One-Bit Precoding Constellation Design via Autoencoder-Based Deep Learning

Foad Sohrabi and Wei Yu
Department of Electrical and Computer Engineering
University of Toronto, Toronto, Ontario M5S 3G4, Canada
Emails: {fsohrabi, weiyu}@ece.utoronto.ca

***Abstract*—This paper considers a multicasting system in which the base station has a large number of antennas with cost-effective one-bit digital-to-analog converters and aims to send a common symbol to multiple remote users. Unlike the existing literature which seeks to design the one-bit precoder for a given constellation, e.g., quadrature amplitude modulation (QAM) or phase shift keying (PSK), this paper aims to jointly design the transmit one-bit precoder and the receive constellation by leveraging the concept of autoencoder, wherein the end-to-end multicasting system is modeled using a deep neural network with the one-bit percoding constraint represented by a binary thresholding layer. To deal with the issue that such a binary layer always produces a gradient of zero, and thus prevents an effective end-to-end training when using the conventional back-propagation method, this paper uses a variant of *straight-through* estimator which approximates the thresholding function with a properly scaled sigmoid function in the back-propagation phase. Numerical results show that, for a fixed channel scenario, the proposed autoencoder-based constellation design is superior to the conventional QAM and PSK constellations. Using the insights obtained from fixed channel scenarios, we also propose a constellation design for varying channel scenarios and numerically show that the proposed design achieves a better performance as compared to the conventional constellations.**

***Index Terms*—autoencoder, binary neurons, deep neural network, one-bit precoding, multiple-input multiple-output.**

I. Introduction

Massive multiple-input multiple-output (MIMO) systems, in which a base station (BS) with large-scale antenna arrays serves single-antenna users, is one of the promising technologies for the next generation of the wireless systems for achieving high spectral efficiency, reliability, and connectivity requirements [1]. One of the main challenges in implementing massive MIMO systems is the extensive power consumption of the high-resolution digital-to-analog converters (DACs) required in conventional MIMO precoding schemes, e.g., [2].

One-bit precoding is a novel precoding scheme in which two DACs are dedicated for each antenna element but with only one-bit resolution. The one-bit precoding architecture, which can significantly reduce the transmitter's circuit power consumption, has recently attracted lots of attention. The early works on one-bit precoding adopt linear-quantized precoding schemes in which the precoder is designed by quantizing the conventional linear precoders [3], [4]. To address the problem of high symbol error floor of linear-quantized precoding schemes in the high signal-to-noise ratio (SNR) regime, more sophisticated non-linear precoding schemes are proposed [5]–[10]. In all these works, the non-linear precoder is designed under the assumption that the receive constellation is fixed to one of the conventional constellations, e.g., QAM constellations [5]–[8] or PSK constellations [9], [10].

In this paper, we point out that the optimal constellation design is a crucial component of the overall problem, and further both the optimal precoder and the optimal constellation can be designed using machine learning techniques. In particular, this paper models the end-to-end multicasting system as a deep neural network (DNN) autoencoder with one-bit constraint at the output of the precoder represented by a binary thresholding layer, and aims to get insights into the structure of the optimal constellation design by leveraging a novel training strategy from the machine learning literature.

In the machine learning literature, the goal of an autoencoder is typically to find a low-dimensional representation of its input at some intermediate layers which allows reconstruction at the output with minimal error. In communication systems, the concept of autoencoder has recently been used to jointly learn the transmitter and receiver components so that the intended message or symbol can be recovered with small probability of error [11]–[13]. The concept of autoencoder is ideally suited for the one-bit precoding problem, because the 1-bit DAC output can be thought of as a low-dimensional representation of the transmitted information. However, the training of the autoencoder is also challenging, because the gradient of the binary thresholding layer (representing the one-bit constraint) is always zero, so the conventional back-propagation used in [11]–[13] does not work. To get around with this problem, we approximate the gradients of the binary layer with a variant of the *straight-through* (ST) estimator [14]–[16]. In particular, we employ the sigmoid-adjusted ST estimator with annealing trick proposed in [16], where the thresholding function in the back-propagation phase is approximated with a properly scaled sigmoid function.

For the fixed-channel scenario, we numerically show that the proposed trained autoencoder can achieve a good performance in terms of the average symbol error rate (SER), and the constellations designed by autoencoder is superior to QAM and PSK constellations. Further, using the insights obtained from fixed channel scenarios, we propose a constellation design for varying channel scenarios and numerically show

This work is supported by Huawei Technologies Canada.

that the proposed constellation can achieve a better SER performance as compared to the conventional constellations.

II. System Model

Consider the downlink transmission of a multicasting system in which a BS with M transmit antennas aims to simultaneously communicate one single common symbol from a constellation of size $|\mathcal{C}|$, i.e., $s \in \mathcal{C}$, to K single-antenna users in each channel use. For such a system, if we denote the vector of channel gains between the BS and user k by $\mathbf{h}_k \in \mathbb{C}^M$, the M-dimensional transmitted signal can be written as a function of the instantaneous channel state information (CSI), $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_K]^H$, and the intended symbol s as:

$$\bar{\mathbf{x}} = \mathcal{P}\left(s, \mathbf{H}\right), \tag{1}$$

where $\mathcal{P} : \mathcal{C} \times \mathbb{C}^{K \times M} \to \mathbb{C}^M$ represents the precoder. The received signal at user k can then be modeled as:

$$y_k = \rho\, \mathbf{h}_k^H \mathbf{x} + z_k, \tag{2}$$

where $\mathbf{x} = \rho^{-1}\bar{\mathbf{x}} \in \mathbb{C}^M$ is the normalized transmitted signal with $\rho = \sqrt{P/2M}$, $z_k \sim \mathcal{CN}(0, 2\sigma^2)$ is the white Gaussian noise, and P is the total transmit power budget. Upon reception of y_k, user k seeks to recover the intended symbol by mapping its received signal to the nearest constellation point, i.e., $\hat{s}_k = \mathcal{Q}(y_k)$. In this paper, we consider the multicast system for simplicity. However, the proposed autoencoder framework can be extended to the more general unicast scenario with independent symbols transmitted to each user.

One-bit precoding refers to the scenario in which one-bit DACs are employed at the transmit antennas. This means that $\mathbf{x}$ must come from a finite alphabet, i.e., $\mathbf{x} \in \mathcal{X}^M$, where $\mathcal{X} = \{\pm 1 \pm \imath\}$, with $\imath$ denoting the imaginary unit. In this case, the process for designing $\mathbf{x}$ is combinatorial and for this reason most existing design strategies are based on discrete optimization heuristics, e.g., the precoding algorithm proposed in [8] is a combination of greedy and exhaustive search.

Moreover, existing one-bit precoding methods [3]–[10] always consider a fixed receive constellation, e.g., QAM or PSK constellations. The investigation of the optimal constellation design, which can potentially impact the eventual system performance, is still not yet available. The main goal of this paper is to obtain insights into the structure of the optimal constellation design by jointly designing the transmitted signal and the receive constellation via the autoencoder framework.

In this paper, we consider the conventional massive MIMO setup with time-division duplex assuming uplink-donwlink channel reciprocity, in which the BS can obtain an estimate of CSI via training on the uplink, and then use that CSI for the purpose of downlink precoding. In order to focus on the impact of constellation design and one-bit precoding scheme, similar to the existing literature on one-bit precoding [3]–[10], we assume perfect uplink-downlink reciprocity so that full CSI can be obtained at the BS. At the receivers' side, we do not assume the availability of the CSI, and instead we assume that the required information about the constellation is available at the users. In this paper, we first restrict our attention to a fixed channel scenario in which the constellation is designed at the BS for that particular channel, and the information about the designed constellation is then fed back to the users. By applying the insights obtained from designed constellations in the fixed channel scenario, we then propose a constellation design for more realistic varying channel scenarios. In that design, the general shape of the constellation is fixed for different fading blocks and only a single parameter needs to be fed back to the users in each coherence time.

III. End-to-End One-Bit Precoding System Design using Autoencoder

In this section, we show how to represent an end-to-end multicasting communications system with one-bit precoder as an autoencoder and how to train that autoencoder to find the constellation design and the corresponding one-bit precoding scheme for a given channel matrix $\mathbf{H}$.

A. Autoencoder Representation

Since most of the existing deep learning libraries only support real-value operations, we first need to transform the complex model in (2) to the following equivalent real model:

$$\underbrace{\begin{bmatrix}\Re\{y_k\} \\ \Im\{y_k\}\end{bmatrix}}_{\widetilde{\mathbf{y}}_k} = \rho \underbrace{\begin{bmatrix}\Re\{\mathbf{h}_k^H\} & -\Im\{\mathbf{h}_k^H\} \\ \Im\{\mathbf{h}_k^H\} & \Re\{\mathbf{h}_k^H\}\end{bmatrix}}_{\widetilde{\mathbf{H}}_k} \underbrace{\begin{bmatrix}\Re\{\mathbf{x}\} \\ \Im\{\mathbf{x}\}\end{bmatrix}}_{\widetilde{\mathbf{x}}} + \underbrace{\begin{bmatrix}\Re\{z_k\} \\ \Im\{z_k\}\end{bmatrix}}_{\widetilde{\mathbf{z}}_k}.$$

Now, let $m \in \{1, \ldots, |\mathcal{C}|\}$ denote the index of the intended symbol and let $\mathbf{1}_m \in \mathbb{R}^{|\mathcal{C}|}$ denote the one-hot representation of m, i.e., a $|\mathcal{C}|$-dimensional vector with the m^{th} element being one and the other elements being zero. By considering $\mathbf{1}_m$ as the input of the autoencoder, the non-linear one-bit precoder in (1) can be modeled by a DNN with multiple dense layers followed by a binary layer ensuring that the one-bit constraints on elements of $\widetilde{\mathbf{x}}$ are met. In this model, the $2M$-dimensional real-valued normalized transmitted signal $\widetilde{\mathbf{x}}$ can be written as:

$$\widetilde{\mathbf{x}} = \operatorname{sgn}\left(\mathbf{W}_{\text{T}}\sigma_{\text{T-1}}\left(\cdots \mathbf{W}_2\sigma_1\left(\mathbf{W}_1\mathbf{1}_m + \mathbf{b}_1\right) + \cdots \mathbf{b}_{\text{T-1}}\right) + \mathbf{b}_{\text{T}}\right),$$

where T is the number of layers, σ_{t} is the activation function for the t^{th} layer, $\Theta_{\text{T}} = \{\mathbf{W}_{\text{t}}, \mathbf{b}_{\text{t}}\}_{\text{t}=1}^{\text{T}}$ is the set of the transmitter's trainable parameters, and $\operatorname{sgn}(\cdot)$ denotes the sign function. Here, the dimensions of the trainable matrices and the bias vectors are respectively:

$$\dim\left(\mathbf{W}_{\text{t}}\right) = \begin{cases} \ell_{\text{t}} \times |\mathcal{C}|, & t = 1, \\ \ell_{\text{t}} \times \ell_{\text{t-1}}, & t = 2, \ldots, T-1, \\ 2M \times \ell_{\text{t-1}}, & t = T, \end{cases} \tag{3}$$

and

$$\dim\left(\mathbf{b}_{\text{t}}\right) = \begin{cases} \ell_{\text{t}} \times 1, & t = 1, \ldots, T-1, \\ 2M \times 1, & t = T, \end{cases} \tag{4}$$

where ℓ_{t} is the number of neurons in the t^{th} hidden layer.

Analogously, the receivers' operations are modeled by another DNN with R dense layers, where the r^{th} layer includes ℓ'_{r} neurons. In this paper, one common DNN is used to represent

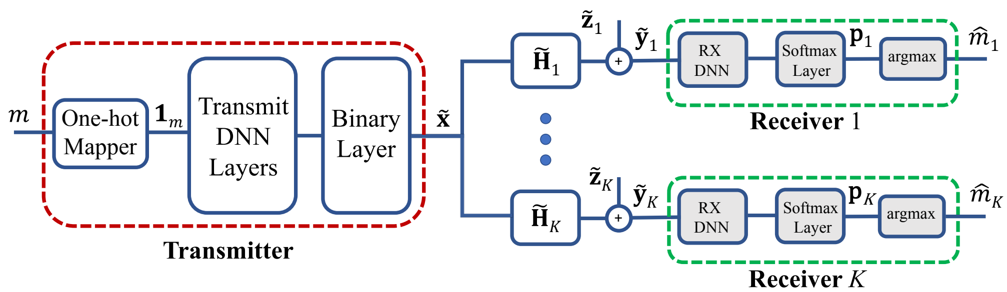

Fig. 1: A block diagram of the end-to-end autoencoder representing a K-user multicasting system in which the BS has one-bit precoding architecture.

the decoding procedure of different users. The motivation of considering such a common receive DNN structure is two-folded. First, it reduces the dimensions of the receivers' trainable parameters which can potentially lead to a faster training procedure. Second, since in this model different users employ the same decision rules to recover m, the final decision boundaries and consequently the designed receive constellation will be the same for all the users. Therefore, after designing the constellation, the BS only needs to broadcast the common constellation parameters to all the users which significantly reduces the amount of required feedback compared to the case that different users adopt different constellations.

In the receive DNN, the activation function of the last layer is set to be a softmax activation in order to generate the probability vector at user k, $\mathbf{p}_k \in (0,1)^{|\mathcal{C}|}$, where its i^{th} element indicates the probability for the index of the intended symbol to be i. Finally, receiver k declares $\hat{m}_k$, which corresponds to the index of the element of $\mathbf{p}_k$ with the highest probability, as the decoded index of the intended symbol. The block diagram of the proposed end-to-end autoencoder that represents a multicasting system with one-bit precoder is shown in Fig. 1. The next step is to learn the transmitter's precoding scheme and the receivers' decision rules by properly training the autoencoder in Fig. 1.

B. Training Autoencoder with a Binary Layer

The proposed autoencoder can be trained end-to-end using stochastic gradient descent (SGD) on the set of all possible symbol indices using the average cross entropy loss function between the input layer $\mathbf{1}_m$ and the probability vectors generated by the users, which can be written as:

$$\mathcal{L}_{\text{CE}} = -\mathbb{E}_{\tilde{\mathbf{z}}}\left[\frac{1}{K|\mathcal{C}|}\sum_{k=1}^{K}\sum_{m=1}^{|\mathcal{C}|}\log p_{k,m}\right], \tag{5}$$

where $p_{k,m}$ is the m^{th} element of the probability vector generated by the k^{th} user, $\mathbf{p}_k$, and the expectation is over the distribution of the noise.

SGD-based training methods require partial derivatives of the loss function with respect to all the trainable parameters in order to update those parameters in each iteration. A common practice in SGD-based training to obtain those partial derivatives is *back-propagation*, which is an efficient method of computing gradients in directed graphs of computations. However, the binary layer that is adopted as the last layer of the transmitter impedes the use of simple conventional back-propagation method. This is because the derivative of the output of a binary neuron with $\text{sgn}(\cdot)$ activation function is zero almost everywhere with the exception of the origin at which the function is not even differentiable. As a result, gradients can never flow through a binary layer and consequently any neural layers before a binary layer cannot be trained with the conventional back-propagation method.

This issue has been tackled in machine learning literature by approximating $\text{sgn}(\cdot)$ by another differentiable function in the back-propagation phase. In [14], Hinton proposes *straight-through* (ST) estimator in which a binary neuron is treated as an identity function during the back-propagation phase. This means that the ST estimator simply estimates the gradient of a binary neuron as 1. A variant of the ST estimator, called *sigmoid-adjusted ST*, replaces the derivative factor with the gradient of $2\,\text{sigm}(u)-1$, where $\text{sigm}(u) = 1/(1+\exp(-u))$ is the sigmoid function [15]. The performance of the sigmoid-adjusted ST estimator can be further improved by using the *slope-annealing* trick, in which the slope of the sigmoid function is slowly increased as training progresses [16]. Particularly, the sigmoid-adjusted ST with slope annealing estimator approximates $\text{sgn}(u)$ in the back-propagation phase with

$$2\,\text{sigm}(\alpha^{(i)}u) - 1 = \frac{2}{1+\exp(-\alpha^{(i)}u)} - 1, \tag{6}$$

where $\alpha^{(i)}$ is the annealing factor in the i^{th} epoch satisfying $\alpha^{(i)} \geq \alpha^{(i-1)}$. In this paper, we employ sigmoid-adjusted ST with annealing during the back-propagation phase to compute the gradients of the binary layer considered at the last stage of the transmitter in the autoencoder in Fig. 1. In the next section, we explain the adopted annealing strategy and the implementation details of the end-to-end autoencoder training.

C. Implementation Details

We implement the autoencoder network in Fig. 1 on TensorFlow which is an open source Python-based machine learning framework [17]. To train such an autoencoder, we use a variant of the stochastic gradient descent method for optimizing deep networks, called *Adam optimizer* [18] with an adaptive learning rate whose initial value is set to be 0.001.

In our implementation, the number of transmit and receive hidden layers are set to $T = 12$ and $R = 5$, respectively,

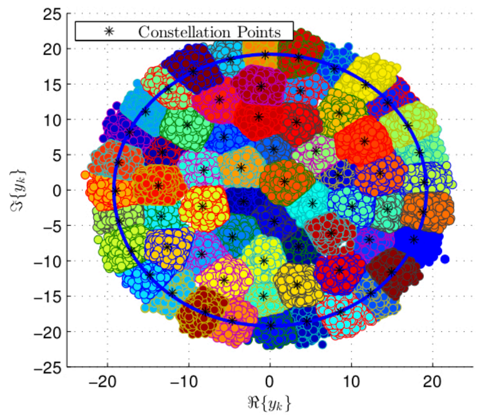

Fig. 2: The receive constellation points and their corresponding decision boundaries obtained from a trained autoencoder. The circle depicted with a blue solid line is centered at the origin and has radius $d^\star$.

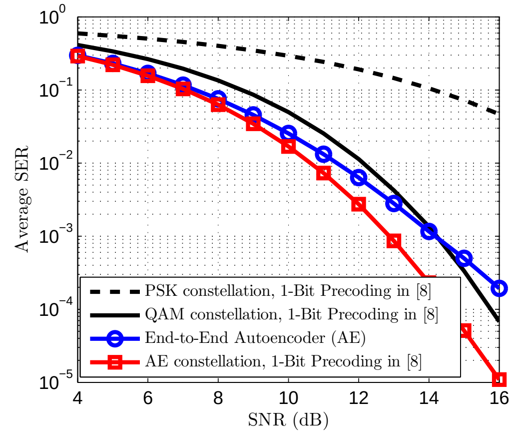

Fig. 3: Average SER versus SNR in a fixed channel scenario.

while the number of transmit and receive hidden neurons are set to $\ell_{\rm t} = 6M$, $\forall t$ and $\ell'_{\rm r} = 2M$, $\forall r$, respectively. These large values for the network dimensions are chosen in order to illustrate the ultimate performance of the proposed autoencoder. Further, we employ exponential linear units (ELUs) [19] as the activation function of the hidden layers. In the training stage, we assume that the noise variance is unknown and it is randomly generated so that the signal-to-noise ratio, $\text{SNR} \triangleq 10\log_{10}(\frac{P}{2\sigma^2})$, is uniformly distributed in a reasonable range. This approach allows us to train the network such that it can operate on a wide range of SNRs. Moreover, we replace the expectation in (5) by the empirical average over the training samples to compute the average cross entropy loss function. Finally, we remark that numerical experiments suggest that considering a sigmoid-adjusted ST estimator with slow annealing strategy leads to the best performance. Therefore, in our simulations, we update the annealing parameter in the i^{th} epoch as $\alpha^{(i)} = 1.002\alpha^{(i-1)}$ where $\alpha^{(0)} = 1$ so that the annealing parameter after 2000 epochs becomes $1.002^{2000} \approx 55$ which is reasonably large to accurately approximate the $\text{sgn}(\cdot)$ function.

After the autoencoder is trained, we obtain the one-bit precoding scheme and constellation design obtained by the autoencoder. In particular, the transmitted signal in (1) can simply be constructed from $\widetilde{\mathbf{x}}_m \in \{\pm 1\}^{2M}$, which is the output of the binary layer in Fig. 1 when the input is set to be m. Further, the receive constellation points can be obtained by averaging the noiseless received signals of different users.

IV. Numerical Results

In the numerical experiments, we consider a multicasting system in which a BS with $M = 128$ antennas serves $K = 4$ users by transmitting symbols from a constellation of size $|C| = 64$ in an environment with $\mathbf{h}_k \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$, $\forall k$.

A. Fixed Channel Scenario

In this experiment, we randomly generate a realization of $\mathbf{H}$ for which we numerically evaluate the performance of the proposed autoencoder-based one-bit precoding and constellation design. As a competitor, we consider one of the existing one-bit precoding algorithms [8] in combination with QAM and PSK constellations. The one-bit precoding algorithm in [8] involves two steps. In the first step the transmitted signal of M_1 antennas is designed in a greedy fashion to minimize mean squared error, while in the second step the transmitted signal of remaining $M_2 = M - M_1$ antennas are designed using an exhaustive search method. Here, we set $M_1 = 120$ and $M_2 = 8$. Further, [8] suggests to design the constellation range for a one-bit unicasting system such that the condition

$$\mathbf{s}^H \left(\mathbf{H}\mathbf{H}^H\right)^{-1} \mathbf{s} \leq \tfrac{2}{\pi} P, \tag{7}$$

in which $\mathbf{s}$ is the vector of intended symbols, is almost always satisfied. By applying (7) to a multicasting system, we set the range in QAM and PSK constellations so that the distance of furthest constellation points from the origin becomes

$$d^\star = \sqrt{\frac{\frac{2}{\pi}P}{\mathbf{1}^T \left(\mathbf{H}\mathbf{H}^H\right)^{-1} \mathbf{1}}}. \tag{8}$$

Fig. 2 depicts the constellation points as well as their corresponding decision boundaries obtained from a trained autoencoder. Interestingly, we observe that $d^\star$ in (8) can accurately approximate the distance of the furthest constellation points from the origin. We would like to remark that, although we only present the obtained constellation for one particular realization of $\mathbf{H}$, in the simulations we observe that the general shapes of the designed constellations are similar for different channel matrix realizations. In particular, constellations designed by the autoencoder suggest that constellation points should be placed in a circle with radius $d^\star$ in a way that the minimum distance in the constellation is maximized. We use this observation in the next subsection to tackle the constellation design problem for varying channel scenarios.

Fig. 3 plots the average SER against SNR. It can be seen that the proposed autoencoder-based scheme achieves a better performance over a reasonable range of SNRs as compared to the one-bit precoding method [8] in combination with either 64-QAM or 64-PSK. However, in high SNR regime,

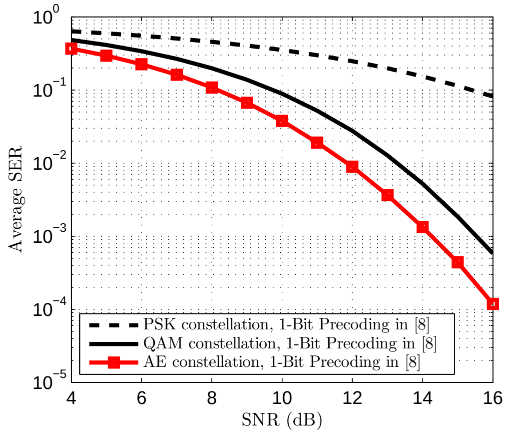

Fig. 4: Average SER versus SNR in the varying channel scenario.

one-bit precoding algorithm in [8] with QAM constellation outperforms the proposed autoencoder-based method. To investigate the reason of such a performance behavior, Fig. 3 also plots the performance of the one-bit precoding method [8] adopted for the receive constellation designed by the autoencoder. Interestingly as shown in Fig. 3, such a method achieves the best performance as compared to all the previous schemes. This suggests that the proposed autoencoder can indeed help find a better receive constellation design, however, the performance of its precoding scheme is not necessarily better than [8] with $M_2 = 8$ in its exhaustive search step.

B. Varying Channel Scenario

The proposed autoencoder-based one-bit precoding and constellation design is restricted to scenarios that the channel is fixed for a long period of time so that it is feasible to consider constellation design for that particular channel. However, in more realistic scenarios, the channel is changing, and we need to design a common receive constellation (with few tunable parameters) for all channel realizations. To tackle the constellation design problem for such scenarios, this paper uses the insights obtained from designed constellations in the fixed channel scenario. Particularly, we observe that the general shapes of the designed constellations are similar regardless of the realization of $\mathbf{H}$ and the only parameter of the constellation that depends on realization of $\mathbf{H}$ is its range which can be accurately characterized by $d^\star$ in (8). Inspired by those observations, we propose to use a constellation designed for one particular $\mathbf{H}$, e.g., Fig. 2 for a constellation with size 64, and for other $\mathbf{H}$ only to rescale that constellation properly so that the range of the constellation becomes $d^\star$.

Fig. 4 plots the average SER against SNR for the varying channel scenario. Fig. 4 shows that the proposed constellation design with one-bit precoding algorithm [8] can improve the performance by about 1.5dB and 8dB as compared to the QAM and PSK constellations with the same precoding scheme, respectively. This indicates that the proposed constellation design, which is inspired by the autoencoder-based constellation design in the fixed channel scenario, is indeed effective.

V. Conclusion

This paper first proposes a joint one-bit precoding and constellation design for a multicasting system with a fixed channel matrix, by training a DNN autoencoder. Using the insights obtained from the fixed channel scenario, this paper then extends its constellation design to the varying channel scenario. Numerical results show that the proposed autoencoder-based constellation design for one-bit precoding can achieve a better performance as compared to the conventional constellations.

References

[1] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[2] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4331–4340, Sept. 2011.

[3] A. Mezghani, R. Ghiat, and J. A. Nossek, "Transmit processing with low resolution D/A-converters," in *Proc. IEEE Int. Conf. Electron., Circuits, Syst. (ICECS)*, Yasmine Hammamet, Tunisia, Dec. 2009, pp. 683–686.

[4] A. K. Saxena, I. Fijalkow, and A. L. Swindlehurst, "Analysis of one-bit quantized precoding for the multiuser massive MIMO downlink," *IEEE Trans. Signal Process.*, vol. 65, no. 17, pp. 4624–4634, Sept. 2017.

[5] O. Castañeda, T. Goldstein, and C. Studer, "POKEMON: A non-linear beamforming algorithm for 1-bit massive MIMO," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, New Orleans, USA, Mar. 2017, pp. 3464–3468.

[6] S. Jacobsson, G. Durisi, M. Coldrey, T. Goldstein, and C. Studer, "Quantized precoding for massive MU-MIMO," *IEEE Trans. Commun.*, vol. 65, no. 11, pp. 4670–4684, Nov. 2017.

[7] M. Shao, Q. Li, W.-K. Ma, and A. M.-C. So, "A framework for one-bit and constant-envelope precoding over multiuser massive MISO channels," *IEEE Trans. Signal Process.*, vol. 67, no. 20, pp. 5309–5324, Oct. 2019.

[8] F. Sohrabi, Y.-F. Liu, and W. Yu, "One-bit precoding and constellation range design for massive MIMO with QAM signaling," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 3, pp. 557–570, June 2018.

[9] H. Jedda, A. Mezghani, J. A. Nossek, and A. L. Swindlehurst, "Massive MIMO downlink 1-bit precoding with linear programming for PSK signaling," in *Proc. IEEE Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Sapporo, Japan, Jul. 2017.

[10] A. Li, C. Masouros, F. Liu, and A. L. Swindlehurst, "Massive MIMO 1-bit DAC transmission: A low-complexity symbol scaling approach," *IEEE Trans. Wireless Commun.*, vol. 17, no. 11, pp. 7559–7575, 2018.

[11] 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.

[12] 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.

[13] 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 Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Kalamata, Greece, June 2018, pp. 1–5.

[14] G. Hinton, "Neural networks for machine learning," *Coursera lectures*.

[15] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," Aug. 2013. [Online]. Available: https://arxiv.org/abs/1308.3432

[16] J. Chung, S. Ahn, and Y. Bengio, "Hierarchical multiscale recurrent neural networks," Sept. 2016. [Online]. Available: https://arxiv.org/abs/1609.01704

[17] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," Mar. 2016. [Online]. Available: https://arxiv.org/abs/1603.04467

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

[19] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," Nov. 2015. [Online]. Available: https://arxiv.org/abs/1511.07289