. 2018.

[12] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018.

[13] N. Farsad, M. Rao, and A. Goldsmith, "Deep learning for joint source-channel coding of text," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 1–5.

[14] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[15] S. Dörner, S. Cammerer, J. Hoydis, and S. ten Brink, "Deep learning-based communication over the air," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 132–143, Feb. 2018.

[16] A. Felix, S. Cammerer, S. Dörner, J. Hoydis, and S. ten Brink, "OFDM-autoencoder for end-to-end learning of communications systems," in *Proc. IEEE 19th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jun. 2018, pp. 1–5.

[17] F. A. Aoudia and J. Hoydis, "End-to-end learning of communications systems without a channel model," 2018, *arXiv:1804.02276*. [Online]. Available: https://arxiv.org/abs/1804.02276

[18] V. Raj and S. Kalyani, "Backpropagating through the air: Deep learning at physical layer without channel models," *IEEE Commun. Lett.*, vol. 22, no. 11, pp. 2278–2281, Nov. 2018.

[19] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, Dec. 2010.

[20] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. ICML*, Jul. 2008, pp. 1096–1103.

[21] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*. [Online]. Available: https://arxiv.org/abs/1312.6114

[22] H. Ye, G. Y. Li, B.-H. Juang, and K. Sivanesan, "Channel agnostic end-to-end learning based communication systems with conditional GAN," in *Proc. IEEE Globecom Workshops*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–5.

[23] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*. [Online]. Available: https://arxiv.org/abs/1411.1784

[24] T. O'Shea, T. Roy, and N. West, "Approximating the void: Learning stochastic channel models from observation with variational generative adversarial networks," 2018, *arXiv:1805.06350*. [Online]. Available: https://arxiv.org/abs/1805.06350

[25] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. NIPS*, Dec. 2014, pp. 2672–2680.

[26] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*. [Online]. Available: https://arxiv.org/abs/1701.07875

[27] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2017, pp. 5767–5777.

[28] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," 2018, *arXiv:1809.11096*. [Online]. Available: https://arxiv.org/abs/1809.11096

[29] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4681–4690.

[30] B. Zhu, J. Wang, L. He, and J. Song, "Joint transceiver optimization for wireless communication PHY with convolutional neural network," 2018, *arXiv:1808.03242*. [Online]. Available: https://arxiv.org/abs/1808.03242

[31] A. Hamalainen and J. Henriksson, "A recurrent neural decoder for convolutional codes," in *Proc. IEEE ICC*, vol. 2, Jun. 1999, pp. 1305–1309.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

[33] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.

[34] Y. Lecun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[35] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 2, pp. 260–269, Apr. 1967.

[36] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.

[37] Q. Wang, S. R. Kulkarni, and S. Verdu, "Divergence estimation for multidimensional densities via k-nearest-neighbor distances," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2392–2405, May 2009.

[38] P. Kyösti *et al.*, "WINNER II channel models," Eur. Commission, Brussels, Belgium, IST-WINNER, Tech. Rep. D1.1.2, Sep. 2007.

Hao Ye (Student Member, IEEE) received the B.E. degree in information engineering from Southeast University, Nanjing, China, in 2012, and the M.E. degree in electrical engineering from Tsinghua University, Beijing, China, in 2015. He is currently pursuing the Ph.D. degree in electrical engineering with the Georgia Institute of Technology, Atlanta, GA, USA. His research interests include machine learning for physical and MAC layer communications.

Le Liang (Member, IEEE) received the B.E. degree in information engineering from Southeast University, Nanjing, China, in 2012, the M.A.Sc. degree in electrical engineering from the University of Victoria, Victoria, BC, Canada, in 2015, and the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2018.

Since 2019, he has been a Research Scientist with Intel Labs, Hillsboro, OR, USA. His general research interests include wireless communications, signal processing, and machine learning. He received the Best Paper Award from IEEE/CIC ICCC in 2014. He is also an Editor of the IEEE COMMUNICATIONS LETTERS.

Geoffrey Ye Li (Fellow, IEEE) is currently a Professor with the Georgia Institute of Technology, Atlanta, GA, USA. Before moving to Georgia Tech, he was with AT&T Labs—Research at Red Bank, New Jersey, as a Senior Member Technical Staff and then a Principal Technical Staff Member, from 1996 to 2000, and a Post-Doctoral Research Associate with the University of Maryland at College Park, Maryland, from 1994 to 1996. His general research interests include statistical signal processing and machine learning for wireless communications. In these areas, he has published over 500 journal and conference papers in addition to over 40 granted patents. His publications have been cited around 40,000 times and he has been recognized as the World's Most Influential Scientific Mind, also known as a Highly Cited Researcher, by Thomson Reuters almost every year.

He was awarded IEEE Fellow for his contributions to signal processing for wireless communications in 2005. He received several prestigious awards from the IEEE Signal Processing Society (Donald G. Fink Overview Paper Award in 2017), the IEEE Vehicular Technology Society (James Evans Avant Garde Award in 2013 and Jack Neubauer Memorial Award in 2014), and the IEEE Communications Society (Stephen O. Rice Prize Paper Award in 2013, Award for Advances in Communication in 2017, and Edwin Howard Armstrong Achievement Award in 2019). He also received 2015 Distinguished Faculty Achievement Award from the School of Electrical and Computer Engineering, Georgia Tech. He has been involved in editorial activities for over 20 technical journals, including the founding Editor-in-Chief of IEEE 5G Tech Focus. He has organized and chaired many international conferences, including a technical program Vice-Chair of IEEE ICC'03, a technical program Co-Chair of IEEE SPAWC'11, a General Chair of IEEE GlobalSIP'14, a technical program Co-Chair of IEEE VTC'16 (Spring), and a General Co-Chair of IEEE VTC'19 (Fall).

Biing-Hwang Juang (Life Fellow, IEEE) received the Ph.D. degree from the University of California, Santa Barbara, CA, USA. He is currently the Motorola Foundation Chair Professor and a Georgia Research Alliance Eminent Scholar with the Georgia Institute of Technology, Atlanta, GA, USA. He is also enlisted as an Honorary Chair Professor with several renowned universities. He had conducted research work with Speech Communications Research Laboratory and Signal Technology, Inc., in the late 1970 on a number of government supported research projects and with Bell Labs from 1980 to 1990 until he joined Georgia Tech in 2002. His notable accomplishments include development of vector quantization for voice applications, voice coders at extremely low bit rates (800 bps and 300 bps), robust vocoders for satellite communications, fundamental algorithms in signal modeling for automatic speech recognition, mixture hidden Markov models, discriminative methods in pattern recognition and machine learning, stereo-and multiphonic teleconferencing, and a number of voice-enabled interactive communication services. He was the Director of Acoustics and Speech Research at Bell Labs from 1996 to 2001. He has authored extensively, including the book-Fundamentals of Speech Recognition, coauthored with L. R. Rabiner, and holds nearly two dozen patents. He was elected as a Bell Labs Fellow in 1999, a member of the U.S. National Academy of Engineering in 2004, and an Academician of the Academia Sinica in 2006. He was named recipient of the IEEE Field Award in Audio, Speech and Acoustics, the J. L. Flanagan Medal, and a Charter Fellow of the National Academy of Inventors, in 2014. He was a recipient of the Technical Achievement Award from the IEEE Signal Processing Society in 1998 for contributions to the field of speech processing and communications and the Third Millennium Medal from the IEEE in 2000. He was a recipient of two Best Senior Paper awards in 1993 and 1994, respectively, and a Best Paper Award in 1994, from the IEEE Signal Processing Society. He served as the Editor-in-Chief of the IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING from 1996 to 2002.