Linear MIMO Precoders With Finite Alphabet Inputs via Stochastic Optimization and Deep Neural Networks (DNNs)

Shusen Jing, *Student Member, IEEE*, and Chengshan Xiao, *Fellow, IEEE*

***Abstract*—In this paper, we investigate designs of linear precoders for vector *Gaussian* channels via stochastic optimizations and deep neural networks (DNNs). We assume that the channel inputs are drawn from practical finite alphabets, and we search for precoders maximizing the mutual information between channel inputs and outputs. Though the problem is generally non-convex, we prove that when the right singular matrix of precoder is fixed, any local optima of this problem is a global optima. Based on this fact, an efficient projected stochastic gradient descent (PSGD) algorithm is designed to search the optimal precoders. Moreover, to reduce the complexity of calculating *a posterior* means involved in gradients calculation, K-best algorithm is adopted to make approximations of *a posterior* means with negligible loss of accuracy. Furthermore, to avoid explicit calculation of mutual information and its gradients, DNN-based autoencoders (AEs) are constructed for this precoding task, and an efficient training algorithm is proposed. We also prove that the AEs, with 'softmax' activation function and 'categorical cross entropy' loss, maximize the mutual information under reasonable assumptions. Then, in order to extend the AE methods to large scale systems, 'sigmoid' activation function and 'binary cross entropy' loss are used such that the size of AEs will not grow prohibitively large. We prove that this maximizes a lower bound of the mutual information under reasonable assumptions. Finally, to make the precoders practical for high speed wireless scenarios, we propose an offline training paradigm which trains DNNs to infer optimal precoders given channel state information instead of training online for every different channel. Simulation results show that all the proposed methods work well in maximizing mutual information and improving bit error rate (BER) performance.**

***Index Terms*—MIMO, linear precoders, finite alphabet, deep neural networks, autoencoders.**

I. Introduction

LINEAR precoding is an important way to improve the reliability and transmission rate of multiple-input multiple-output (MIMO) systems, and it has attracted much attention from researchers for recent decades [1], [2]. Under *Gaussian* noise assumption, the capacity of vector *Gaussian* channels is achieved with *Gaussian* channel inputs, and water-filling (WF) [3] is the optimal precoding method which maximizes mutual information and channel capacity. However, *Gaussian* channel inputs are rarely used in practical communication systems, instead of which finite alphabet channel inputs, such as quadrature amplitude modulation (QAM) and phase shift keying (PSK), are usually adopted. For this reason, more and more work began to study the precoding problem under the assumptions of finite alphabet inputs [3]–[9].

In [10], [11], an approximation of mutual information at low signal-to-noise ratio (SNR) is obtained, which implies that the optimal precoder is closed to the solution given by WF when SNR is low. [5] shows by simulations that it is inappropriate to treat finite alphabet inputs as *Gaussian* at high SNR in the precoding problems. It is shown that the WF precoder can even reduce the amount of mutual information of a system with finite alphabet inputs. [12] proposes mercury/water-filling (MWF), which is the optimal power allocation method for parallel additive white *Gaussian* noise (AWGN) channels with independent parallel transceivers and channel inputs. Let $\mathbf{H}$, $\mathbf{G}$ denote the channel matrix and precoding matrix, respectively. For a general vector *Gaussian* channel, MWF is equivalent to maximizing the mutual information $\mathcal{I}(\mathbf{x};\mathbf{y})$ with respect to the singular values $\mathbf{\Sigma_G}$ of $\mathbf{G}$, while fixing its left singular matrix $\mathbf{U_G}$ as the right singular matrix $\mathbf{V_H}$ of $\mathbf{H}$, and fixing its right singular matrix $\mathbf{V_G}$ as identity matrix $\mathbf{I}$. Apparently, MWF gives up the part of searching space related to $\mathbf{V_G}$, so its performance for general vector *Gaussian* channels are not satisfactory as shown in [5].

To design better precoders, some results and methods have been recently proposed. [13] points out that the $\mathbf{W}=\mathbf{G}$ is a sufficient statistic of $\mathcal{I}(\mathbf{x};\mathbf{y})$, so it is sufficient to do optimization with respect to $\mathbf{W}$. [5] proves that $\mathcal{I}(\mathbf{x};\mathbf{y})$ is strictly concave to $\mathbf{\Sigma_G^2}$, and designs block coordinate descent algorithm with respect to $\mathbf{\Sigma_G^2}$ and $\mathbf{V_G}$. [14] derives optimality conditions of $\mathbf{G}$ based on manifolds approaches and does optimization with respect to $\mathbf{\Sigma_G^2}$ and $\mathbf{V_G}$. [15] proposes to maximize a lower bound of $\mathcal{I}$ to reduce the complexity.

Though the previous works show great performances in the precoding tasks, designing efficient and flexible algorithms is still challenging. Since the closed formed expression of $\mathcal{I}(\mathbf{x};\mathbf{y})$ with common finite alphabet inputs has not been found (if exists), the calculation of $\mathcal{I}(\mathbf{x};\mathbf{y})$ and its gradients relies on Monte-Carlo methods, which is extremely expensive for large systems. Most of existing works require accurate gradients information, so the gradients calculation in their algorithms takes a very long time

Manuscript received December 3, 2020; revised June 10, 2021; accepted July 5, 2021. Date of publication July 14, 2021; date of current version August 27, 2021. The associate editor coordinating the review of this manuscript and approving it for publication was Miss Abla Kammoun. This work was supported in part by National Science Foundation under Grant ECCS-1827592. *(Corresponding author: Chengshan Xiao.)*

The authors are with the Department of Electrical and Computer Engineering, Lehigh University, Bethlehem, PA 18015 USA (e-mail: shj218@lehigh.edu; xiaoc@lehithigh.edu).

Digital Object Identifier 10.1109/TSP.2021.3096466

with large systems. Moreover, most of the existing algorithms are not flexible in time, since the time interval between two updates of precoder is large due to the gradients calculation in each iterations.

To the best knowledge of the authors, none of the existing work considers stochastic nature of the optimization process. Since Monte-Carlo method is used to calculate mutual information and its gradients, the algorithm should be designed in a view of stochastic optimization. In this paper, we prove that doing optimization directly with respect to $\mathbf{G}$ has similar efficacy with algorithms in [5]. Then a projected stochastic gradient descent (PSGD) [16] algorithm is adopted to find optimal precoders without evaluating $\mathcal{I}(\mathbf{x};\mathbf{y})$. Note that, the projection operations in the PSGD can be easily conducted by dividing $\mathbf{G}$ with a multiple of its Frobenius norm $||\mathbf{G}||_F$, and the algorithm is guaranteed for convergence.

To calculate stochastic gradients of mutual information, *a posterior* means conditioned on different received signal samples are required. However, the complexity of calculating *a posterior* mean grows exponentially with the number of transmitting antennas, which makes existing precoding algorithms intractable. For this reason, K-best [17], [18] algorithm is adopted to get close approximations of *a posterior* means with a tolerable complexity, and simulation results show that this approximation has negligible performance loss.

With the help of stochastic optimization techniques in training, deep neural networks (DNNs) show a great success in a variety of tasks [19], [20]. Among the studies of DNNs, autoencoder (AE) is one of the most promising topics todate [21]–[25]. It can automatically design encoders and decoders in pair for the purpose of messages compression and decompression. [26] shows that AE can also be used in a communication system for reliable and high rate transceiver designs, and optimal transceiver can be obtained via training. In this paper, we explore the possibility of designing linear precoders with AEs. An advantage of the method is that explicitly computation of $\mathcal{I}(\mathbf{x};\mathbf{y})$ and its gradients can be completely avoided, and modern software and hardware technologies help accelerate the training processes [27]. We show that, under reasonable assumption, training an AE, with 'softmax' and 'categorical cross entropy' as activation function and loss, respectively, is equivalent to maximizing the mutual information $\mathcal{I}(\mathbf{x};\mathbf{y})$. For larger MIMO systems, we modify the AEs by using 'sigmoid' and 'binary cross entropy' as activation function and loss, respectively, so the size of AEs will not grow prohibitively large. We show that training such an AE is equivalent to maximizing a lower bound of the mutual information. Though greatly improve the efficiency of finding optimal precoders, PSGD and AEs require online training with new generated samples for every different channel, which is sometimes still impractical. To make mutual information-driven precoder more practical, we propose an offline training paradigm, in which DNNs are trained to infer optimal precoders given channel state information.

Simulation results show that PSGD, AEs and the proposed DNN successfully improve the mutual information, and the proposed DNN significantly reduced the time complexity.

The rest of the paper is organized as follows. Section II gives the system model and preliminaries of precoded MIMO systems. Section III shows the theoretical results obtained in this paper. Section IV shows the PSGD based precoder design. Section V shows the AE and DNN based precoder design. Section VI shows the numerical results. Section VII finally concludes this paper.

II. System Model and Preliminaries

Consider a linear precoder involved MIMO transmission process described by

$$\mathbf{y} = \mathbf{HGx} + \mathbf{n} \tag{1}$$

where $\mathbf{y} \in \mathbb{C}^{N_r \times 1}$ is the received signal; $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$ is the channel matrix; $\mathbf{G} \in \mathbb{C}^{N_t \times N_t}$ is the linear precoding matrix; $\mathbf{x} \in \mathbb{C}^{N_t \times 1}$ is the transmitted signal with $\mathsf{E}[\mathbf{x}\mathbf{x}^h] = \mathbf{I}$ where $\mathbf{I}$ is the identity matrix; $\mathbf{n} \in \mathbb{C}^{N_r \times 1}$ is additive white *Gaussian* noise (AWGN) with zero mean and covariance $\sigma^2\mathbf{I}$.

In this process, we assume that the entries of $\mathbf{x}$ are drawn from a constellation $\mathcal{W}$ independently, such as phase-shift keying (PSK) and quadrature amplitude modulation (QAM), so the alphabet of channel inputs can be expressed as $\mathcal{X} = \mathcal{W}^{N_t}$. In this way, every $\log_2|\mathcal{W}|$ bits are mapped to a symbol in $\mathcal{W}$, and every $N_t \log_2|\mathcal{W}|$ bits are mapped to a vector in $\mathcal{X}$. With discrete finite alphabet $\mathcal{X}$, the mutual information $\mathcal{I}(\mathbf{x};\mathbf{y})$ is expressed as

$$\mathcal{I}(\mathbf{x};\mathbf{y}) = N_t \log_2|\mathcal{W}| - \frac{1}{|\mathcal{W}|^{N_t}} \sum_{\mathbf{x}_a \in \mathcal{X}} \mathsf{E}_{\mathbf{n}} \left\{ \log_2 \sum_{\mathbf{x}_b \in \mathcal{X}} e^{-d_{a,b}} \right\} \tag{2}$$

where $d_{a,b} = \frac{1}{\sigma^2}(||\mathbf{HG}(\mathbf{x}_a - \mathbf{x}_b) + \mathbf{n}||^2 - ||\mathbf{n}||^2)$.

The task of designing the precoding matrix $\mathbf{G}$ to maximize the mutual information $\mathcal{I}(\mathbf{x};\mathbf{y})$ (minimize minus mutual information) can be formulated as

$$\begin{aligned} &\min_{\mathbf{G}} -\mathcal{I}(\mathbf{x};\mathbf{HGx}+\mathbf{n}) \\ &\text{s.t. } \mathsf{Tr}\{\mathbf{G}^h\mathbf{G}\} \le N_t \end{aligned} \tag{3}$$

where the constraint is used to limit the average energy of channel inputs. We further define two important identities: conditional minimum mean square error (MMSE) matrix

$$\mathbf{\Phi}(\mathbf{y}) = \mathsf{E}[(\mathbf{x} - \mathsf{E}[\mathbf{x}|\mathbf{y}])(\mathbf{x} - \mathsf{E}[\mathbf{x}|\mathbf{y}])^h|\mathbf{y}] \tag{4}$$

and MMSE matrix

$$\mathbf{\Phi} = \mathsf{E}[\mathbf{\Phi}(\mathbf{y})]. \tag{5}$$

Denote the the singular value decomposition (SVD) of $\mathbf{H}$ and $\mathbf{G}$ as $\mathbf{H} = \mathbf{U_H}\mathbf{\Sigma_H}\mathbf{V_H}^h$ and $\mathbf{G} = \mathbf{U_G}\mathbf{\Sigma_G}\mathbf{V_G}^h$, respectively. [5, Thm 1] proves that $\mathbf{W} = \mathbf{G}^h\mathbf{H}^h\mathbf{HG}$ is a sufficient statistic of $\mathcal{I}(\mathbf{x};\mathbf{HGx}+\mathbf{n})$, and the the problem (3) is equivalent to

$$\begin{aligned} \min_{\mathbf{V_G},\mathbf{\Sigma_G}} \; & -\mathcal{I}(\mathbf{x};\mathbf{\Sigma_H}\mathbf{\Sigma_G}\mathbf{V_G}^h\mathbf{x}+\mathbf{n}) \\ s.t. \; & \mathbf{V_G} \in o(N_t) \\ & \mathbf{\Sigma_G} \in d(N_t) \\ & \mathsf{Tr}\{\mathbf{\Sigma_G}^2\} \le N_t \end{aligned} \tag{6}$$

by letting $\mathbf{U_G} = \mathbf{V_G}^h$ without loss of generality. According to [5, Thm 2], $\mathcal{I}(\mathbf{x};\mathbf{y})$ is concave to $\mathbf{\Sigma_G}^2$, and the feasible set of

$\mathbf{\Sigma}_\mathbf{G}^2$ is convex, so gradient descent converges to the optimal $\mathbf{\Sigma}_\mathbf{G}^2$ with a fixed $\mathbf{V_G}$. Based on this fact, [5] proposes a block coordinates descent algorithm with respect to $\mathbf{\Sigma}_\mathbf{G}^2$ and $\mathbf{V_G}$ to solve the following the equivalent problem

$$\begin{aligned} \min_{\mathbf{V_G},\mathbf{\Sigma}_\mathbf{G}^2} \; & -\mathcal{I}(\mathbf{x};\mathbf{\Sigma_H}\mathbf{\Sigma_G}\mathbf{V}_\mathbf{G}^h\mathbf{x}+\mathbf{n}) \\ s.t. \quad & \mathbf{V_G} \in o(N_t) \\ & \mathbf{\Sigma}_\mathbf{G}^2 \in d(N_t) \\ & \mathbf{\Sigma}_\mathbf{G}^2 \succcurlyeq 0 \\ & \mathsf{Tr}\{\mathbf{\Sigma}_\mathbf{G}^2\} \le N_t \end{aligned} \tag{7}$$

where $o(N_t)$ is the set of N_t dimensional unit matrices; $d(N_t)$ is the set of N_t dimensional diagonal matrices.

III. Theoretical Results

In this section, we will show that maximizing $\mathcal{I}(\mathbf{x};\mathbf{y})$ with respect to $\mathbf{G}$ directly has similar efficacy with maximizing $\mathcal{I}(\mathbf{x};\mathbf{y})$ along $\mathbf{\Sigma}_\mathbf{G}^2$ and $\mathbf{V_G}$. Denote the eigenvalue decomposition (EVD) of $\mathbf{\Phi}$ as $\mathbf{\Phi}=\mathbf{U}_\mathbf{\Phi}\mathbf{\Sigma}_\mathbf{\Phi}\mathbf{U}_\mathbf{\Phi}^h$, the following proposition shows the structure of optimal $\mathbf{G}$:

Proposition 1: Let $\mathbf{G}$ be a critical point of problem described in eq. (3), then

$$\mathbf{U_G} = \mathbf{V_H}\mathbf{\Pi}_1\bar{\mathbf{I}}_1 \tag{8}$$

where $\mathbf{\Pi}_1$ is any permutation matrix and $\bar{\mathbf{I}}_1$ is any matrix in a form of

$$\bar{\mathbf{I}}_1 = \begin{bmatrix} e^{j\theta_1} & 0 & 0 \\ 0 & e^{j\theta_2} & 0 \\ 0 & 0 & e^{j\theta_3} \end{bmatrix} \tag{9}$$

where θ_1, θ_2 and θ_3 are arbitrary real number. Additionally, we have

$$\mathbf{V_G} = \mathbf{U}_\mathbf{\Phi}\mathbf{\Pi}_2\bar{\mathbf{I}}_2 \tag{10}$$

where $\mathbf{\Pi}_2$ and $\bar{\mathbf{I}}_2$ have similar definition with $\mathbf{\Pi}_1$ and $\bar{\mathbf{I}}_1$, respectively.

Proof: The proof is similar to that of [14, Thm 2], and we omit details here for brevity.

We next show a simple verification to the Proposition 1. Let the covariance matrix of $\mathbf{n}$ be $\mathbf{Q_n}=\sigma^2\mathbf{I}$, according to [28], the derivative of $\mathcal{I}$ is

$$\nabla_\mathbf{G}\mathcal{I}(\mathbf{x};\mathbf{HGx}+\mathbf{n}) = \frac{1}{\sigma^2}\mathbf{H}^h\mathbf{HG}\mathbf{\Phi}. \tag{11}$$

A necessary condition for $\mathbf{G}$ to be critical point of problem (3) is

$$\frac{1}{\sigma^2}\mathbf{H}^h\mathbf{HG}\mathbf{\Phi} = \lambda\mathbf{G} \tag{12}$$

where $\lambda>0$ is *Lagrangian* multiplier. (It can be easily verified that $\lambda\neq 0$ since the constraint must be active in this problem.) Plug in the decomposition of $\mathbf{H}$, $\mathbf{G}$ and $\mathbf{\Phi}$ to eq. (12) and make an arrangement, we have

$$\frac{1}{\sigma^2}\mathbf{U}_\mathbf{G}^h\mathbf{V_H}\mathbf{\Sigma}_\mathbf{H}^2\mathbf{V}_\mathbf{H}^h\mathbf{U_G}\mathbf{\Sigma_G}\mathbf{V}_\mathbf{G}^h\mathbf{U}_\mathbf{\Phi}\mathbf{\Sigma}_\mathbf{\Phi}\mathbf{U}_\mathbf{\Phi}^h\mathbf{V_G} = \lambda\mathbf{\Sigma_G}. \tag{13}$$

According to eq. (13), the left hand side (LHS) of it needs to be diagonal. Now let us plug in the results in Proposition 1 to check their correctness. Plug in eq. (8) and (10) to eq. (13), we obtain

$$\frac{1}{\sigma^2}\bar{\mathbf{I}}_1^h\mathbf{\Pi}_1^h\mathbf{\Sigma}_\mathbf{H}^2\mathbf{\Pi}_1\bar{\mathbf{I}}_1\mathbf{\Sigma_G}\bar{\mathbf{I}}_2^h\mathbf{\Pi}_2^h\mathbf{\Sigma}_\mathbf{\Phi}\mathbf{\Pi}_2\bar{\mathbf{I}}_2 = \lambda\mathbf{\Sigma_G}. \tag{14}$$

Denote $\tilde{\mathbf{\Sigma}}_\mathbf{H}^2 = \bar{\mathbf{I}}_1^h\mathbf{\Pi}_1^h\mathbf{\Sigma}_\mathbf{H}^2\mathbf{\Pi}_1\bar{\mathbf{I}}_1$, $\tilde{\mathbf{\Sigma}}_\mathbf{\Phi} = \bar{\mathbf{I}}_2^h\mathbf{\Pi}_2^h\mathbf{\Sigma}_\mathbf{\Phi}\mathbf{\Pi}_2\bar{\mathbf{I}}_2$, and it is easy to observe that $\tilde{\mathbf{\Sigma}}_\mathbf{H}^2$ and $\tilde{\mathbf{\Sigma}}_\mathbf{\Phi}$ are diagonal, then eq. (14) can be simplified to

$$\frac{1}{\sigma^2}\tilde{\mathbf{\Sigma}}_\mathbf{H}^2\mathbf{\Sigma_G}\tilde{\mathbf{\Sigma}}_\mathbf{\Phi} = \lambda\mathbf{\Sigma_G} \tag{15}$$

whose LHS turns out to be diagonal. In this way, we checked the results in Proposition 1 make sense.

Next we provide a relationship between $\mathbf{\Sigma_G}$ and $\mathcal{I}(\mathbf{x};\mathbf{y})$ via the following theorem.

Corollary 1: For the model in eq. (6), let $\mathbf{V_G}$ be fixed, then any local minima of the problem

$$\begin{aligned} \min_{\mathbf{\Sigma_G}} \; & -\mathcal{I}(\mathbf{x};\mathbf{\Sigma_H}\mathbf{\Sigma_G}\mathbf{V}_\mathbf{G}^h\mathbf{x}+\mathbf{n}) \\ s.t. \quad & \mathbf{\Sigma_G} \in d(N_t) \\ & \mathsf{Tr}\{\mathbf{\Sigma}_\mathbf{G}^2\} \le N_t \end{aligned} \tag{16}$$

is a global minima. Furthermore, if $\det(\mathbf{\Sigma}_\mathbf{H}^2)>0$, then the only global maximum is $\mathbf{\Sigma_G}=\mathbf{0}$, and other critical points (only including saddle points and local minima) are on the boundary $\mathsf{Tr}\{\mathbf{\Sigma}_\mathbf{G}^2\}=N_t$.

Proof: See Appendix. □

This result is an extension of [5, Thm 2]. It reveals the symmetry of objective landscape in problem (3) and relationship between local minimas.

Corollary 2: Under the model expressed in eq. (6), $\mathbf{G}$ is a local minima of problem (3) if and only if $\mathbf{G}$ is a local minima of problem (7).

Proof: Apply Proposition 1 and Corollary 1, the results in straight forward. Details are omitted here for brevity. □

According to Corollary 1 and Proposition 1, if the gradient descent algorithm is designed properly such that it can escape saddle points efficiently, then at any local minima $\mathbf{G}^\star$ of problem (3), $\mathbf{\Sigma}_\mathbf{G}^\star$ is the global minima when the right singular matrix is fixed as $\mathbf{V}_\mathbf{G}^\star$.

IV. Precoder Design via Projected Stochastic Gradient Descent

In this section, we design efficient projected stochastic gradient descent (PSGD) algorithms to solve the problem (3). In the algorithms, we update $\mathbf{G}$ directly in each iteration instead of updating $\mathbf{U_G}$, $\mathbf{\Sigma_G}$ and $\mathbf{V_G}$ separately. This design is motivated by Corollary 2 and the fact that PSGD escapes saddle points efficiently. Besides, the projection operations involved in the algorithms are simple, so negligible additional complexity is introduced.

Algorithm 1: PSGD Precoding.

Initialization: Initialize $\mathbf{G}_0$ as a complex value matrix
for $t \leftarrow 1$ **to** $itermax$ **do**
  Generate B independent samples $\{\mathbf{y}^j, \mathbf{x}^j\}_{j=1}^B$ through eq. (1) with $\mathbf{G}_{t-1}$
  $\mathbf{\Omega}_t = \frac{1}{\sigma^2}\mathbf{H}^h\mathbf{H}\mathbf{G}\frac{1}{B}\sum_{j=1}^B(\mathbf{x}^j - \mathsf{E}[\mathbf{x}|\mathbf{y}^j])(\mathbf{x}^j - \mathsf{E}[\mathbf{x}|\mathbf{y}^j])^h$
  $\mathbf{P}_t = \mathbf{G}_{t-1} + \frac{\mu}{\sqrt{t}}\mathbf{\Omega}_t$
  $\mathbf{G}_t = \sqrt{N_t}\mathbf{P}_t/||\mathbf{P}_t||_F$
  $\bar{\mathbf{P}}_t = \frac{1}{t}\mathbf{G}_t + \frac{t-1}{t}\mathbf{G}_{t-1}$
end for
$\bar{\mathbf{G}}_{itermax} = \sqrt{N_t}\bar{\mathbf{P}}_{itermax}/||\bar{\mathbf{P}}_{itermax}||_F$
Output: $\bar{\mathbf{G}}_{itermax}$

Algorithm 2: K-Best *a Posterior* Mean Calculation.

Initialization: Initialize a tree root and set $D_0 = 0$ for all paths.
for $i \leftarrow 1$ **to** N_t **do**
  Extend each reserved path with all elements in $\mathcal{W}$.
  Calculate D_i for each path according to eq. (21).
  Reserve K paths with smallest D_i.
end for
Let $\{\hat{\mathbf{x}}_k\}_{k=1}^K$ and $\{D_{k,N_t}\}_{k=1}^K$ be the paths and their corresponding cost at the N_t-th level.
Output: $\hat{\mathbf{x}} = \frac{\sum_{k=1}^K \hat{\mathbf{x}}_k e^{-D_{k,N_t}/\sigma^2}}{\sum_{k=1}^K e^{-D_{k,N_t}/\sigma^2}}$

A. Projected Stochastic Gradient Descent

Notice that the MMSE matrix $\mathbf{\Phi}$ can be written as

$$\begin{aligned}\mathbf{\Phi} &= \mathsf{E}_{\mathbf{x},\mathbf{y}}\left[(\mathbf{x} - \mathsf{E}[\mathbf{x}|\mathbf{y}])(\mathbf{x} - \mathsf{E}[\mathbf{x}|\mathbf{y}])^h\right] \\ &= \mathbf{I} - \mathsf{E}_{\mathbf{y}}\left[\mathsf{E}[\mathbf{x}|\mathbf{y}]\mathsf{E}[\mathbf{x}|\mathbf{y}]^h\right]. \end{aligned} \tag{17}$$

Given a batch of B independent samples $\{\mathbf{x}^j, \mathbf{y}^j\}_{j=1}^B$ generated by the model in eq. (1), a noise gradient $\mathbf{\Omega}$ can be calculated according to eq. (11) as

$$\mathbf{\Omega} = \frac{1}{\sigma^2}\mathbf{H}^h\mathbf{H}\mathbf{G}\frac{1}{B}\sum_{j=1}^B(\mathbf{x}^j - \mathsf{E}[\mathbf{x}|\mathbf{y}^j])(\mathbf{x}^j - \mathsf{E}[\mathbf{x}|\mathbf{y}^j])^h. \tag{18}$$

One can easily verify that $\mathsf{E}[\mathbf{\Omega}] = \nabla_{\mathbf{G}}\mathcal{I}(\mathbf{x}; \mathbf{H}\mathbf{G}\mathbf{x} + \mathbf{n})$, and $\mathsf{E}[\mathbf{x}|\mathbf{y}]$ is calculated as

$$\mathsf{E}[\mathbf{x}|\mathbf{y}] = \sum_{\mathbf{x}_a \in \mathcal{X}} \frac{\mathbf{x}_a e^{-||\mathbf{y} - \mathbf{H}\mathbf{G}\mathbf{x}_a||^2/\sigma^2}}{\sum_{\mathbf{x}_b \in \mathcal{X}} e^{-||\mathbf{y} - \mathbf{H}\mathbf{G}\mathbf{x}_b||^2/\sigma^2}}. \tag{19}$$

Let $itermax$ denote the maximum number of iterations, and μ denote the initial step size. The PSGD algorithm is summarized as follows.

In the algorithm, $||*||_F$ refers to as Frobenius norm, which is defined as $||\mathbf{G}||_F = \sqrt{\mathsf{Tr}\{\mathbf{G}^h\mathbf{G}\}}$. Note that $\bar{\mathbf{P}}_T$ is the average of $\mathbf{G}_t$ the in the former T iterates, and step size $\frac{\mu}{\sqrt{t}}$ decreases with iterations index t. This two tricks help reduce the fluctuation of iterates around the local solutions.

Since the norm of each entry of $\mathbf{\Phi}$ is bounded (because $\mathcal{W}$ is bounded), the gradient $\frac{1}{\sigma^2}\mathbf{H}^h\mathbf{H}\mathbf{G}\mathbf{\Phi}$ is bound, so the variance of stochastic gradient is bounded. Therefore according to [16], when $\bar{\mathbf{G}}_t$ is in a region with local convexity, convergence is guaranteed.

It is point out that this method works with any finite alphabet $\mathcal{X}$ (not necessarily $\mathcal{W}^{N_t}$) with arbitrarily probability mass distribution.

B. Simplified PSGD

In Algorithm 1, calculating $\mathsf{E}[\mathbf{x}|\mathbf{y}]$ is very expensive in large systems, because it involves enumeration of all possible transmitted signals in $\mathcal{X}$, whose size $|\mathcal{X}| = |\mathcal{W}|^{N_t}$ grows exponentially with N_t. For this reason, K-best [17] algorithm is adopted to find close approximations of $\mathsf{E}[\mathbf{x}|\mathbf{y}]$ with an affordable complexity.

Denote the QR decomposition of $\mathbf{H}\mathbf{G}$ as $\mathbf{H}\mathbf{G} = \mathbf{Q}\mathbf{R}$, where $\mathbf{Q} \in \mathbb{C}^{N_r \times N_t}$ consists of columns of a unit matrix, and $\mathbf{R} \in \mathbb{C}^{N_t \times N_t}$ is a lower triangular matrix. Then eq. (1) can be rewritten as

$$\mathbf{z} = \mathbf{R}\mathbf{x} + \tilde{\mathbf{n}} \tag{20}$$

where $\mathbf{z} = \mathbf{Q}^h\mathbf{y}$ and $\tilde{\mathbf{n}} = \mathbf{Q}^h\mathbf{n}$. By the definition of $\mathbf{Q}$, the covariance matrix of $\tilde{\mathbf{n}}$ is also $\sigma^2\mathbf{I}$. The K-best algorithm is a width-first pruned tree search process. At each level of the tree, only K paths with the smallest cost are reserved. When the tree is extended to the next level, each one of the K reserved paths at the current level are extended to $|\mathcal{W}|$ new paths, so there are totally $K|\mathcal{W}|$ paths at the next level. Then only K paths with smallest cost among all $K|\mathcal{W}|$ paths are reserved. After that the tree can be extended again similarly. Denote the cost a path $\mathbf{x}$ at the i-th level as D_i, then we have

$$D_i = D_{i-1} + \left| z(i) - \sum_{l=1}^{i} R(i,l)x(l) \right|^2 \tag{21}$$

where $R(i,l)$ refers to the entry of $\mathbf{R}$ at the i-th row and l-th column, and the same rule can be extended to other variables. Note that at the i-th level, the summation in eq. (21) stopped at $l = i$ instead of N_t, because $\mathbf{R}$ is lower triangular. The algorithm is summarized as follows:

Fig. 1 shows an example of this tree search process with $\mathcal{W} = \{-1, +1\}$ and $K = 4$. The tree is first extended to 2 and 4 paths at the first and second level, respectively. The 4 paths at the second level are extended to 8 paths at the third level, but only the 4 paths with the smallest cost are reserved at the third level, and others are pruned. Similar things happen when it is extended to other levels. The main idea of this algorithm is to enumerate only K most likely channel inputs instead of all elements in $\mathcal{X}$ when calculating the $\mathsf{E}[\mathbf{x}|\mathbf{y}]$.

V. Precoder Design via Deep Neural Networks

In this section, we first explore the possibility of finding the optimal precoders with autoencoders (AEs) based on deep neural networks (DNNs). The benefits of using AEs are as follows: first, after training the AE, it can obtain not only optimal precoders

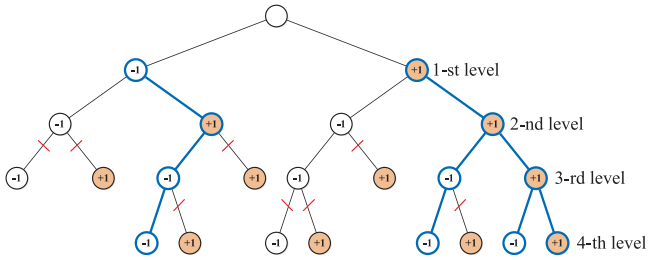


Fig. 1. An example of K-best algorithm with $K = 4$ and $\mathcal{W} = \{-1, +1\}$. The 1-st level reserved paths are $[-1]$ and $[+1]$; the 2-nd level reserved paths are $[-1, -1]$, $[-1, +1]$, $[+1, -1]$ and $[+1, +1]$; the 3-rd level reserved paths are $[-1, -1, +1]$, $[+1, -1, -1]$, $[+1, +1, -1]$ and $[+1, +1, +1]$; the 4-th level reserved paths are $[-1, -1, +1, -1]$, $[+1, +1, -1, -1]$, $[+1, +1, +1, -1]$ and $[+1, +1, +1, +1]$, which are marked with bold blue line.

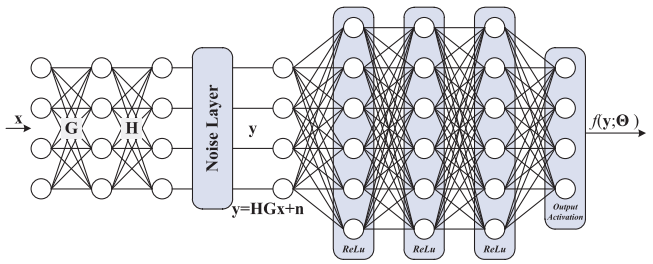


Fig. 2. Structure of DNN-based AE for precoding. The trainable parameters are $\mathbf{G}$ and $\mathbf{\Theta}$, where $\mathbf{\Theta}$ refers to all the weights and biases in layers at the right hand side (receiver side) of the AE.

but also optimal receivers; second, the explicit calculation of MMSE matrix $\mathbf{\Phi}$ is avoided; third, increasing number of modern technologies can be used to accelerate the training process. Note that the $\mathcal{X}$ can be any finite set (not necessarily $\mathcal{W}^{N_t}$), and noise is not required to be *Gaussian* in this method. After that, we design an offline training paradigm which trains DNNs to infer optimal precoder given channel state information. This method significantly reduces the time complexity of precoding and is more practical for high speed wireless scenarios.

Fig. 2 shows the structure of a DNN-based AE. As the input of AE, $\mathbf{x}$ is first multiplied with precoding matrix $\mathbf{G}$, which can be viewed as a group of trainable weights of a layer without bias and activation function. Then this temporary result $\mathbf{Gx}$ is multiplied with channel $\mathbf{H}$, which is non-trainable and then added with *Gaussian* noise. The receiver in this AE consists of multiple dense layers, and a particular activation function is implemented at the output layer to give the corresponding type of estimations of $\mathbf{x}$. To simplify the analysis of DNNs' behaviours, we made the following assumption.

Assumption 1: Let $f(\mathbf{y}; \mathbf{\Theta})$ be the parameterized function at the receiver of the AE, where $\mathbf{\Theta}$ are the trainable parameters (weights and biases) at the receiver. Let $f^\star(\mathbf{y})$ be the function that minimize the loss given $\mathbf{H}$ and $\mathbf{G}$. We assume that the distance between $f(\mathbf{y}; \mathbf{\Theta}^\star)$ and $f^\star(\mathbf{y})$ can be arbitrarily small, where $\mathbf{\Theta}^\star$ are the parameters obtained after training.

In this assumption, the 'loss' and 'distance' are not defined specifically. We assume that it holds for any losses and distances used in the following discussions. The assumption is based on the fact that the regression capability of DNNs is strong, and the regression error can be negligibly small.

Algorithm 3: Training the Autoencoder.

Initialization: Randomly initialize $\mathbf{\Theta}$ and $\mathbf{G}$ such that $\mathsf{Tr}\{\mathbf{G}^h\mathbf{G}\} \leq 1$.
for $i \leftarrow 1$ **to** $itermax$ **do**
  Generate B independent samples $\{\mathbf{x}^j\}_{j=1}^B$.
  Update $\mathbf{\Theta}$ and $\mathbf{G}$ with one Adam step.
  Update $\mathbf{G}$ with $\sqrt{N_t}\mathbf{G}/||\mathbf{G}||_F$
  Generate B independent samples $\{\mathbf{x}^j\}_{j=1}^B$.
  Update $\mathbf{G}$ with one SGD step.
  Update $\mathbf{G}$ with $\sqrt{N_t}\mathbf{G}/||\mathbf{G}||_F$
end for
Output: $\mathbf{G}$

A. Autoencoder Maximizing Mutual Information

The following proposition is proposed to show the equivalence between training the AE and maximizing the mutual information $\mathcal{I}(\mathbf{x}; \mathbf{y})$.

Proposition 2: Suppose Assumption 1 holds. Let the activation function of the output layer be 'softmax' and loss function be 'categorical cross entropy', then training the AE is nearly same with maximizing the mutual information $\mathcal{I}(\mathbf{x}; \mathbf{y})$.

Proof: See Appendix. □

To maximize the mutual information, the activation function of the output layer is chosen as 'softmax' and loss function is chosen as 'categorical cross entropy'. The training algorithm is summarized as follows Algorithm 3.

In the algorithm, the target is chosen as the one-hot representation of $\mathbf{x}$. Note that in each iteration, we update $\mathbf{G}$ with an additional SGD step after an Adam [29] step. The network is trained in this way due to the fact that training former layers of AEs requires more efforts.

B. Large Size Precoder Design via Autoencoder

When 'softmax' and 'categorical cross entropy' are adopted, the sizes of layers in the AE grow exponentially with the number of transmitting antennas N_t, which is very expensive in large systems. For this reason, 'sigmoid' and 'binary cross entropy' are chosen as activation function and loss in this subsection, respectively, so the sizes of layers will only grow linearly with N_t. Let the vector $\mathbf{b} \in \mathbb{B}^{N_t \log_2 |\mathcal{W}| \times 1}$ denote the binary messages corresponding to $\mathbf{x}$. We use $\mathbf{b}$ as target and train the AE with Algorithm 3. The following proposition points out the relationship between training this AE and maximizing mutual information $\mathcal{I}(\mathbf{x}; \mathbf{y})$.

Proposition 3: Suppose Assumption 1 holds. Let the activation function of the output layer be 'sigmoid' and loss function be 'binary cross entropy', then training the AE is nearly same with maximizing the following lower bound of $\mathcal{I}(\mathbf{x}; \mathbf{y})$

$$\mathcal{I}(\mathbf{x}; \mathbf{y}) \geq \mathcal{H}(\mathbf{x}) - \sum_{i=1}^{N_t \log_2 |\mathcal{W}|} \mathcal{H}(b(i)|\mathbf{y}) \tag{22}$$

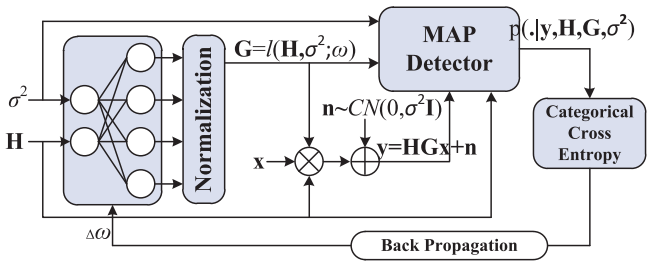


Fig. 3. Paradigm of training a DNN offline to infer optimal precoder given channel matrix $\mathbf{H}$ and σ^2. The precoder is parameterized with DNN, whose weights are denote as ω.

where $b(i)$ refers to the i-th entry of vector $\mathbf{b}$.

Proof: See Appendix. □

C. Learn How to Precode via Deep Nerual Networks

In previous subsections, we investigated the possibility of learning precoding matrices $\mathbf{G}$ with AE. Although it works theoretically, for every channel matrix $\mathbf{H}$ and noise variance σ^2, AEs need to be trained online with new samples related to $\mathbf{H}$ and σ^2, which makes it impractical in high speed wireless scenarios.

To make the precoder keep up with the changes of channel environments, we propose a novel offline training paradigm in which DNNs are trained to infer optimal $\mathbf{G}$ from $\mathbf{H}$ and σ^2. In this paradigm, the precoder $\mathbf{G}$ is parameterized with a DNN followed by a normalization layer to ensure $||\mathbf{G}||_F^2 = N_t$. The inputs of the DNN are $\mathbf{H}$ and σ^2, and we denote function of DNN as $\mathbf{G} = l(\mathbf{H}, \sigma^2; \omega)$, where ω refers to as all the weights in this DNN. In each update of the training process, $\mathbf{H}$, σ^2, $\mathbf{x}$ and $\mathbf{n}$ are generated as training samples, in which $\mathbf{n} \sim CN(0, \sigma^2\mathbf{I})$ and $\mathbf{x}$ are drawn from a constellation set. Then $\mathbf{y} = \mathbf{HGx} + \mathbf{n}$ is calculated and sent to an maximum *a posterior* probability (MAP) detector together with $\mathbf{H}$, $\mathbf{G}$ and σ^2. The MAP detector then calculates the *a posterior* distribution of $\mathbf{x}$, and the results are used to calculate categorical cross entropy loss. Based on the loss, weights ω is updated through back propagation. The paradigm is shown in Fig. 3.

The principles of this design are explained here. According to the proof of Proposition 2, training such a network with the paradigm described above is equivalent to the following optimization problem

$$\max_{\omega} \mathsf{E}_{\sigma^2}\mathsf{E}_{\mathbf{H}}\mathcal{I}(\mathbf{x}; \mathbf{H}l(\mathbf{H}, \sigma^2; \omega)\mathbf{x} + \mathbf{n}|\mathbf{H}, \sigma^2). \tag{23}$$

Note that eq. (23) is upper bound by

$$\begin{aligned}&\mathsf{E}_{\sigma^2}\mathsf{E}_{\mathbf{H}} \max_{\omega} \mathcal{I}(\mathbf{x}; \mathbf{H}l(\mathbf{H}, \sigma^2; \omega)\mathbf{x} + \mathbf{n}|\mathbf{H}, \sigma^2)\\ \leq\ &\mathsf{E}_{\sigma^2}\mathsf{E}_{\mathbf{H}} \max_{\mathbf{G}:||\mathbf{G}||_F^2 = N_t} \mathcal{I}(\mathbf{x}; \mathbf{HGx} + \mathbf{n}|\mathbf{H}, \sigma^2)\end{aligned} \tag{24}$$

so the training process encourage $l(\mathbf{H}, \sigma^2; \omega)$ to approach $\arg\max_{\mathbf{G}:||\mathbf{G}||_F^2 = N_t} \mathcal{I}(\mathbf{x}; \mathbf{HGx} + \mathbf{n}|\mathbf{H}, \sigma^2)$, and the distribution of $\mathbf{H}$ and σ^2 during training can be set arbitrary as long as their sample spaces are large enough.

During the training, $\mathbf{H}$ is diagonal with normalized power $||\mathbf{H}||_F^2 = N_t$. To obtain precoder for a general $\mathbf{H}$ after training, its singular value matrix $\Sigma_{\mathbf{H}}$ is used as a input of the network. Then the desired precoder will be obtained by multiplying the output of the network with $\mathbf{V_H}$ on the left hand side.

VI. Numeric Results

In this section, we will compare the performances of different linear precoding methods in terms of mutual information $\mathcal{I}(\mathbf{x}; \mathbf{y})$ between channel inputs and outputs. We will also show the influence of precoder on bit error rate (BER) for MIMO systems and the convergence of proposed algorithms.

Fig. 4(a) shows the mutual information with different precodering schemes under channel $\mathbf{H}_1 = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$, and SNR is defined as $\mathrm{SNR} = \frac{\mathsf{Tr}\{\mathbf{H}^h\mathbf{H}\}}{N_t\sigma^2}$. In this figure we make the following observations: 1) When the channel inputs are *Gaussian*, the WF improves the mutual information; 2) When QPSK is adopted, our PSGD algorithm improves the mutual information; 3) At low SNR regions, the problems of designing precoders with finite alphabets inputs can be approximated and reduced to WF problems based on the following approximation in [11] and *Taylor* expansion $\ln(1 + x) = x + o(x)$;

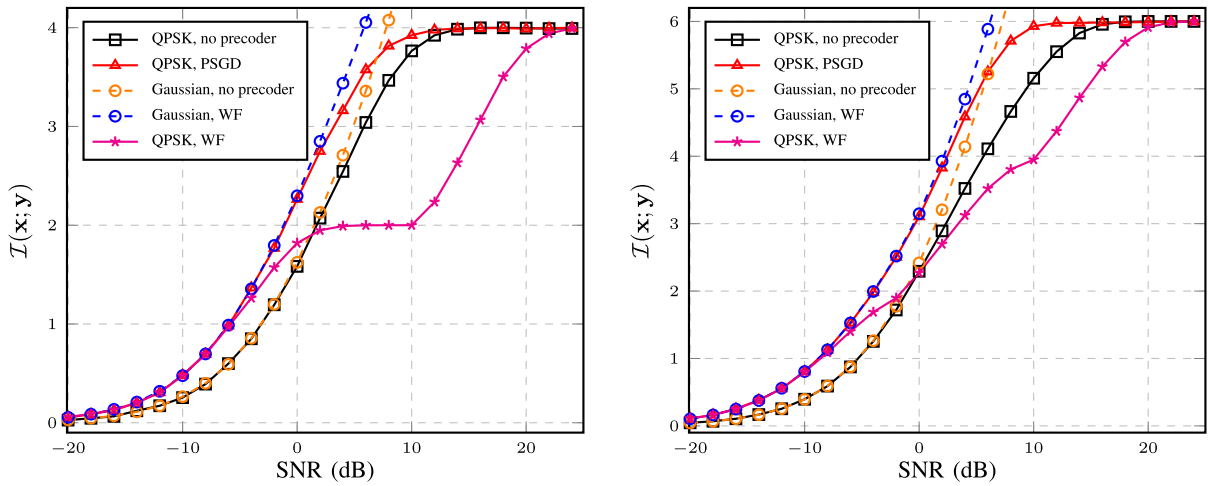
$$\mathcal{I}(\mathbf{x}; \mathbf{y}) = \sigma^{-2}\mathsf{Tr}\{\mathbf{G}^h\mathbf{H}^h\mathbf{HG}\} + o(\sigma^{-2}). \tag{25}$$

4) When we have the wrong assumption of channel inputs, the precoder can reduce the amount mutual information. As the star marked magenta curve suggests, WF precoders reduce the amount of mutual information when QPSK is adopted as channel inputs. Fig. 4(b) shows the comparison under channel $\mathbf{H}_2$, where

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 0.5j & 0.3 \\ -0.5j & 1.5 & -0.1j \\ 0.3 & 0.1j & 0.5 \end{bmatrix}. \tag{26}$$

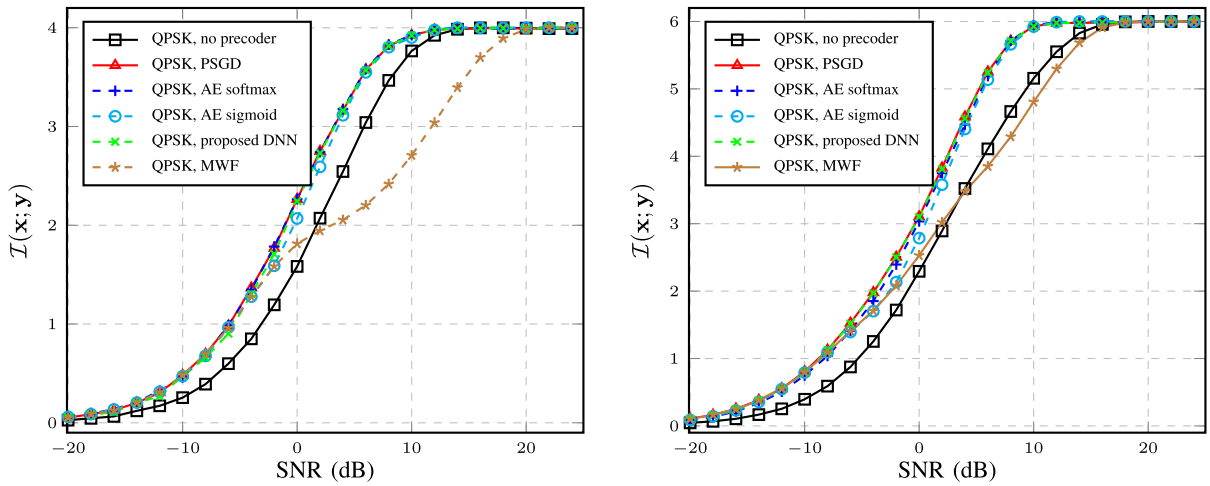
In Fig. 4(b), similar observations can be made.

In Fig. 5, we compare the performances of different precoding methods including PSGD, AE with softmax activation (AE softmax), AE with sigmoid activation (AE sigmoid), proposed DNN and mercury/waterfilling (MWF). As shown in the figure, PSGD, AE softmax, AE sigmoid and proposed DNN have almost same performance. PSGD, AE softmax and proposed DNN perform slightly better than AE sigmoid because AE sigmoid maximizes the lower bound of mutual information instead of mutual information itself. MWF is the optimal power allocation method for parallel AWGN channels, but it is sub-optimal for vector *Gaussian* channels, since MWF is only a special case in which $\mathbf{U_G} = \mathbf{V_H}$ and $\mathbf{V_G} = \mathbf{I}$, and $\mathcal{I}(\mathbf{x}; \mathbf{y})$ is only optimized with respect to $\mathbf{\Sigma}_{\mathbf{G}}^2$. As suggested by eq. (25), all methods have similar optimal performance at low SNR region. However, at high SNR regime, MWF performs much worse than the other methods since it gives up the part of feasible set corresponding to $\mathbf{V_G}$. The receivers in AEs are parameterized with a 3-layer DNN with size 128 in this experiment. The networks are trained according to Algorithm 3, in which the $itermax$ is set as 10,000,



(a) Comparison under channel $\mathbf{H}_1$ (b) Comparison under channel $\mathbf{H}_2$

Fig. 4. Mutual information for *Gaussian* and QPSK channel inputs. The WF and PSGD improve the mutual information for *Gaussian* and QPSK channel inputs, respectively. However, WF reduce the amount of mutual information when QPSK inputs are assumed.



(a) Comparison under channel $\mathbf{H}_1$ (b) Comparison under channel $\mathbf{H}_2$

Fig. 5. Precoding performances of proposed methods and MWF. All the proposed methods show successes in improving mutual information, while MFW fails to improve it at high SNR regime. Compared with other proposed methods, AE sigmoid has a small performance between due to the mismatch between training loss and exact mutual information.

and the batch size $B = 32$. The learning rate for Adam step and SGD step are set as 10^{-3} and 10^{-4}, respectively.

Fig. 6 shows the performance of precoder under different modulation schemes. It can be observed that different channel inputs achieve nearly same amount of mutual information at low SNR region, which verifies the eq. (25) again. At high SNR, the amount of mutual information differs a lot under different modulation schemes. Note that the higher the order of modulation, the higher the SNR is required to get close to their corresponding maxima of mutual information. As we can observe, BPSK, QPSK and 16-QAM get close to their maxima at 1, 10 and 20 dB, respectively.

Fig. 7 shows the average convergence of PSGD with random initialization. The vertical error bar refers to as the standard deviation of the mutual information at the corresponding iteration. According to the figure, the algorithm converges at the first 50 iterations under different initial values. After that, the standard deviation decrease with the number of iterations. In the simulations, the batch size is set as $B = 32$, and the initial step size is set as $\mu = 0.1$. It can be observed that our work takes more iterations than exiting works [5], however each iteration of our work is much simpler. Existing works, requiring accurate gradient information or evaluations of $\mathcal{I}(\mathbf{x};\mathbf{y})$, need hundreds of samples of $\{\mathbf{x},\mathbf{y}\}$ and their corresponding *a posterior* means at each iteration. While in our work, only $B = 32$ samples of *a posterior* means are required at each iteration. The overall complexity of our algorithm and existing ones are similar, but less work at each iteration makes our algorithm more flexible.

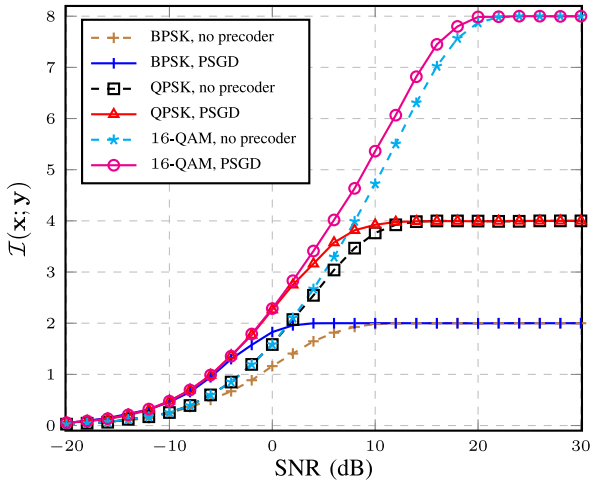


Fig. 6. Comparison of percoder performances with different modulation schemes under channel $\mathbf{H}_1$.

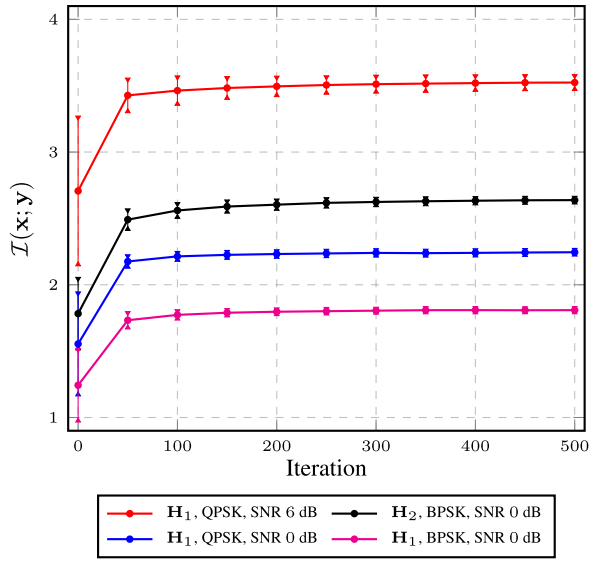


Fig. 7. Convergence of PSGD with different channels and channel inputs at different SNRs.

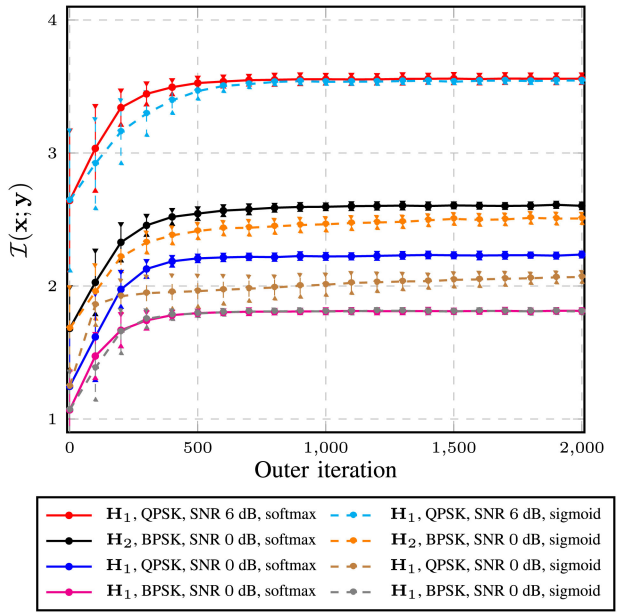


Fig. 8. Convergence of AE with different channels and channel inputs at different SNRs.

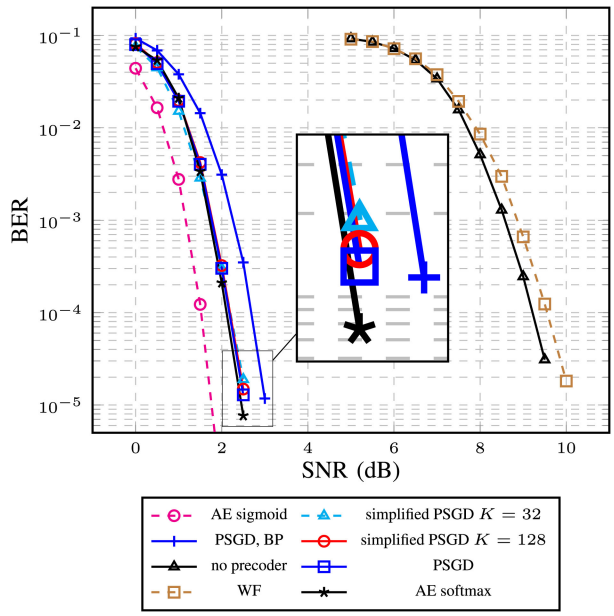


Fig. 9. Coded BER performance of proposed methods under channel $\mathbf{H}_3$ and BPSK modulation.

Fig. 8 shows the average convergence of the training algorithm for different AEs. The vertical error bar is the standard deviation at the corresponding iteration. As we can observed, the algorithms converge quickly at the first 500 iterations, and then the deviation decreases gradually as iteration number increases. It can be seen that performance of AEs with 'softmax' and 'sigmoid' might have a small performance gap, since AE sigmoid maximize a lower bound of mutual information instead of itself.

Fig. 9 shows the coded BER performance of proposed methods under $\mathbf{H}_3$ and BPSK modulation, where $\mathbf{H}_3$ is a 10×10 diagonal matrix with 1, 2, ...,10 being diagonal entries. In the simulation, the (648,486) LDPC code in IEEE 802.11 is adopted for error correction, and MAP is used for detection (if not specified). According to the figure, we made following observations: 1) The BER performance is very poor when there is no preocder or WF is applied; 2) All the proposed methods have huge performance gain; 3) Belief propagation (BP) detection has a small performance loss compared with MAP detection; 4) PSGD, simplified PSGD and AE softmax precoders have nearly same performance; 5) AE sigmoid has the best performance among all the precoding methods. For the last observation, we provide the following explanation. According to eq. (22), using 'sigmoid' and 'binary cross entropy' results in minimizing the summation

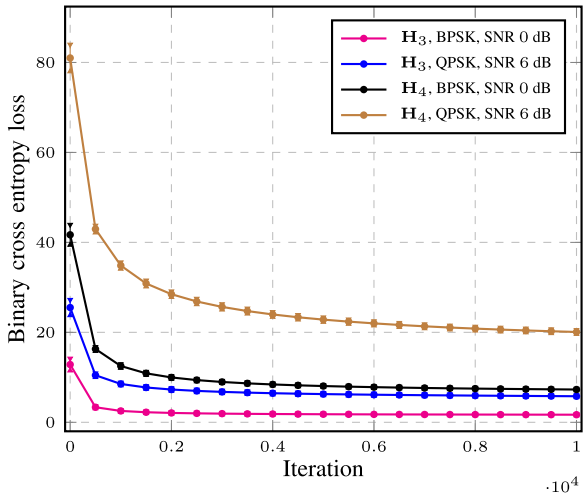


Fig. 10. Convergence of AE with sigmoid for large scale system.

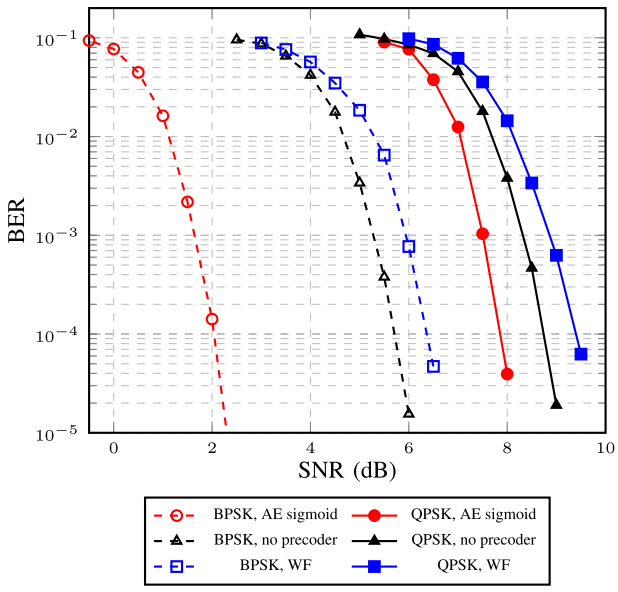


Fig. 11. Coded BER with different precoding strategies under $\mathbf{H}_4$.

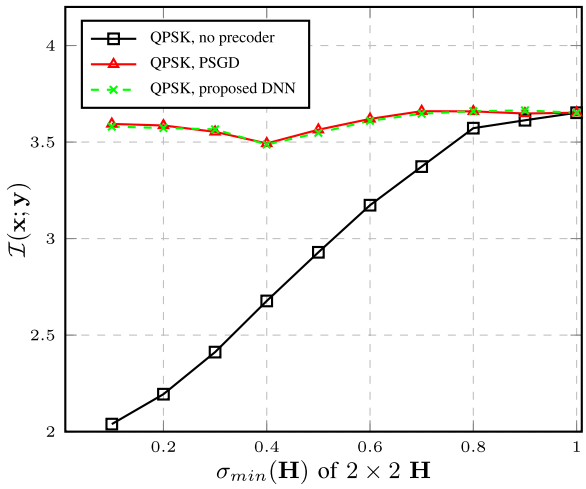


Fig. 12. Achieved mutual information vs. $\sigma_{\min}(\mathbf{H})$. $\mathbf{H}$ is diagonal and it is normalized to satisfy $||\mathbf{H}||_F^2 = N_t$ without loss of generality.

of conditional entropy of bits, i.e. $\sum_{i=1}^{N_t \log_2 |\mathcal{W}|} \mathcal{H}(b(i)|\mathbf{y})$, which coincides with the metric of BER. In this simulation, AEs consist of two hidden layers with size 512, and they are trained at SNR 2 dB.

Fig. 10 shows the convergence of Algorithm 3 under channel $\mathbf{H}_3$ and $\mathbf{H}_4$ with 'sigmoid', where $\mathbf{H}_4$ is a 32×32 diagonal matrix with 1.1, 1.2, 1.3, ...,4.2 being diagonal entries. Since the exact lower bound in eq. (22) is computationally prohibitive, we use the binary cross entropy loss as a metric of convergence, which is calculated by taking the average of the loss of multiple batch of data in all former iterations. As shown in the Appendix C, the binary cross entropy is lower bounded by the term $\sum_{i=1}^{N_t \log_2 |\mathcal{W}|} \mathcal{H}(b(i)|\mathbf{y})$ in eq. (22), so subtracting binary cross entropy from source entropy gives a lower bound of mutual information. In the figure, we can observe that the average binary cross entropy decreases gradually as the number of iterations increase, and the algorithm converges well in different cases. We also noticed that more iterations are needed for the convergence in large systems.

Fig. 11 shows the coded BER performance of no precoder, AE sigmoid and WF under $\mathbf{H}_4$. In the simulation, BP is adopted for detection, since the complexity of MAP is not affordable for channel matrix with such size. The precoder for QPSK is obtained by training AE at 8 dB, and that for BPSK is trained at 5 dB. It can be observed that AE sigmoid can achieve significant performance gain compared with no precoders and WF. It can also be observed that precoder for BPSK can achieve more gain than that for QPSK.

Fig. 12 shows the relationship between achieved mutual information and $\sigma_{\min}(\mathbf{H})$ of 2×2 $\mathbf{H}$, where $\sigma_{\min}(\mathbf{H})$ is the smaller singular value of $\mathbf{H}$. Without loss of generality, $\mathbf{H}$ is diagonal and it is normalized to satisfy $||\mathbf{H}||_F^2 = N_t$. In the figure, $N_t = 2$, so the range of $\sigma_{\min}(\mathbf{H})$ is 0 to 1. SNR is set as 6 dB and QPSK modulation is assumed. From Fig. 12, we observe that 1) the mutual information is higher and more stable after precoding; 2) the smallest value of mutual information after precoding appears around $\sigma_{\min}(\mathbf{H}) = 0.4$ instead of $\sigma_{\min}(\mathbf{H}) = 0$. For the second observation, we provide the following explanation. Based on the proof of Corollary 1, when $\sigma_{\min}(\mathbf{H})$ is small enough, the optimal precoder will not allocate power to the sub-channel corresponding to $\sigma_{\min}(\mathbf{H})$. In other word, the power of the sub-channel is wasted. From Fig. 12, we infer that when $\sigma_{\min}(\mathbf{H})$ is around 0.4, the precoder waste the largest amount of power of $\mathbf{H}$, so we obtain a small value of mutual information.

In Table I, we make a comparison among the proposed methods, [5] and [15]. The code is written in Python 3.6 and is run on intel i7-8700 CPU. From the table, we make the following observations: 1) [5] achieves the optimal values of mutual information but spends much more time than other methods; 2) [15] achieves sub optimal values and it is much faster than [5]; 3) PSGD achieves optimal values and takes much less time than [5]; 4) AEs achieve near optimal values with executing time less than [5] but more than PSGD; 5) In the cases corresponding to the first two columns, [15] is faster than PSGD and AE softmax, but it is

TABLE I
COMPARISON AMONG PROPOSED METHODS AND EXISTING WORKS: *MUTUAL INFORMATION/EXECUTING TIME*

References	$\mathbf{H}_1$, BPSK, 0 dB	$\mathbf{H}_1$, QPSK, 6 dB	$\mathbf{H}_2$, QPSK, 8 dB
[5]	$1.81/9.2s$	$3.56/97.2s$	$5.72/5763s$
[15]	$1.73/0.01s$	$3.50/0.13s$	$5.46/11.3s$
PSGD	$1.81/2.2s$	$3.55/2.3s$	$5.72/2.7s$
AE softmax	$1.80/4.4s$	$3.54/4.4s$	$5.64/5.5s$
proposed DNN	$1.81/0.002s$	$3.55/0.002s$	$5.72/0.002s$

reversed in the third case; 6) The proposed DNN, introduced in Section V-C, achieves optimal values in all cases, and run faster than all other methods.

In [15], the complexity of evaluating lower bounds of $\mathcal{I}$ is $\mathcal{O}(|\mathcal{X}|^2)$, while the complexity of calculating $\mathsf{E}[\mathbf{x}|\mathbf{y}]$ in PSGD is $\mathcal{O}(|\mathcal{X}|)$. Besides, the power allocation used in [15] is determined by exhaustive search, whose complexity is exponential to N_t. Therefore as N_t increases, the complexity of [15] grows faster, which results in a longer executing time in the third case in Table I. The proposed DNN calculates the precoder by evaluating the explicit function $\mathbf{G} = l(\mathbf{H}, \sigma^2; \omega)$, so it is faster than other iterative methods.

VII. CONCLUSION

In this paper, we considered the problem of finding linear precoders that maximize the mutual information with finite alphabet channel inputs. We proved that if the right singular matrix of the precoder is fixed, then any local optimal of this problem is a global optimal. Based on this result, a flexible PSGD algorithm and its simplified version were proposed to solve this problem. Moreover, motivated by modern technologies for training accelerations of DNNs, AE based precoding methods were proposed, which works for MIMO systems with tens of antennas. To make the precoder practical for high speed wireless scenarios, a novel offline training paradigm of DNNs was proposed, in which DNNs are trained to infer optimal precoder given channel state information. Simulation results showed that all the proposed methods performed well in terms of both mutual information and BER performance, and the proposed DNN significantly reduced the time complexity of precoding.

APPENDIX A
APPENDIX PROOF OF COROLLARY 1

Let p_i be the i-th diagonal element of $\mathbf{\Sigma}_{\mathbf{G}}^2$. Then the problem of maximizing $\mathcal{I}(\mathbf{x};\mathbf{y})$ with respect to $\mathbf{\Sigma}_{\mathbf{G}}^2$ can be reformulated as

$$\begin{aligned} &\min_{p_1,p_2,\ldots,p_{N_t}} \quad -\mathcal{I}(\mathbf{x};\mathbf{y}) \\ &s.t.\ \ \forall i, p_i \geq 0 \\ &\quad \sum_{i=1}^{N_t} p_i \leq N_t. \end{aligned} \tag{27}$$

Then according to [5, Thm 2], the gradient of $\mathcal{I}(\mathbf{x};\mathbf{y})$ with respect to p_i can be written as

$$\frac{\partial \mathcal{I}}{\partial p_i} = \psi_i \gamma_i \tag{28}$$

where ψ_i refers to as the i-th diagonal element of $\mathbf{V}_{\mathbf{G}}^h \mathbf{\Phi} \mathbf{V}_{\mathbf{G}}$ (which is still a function of $p_1, p_2, ..., p_{N_t}$), and γ_i is the i-th diagonal element of $\mathbf{\Sigma}_{\mathbf{H}}^2$. Since $-\mathcal{I}(\mathbf{x};\mathbf{y})$ is a convex function of $p_1, p_2, ..., p_{N_t}$ according to [5, Thm 2] and the constraints of problem (27) is convex, the following Karush $-$ Kuhn$-$Tucker (KKT) conditions are sufficient and necessary for optimal solutions of eq. (27):

$$\begin{cases} \forall i, -\psi_i \gamma_i + \lambda - \lambda_i = 0 \\ \forall i, p_i \geq 0, \lambda_i \geq 0, \lambda_i p_i = 0 \\ \sum_{i=1}^{N_t} p_i \leq N_t, \lambda \geq 0, \lambda(\sum_{i=1}^{N_t} p_i - N_t) = 0. \end{cases} \tag{29}$$

Notice the fact that $\psi_i > 0$ for all i and there exists at least a $\gamma_i > 0$, eq. (29) is equivalent to

$$\begin{cases} \forall p_i > 0,\ \psi_i \gamma_i = \lambda \\ \forall p_i = 0,\ \psi_i \gamma_i \leq \lambda \\ \sum_{i=1}^{N_t} p_i = N_t,\ \lambda > 0. \end{cases} \tag{30}$$

Denote g_i as the i-th diagonal element of $\mathbf{\Sigma}_{\mathbf{G}}$, then the problem (16) can be reformulated as

$$\begin{aligned} &\min_{g_1,g_2,\ldots,g_{N_t}} \quad -\mathcal{I}(\mathbf{x};\mathbf{y}) \\ &s.t.\ \ \sum_{i=1}^{N_t} g_i^2 \leq N_t \end{aligned} \tag{31}$$

and the *Lagrangian* of eq. (31) is given as

$$\mathcal{L}(-\mathcal{I}, \eta) = -\mathcal{I} + \eta \left(\sum_{i=1}^{N_t} g_i^2 - N_t \right). \tag{32}$$

Apply chain rule of derivative to eq. (28) with $p_i = g_i^2$, the derivative of $\mathcal{I}(\mathbf{x};\mathbf{y})$ with respect to g_i is given as

$$\frac{\partial \mathcal{I}}{\partial g_i} = 2 g_i \psi_i \gamma_i. \tag{33}$$

We list the following KKT conditions, which are sufficient and necessary for critical points of eq. (31):

$$\begin{cases} \forall i, -2 g_i \psi_i \gamma_i + 2\eta g_i = 0 \\ \sum_{i=1}^{N_t} g_i^2 \leq N_t,\ \eta \geq 0,\ \eta(\sum_{i=1}^{N_t} g_i^2 - N_t) = 0 \end{cases} \tag{34}$$

in which the first line is equivalent to

$$\forall i, g_i = 0 \text{ or } \psi_i \gamma_i = \eta. \tag{35}$$

If $\eta = 0$, then for any $\gamma_i > 0$, $g_i = 0$, and this gives the global maximum with $\mathcal{I} = 0$. For the case $\eta > 0$, eq. (34) is equivalent to

$$\begin{cases} \forall i, g_i = 0 \text{ or } \psi_i \gamma_i = \eta \\ \sum_{i=1}^{N_t} g_i^2 = N_t, \ \eta > 0. \end{cases} \tag{36}$$

For any $i = 1, 2, ..., N_t$, the second order derivative of $\mathcal{L}(-\mathcal{I}, \eta)$ with respect to g_i is given by

$$\frac{\partial^2 \mathcal{L}(-\mathcal{I}, \eta)}{\partial g_i^2} = 2\left(\eta - \frac{\partial \mathcal{I}}{\partial p_i}\right) - 2g_i^2 \frac{\partial^2 \mathcal{I}}{\partial p_i^2} \tag{37}$$

which is the i-th diagonal element of the *Hessian* of $\mathcal{L}(-\mathcal{I}, \eta)$. Let $\bar{g}_1, \bar{g}_2, ..., \bar{g}_{N_t}$ satisfy eq. (35) but violate eq. (30) with corresponding $\bar{\psi}_1, \bar{\psi}_2, ..., \bar{\psi}_{N_t}$ and $\bar{\eta}$, then there exists an index m, such that $\bar{g}_m = 0$ and $\psi_m \gamma_m > \eta$. Then we have

$$\frac{\partial^2 \mathcal{L}(-\mathcal{I}, \bar{\eta})}{\partial \bar{g}_m^2} = 2(\bar{\eta} - \bar{\psi}_m \gamma_m) < 0 \tag{38}$$

which means the *Hessian* at the point $\bar{g}_1, \bar{g}_2, ..., \bar{g}_{N_t}, \bar{\eta}$ is not positive semi-definite, so the $\bar{g}_1, \bar{g}_2, ..., \bar{g}_{N_t}$ is not a local minimum. Therefore any local minimum has to satisfies eq. (30), so any local minimum is global minimum.

If $\gamma_i > 0$ for all i, then the only global maximum is $g_i = 0$ for all i. If, in addition, there exists an index n such that $\bar{g}_n > 0$, then

$$\frac{\partial^2 \mathcal{L}(-\mathcal{I}, \bar{\eta})}{\partial \bar{g}_n^2} = -2g_n^2 \frac{\partial^2 \mathcal{I}}{\partial p_n^2} > 0 \tag{39}$$

where $\frac{\partial^2 \mathcal{I}}{\partial p_n^2} < 0$ due to concavity of $\mathcal{I}$ with respect to $p_1, ..., p_{N_t}$. Therefore in this case, the $\bar{g}_1, \bar{g}_2, ..., \bar{g}_{N_t}$ is neither a local minimum nor a local maximum, so it is a saddle point.

Appendix B
Appendix Proof of Proposition 2

In this section, we use $\mathbf{X}$ to denote the random variable, and let $\mathbf{x}_a$ enumerate samples space $\mathcal{X}$ for $a = 1, 2, ..., |\mathcal{W}|^{N_t}$. Similar notation is applied to $\mathbf{Y}$.

With one-hot encoding, the target value is $\mathbf{e}_a = onehot(\mathbf{x}_a)$ if $\mathbf{x}_a$ is generated as sample, where $\mathbf{e}_a$ is the a-th column of an identity matrix. If the loss is chosen as 'categorical cross entropy', then training the autoencoder is equivalent to solve the following problem:

$$\begin{aligned} \min_{\mathbf{G}, \mathbf{\Theta}} \ & \mathsf{E}_{\mathbf{X},\mathbf{Y}}[-\log_2 (f(\mathbf{Y}; \mathbf{\Theta}))^T onehot(\mathbf{X})] \\ s.t. \ & \mathsf{Tr}\{\mathbf{G}^h \mathbf{G}\} \le N_t. \end{aligned} \tag{40}$$

Note that the above objection function

$$\begin{aligned} & \mathsf{E}_{\mathbf{X},\mathbf{Y}}[-\log_2(f(\mathbf{Y}; \mathbf{\Theta}))^T onehot(\mathbf{X})] \\ &= \mathsf{E}_{\mathbf{Y}}\left[-\sum_a \mathrm{pr}(\mathbf{X} = \mathbf{x}_a | \mathbf{Y}) \log_2 (f(\mathbf{Y}; \mathbf{\Theta}))^T onehot(\mathbf{x}_a)\right] \\ &= \mathsf{E}_{\mathbf{Y}}\left[-\sum_a \mathrm{pr}(\mathbf{X} = \mathbf{x}_a | \mathbf{Y}) \log_2 (f(\mathbf{Y}; \mathbf{\Theta})_a)\right] \end{aligned} \tag{41}$$

where $f(\mathbf{Y}; \mathbf{\Theta})_a$ refers to the a-th entry of $f(\mathbf{Y}; \mathbf{\Theta})$. Note that $\sum_a f(\mathbf{Y}; \mathbf{\Theta})_a = 1$ always holds, so $f(\mathbf{Y}; \mathbf{\Theta})$ returns a distribution. By the non-negativity of Kullback Leibler divergence (KL divergence), we have

$$\begin{aligned} & \mathsf{E}_{\mathbf{Y}}[-\sum_a \mathrm{pr}(\mathbf{X} = \mathbf{x}_a | \mathbf{Y}) \log_2 (f(\mathbf{Y}; \mathbf{\Theta})_a)] \\ &\ge \mathsf{E}_{\mathbf{Y}}[-\sum_a \mathrm{pr}(\mathbf{X} = \mathbf{x}_a | \mathbf{Y}) \log_2 \mathrm{pr}(\mathbf{X} = \mathbf{x}_a | \mathbf{Y})] \\ &\equiv \mathcal{H}(\mathbf{X}|\mathbf{Y}). \end{aligned} \tag{42}$$

Suppose the Assumption 1 holds and the 'distance' is chosen as KL divergence, then for any $\mathbf{G}$, we can obtain a parameter $\mathbf{\Theta}^\star$ via training such that

$$\mathsf{E}_{\mathbf{X},\mathbf{Y}}[-\log_2(f(\mathbf{Y}; \mathbf{\Theta}^\star))^T onehot(\mathbf{X})] = \mathcal{H}(\mathbf{X}|\mathbf{Y}) + \epsilon \tag{43}$$

where $\epsilon > 0$ is a small training error. Then we have

$$\begin{aligned} & \min_{\mathsf{Tr}\{\mathbf{G}^h \mathbf{G}\} \le N_t} \min_{\mathbf{\Theta}} \mathsf{E}_{\mathbf{X},\mathbf{Y}}[-\log_2(f(\mathbf{Y}; \mathbf{\Theta}))^T onehot(\mathbf{X})] \\ \le & \min_{\mathsf{Tr}\{\mathbf{G}^h \mathbf{G}\} \le N_t} \mathsf{E}_{\mathbf{X},\mathbf{Y}}[-\log_2(f(\mathbf{Y}; \mathbf{\Theta}^\star))^T onehot(\mathbf{X})] \\ = & \min_{\mathsf{Tr}\{\mathbf{G}^h \mathbf{G}\} \le N_t} \mathcal{H}(\mathbf{X}|\mathbf{Y}) + \epsilon. \end{aligned} \tag{44}$$

Therefore problem (40) is equivalent to

$$\begin{aligned} \min_{\mathbf{G}} \ & \mathcal{H}(\mathbf{X}|\mathbf{Y}) + \epsilon(\mathbf{G}) \\ s.t. \ & \mathsf{Tr}\{\mathbf{G}^h \mathbf{G}\} \le N_t \end{aligned} \tag{45}$$

where $\epsilon(\mathbf{G})$ can be viewed as function of $\mathbf{G}$ with very small value. Since $\mathcal{I}(\mathbf{X}; \mathbf{Y}) = \mathcal{H}(\mathbf{X}) - \mathcal{H}(\mathbf{X}|\mathbf{Y})$, the problem (45) is equivalent to maximizing mutual information.

Appendix C
Appendix Proof of Proposition 3

In this section, we use notation rules same with that in last section. We further denote random variable $\mathbf{B} = bin(\mathbf{X})$, where $bin(\mathbf{X})$ returns the binary representation of $\mathbf{X}$. If $\mathbf{x}_a$ is generated as sample, the target is $\mathbf{b}_a = bin(\mathbf{x}_a)$, where $\mathbf{b}_a$ is the binary representation of $(a-1)$. With 'sigmoid' being activation function, training the AE is equivalent to solving the following problem:

$$\begin{aligned} \min_{\mathbf{G}, \mathbf{\Theta}} \ & \mathsf{E}_{\mathbf{X},\mathbf{Y}}[-\log_2(f(\mathbf{Y}; \mathbf{\Theta}))^T bin(\mathbf{X}) \\ & - \log_2(\mathbf{1} - f(\mathbf{Y}; \mathbf{\Theta}))^T (\mathbf{1} - bin(\mathbf{X}))] \\ s.t. \ & \mathsf{Tr}\{\mathbf{G}^h \mathbf{G}\} \le N_t \end{aligned} \tag{46}$$

where $\mathbf{1}$ refers to an vector with all entries being 1. Make an arrangement of the objective function, we obtain eq. (47), shown

$$
\begin{aligned}
&\mathsf{E}_{\mathbf{X},\mathbf{Y}}[-\log_2(f(\mathbf{Y};\mathbf{\Theta}))^T bin(\mathbf{X}) - \log_2(\mathbf{1}-f(\mathbf{Y};\mathbf{\Theta}))^T(\mathbf{1}-bin(\mathbf{X}))] \\
&= \mathsf{E}_{\mathbf{B},\mathbf{Y}}\left[\sum_{i=1}^{N_t\log_2|\mathcal{W}|} -\log_2(f(\mathbf{Y};\mathbf{\Theta})_i)B(i) - \log_2(1-f(\mathbf{Y};\mathbf{\Theta})_i)(1-B(i))\right] \\
&= \mathsf{E}_{\mathbf{Y}}\mathsf{E}_{B(1),B(2),\ldots|\mathbf{Y}}\left[\sum_{i=1}^{N_t\log_2|\mathcal{W}|} -\log_2(f(\mathbf{Y};\mathbf{\Theta})_i)B(i) - \log_2(1-f(\mathbf{Y};\mathbf{\Theta})_i)(1-B(i))\right] \\
&= \mathsf{E}_{\mathbf{Y}}\left[\sum_{i=1}^{N_t\log_2|\mathcal{W}|} -\log_2(f(\mathbf{Y};\mathbf{\Theta})_i)\mathrm{pr}(B(i)=1|\mathbf{Y}) - \log_2(1-f(\mathbf{Y};\mathbf{\Theta})_i)\mathrm{pr}(B(i)=0|\mathbf{Y})\right] \\
&\le \mathsf{E}_{\mathbf{Y}}\left[\sum_{i=1}^{N_t\log_2|\mathcal{W}|} -\log_2(\mathrm{pr}(B(i)=1|\mathbf{Y}))\mathrm{pr}(B(i)=1|\mathbf{Y}) - \log_2(\mathrm{pr}(B(i)=0|\mathbf{Y}))\mathrm{pr}(B(i)=0|\mathbf{Y})\right] \\
&\equiv \sum_{i=1}^{N_t\log_2|\mathcal{W}|} \mathcal{H}(B(i)|\mathbf{Y}). \qquad (47)
\end{aligned}
$$

$$
\begin{aligned}
&\min_{\mathsf{Tr}\{\mathbf{G}^h\mathbf{G}\}\le N_t}\ \min_{\mathbf{\Theta}} \mathsf{E}_{\mathbf{X},\mathbf{Y}}\left[-\log_2(f(\mathbf{Y};\mathbf{\Theta}))^T bin(\mathbf{X}) - \log_2(\mathbf{1}-f(\mathbf{Y};\mathbf{\Theta}))^T(\mathbf{1}-bin(\mathbf{X}))\right] \\
&\le \min_{\mathsf{Tr}\{\mathbf{G}^h\mathbf{G}\}\le N_t} \mathsf{E}_{\mathbf{X},\mathbf{Y}}\left[-\log_2(f(\mathbf{Y};\mathbf{\Theta}^\star))^T bin(\mathbf{X}) - \log_2(\mathbf{1}-f(\mathbf{Y};\mathbf{\Theta}^\star))^T(\mathbf{1}-bin(\mathbf{X}))\right] \\
&= \min_{\mathsf{Tr}\{\mathbf{G}^h\mathbf{G}\}\le N_t} \sum_{i=1}^{N_t\log_2|\mathcal{W}|} \mathcal{H}(B(i)|\mathbf{Y}) + \epsilon(\mathbf{G}). \qquad (49)
\end{aligned}
$$

at the top of this page. By Assumption 1, we can obtain a $\mathbf{\Theta}^\star$ via training such that

$$
\begin{aligned}
&\mathsf{E}_{\mathbf{X},\mathbf{Y}}[-\log_2(f(\mathbf{Y};\mathbf{\Theta}^\star))^T bin(\mathbf{X}) \\
&-\log_2(\mathbf{1}-f(\mathbf{Y};\mathbf{\Theta}^\star))^T(\mathbf{1}-bin(\mathbf{X}))] \\
&= \sum_{i=1}^{N_t\log_2|\mathcal{W}|} \mathcal{H}(B(i)|\mathbf{Y}) + \epsilon \qquad (48)
\end{aligned}
$$

where $\epsilon > 0$ is a small training error. Then we have eq. (49), shown at the top of this page, and the problem (46) is equivalent to

$$
\begin{aligned}
&\min_{\mathbf{G}} \sum_{i=1}^{N_t\log_2|\mathcal{W}|} \mathcal{H}(B(i)|\mathbf{Y}) + \epsilon(\mathbf{G}) \\
&\quad s.t.\ \mathsf{Tr}\{\mathbf{G}^h\mathbf{G}\} \le N_t \qquad (50)
\end{aligned}
$$

where $\epsilon(\mathbf{G}) > 0$ is small value and it is a function of $\mathbf{G}$. By inequality of entropy, we have

$$
\mathcal{H}(\mathbf{B}|\mathbf{Y}) \le \sum_{i=1}^{N_t\log_2|\mathcal{W}|} \mathcal{H}(B(i)|\mathbf{Y}). \qquad (51)
$$

Notice the fact that $\mathcal{H}(\mathbf{B}|\mathbf{Y}) = \mathcal{H}(\mathbf{X}|\mathbf{Y})$, we have

$$
\mathcal{I}(\mathbf{X};\mathbf{Y}) \ge \mathcal{H}(\mathbf{X}) - \sum_{i=1}^{N_t\log_2|\mathcal{W}|} \mathcal{H}(B(i)|\mathbf{Y}). \qquad (52)
$$

REFERENCES

[1] N. Fatema, G. Hua, Y. Xiang, D. Peng, and I. Natgunanathan, "Massive MIMO linear precoding: A survey," *IEEE Syst. J*, vol. 12, no. 4, pp. 3920–3931, Dec. 2018.

[2] A. Wiesel, Y. C. Eldar, and S. Shamai, "Linear precoding via conic optimization for fixed MIMO receivers," *IEEE Trans. Signal Process.*, vol. 54, no. 1, pp. 161–176, Jan. 2006.

[3] G. Scutari, D. P. Palomar, and S. Barbarossa, "The MIMO iterative waterfilling algorithm," *IEEE Trans. Signal Process.*, vol. 57, no. 5, pp. 1917–1935, May 2009.

[4] F. Perez-Cruz, M. R. D. Rodrigues, and S. Verdu, "MIMO Gaussian channels with arbitrary inputs: Optimal precoding and power allocation," *IEEE Trans. Inf. Theory*, vol. 56, no. 3, pp. 1070–1084, Mar. 2010.

[5] C. Xiao, Y. R. Zheng, and Z. Ding, "Globally optimal linear precoders for finite alphabet signals over complex vector Gaussian channels," *IEEE Trans. Signal Process.*, vol. 59, no. 7, pp. 3301–3314, Jul. 2011.

[6] C. Xiao and Y. R. Zheng, "On the mutual information and power allocation for vector Gaussian channels with finite discrete inputs," in *Proc. IEEE Glob. Commun. Conf.*, 2008, pp. 1–5.

[7] W. Zeng, C. Xiao, M. Wang, and J. Lu, "Linear precoding for finite-alphabet inputs over MIMO fading channels with statistical CSI," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 3134–3148, Jun. 2012.

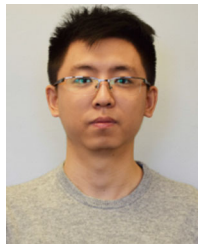
[8] Y. Wu, C. Xiao, X. Gao, J. D. Matyjas, and Z. Ding, "Linear precoder design for MIMO interference channels with finite-alphabet signaling," *IEEE Trans. Commun.*, vol. 61, no. 9, pp. 3766–3780, Sep. 2013.

[9] Y. Wu, C. Xiao, Z. Ding, X. Gao, and S. Jin, "Linear precoding for finite-alphabet signaling over MIMOME wiretap channels," *IEEE Trans. Veh. Technol.*, vol. 61, no. 6, pp. 2599–2612, Jul. 2012.

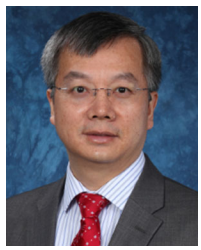
[10] S. Verdu, "Spectral efficiency in the wideband regime," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1319–1343, Jun. 2002.

[11] Dongning Guo, S. Shamai, and S. Verdu, "Mutual information and minimum mean-square error in Gaussian channels," *IEEE Trans. Inf. Theory*, vol. 51, no. 4, pp. 1261–1282, Apr. 2005.

[12] A. Lozano, A. M. Tulino, and S. Verdu, "Optimum power allocation for parallel Gaussian channels with arbitrary input distributions," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 3033–3051, Jul. 2006.
[13] M. Payaro and D. P. Palomar, "On optimal precoding in linear vector Gaussian channels with arbitrary input distribution," in *Proc. IEEE Int. Symp. Inf. Theory*, 2009, pp. 1085–1089.
[14] A. Lu, X. Gao, Y. R. Zheng, and C. Xiao, "Linear precoder design for SWIPT in MIMO broadcasting systems with discrete input signals: Manifold optimization approach," *IEEE Trans. Commun.*, vol. 65, no. 7, pp. 2877–2888, Jul. 2017.
[15] Y. R. Zheng, M. Wang, W. Zeng, and C. Xiao, "Practical linear precoder design for finite alphabet multiple-input multiple-output orthogonal frequency division multiplexing with experiment validation," *IET Commun.*, vol. 7, no. 9, pp. 836–847, 2013.
[16] K. Cohen, A. Nedić, and R. Srikant, "On projected stochastic gradient descent algorithm with weighted averaging for least squares regression," *IEEE Trans. Autom. Control*, vol. 62, no. 11, pp. 5974–5981, Nov. 2017.
[17] Z. Guo and P. Nilsson, "Algorithm and implementation of the k-best sphere decoding for MIMO detection," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 3, pp. 491–503, Mar. 2006.
[18] M. Wenk, M. Zellweger, A. Burg, N. Felber, and W. Fichtner, "K-best MIMO detection VLSI architectures achieving up to 424 Mbps," in *Proc. IEEE Int. Symp. Circuits Syst*, 2006, pp. 1151–1154.
[19] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5438–5453, Oct. 2018.
[20] Y. Liao, N. Farsad, N. Shlezinger, Y. C. Eldar, and A. J. Goldsmith, "Deep neural network symbol detection for millimeter wave communications," in *Proc. IEEE Glob. Commun. Conf.*, 2019, pp. 1–6.
[21] Y. Pu *et al.*, "Variational autoencoder for deep learning of images, labels and captions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2352–2360.
[22] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *Proc. IEEE Inf. Theory Workshop*, 2015, pp. 1–5.
[23] J. Tao, J. Chen, J. Xing, S. Fu, and J. Xie, "Autoencoder neural network based intelligent hybrid beamforming design for mmWave massive MIMO systems," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 3, pp. 1019–1030, Sep. 2020.
[24] 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 Int. Workshop Signal Process. Adv. Wireless Commun.*, 2018, pp. 1–5.
[25] Y. Pu *et al.*, "Variational autoencoder for deep learning of images, labels and captions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2352–2360.
[26] 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.
[27] S. Amini and S. Ghaemmaghami, "A new framework to train autoencoders through non-smooth regularization," *IEEE Trans. Signal Process.*, vol. 67, no. 7, pp. 1860–1874, Apr. 2019.
[28] D. P. Palomar and S. Verdu, "Gradient of mutual information in linear vector Gaussian channels," *IEEE Trans. Inf. Theory*, vol. 52, no. 1, pp. 141–154, Jan. 2006.
[29] D. P. Kingma and J. Ba, "ADAM: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2014.



Shusen Jing (Student Member, IEEE) received the B.E. degree in information science and engineering from Southeast University, Nanjing, China, in 2017. He is currently working toward the Ph.D. degree with Lehigh University, Bethlehem, PA, USA. His interests include wireless communication and machine learning.



Chengshan Xiao (Fellow, IEEE) received the Bachelor of Science degree in electronic engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 1987, the Master of Science degree in electronic engineering from Tsinghua University, Beijing, China, in 1989, and the Ph.D. degree in electrical engineering from The University of Sydney, Sydney, NSW, Australia, in 1997.

He is currently the Chandler Weaver Professor and Chair of the Department of Electrical and Computer Engineering, Lehigh University, Bethlehem, PA, USA. He is a Fellow of the Canadian Academy of Engineering. He was the Program Director of the Division of Electrical, Communications and Cyber Systems at the USA National Science Foundation. He was a Senior Member of Scientific Staff with Nortel Networks, Ottawa, ON, Canada, a Faculty Member with Tsinghua University, the University of Alberta, Edmonton, AB, Canada, the University of Missouri, Columbia, MO, USA, and Missouri University of Science and Technology, Rolla, MO, USA. He has also held visiting professor positions in Germany and Hong Kong. His research interests include wireless communications, signal processing, and underwater acoustic communications. He is the holder of various patents granted in USA, Canada, China, and Europe. His invented algorithms waere implemented into Nortel's base station radio products after successful technical field trials and network integration.

Dr. Xiao is the Awards Committee Chair and elected Member-at-Large of Board of Governors of IEEE Communications Society. Previously, he was on the IEEE Technical Activity Board Periodical Committee, he was an elected Member-at-Large of Board of Governors, a Member of Fellow Evaluation Committee, the Director of Conference Publications, a Distinguished Lecturer of the IEEE Communications Society, and a Distinguished Lecturer of the IEEE VEHICULAR TECHNOLOGY. He was also the Editor, Area Editor, and the Editor-in-Chief of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-I. He was the Technical Program Chair of the 2010 IEEE International Conference on Communications, Cape Town, South Africa, the Technical Program Co-Chair of the 2017 IEEE Global Communications Conference, Singapore. He was the Founding Chair of the IEEE Wireless Communications Technical Committee. He was the recipient of the distinguished awards, including the 2014 Humboldt Research Award, the 2014 IEEE Communications Society Joseph LoCicero Award, the 2015 IEEE Wireless Communications Technical Committee Recognition Award, and the 2017 IEEE Communications Society Harold Sobol Award.