Deep Autoencoder-based Z-Interference Channels

Xinliang Zhang and Mojtaba Vaezi
Department of Electrical and Computer Engineering
Villanova University, Villanova, PA 19085, USA
Email: {xzhang4, mvaezi}@villanova.edu

2023 IEEE Wireless Communications and Networking Conference (WCNC) | 978-1-6654-9122-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/WCNC55385.2023.10118986

***Abstract*—A deep autoencoder (DAE)-based communication over the two-user Z-interference channel (ZIC) is introduced in this paper. The proposed DAE-ZIC is designed to minimize the bit error rate (BER) in the presence of interference by jointly optimizing the encoders and decoders. Effectively, this is an end-to-end communication that designs new constellations for the ZIC. Normalization layers are embedded in the proposed DAE design to realize an average power constraint so that there are no regular shape restrictions on the constellation symbols. We compare the performance of the DAE-ZIC with two baseline methods, which are ZIC with regular and rotated constellations. Simulation results show a significant gain in BER reduction. On average, in weak, moderate, and strong regimes, 31%–75% BER improvement is achieved compared to the best existing methods.**

I. Introduction

Interference is a central issue in today's *multi-cell* networks. The information-theoretic model for a multi-cell network is the *interference channel* (IC). The capacity region of the two-user IC is only known for *strong* and *very strong* interference [1] where decoding and canceling the interference is optimal [2]. Also, at *very weak* interference, sum-capacity is achievable by treating interference as noise [3]–[5]. In general, Han-Kobayashi encoding, which decodes part of the interference and treats the remaining as noise, is the best achievable scheme [6]. The two-user Z-interference channel (ZIC), also known as one-sided IC [7], is a special case of the two-user IC in which only one user suffers from interference. The capacity region of the ZIC is only determined in the strong and very strong interference regimes with Gaussian signaling [7], [8].

The above Shannon-theoretic works are based on Gaussian inputs. Despite being theoretically appealing, Gaussian alphabets are continuous and unbounded, and thus, are rarely applied in real-world communication systems. In practice, signals are generated using finite alphabet sets and the common practical approaches for the IC are *orthogonalizing* the time/frequency and *treating interference as noise*. Although treating interference as noise is sum-capacity optimal at very weak interference [3]–[5], such an approach is very inefficient when interference is moderate or strong.

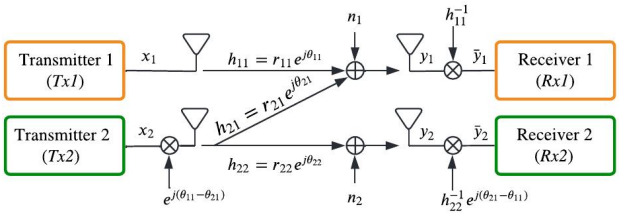
Some work has considered the ZIC when the channel inputs are restricted to finite alphabet sets for specific regimes. In [9], it is shown that rotating one input alphabet can improve the sum-rate of the two-user IC in strong/very strong interference regimes. The performance of the ZIC with regular constellations can be improved by the rotation constellations. Later, an exhaustive search for finding the optimal rotation of the signal constellation was presented in [10]. Transmission strategies in both low and high SNRs for the multi-antenna IC are studied in [11]. The focus of the aforementioned works is to maximize the mutual information and sum-rate, while bit error rate (BER) performance has rarely been analyzed.

Recently, several researchers have studied the performance of the IC using new tools from deep learning. Notably, an end-to-end learning-based approach is introduced for the two-user IC [12], [13] by using a channel autoencoder-based deep learning technique. By jointly training autoencoders for very short blocklengths, it was shown that neural network-based systems can outperform time-sharing schemes [13]. An adaptive deep learning algorithm for the multi-user IC is proposed in [14]. This approach can learn and predict dynamic interference by utilizing pilot signals to estimate the strength of interference. This method is shown to outperform the conventional system using *uncoded* phase-shift keying (PSK) or quadrature amplitude modulations (QAM). However, the above papers focus on symbols with equal power and compare their results with baselines that use QAM signals, whereas there are more competitive conventional methods like rotated QAM in the literature.

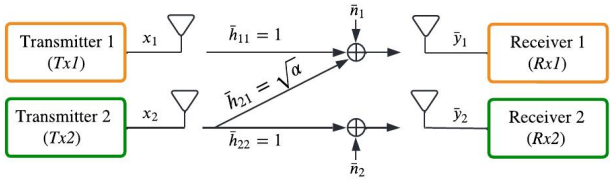
In this paper, we introduce using deep autoencoder (DAE)-based communication in the two-user ZIC with finite-alphabet inputs. In our design, two DAE pairs are considered for the two transmitters and receivers. By adapting their constellations to the interference intensity, the two DAEs cooperate to avoid interference as far as possible. The main contributions of the paper are as follows:

- We design a DAE-based transmission for the ZIC with finite alphabets at different interference regimes, including weak, moderate, and strong interference. The constellations designed by the DAE-ZIC are adaptive to the interference intensity at a variety of channel conditions.
- Different from the existing DAE designs [12]–[14], an average power constraint normalization layer is designed which allows the nonuniform distribution of constellations. Thus, the in-phase and quadrature-phase (I/Q) plane is used more efficiently.
- The BER of the proposed DAE-ZIC is noticeably lower than the ZIC with a rotated QAM constellation which is the best conventional ZIC. Specifically, there is 31%-75% reduction in BER depending on the interference regime (weak, moderate, and strong interference).

The remainder of this paper is organized as follows. We



(a) Original system model



(b) Equivalent system model

Fig. 1: The equivalent system model of the ZIC.

elaborate on the ZIC system model in Section II. The DAE design and the training approach are introduced in Section III. Numerical results are presented in Section IV, and the paper is concluded in Section V.

II. System Model of the ZIC

Figure 1(a) shows the system model of a two-user single-input single-output (SISO) ZIC. The two transmitter-receiver pairs wish to reliably transmit their messages while the transmission of the first pair interferes with the transmission of the second. The four nodes are named *Tx1*, *Tx2*, *Rx1*, and *Rx2*, as shown in Fig. 1(a). h_{ij} is the channel coefficients from the ith transmitter to the jth receiver and $i, j \in \{1, 2\}$. For ZIC, $h_{21} = 0$. The received signals at the receivers can be written as

$$y_1 = h_{11}x_1 + h_{21}x_2 + n_1, \tag{1a}$$
$$y_2 = h_{22}x_2 + n_2, \tag{1b}$$

in which x_1 and x_2 denote the transmitted symbols of *Tx1* and *Tx2*. The transmitted signals are complex-valued with finite-alphabets and variances $\mathbb{E}\{|x_1|^2\} = P_1$ and $\mathbb{E}\{|x_2|^2\} = P_2$ in which P_1 and P_2 are the power budgets of the two transmitters. The channel coefficients are

$$h_{ij} \triangleq r_{ij}e^{j\theta_{ij}} \sim \mathcal{CN}(\mu_H, \sigma_H^2), \tag{2}$$

where μ_H and σ_H^2 are the mean and variance of the channel. n_1 and n_2 are the complex-valued independent and identically distributed (i.i.d.) additive white Gaussian noise with zero means and variances σ_1^2 and σ_2^2.

In some papers [8], [15], [16], the channel gains of the direct transmission links are directly modeled as one, shown in Fig. 1(b). The interference gain is also real-valued for both real- and complex-valued systems. If channel coefficients are available and we apply pre- and post-processing illustrated in Fig. 1(a), the system model in Fig. 1(a) is equivalent to that of Fig. 1(b). The *Tx2* applies $e^{j(\theta_{11}-\theta_{21})}$ to cancel the phase of h_{21}. The *Rx1* and *Rx2* apply h_{11}^{-1} and $h_{22}^{-1}e^{j\theta_{21}}$ to normalize the channel gain to one. Then, the received post-processed signals are

$$\bar{y}_1 = h_{11}^{-1}y_1 = x_1 + r_{21}r_{11}^{-1}x_2 + n_1h_{11}^{-1}, \tag{3a}$$
$$\bar{y}_2 = h_{22}^{-1}e^{j\theta_{21}}y_2 = x_2 + e^{j\theta_{21}}n_1h_{22}^{-1}. \tag{3b}$$

By defining $\sqrt{\alpha} \triangleq r_{21}r_{11}^{-1}$, $\bar{n}_1 \triangleq n_1h_{11}^{-1}$, and $\bar{n}_2 \triangleq e^{j\theta_{21}}n_1h_{22}^{-1}$, we have the system model in Fig. 1(b) as

$$\bar{y}_1 = \bar{h}_{11}x_1 + \sqrt{\alpha}x_2 + \bar{n}_1, \tag{4a}$$
$$\bar{y}_2 = \bar{h}_{22}x_2 + \bar{n}_2. \tag{4b}$$

where $\bar{h}_{11} = \bar{h}_{22} = 1$ and $\bar{h}_{21} = \sqrt{\alpha}$ are the equivalent channel gains.

Thus, the two system models in Fig. 1 are equivalent. Hence, we follow existing studies and use the system model in Fig. 1(b), and consider a fixed $\sqrt{\alpha}$ at each time. It is worth mentioning that both actual channel gains and noise (h_{ij} and n_i, $i, j \in \{1, 2\}$) are Gaussian. In this paper, we assume a slow fading scenario. The $\bar{n}_i$ is normal and

$$\bar{n}_i \sim \mathcal{CN}(0, \sigma_i^2 r_{ii}^{-2}). \tag{5}$$

III. Deep Autoencoder for the ZIC

Existing studies on finite-alphabet ZIC [9], [10] use standard QAM constellations. Such constellations are fixed and are not adjustable according to the interference intensity. To further improve the transmission performance, we propose a DAE-based transmission for the two-user ZIC, named the DAE-ZIC. The architecture is shown in Fig. 2.

A. The Architecture of the DAE-ZIC

The DAE-ZIC consists of two pairs of DAEs. Each pair performs an end-to-end transmission, which includes input bits and $\bar{h} = \sqrt{\alpha}$, autoencoder at the transmitter, channel and noise layers, autoencoder at the receiver, and final output bits.

1) Network Input: Each transmitter sends N_s bits to the corresponding receiver. The interference channel coefficient $\sqrt{\alpha}$ is known at the transmitter and receiver and is appended to the input bit vector. Then, both the transmitters and receivers know the channel coefficients. The two transmitters are expected to jointly design their constellations and the receivers will decode correspondingly.

2) Transmitter DAE: As shown in Fig. 2, the DAE of the transmitter contains two sub-networks: *Sub-network 1* and *Sub-network 2*. *Sub-network 1* converts the input bit-vector to symbols taking the value of h_{21} into account. *Sub-network 2* performs power allocation, which controls the power of the I/Q components. The batch normalization in *sub-network 1* and *sub-network 2* together realize the average power constraint at the transmit antenna. This is different from the existing DAEs designs [12]–[14], [17] which generate symbols with fixed power. Having an average power constraint is necessary especially for SISO systems. In this way, the I/Q plane is used efficiently, like QAM. Otherwise, the DAE can only use pulse-amplitude keying (PAM) which is not energy efficient.

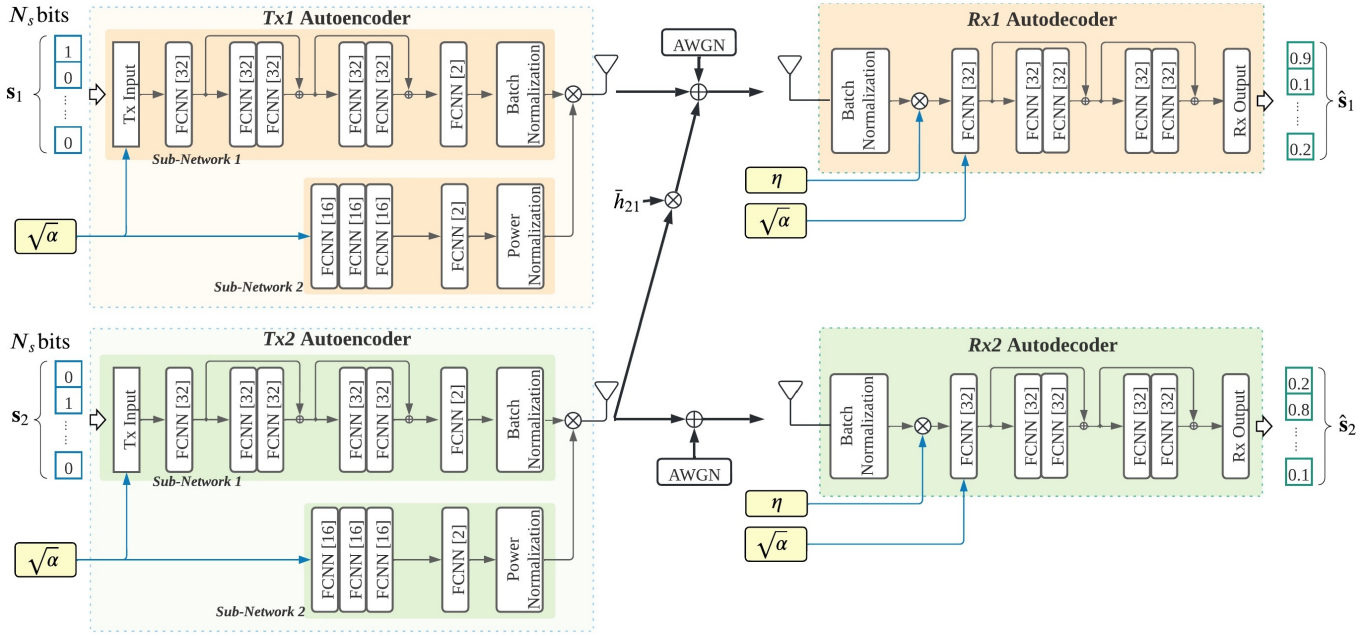


Fig. 2: The architecture of the two-user DAE-ZIC implemented by two pairs of deep autoencoders. Each transmitter of the ZIC contains two sub-networks. *Sub-network 1* mainly generates the constellation and *sub-network 2* is used to implement the average power constraint. The receivers decode their bits from the received signal. This architecture is based on the system model in Fig. 1(b). η is a power control parameter defined in (12).

The main components of *sub-network 1* are fully connected layers (FCNN), residual connections, and the output batch normalization layer.[1] The activation function of the FCNN layers is *tanh* except for the last layer, which has two hidden nodes and no activation function. Assume the batch size is N_B, and the output of the last FCNN is

$$\mathbf{X}_{\text{fcnn}} \triangleq [\mathbf{x}^{\text{I}}_{\text{fcnn}},\ \mathbf{x}^{\text{Q}}_{\text{fcnn}}], \tag{6}$$

where $\mathbf{x}^{\text{I}}_{\text{fcnn}}$ and $\mathbf{x}^{\text{Q}}_{\text{fcnn}} \in \mathbb{R}^{N_B \times 1}$ are the outputs of the two hidden nodes and represent I/Q of the complex-valued signal.

Since the FCNN cannot achieve any power constraint for the transmission, we propose a DAE design to achieve the average power constraint at the antenna. First, we use batch normalization in *sub-network 1* to unify the statistics of the input to the next layer. The batch normalization layer linearly normalizes the average power of $\mathbf{x}^{\text{I}}_{\text{fcnn}}$ and $\mathbf{x}^{\text{Q}}_{\text{fcnn}}$ independently. The normalized vectors $\mathbf{x}^{\text{I}}_B$ and $\mathbf{x}^{\text{Q}}_B$ are given by

$$\mathbf{x}^{\text{I}}_B \triangleq \beta^{\text{I}} \cdot \mathbf{x}^{\text{I}}_{\text{fcnn}},\ \ \mathbf{x}^{\text{Q}}_B \triangleq \beta^{\text{Q}} \cdot \mathbf{x}^{\text{Q}}_{\text{fcnn}}, \tag{7}$$

where $\boldsymbol{\beta} \triangleq [\beta^{\text{I}},\ \beta^{\text{Q}}]^T$ contains two factors for normalization.

Then, the powers of $\mathbf{x}^{\text{I}}_B$ and $\mathbf{x}^{\text{Q}}_B$ are modified by *sub-network 2*. *Sub-network 2* has two output values: γ^{I} and γ^{Q}. The FCNN layers in *sub-network 2* determine the power allocated of the I/Q components based on the input value $\sqrt{\alpha}$. The power normalization block in *sub-network 2* limits the total power to P_t. Defining $\boldsymbol{\gamma} \triangleq [\gamma^{\text{I}},\ \gamma^{\text{Q}}]^T \in \mathbb{R}^{2\times 1}$, we should have $\boldsymbol{\gamma}^T\boldsymbol{\gamma} = P_t$. Such an operation can be done via the *Lambda layer* in KERAS. Finally, the outputs of the batch normalization and power normalization are multiplied together, i.e.,

$$\mathbf{x}^{\text{I}}_{\text{out}} \triangleq \gamma^{\text{I}} \cdot \mathbf{x}^{\text{I}}_B,\ \ \mathbf{x}^{\text{Q}}_{\text{out}} \triangleq \gamma^{\text{Q}} \cdot \mathbf{x}^{\text{Q}}_B. \tag{8}$$

The powers of $\mathbf{x}^{\text{I}}_{\text{out}}$ and $\mathbf{x}^{\text{Q}}_{\text{out}}$ are γ^{I} and γ^{Q}, respectively. To summarize, the batch normalization is applied to the I/Q components along the time axis, while the power normalization normalizes the I/Q components at each time. Hence, the two normalization operations are implemented in different dimensions. In this way, a average power constraint is reached. The final output of the transmitter is

$$\begin{aligned}\mathbf{X} &= [\mathbf{x}^{\text{I}}_B,\ \mathbf{x}^{\text{Q}}_B] \cdot \text{diag}(\boldsymbol{\gamma}) \\ &= [\mathbf{x}^{\text{I}}_{\text{fcnn}},\ \mathbf{x}^{\text{Q}}_{\text{fcnn}}] \cdot \text{diag}(\boldsymbol{\beta}) \cdot \text{diag}(\boldsymbol{\gamma}).\end{aligned} \tag{9}$$

where $\mathbf{x}^{\text{I}}_{\text{fcnn}}$ and $\mathbf{x}^{\text{Q}}_{\text{fcnn}}$ are the output of the FCNN layer in *sub-network 1* and represent the preliminary I/Q signals, $\boldsymbol{\beta}$ normalizes the power, and $\boldsymbol{\gamma}$ controls the power of the I/Q signals.

3) Receiver DAE: The received signals are $\bar{y}_1$ and $\bar{y}_2$. To ensure the receiver networks have a finite input range, we use *Batch Normalization layers* in KERAS unifying the power of the received signals, i.e.,

$$y_{\text{B},i} = \xi \cdot \bar{y}_i,\ \ E\{|y_{\text{B},i}|^2\} = 1,\ \ \forall i \in \{1, 2\}, \tag{10}$$

where ξ is a coefficient to reach the unit power. The process details and settings are the same as the ones in the transmitter.

We further define the *desired signal* for *Rx1* as

$$x_{\text{D},1} \triangleq x_1 + \sqrt{\alpha} x_2. \tag{11}$$

[1]The FCNN and residual connections inherit the design of the point-to-point MIMO transmission in [17].

$x_{\mathrm{D},1}$ contains the truly desired signal x_1 and the interference $\sqrt{\alpha}x_2$. The goal of the receiver is to decode x_1 for an arbitrary x_2 in its constellation. The *desired signal* of *Rx2* is $x_{\mathrm{D},2} \triangleq x_2$. Since the normalization in (10) is performed on the *desired signal* with AWGN, the autoencoder should adjust the decoding boundary according to the noise power, which is an extra burden. To eliminate the effect of the noise, we use a linear factor, η, multiplied by the batch normalization output, i.e.,

$$y_{\mathrm{D},i} = \eta \cdot y_{\mathrm{B},i},\ \eta \triangleq \sqrt{1 + P_{\mathrm{D,i}}\sigma_N^{-2}}, \quad \forall i \in \{1,2\}, \tag{12}$$

where $P_{\mathrm{D},i}$ is the power of the *desired signal* $x_{\mathrm{D},i}$ and σ_N^2 is the noise power. In short, batch normalization normalizes the *desired signals* using pre-processing η. Then, the normalized signal, $y_{\mathrm{D},i}$, together with the feature of the ZIC, $h_{21} = \sqrt{\alpha}$, are sent to the rest of the FCNN layers. The final output of the DAE is an estimation of the transmitted bit-vectors, $\hat{\mathbf{s}}_1$ and $\hat{\mathbf{s}}_2$, as shown in Fig. 2. The output layer uses soft-max. More specifically, the activation function is sigmoid.

4) Loss Function: In our DAE-ZIC, each receiver has their own estimation of the transmitted bits. Then, the overall loss function of the DAE-ZIC is $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2$, where $\mathcal{L}_1$ and $\mathcal{L}_2$ are the loss at *Rx1* and *Rx2*. In this paper, we use binary cross-entropy as the loss function, i.e.,

$$\mathcal{L}_i = \frac{1}{N_B}\sum_{n=1}^{N_B}(\mathbf{s}_{i,n})^T \log \hat{\mathbf{s}}_{i,n} + (1-\mathbf{s}_{i,n})^T \log(1-\hat{\mathbf{s}}_{i,n}), \tag{13}$$

where $i \in \{1,2\}$ distinguishes the users, N_B is the batch size, $\mathbf{s}_{i,n}$ is the nth input bit-vector in the batch, and $\hat{\mathbf{s}}_{i,n}$ is corresponding the output. The loss function treats each element of the DAE output as a zero/one classification task. Cross-entropy is used to evaluate each classification task. Finally, the loss is the summation of the loss of N_s tasks, where N_s is the number of bits in the transmission. In the training process, the back propagation algorithm passes $\mathcal{L}_1$ to *Rx1* and it will further go to *Tx1* and *Tx2*. The $\mathcal{L}_2$ affects the *Rx2* and *Tx2*.

B. Training Procedure of the DAE-ZIC

To reduce the difficulty in training, we use separate instances of DAEs to realize a wide choice of interference gains α. In each training, we select an N_s and the desired range for $\alpha \in [\alpha_{\min}, \alpha_{\max}]$. We train the DAE repeatedly using random values of α in this interval. For each α, the DAE is trained through epochs E_p, mini-batch size N_B, and a constant learning rate l_r. After training the DAE for N_d different values of α, the learning rate is reduced to $d_r l_r$. The detailed training procedure is summarized in Algorithm 1. Also, we choose the best DAEs from five realizations trained alone with the same hyperparameters. The best is defined on the average loss value on ten random generated αs in $[\alpha_{\min}, \alpha_{\max}]$.

Algorithm 1 Training Procedure for the DAE-ZIC

```
Fix N_s, α_min, and α_max.
Set P_t = 1W, SNR= 10dB, and N_α = 30,000.
Set E_p = 10, and N_B = 10^4.
Set l_r = 10^{-2}, the initial learning rate, which will drop
    to d_r l_r = 0.95 l_r after every N_d = 200 trained channels.
Initialize the DAE-ZIC network.
for index i_α from 1 to N_α do
    Uniformly and randomly select one α ∈ [α_min, α_max].
    Randomly generate h_11 and h_22 using (2).
    Normalize the channel using (4).
    Set the variance of the noise layer according to (5).
    for index i_e from 1 to E_p do
        Randomly generate N_B bit vectors.
        Update the weights of the DAE-ZIC using Adam.
    end for
    Set learning rate l_r = d_r l_r if i_α/N_d is an integer.
end for
```

IV. Performance Analysis

The performance is evaluated and compared for the three methods below:

- *DAE-ZIC*: The proposed method which designs new constellations based on the interference intensity.
- *Baseline-1*: The transmitters directly use standard QAM.
- *Baseline-2*: *Tx1* uses standard QAM, while *Tx2* rotates the standard QAM symbols based on the interference intensity [9], [10].

An illustrative example of the difference between the two baselines is shown in Fig. 3 in which $M_1 = M_2 = 4$ and $\alpha = 0.2$. The optimal angle of the rotation in *Baseline-2* is 13°. The distance among the symbols at *Rx1* is enlarged by the rotation.

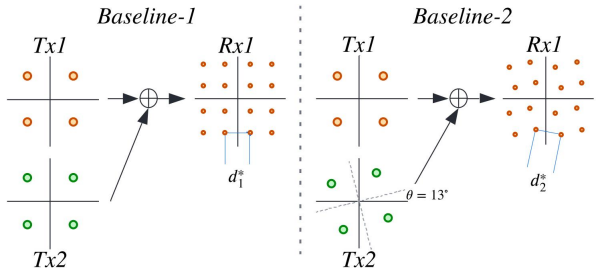
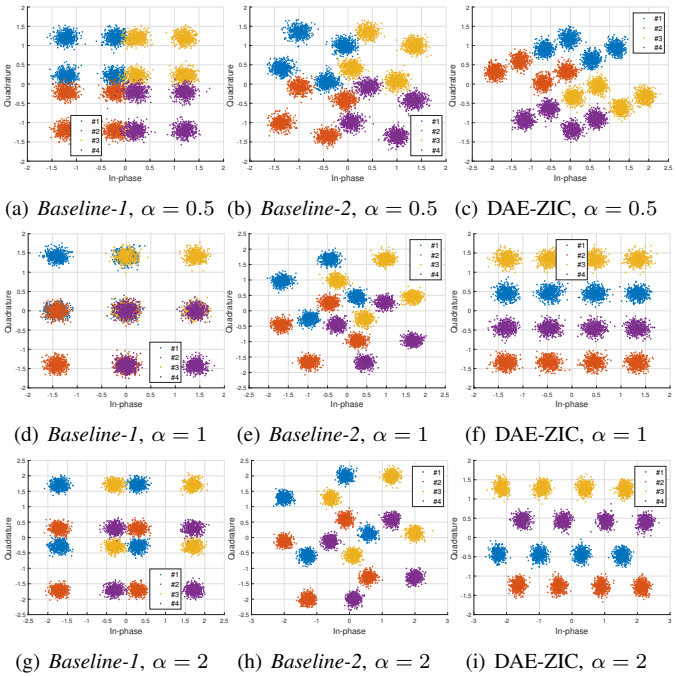


Fig. 3: An example constellations in *Baseline-1* and *Baseline-2* when $M_1 = M_2 = 4$, $\alpha = 0.2$, and $\theta = 13^\circ$. In the two methods, $d_1^* < d_2^*$ are minimum symbol distances in which the angles are 0° and 13°, respectively.

Next, we illustrate the constellations given by the proposed DAE-ZIC method and analyze its performance.

A. Constellation analysis for the DAE-ZIC

The received constellations at *Rx1* generated by the baselines and the proposed DAE-ZIC are shown in Fig. 4. In this simulation, we set $N_s = 2$ so that each user has $2^{N_s} = 4$ information symbols. The channels are known as $h_{11} = h_{22} = 1$ and $h_{21} = \sqrt{\alpha}$. The transmit power is unit one and the SNR is 8dB.



(a) *Baseline-1*, $\alpha = 0.5$ (b) *Baseline-2*, $\alpha = 0.5$ (c) DAE-ZIC, $\alpha = 0.5$

(d) *Baseline-1*, $\alpha = 1$ (e) *Baseline-2*, $\alpha = 1$ (f) DAE-ZIC, $\alpha = 1$

(g) *Baseline-1*, $\alpha = 2$ (h) *Baseline-2*, $\alpha = 2$ (i) DAE-ZIC, $\alpha = 2$

Fig. 4: Constellations of the DAE-ZIC and the two baselines at *Rx1* for three values of α.

Each sub-figure of Fig. 4 contains four symbol clusters differentiated by different colors. Each cluster refers to a symbol transmitted to *Rx1*. Within each cluster, there are four symbols, which are the symbols of *Rx2*. For example, the blue colors together denote symbol 1 for *Rx1*, and the four blue symbols individually correspond to the four symbols of *Rx2*. These symbols are polluted by interference from the other user and AWGN noise. It can be seen that the location and distribution of symbols are different in each method. The constellations of *Baseline-1* (left column) are very crowded and even overlapped when $\alpha = 1$ in Fig 4(d). This is because *Tx2* strongly interferes with the transmission between *Tx1* and *Rx1* by directly applying 4-QAM. *Baseline-2* (middle column) rotates the constellation of *Tx2*, which enlarges the space between symbols and thus makes the decoding easier. The proposed DAE-ZIC (right-column) creates the most separable constellations. It can better make use of the I/Q plane in constellation design based on the interference intensity. The two cooperating DAEs can intelligently choose and adjust various scaled constellation types to avoid constellation overlapping. When $\alpha = 0.5$, the DAE-ZIC designs a parallelogram-shape constellation compared with the square-shape constellations in the baselines. When $\alpha = 1$, both *Tx1* and *Tx2* choose PAM. The two PAM constellations are perpendicular to each other hence the overlapping is eliminated. When $\alpha = 2$, *Tx1* uses a parallelogram-shape QAM and *Tx2* uses a PAM. By adapting their constellations to the interference intensity, the two DAEs cooperate to avoid constellation overlap as much as possible. This is the main reason that the DAE-ZIC outperforms the baselines.

B. BER Performance of the DAE-ZIC

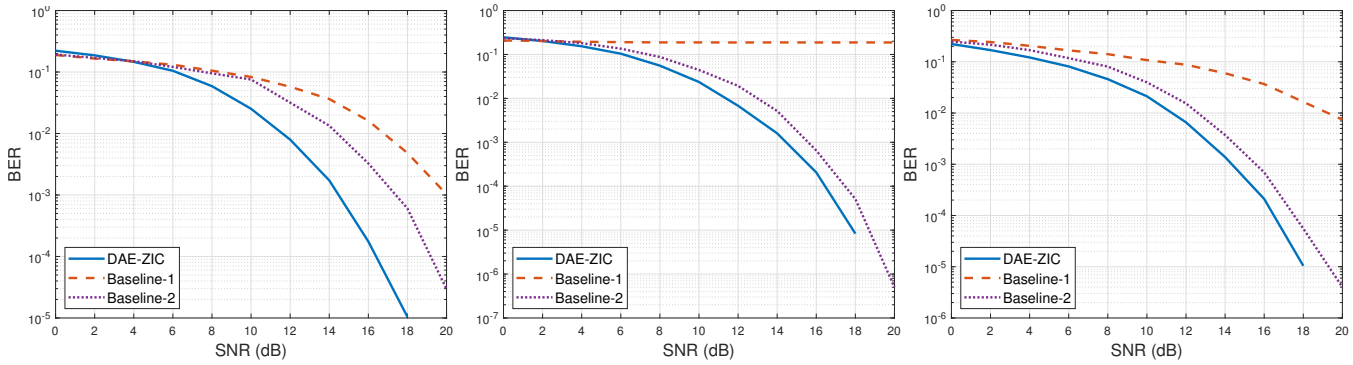
To evaluate the effectiveness of the DAE-ZIC, we compare the BER of the three methods over SNR in $[0, 20]$dB and $0 \leq \alpha \leq 3$. For the DAE-ZIC, we divide interval $\alpha \in [0, 3]$ into six sub-intervals, i.e, $[0, 0.5)$, $[0.5, 1)$, ..., $[2.5, 3]$. For each sub-interval of α, we train a DAE-ZIC through Algorithm 1.

The BER versus the SNR is shown in Fig. 5. In each sub-figure, α is a fixed value. In general, the DAE-ZIC outperforms the two baseline models, especially in moderate and strong interference regimes. With $N_s = 3$, the performance of DAE-ZIC drops at 0dB. The reason is we have trained the network at SNR $= 10$dB but have tested it for a range of SNRs from 0 to 20dB. A potential way to improve is to train the DAE-ZIC with a variety of SNRs.

The BER performance versus α at SNR $= 10$dB is shown in Fig. 6. In Fig. 6(a) and Fig. 6(b), we set $N_s = 2$ and $N_s = 3$, i.e., $M_1 = M_2 = 4$ and $M_1 = M_2 = 8$. When interference is very weak, i.e., $\alpha \in [0, 0.25]$, the three methods have similar BERs. The proposed DAE-ZIC noticeably reduces the BER in weak, moderate, and strong interference cases, where $\alpha \in [0.5, 2]$. From Fig. 6(a), when using DAE-ZIC over $\alpha \in [0, 3]$, BER is reduced 75.7% and 44.29%, compared to *Baseline-1* and *Baseline-2*. When $\alpha \in [0.5, 2]$, the improvement becomes 80.3% and 51.5%. When the interference gain is very strong, e.g., $\alpha > 2.5$, *Baseline-2* outperforms DAE-ZIC. The reason could be that 4-QAM with rotation may achieve optimal performance [9]. For $N_s = 3$ in Fig. 6(b), 44.4% and 31.5% BER reduction is reached by the DAE-ZIC over $\alpha \in [0, 3]$. For $\alpha \in [0.5, 2]$, DAE-ZIC outperforms the other two methods with 45.0% and 33.4%. *Baseline-1* performs poorly for $\alpha \in [0.5, 2]$, because the two added QAM constellations may get crossed and overlapped. Thus, *Rx1* cannot decode its message. Such a phenomenon is alleviated in *Baseline-2* which simply rotates one QAM constellation and the added constellations still have a reasonable symbol distance. The normalization layer allows the DAE-ZIC to design constellations without any regular-shape restrictions. Thus, the minimum distance at receivers can be enlarged which results in a lower BER.

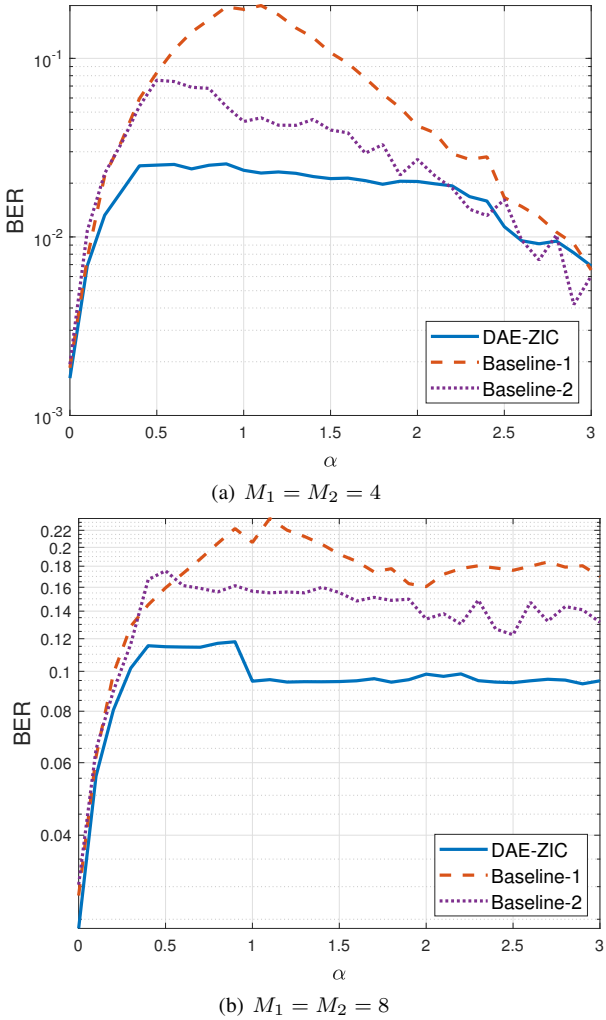
V. Conclusion

A DAE design for communication over the single-antenna two-user ZIC has been proposed. The DAE-ZIC minimizes the BER by jointly designing transmit and receive DAEs and optimizing them. The average power constraint is achieved by a designed normalization layer. Thus, the DAE-ZIC is enabled to design more efficient symbols to achieve a lower BER. The effectiveness of the proposed structure is verified by simulations. We have compared the performance of the proposed DAE-ZIC with two baseline models, and the DAE-ZIC outperforms both. In general, there are significant improvements by exploiting DAE in different interference regimes. Intuitively, our DAE outperforms the baseline methods because it resigns constellations to make the symbols separable at the interfered receiver.



(a) $M_1 = M_2 = 4$, $\alpha = 0.5$ (b) $M_1 = M_2 = 4$, $\alpha = 1$ (c) $M_1 = M_2 = 4$, $\alpha = 1.5$

Fig. 5: The maximum (worst) BER performance among the two users of DAE-ZIC versus SNR. The BER is averaged over different interference gains which is uniformly distributed in the given interval.



(a) $M_1 = M_2 = 4$

(b) $M_1 = M_2 = 8$

Fig. 6: The maximum (worst) BER performance versus interference gains. The SNR is fixed as 10dB.

REFERENCES

[1] A. Carleial, "A case where interference does not reduce capacity (corresp.)," *IEEE Transactions on Information Theory*, vol. 21, no. 5, pp. 569–570, 1975.

[2] Z. K. Ho and E. Jorswieck, "Improper Gaussian signaling on the two-user SISO interference channel," *IEEE Transactions on Wireless Communication*, vol. 11, no. 9, pp. 3194–3203, 2012.

[3] A. S. Motahari and A. K. Khandani, "Capacity bounds for the Gaussian interference channel," *IEEE Transactions on Information Theory*, vol. 55, no. 2, pp. 620–643, 2009.

[4] X. Shang, G. Kramer, and B. Chen, "A new outer bound and the noisy-interference sum–rate capacity for Gaussian interference channels," *IEEE Transactions on Information Theory*, vol. 55, no. 2, pp. 689–699, 2009.

[5] V. S. Annapureddy and V. V. Veeravalli, "Gaussian interference networks: Sum capacity in the low-interference regime and new outer bounds on the capacity region," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3032–3050, 2009.

[6] T. Han and K. Kobayashi, "A new achievable rate region for the interference channel," *IEEE Transactions on Information Theory*, vol. 27, no. 1, pp. 49–60, 1981.

[7] M. Costa, "On the Gaussian interference channel," *IEEE Transactions on Information Theory*, vol. 31, no. 5, pp. 607–615, 1985.

[8] M. Vaezi and H. V. Poor, "Simplified Han-Kobayashi region for one-sided and mixed Gaussian interference channels," in *Proc. IEEE International Conference on Communications (ICC)*, pp. 1–6, 2016.

[9] F. Knabe and A. Sezgin, "Achievable rates in two-user interference channels with finite inputs and (very) strong interference," in *Proc. IEEE Asilomar Conference on Signals, Systems and Computers (ACSSC)*, pp. 2050–2054, 2010.

[10] A. Ganesan and B. S. Rajan, "Two-user Gaussian interference channel with finite constellation input and FDMA," *IEEE Transactions on Wireless Communications*, vol. 11, no. 7, pp. 2496–2507, 2012.

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

[12] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[13] T. Erpek, T. J. O'Shea, and T. C. Clancy, "Learning a physical layer scheme for the MIMO interference channel," in *Proc. IEEE International Conference on Communications (ICC)*, pp. 1–5, 2018.

[14] D. Wu, M. Nekovee, and Y. Wang, "Deep learning-based autoencoder for M-user wireless interference channel physical layer design," *IEEE Access*, vol. 8, pp. 174679–174691, 2020.

[15] C. Lameiro, I. Santamaría, and P. J. Schreier, "Rate region boundary of the SISO Z-interference channel with improper signaling," *IEEE Transactions on Communications*, vol. 65, no. 3, pp. 1022–1034, 2016.

[16] G. Kramer, "Review of rate regions for interference channels," in *Proc. IEEE International Zurich Seminar on Communications (IZSC)*, pp. 162–165, 2004.

[17] X. Zhang, M. Vaezi, and T. J. O'Shea, "SVD-embedded deep autoencoder for MIMO communications," in *Proc. IEEE International Conference on Communications (ICC)*, pp. 1–6, 2022.