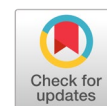


Deep reinforcement learning autoencoder with ra-gan and gan



Hoang-Sy Nguyen ^{a,1}, Cong-Danh Huynh ^{b,2,*}

^a Becamex Business School, Eastern International University, Binh Duong Province, Vietnam

^b Thu Dau Mot University, Binh Duong Province, Vietnam

¹ sy.nguyen@eiu.edu.vn; ² danhhc@tdmu.edu.vn

* corresponding author

ARTICLE INFO

Article history

Received August 23, 2022

Revised November 4, 2022

Accepted November 11, 2022

Available online November 30, 2022

Keywords

Artificial neural networks

Communication systems

Reinforcement learning-based training

Channel models

ABSTRACT

Deep learning utilization to optimize block-structured communication systems has attracted tremendous attention from researchers. Nevertheless, owing to the extensive data transmission between the transmitter and the receiver, communication, in this case, is hard to establish and maintain effectively. As a solution for this, we first investigate typical end-to-end learning for communication system, Generative Adversarial Network (GAN). Then, two problems associated with GAN-based systems, the gradient vanishing and overfitting, are reviewed. Subsequently, a residual aided GAN (RA-GAN) is proposed as means to overcome these problems. In the proposed learning scheme, the residual learning and the regularization method are used to mitigate the gradient vanishing and overfitting problems. In the proposed learning scheme, the residual learning and the regularization method are used to mitigate the gradient vanishing and overfitting problems. Finally, the numerical results performed in MATLAB for simulation and Codelabs for training have proven that the RA-GAN scheme has near-optimal performance and outperforms the conventional GAN scheme. Throughout this case study, readers can understand the issues that would occur when deep learning is applied to a communication system and possible approaches to address them.



This is an open access article under the [CC-BY-SA](#) license.



1. Introduction

Recently, deep learning (DL) has been widely investigated for performance improvement of the conventional block-structured communication systems, i.e., the multiple-input and multiple-output (MIMO) detection [1], channel estimation [2], and channel decoding [3]. Besides, DL shows its potentials in joint optimization of communication blocks, regarding the channel estimation and detection [4] or in the task of encoding channel and source [5].

Utilization of DL in improving the communication blocks is a data-driven approach, since a DL model can learn directly on the provided data by optimizing a given end-to-end loss function without the need of ad-hocs. In view of this, a number of works have investigated a so-called end-to-end (E2E) learning communication system. It should be noted that the transmitters and receivers were considered as deep neural networks (DNNs), which play respectively the role of auto-encoders and auto-decoders [6] [9]. The supervised learning was used to optimize the weights of the models, leading to the improvement of the end-to-end recovery accuracy [6]. It was proven that the performance of this method is performs as well as the traditional block-structured systems operating with additive white Gaussian noise (AWGN) channel. Additionally, the method was developed to counteract hardware imperfection [8], or to boost the orthogonal frequency-division multiplexing (OFDM) systems [9]. Additionally,

authors in [7] proposed a novel hybrid approach where supervised training and reinforcement learning are applied respectively on the receiver and transmitter.

Notably, the above works consider solely the most undemanding AWGN channel case, and the systems were already burdened with the substantial amount of data transmission between the receiver and the transmitter. To overcome this issue, a number of channel imitation-based schemes utilizing Deep Learning (DL) were introduced [10] [11] [12], in which addition modules used to imitate the channels in practice are installed. Particularly, DL models can imitate any arbitrary functions with the help of Deep Neural Network (DNN). It is highly capable of solving large-scale and complex problems [13]. DNN can either serve as a separate signal processing module or be implemented in an existing module to improve its performance [14] [15] [16]. Additionally, it can be used instead of a complex transceiver to remove the hardware burden when the designers attempt to optimize the global performance of wireless networks from the end-to-end (E2E) viewpoint [17].

The received signal was imitated with a generative adversarial network (GAN) in [18]. GAN has two components, which are the generator and the discriminator deployed in the framework of multi-layer DNNs. During the DNNs training, a fake received signal imitating the real received signal is generated by the generator to establish and maintain the training process for the transmitter. Meanwhile, the discriminator trains the generator to ensure that the signal it generates is as close as possible to the real signal. Thereby, the bridge for the standard propagation (BP) for gradient calculation for the transmitter is built. Papers [19] [20] [21] [22] [23] have shown that any arbitrary channel can be imitated with this method, and the hardware complexity of the transceiver can be significantly reduced. However, there are two drawbacks with the method, being the possible gradient vanishing problem with multi-layer generator and the over-fitting problem during iterative training. These problems affect the imitation accuracy of the GAN, which leads to the degrade of the E2E learning, thus the system performance [24].

A wireless system is fundamentally divided into different modules, i.e., source and channel encoder/decoder, (de)modulator, etc. [17]. Therefore, the nonlinear behavior of the system cannot be simply expressed using mathematics and optimizing the individual modules does not guarantee the optimization of the global system [25]. It was proven in [26] that designing modulation and coding separately is sub-optimal and the same principle is applied to designing other components as well, which hinders the global optimization of conventional wireless networks. Thus, there is a need to change the design paradigm with other approaches.

One of the approaches to optimize the communication system is to utilize end-to-end (E2E) learning. However, an E2E learning communication system only be trained via BP with known channel [3]. Thus, it is necessary to deploy GAN, which is capable of imitating any arbitrary channel and reduce effectively the complexity of transceiver's hardware [27] [28] [29]. The GAN system faces its inherent problems being gradient vanishing (when calculation is passed through the multi-layer NN structures) and overfitting (when attempting to optimize several modules and their weights at once). The two problems will cause the downgrade of the system performance.

As a solution, we propose in this paper a so-called residual aided GAN (RA-GAN) scheme, in which generator's structure is modified with the residual neural network (Resnet) studied in [30]. Instead of generating the received signal in GAN, the generator in RA-GAN generates the difference between the signal that the system transmits and receives. Specifically, instead of the layer-by-layer, a skip connection is deployed so that in the generator, the input layers and output layers are connected. Moreover, the loss function is changed in RA-GAN so as to mitigate the over-fitting issue in GAN by adding a regularizer. The benefits of such modifications are presented as the contributions of the study:

- Generator's structure in the conventional GAN-based system is modified using Resnet to counter the gradient vanishing and overfitting problems.
- By employing a skip connection to link the the input and output, it is possible to produce extra gradient (difference between the signals that the system transmits and receives), as a counteraction to the gradient vanishing issue. By adding a regularizer in the loss function in RA-GAN, which is

not computationally complex, the representation ability of the training scheme is limited so that the over-fitting issue can be mitigated.

- In comparison with the GAN scheme, the fake signal that RA-GAN generates is closer to the real received signal, proving that the proposed residual generator we propose outperforms the conventional one;

Following the Introduction in Section I. The method for GAN and RA-GAN training schemes of an E2E learning communication system is reported in Section II. Section III reports the simulation results. Eventually, the paper is concluded in Section IV.

2. Method

Fig.1 illustrates a structure of an E2E learning communication system which have three components: the transmitter T , inter-mediate channel n , and receiver R . Two multi-layer NNs T and R are deployed respectively with trainable weights of θ_T and θ_R . At first, a piece of information x input to T is mapped to a one-hot vector $\mathbf{1}_m$. Notably, $\mathbf{1}_m$ is a M -dimensional vector taken from \mathcal{M} . In this set, the m -th element is 1, and the rest $M-1$ are 0.

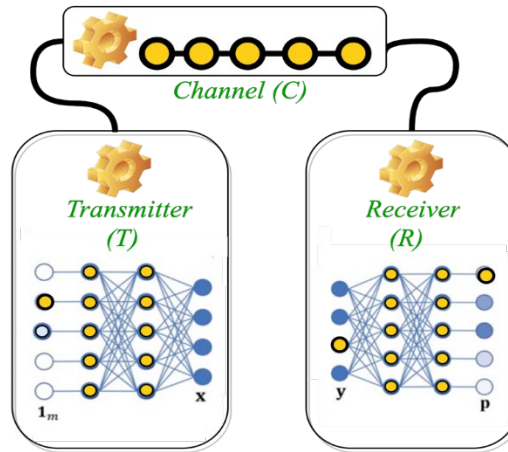


Fig. 1. Deep learning applied on the architecture of E2E learning

As a next step, T is deployed as a function $f_{\theta_T}: \mathcal{M} \mapsto \mathcal{C}^n$ to map the one-hot vector $\mathbf{1}_m$ to the signal $\mathbf{x} \in \mathcal{C}^n$ so that it can be sent via n channels. Similarly, R is a function $f_{\theta_R}: \mathcal{C}^n \mapsto \{\mathbf{p} \in R_+^M \mid \sum_{i=1}^M \mathbf{p}_i = 1\}$, which is used to map the received signal $\mathbf{y} \in \mathcal{C}^n$ to a probability vector $\mathbf{p} \in R_+^M$. Subsequently, we obtain the final decision \hat{x} , corresponding to the input $\mathbf{1}_{\hat{m}}$, with \hat{m} being the maximum value in the probability vector \mathbf{p} . It should be noted that the T 's hardware defines the power constraint put on signal \mathbf{x} , that is, $|\mathbf{x}|^2 = 1$. The final aim is to retrieve $\mathbf{1}_m$ so that it can be as accurate as possible to the signal the system receives.

$$y = \kappa x + v \quad (1)$$

where $\kappa \in \mathcal{C}$ and $v \in \mathcal{C}^n$ are respectively the channel and Gaussian noise. The standard complex Gaussian distribution whose mean and variance are $\mathcal{CN}(0,1)$. We use a loss function following [9] to compare $\mathbf{1}_m$ that is transmitted and the recovered \mathbf{p} :

$$\mathcal{L}(\theta_T, \theta_R, \kappa) \triangleq E \left\{ \int l(f_{\theta_R}(y), \mathbf{1}_m) \mathbf{p}_\kappa(y \mid f_{\theta_T}(\mathbf{1}_m)) dy \right\} \approx \frac{1}{B} \sum_{i=1}^B l(f_{\theta_R}(y^{(i)}), \mathbf{1}_{m^{(i)}}), \quad (2)$$

$$l(\mathbf{p}, \mathbf{1}_m) = -\sum_{i=1}^M (\mathbf{1}_m)_i \ln \mathbf{p}_i + (1 - (\mathbf{1}_m)_i) \ln (1 - \mathbf{p}_i). \quad (3)$$

To acquire the optimal weights θ_T^* and θ_R^* respectively for T and R , we apply a back propagation (BP) algorithm on the loss function $\mathcal{L}(\theta_T, \theta_R, \kappa)$ in (2) to calculate the gradient. Nonetheless, in (2), we can only update θ_R with the below gradient:

$$\nabla_{\theta_R} \tilde{\mathcal{L}}(\theta_R) = \frac{1}{B} \sum_{i=1}^B \nabla_{\theta_R} l(f_{\theta_R}(y^{(i)}), \mathbf{1}_m^{(i)}), \quad (4)$$

where there is the loss function approximation $\tilde{\mathcal{L}}$, derived from (2). For functions $f_x \in \mathbb{R}^n$ and $f_y \in \mathbb{R}^k$ with variable $x, y \in \mathbb{R}^m$, we have their gradient matrices as $\frac{\partial f_y}{\partial f_x} \in \mathbb{R}^{k \times n}$, $\frac{\partial y}{\partial x} \in \mathbb{R}^{m \times m}$, and $\nabla_x f_x \in \mathbb{R}^{n \times m}$.

So as to boost the performance of the system, we have to optimize θ_T as well [2]. Nevertheless, the gradient $\nabla_{\theta_T} \tilde{\mathcal{L}}$ cannot be obtained as proven in [4], because the backpropagation process is obstructed with the unknown channel κ as follows:

$$\begin{aligned} \nabla_{\theta_T} \tilde{\mathcal{L}}(\theta_T) &= \frac{1}{B} \sum_{i=1}^B \nabla_{\theta_T} l(f_{\theta_T}(y^{(i)}), \mathbf{1}_m^{(i)}) = \frac{1}{B} \sum_{i=1}^B \frac{\partial l}{\partial f_{\theta_R}} \frac{\partial f_{\theta_R}}{\partial y^{(i)}} \frac{\partial y^{(i)}}{\partial x^{(i)}} \nabla_{\theta_T} f_{\theta_T}(\mathbf{1}_m^{(i)}) \\ &= \frac{1}{B} \sum_{i=1}^B \kappa \frac{\partial l}{\partial f_{\theta_R}} \frac{\partial f_{\theta_R}}{\partial y^{(i)}} \mathbf{I}_n \nabla_{\theta_T} f_{\theta_T}(\mathbf{1}_m^{(i)}), \end{aligned} \quad (5)$$

where the identity matrix of size n is \mathbf{I}_n

2.1. Generative Adversarial Network scheme (GAN)

To overcome this, a GAN illustrated in Fig. 2 was used for generation of surrogate gradient to subsequently update θ_T [19].

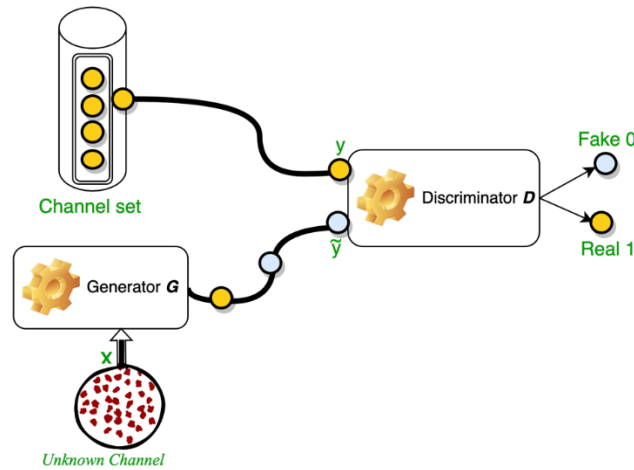


Fig. 2. The scheme of GAN with multi-layer NN generator and discriminator

In general, a GAN includes a generator \mathbf{G} connected to a multi-layer NN discriminator D . Both are with trainable weights respectively symbolized with θ_G and θ_D . We use $\mathbf{G}: f_{\theta_G}: \mathcal{C}^n \mapsto \mathcal{C}^n$ to produce fake received signal \tilde{y} in accordance with the transmitted signal x and random noise z obtained from the standard Gaussian distribution. For simplification, in \mathbf{G} , we consider z as a built-in variable. Meanwhile, the discriminator $f_{\theta_D}: \mathcal{C}^n \mapsto (0,1)$ is deployed for training \mathbf{G} to ensure that the difference between the fake signal and real received signal is minimal.

On the other hand, D is used to recognize the real from the fake signal it receives. Specifically, if the input to D is from the real with signal distribution of $\mathbf{p}_\kappa(y | x)$, output of D will be 1. Contrarily, the output is expected to be 0 for the fake received signal distribution $\mathbf{p}_{\tilde{\kappa}}(\tilde{y} | x)$.

As for \mathbf{G} , to ensure the real and the fake received signals are similar, it has to adjust its own output \tilde{y} so that after inputting into D , the output of D $f_{\theta_D}(\tilde{y})$ is made as close to 1 as possible.

With regard to the GAN's procedure, it is possible to update the weights θ_D of \mathbf{G} as per the loss function of the real received input y as follows:

$$\tilde{\mathcal{L}}(\theta_D) = \frac{1}{B} \sum_{i=1}^B l(f_{\theta_D}(\mathbf{y}^{(i)}, 1) \{l + f_{\theta_D}(\tilde{\mathbf{y}}^{(i)}, 0)\}) \quad (6)$$

Correspondingly, the weights θ_G of D is alternately updated as per the loss function of the fake received input $\tilde{\mathbf{y}}$ with:

$$\tilde{\mathcal{L}}(\theta_G) = \frac{1}{B} \sum_{i=1}^B l(f_{\theta_D}(f_{\theta_G}(\mathbf{x}^{(i)}), 1)). \quad (7)$$

Consequently, we can calculate the gradients with $\nabla_{\theta_G \mathcal{L}}(\theta_G)$ and $\nabla_{\theta_D \mathcal{L}}(\theta_D)$. The loss functions (6) and (7) can be minimized with the Adam gradient descent algorithm in [27], [28] and [29]. Because we can train \mathbf{G} to reproduce the real received signal, we have to ensure that the surrogate gradient is the closest to the gradient (5) and it can be passed back through with:

$$\nabla_{\theta_T} \tilde{\mathcal{L}}(\theta_T) = \frac{1}{B} \sum_{i=1}^B \nabla_{\theta_T} l(f_{\theta_R}(f_{\theta_G}(f_{\theta_T}(\mathbf{1}_m^{(i)}))), \mathbf{1}_m^{(i)}) = \frac{1}{B} \mathcal{J}_1. \quad (8)$$

$$\text{where } \mathcal{J}_1 = \sum_{i=1}^B \frac{\partial l}{\partial f_{\theta_R}} \frac{\partial f_{\theta_R}}{\partial f_{\theta_G}} \frac{\partial f_{\theta_G}}{\partial f_{\theta_T}} \nabla_{\theta_T} f_{\theta_T}(\mathbf{1}_m^{(i)}). \quad (9)$$

However, as mentioned in [24], the training instability will limit the performance of GAN, resulting in the dramatic downgrade of the whole system. As aforementioned, the causes for this are gradient vanishing and over-fitting and would be addressed with a proposed RA-GAN training scheme.

2.2. Ra-gan scheme

Because of the existence of unknown channel, training the transmitter T is a demanding task. As per the GAN training scheme, we can obtain a surrogate gradient using (8) to update T . Nonetheless, the \mathbf{G} output distribution $\mathbf{p}_{\tilde{\kappa}}(\tilde{\mathbf{y}} | \mathbf{x})$ varies much from $\mathbf{p}_{\kappa}(\mathbf{y} | \mathbf{x})$ owing to the problems of gradient vanishing and over-fitting, leading to the proposal of RA-GAN training scheme.

2.2.1. Residual learning to alleviate the gradient vanishing issue

For the traditional GAN, the variables are fed forward through layers and eventually output as fake samples by the multi-layer \mathbf{G} . However, if there are more layers in \mathbf{G} , the gradient would become significantly small because it has been multiplied by the partial derivatives of loss functions when it is passed through the layers in the BP algorithm. If the case the partial derivative is near to 0, it will result in the vanish of the gradient, causing problems to the training process of \mathbf{G} . With inspiration taken from the residual learning in [20], a connection to keep the in-between layers that links the input and the output of \mathbf{G} is designed. Correspondingly, the residual generating function $f_{\theta_G^R}: \mathcal{C}^n \mapsto \mathcal{C}^n$ could be formulated as:

$$f_{\theta_G^R}(\mathbf{x}) = \tilde{\mathbf{y}} - \mathbf{x} = f_{\theta_G}(\mathbf{x}) - \mathbf{x}, \quad (10)$$

where the transmitted and the generated signals are respectively denoted as \mathbf{x} and $\tilde{\mathbf{y}}$, respectively. Additionally, the residual generator $f_{\theta_G^R}(\mathbf{x})$ is implemented to distinguish the transmitted from the received signals according to the conditional input \mathbf{x} .

Then, we can calculate the gradient for updating weights of the T in the RA-GAN scheme as:

$$\nabla_{\theta_T} \tilde{\mathcal{L}}(\theta_T) = \frac{1}{B} \sum_{i=1}^B \frac{\partial l}{\partial f_{\theta_R}} \frac{\partial f_{\theta_R}}{\partial f_{\theta_G}} \frac{\partial f_{\theta_G}}{\partial f_{\theta_T}} \nabla_{\theta_T} f_{\theta_T}(\mathbf{1}_m^{(i)}) = \frac{1}{B} (\mathcal{J}_2 + \mathcal{J}_3), \quad (11)$$

$$\text{where } \mathcal{J}_2 = \sum_{i=1}^B \frac{\partial l}{\partial f_{\theta_R}} \frac{\partial f_{\theta_R}}{\partial f_{\theta_G}} \nabla_{\theta_T} f_{\theta_T}(\mathbf{1}_m^{(i)}) \text{ and } \mathcal{J}_3 = \sum_{i=1}^B \frac{\partial l}{\partial f_{\theta_R}} \frac{\partial f_{\theta_R}}{\partial f_{\theta_G}} \frac{\partial f_{\theta_G^R}}{\partial f_{\theta_T}} \nabla_{\theta_T} f_{\theta_T}(\mathbf{1}_m^{(i)}) \quad (12)$$

2.2.2. Regularization method to mitigate overfitting

The loss function is reconstructed to overcome the over-fitting problem in the E2E learning of the system. As aforementioned, when G and D are added in GAN for training purposes, we can witness the significant increase of additional trainable weights, leading to the over-growing representation ability

thus the over-fitting problem. To mitigate this, a so-called regularizer can be added to the loss function. Being different from the GAN, this regularizer allows the RA-GAN scheme to recover the real signal it receives with the help of regularization method. In particular, for the representation ability restriction, we would need to add a weight $\Omega(\theta)$ for penalty into the loss function of the RA-GAN scheme:

$$\hat{\mathcal{L}}(\theta_i) = \tilde{\mathcal{L}}(\theta_i) + \lambda\Omega(\theta_i), i \in \{R, T, G, D\}, \quad (13)$$

where the loss function that we reconstructed and original loss function are respectively denoted as $\hat{\mathcal{L}}(\theta_i)$ and $\tilde{\mathcal{L}}(\theta_i)$. In addition, there is a hyper-parameter λ deployed to balance $\Omega(\theta)$ and $\tilde{\mathcal{L}}(\theta_i)$.

Fig. 3 depicts the algorithm RA-GAN E2E training schemes. The inputs are iterations number, Epoch, and real channel dataset as inputs, from which the weights of the trained T and R, θ_T and θ_R , are output.

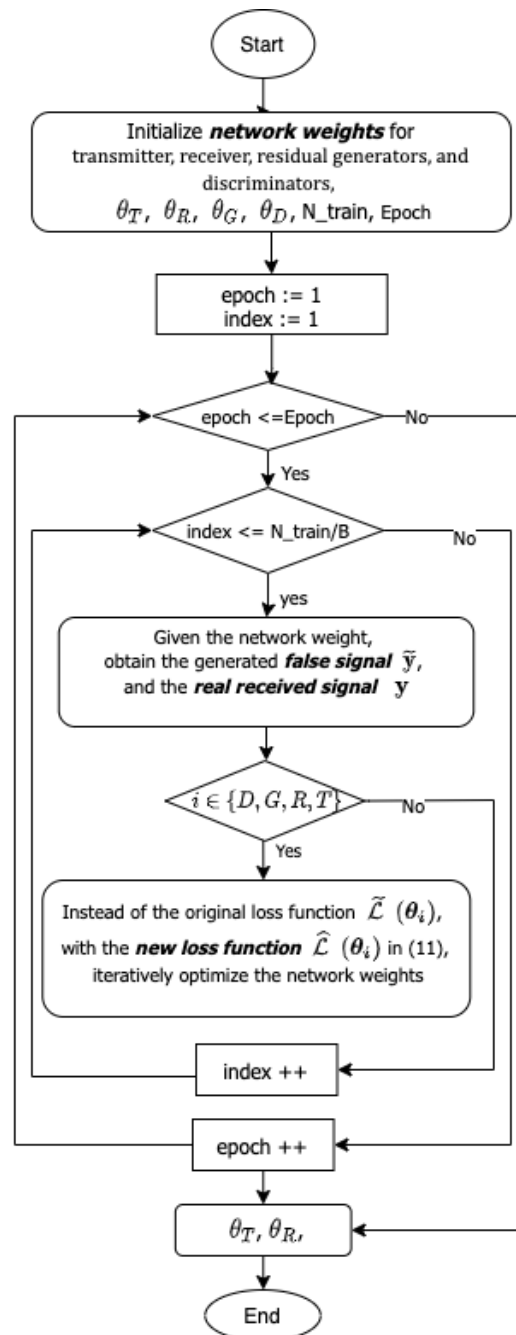


Fig. 3. Algorithm RA-GAN E2E training schemes.

The process is separated to three primary steps: 1. Initializing weights; 2. Generating fake signal and real signal; 3. Training the network weights iteratively. It is worth noting that iterative training is applied for T, R, G, and D in this system. It means that the change in parameters of a module does not lead to the changes of other modules unless it is converged. Additionally, the actual signal transmission is only completed once the parameters of T and R are returned.

3. Results and Discussion

Here, we presents the performance analysis of the RA-GAN scheme in AWGN channel. To evaluate the ability to transmit the data, we focus on the block error rate (BLER). Followingly, we compare the RA-GAN and the conventional GAN scheme in [19], while considering the training method, which has the known channel, as the optimal performance bound.

Specifically, for the optimal case, T is assumed to know the real channel, making the gradient $\nabla_{\theta_T} \tilde{\mathcal{L}}$ available for T training in (10). Moreover, the ability to generate fake received signals of the RA-GAN and the conventional GAN is compared. The simulation parameters are $M = 16, n = 7, B = 320, N_{\text{train}} = 10000, \lambda = 0.01, \delta^2 = \left(\frac{N_0 n}{2E_b \log_2 M}\right)$ in [2]. The simulations are conducted in Matlab with the formulations that are derived previously in this paper.

We used a dataset of 100.000 one-hot vectors that are randomized to validate the BLER performance of the system trained with $\text{SNR} = 5(\text{dB})$. As can be observed in Fig. 4, the GAN-based [19] performs significantly worse than the optimal case (with known channel), owing to the aforementioned gradient vanishing and over-fitting problems happen during GAN training. On the other hand, the RA-GAN scheme is almost identical to the optimal training method. This shows how effectively the existing problems of GAN system is mitigated by employing the skip connection in the generator and regularizer as a part of the loss function.

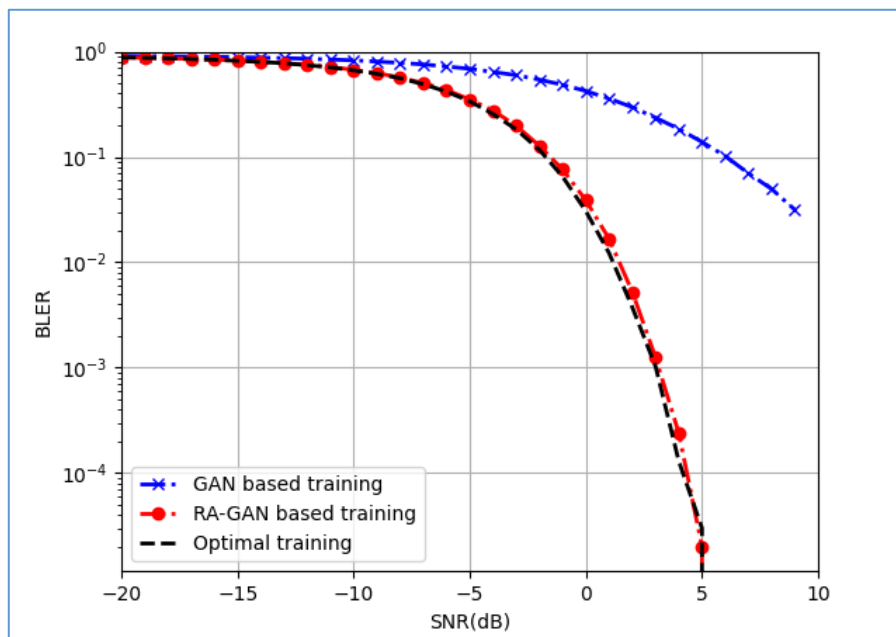


Fig. 4. BLER performance comparison GAN and RA-GAN vs SNR(dB)

In Fig. 5 and Fig. 6, we plotted the results from the loss functions in relation with the training epoch in AWGN channel with $\text{SNR} = 5(\text{db})$. In Fig. 5, we can see that the GAN scheme cannot converge while the RA-GAN one in Fig. 6 can. It should be noted that during the training process, owing to the training randomness, some bad points would exist, however, they could be recovered in the next epoch. Therefore, it can be concluded that the RA-GAN scheme is able to generate signals that are better fit

to the real received one. In other words, the trained residual \mathbf{G} outperforms the conventional \mathbf{G} in terms of generation performance.

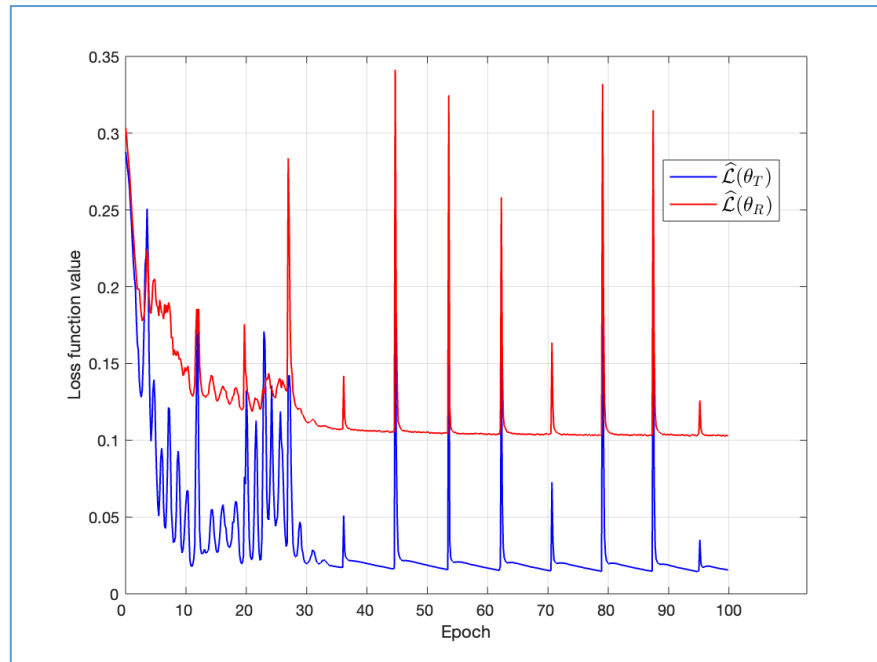


Fig. 5. The original loss function $\tilde{\mathcal{L}}(\theta_R)$ and $\tilde{\mathcal{L}}(\theta_T)$ for GAN schemes

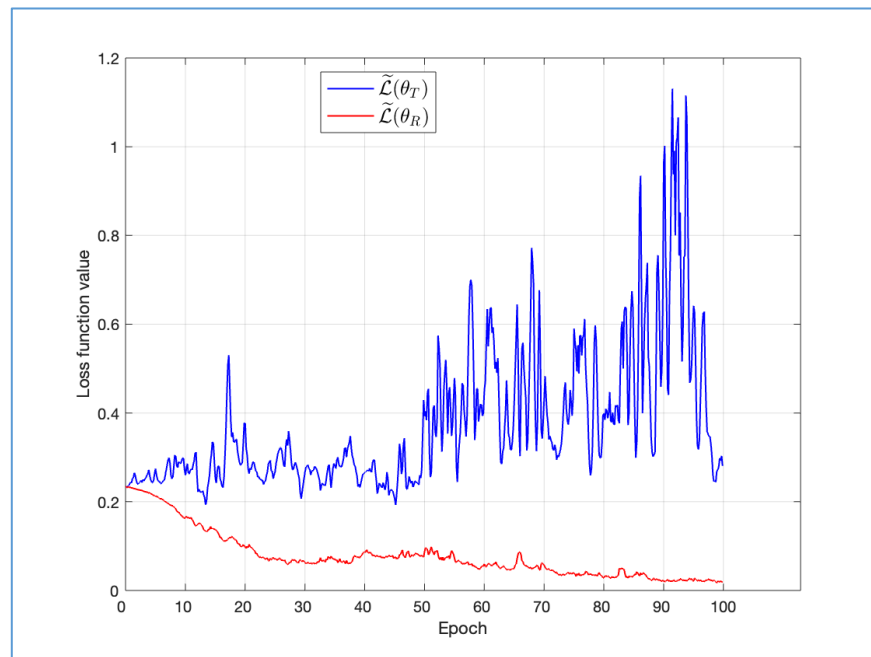


Fig. 6. The reconstructed loss function $\hat{\mathcal{L}}(\theta_R)$, $\hat{\mathcal{L}}(\theta_T)$ for RA-GAN schemes.

From the results, it can be observed that the proposed RA-GAN-based system performs remarkably better than the conventional GAN. This is proven as well from the sensitivity analysis that is conducted. Future works can consider different datasets to assess how consistent the RA-GAN scheme is. Moreover, it is suggested to investigate different deep learning neural networks such as the deep ensemble learning [31], and compare their performance to the basic GAN and RA-GAN in this paper. Besides, broader and more comprehensive view on the application of DL in the wireless networks from both the software and hardware perspectives [32] can be an inspiration for the future research interests.

4. Conclusion

Firstly, the definition of E2E learning system is introduced together with its challenges, one of which is that the transmitter of the system can only be trained with known channel. Based on that, we improve the GAN scheme to a so-called RA-GA scheme. The two primary problems associated with the conventional GAN being the gradient vanishing and over-fitting were solved as RA-GAN delivers more robust, powerful gradients (by skipping connection between the input and output of the generator) and helps to control the representation ability (by added the Regularizer to the loss function). In terms of the BLER, the simulation result shows that the RA-GAN performs comparatively well as the optimal method, which outperforms the conventional GAN.

Declarations

Author contribution. All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding statement. None of the authors have received any funding or grants from any institution or funding body for the research.

Conflict of interest. The authors declare no conflict of interest.

Additional information. No additional information is available for this paper.

References

- [1] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO Detection," *IEEE Work. Signal Process. Adv. Wirel. Commun. SPAWC*, vol. 2017-July, pp. 1–5, Jun. 2017, doi: [10.1109/SPAWC.2017.8227772](https://doi.org/10.1109/SPAWC.2017.8227772).
- [2] H. He, C. K. Wen, S. Jin, and G. Y. Li, "Deep Learning-Based Channel Estimation for BeamSpace mmWave Massive MIMO Systems," *IEEE Wirel. Commun. Lett.*, vol. 7, no. 5, pp. 852–855, Oct. 2018, doi: [10.1109/LWC.2018.2832128](https://doi.org/10.1109/LWC.2018.2832128).
- [3] E. Nachmani, Y. Be'Ery, and D. Burshtein, "Learning to Decode Linear Codes Using Deep Learning," *54th Annu. Allert. Conf. Commun. Control. Comput. Allert. 2016*, pp. 341–346, Jul. 2016, doi: [10.48550/arxiv.1607.04793](https://doi.org/10.48550/arxiv.1607.04793).
- [4] H. Ye, G. Y. Li, and B. H. Juang, "Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems," *IEEE Wirel. Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018, doi: [10.1109/LWC.2017.2757490](https://doi.org/10.1109/LWC.2017.2757490).
- [5] N. Farsad, M. Rao, and A. Goldsmith, "Deep Learning for Joint Source-Channel Coding of Text," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2018-April, pp. 2326–2330, Feb. 2018, doi: [10.48550/arxiv.1802.06832](https://doi.org/10.48550/arxiv.1802.06832).
- [6] Z. Qin, H. Ye, G. Y. Li, and B. H. F. Juang, "Deep learning in physical layer communications," *IEEE Wirel. Commun.*, vol. 26, no. 2, pp. 93–99, Apr. 2019, doi: [10.1109/MWC.2019.1800601](https://doi.org/10.1109/MWC.2019.1800601).
- [7] A. Felix, S. Cammerer, S. Dorner, J. Hoydis, and S. Ten Brink, "OFDM-Autoencoder for End-to-End Learning of Communications Systems," *IEEE Work. Signal Process. Adv. Wirel. Commun. SPAWC*, vol. 2018-June, Mar. 2018, doi: [10.48550/arxiv.1803.05815](https://doi.org/10.48550/arxiv.1803.05815).
- [8] T. Wang, C. K. Wen, H. Wang, F. Gao, T. Jiang, and S. Jin, "Deep Learning for Wireless Physical Layer: Opportunities and Challenges," *China Commun.*, vol. 14, no. 11, pp. 92–111, Oct. 2017, doi: [10.48550/arxiv.1710.05312](https://doi.org/10.48550/arxiv.1710.05312).
- [9] F. A. Aoudia and J. Hoydis, "End-to-End Learning of Communications Systems Without a Channel Model," *Conf. Rec. - Asilomar Conf. Signals, Syst. Comput.*, vol. 2018-October, pp. 298–303, Feb. 2019, doi: [10.1109/ACSSC.2018.8645416](https://doi.org/10.1109/ACSSC.2018.8645416).
- [10] W. Yu, T. Wang, and S. Wang, "Multi-Label Learning Based Antenna Selection in Massive MIMO Systems," *IEEE Trans. Veh. Technol.*, vol. 70, no. 7, pp. 7255–7260, Jul. 2021, doi: [10.1109/TVT.2021.3098888](https://doi.org/10.1109/TVT.2021.3098888).

[10.1109/TVT.2021.3087132](https://doi.org/10.1109/TVT.2021.3087132).

- [11] “Machine learning for 5G and beyond: From model-based to data-driven mobile wireless networks | IEEE Journals & Magazine | IEEE Xplore.” (accessed Jan. 02, 2023), doi : [10.12676/j.cc.2019.01.015](https://doi.org/10.12676/j.cc.2019.01.015).
- [12] M. Singh, “Integrating Artificial Intelligence and 5G in the Era of Next-Generation Computing,” *Proc. - 2021 2nd Int. Conf. Comput. Methods Sci. Technol. ICCMST 2021*, pp. 24–29, 2021, doi: [10.1109/ICCMST54943.2021.00017](https://doi.org/10.1109/ICCMST54943.2021.00017).
- [13] S. Han, I. Chih-Lin, G. Li, S. Wang, and Q. Sun, “Big Data Enabled Mobile Network Design for 5G and beyond,” *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 150–157, Jul. 2017, doi: [10.1109/MCOM.2017.1600911](https://doi.org/10.1109/MCOM.2017.1600911).
- [14] S. Zhang, Y. Zhang, J. Chang, B. Wang, and W. Bai, “DNN-based Signal Detection for Underwater OTFS Systems,” *2022 IEEE/CIC Int. Conf. Commun. China, ICCCWk. 2022*, pp. 348–352, 2022, doi: [10.1109/ICCCWORKSHOPS55477.2022.9896695](https://doi.org/10.1109/ICCCWORKSHOPS55477.2022.9896695).
- [15] O. Agiv and N. Shlezinger, “Learn to Rapidly Optimize Hybrid Precoding,” *IEEE Work. Signal Process. Adv. Wirel. Commun. SPAWC*, vol. 2022–July, 2022, doi: [10.1109/SPAWC51304.2022.9833923](https://doi.org/10.1109/SPAWC51304.2022.9833923).
- [16] T. Diskin, U. Okun, and A. Wiesel, “Learning to Detect with Constant False Alarm Rate,” Jun. 2022, doi: [10.48550/arxiv.2206.05747](https://doi.org/10.48550/arxiv.2206.05747).
- [17] 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, doi: [10.1109/TCCN.2017.2758370](https://doi.org/10.1109/TCCN.2017.2758370).
- [18] H. Ye, G. Y. Li, B. H. F. Juang, and K. Sivanesan, “Channel Agnostic End-to-End Learning based Communication Systems with Conditional GAN,” *2018 IEEE Globecom Work. GC Wkshps 2018 - Proc.*, Jul. 2018, doi: [10.48550/arxiv.1807.00447](https://doi.org/10.48550/arxiv.1807.00447).
- [19] M. Goutay, F. A. Aoudia, and J. Hoydis, “Deep Reinforcement Learning Autoencoder with Noisy Feedback,” *Proc. - 17th Int. Symp. Model. Optim. Mobile, Ad Hoc, Wirel. Networks, WiOpt 2019*, Oct. 2018, doi: [10.48550/arxiv.1810.05419](https://doi.org/10.48550/arxiv.1810.05419).
- [20] H. Ye, L. Liang, G. Y. Li, and B. H. Juang, “Deep Learning-Based End-to-End Wireless Communication Systems with Conditional GANs as Unknown Channels,” *IEEE Trans. Wirel. Commun.*, vol. 19, no. 5, pp. 3133–3143, May 2020, doi: [10.1109/TWC.2020.2970707](https://doi.org/10.1109/TWC.2020.2970707).
- [21] T. J. O’Shea, T. Roy, N. West, and B. C. Hilburn, “Physical Layer Communications System Design Over-the-Air Using Adversarial Networks,” *Eur. Signal Process. Conf.*, vol. 2018–September, pp. 529–532, Mar. 2018, doi: [10.48550/arxiv.1803.03145](https://doi.org/10.48550/arxiv.1803.03145).
- [22] T. J. Orshea, T. Roy, and N. West, “Approximating the Void: Learning Stochastic Channel Models from Observation with Variational Generative Adversarial Networks,” *2019 Int. Conf. Comput. Netw. Commun. ICNC 2019*, pp. 681–686, May 2018, doi: [10.48550/arxiv.1805.06350](https://doi.org/10.48550/arxiv.1805.06350).
- [23] S. Dorner, M. Henninger, S. Cammerer, and S. Ten Brink, “WGAN-based Autoencoder Training Over-the-air,” *IEEE Work. Signal Process. Adv. Wirel. Commun. SPAWC*, vol. 2020–May, Mar. 2020, doi: [10.48550/arxiv.2003.02744](https://doi.org/10.48550/arxiv.2003.02744).
- [24] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein Generative Adversarial Networks.” PMLR, pp. 214–223, Jul. 17, 2017. Accessed: Jan. 02, 2023. [Online]. Available at : <https://proceedings.mlr.press/v70/arjovsky17a.html>
- [25] E. Zehavi, “8-PSK Trellis Codes for a Rayleigh Channel,” *IEEE Trans. Commun.*, vol. 40, no. 5, pp. 873–884, 1992, doi: [10.1109/26.141453](https://doi.org/10.1109/26.141453).

- [26] S. Dorner, S. Cammerer, J. Hoydis, and S. Ten Brink, "Deep Learning Based Communication over the Air," *IEEE J. Sel. Top. Signal Process.*, vol. 12, no. 1, pp. 132–143, Feb. 2018, doi: [10.1109/JSTSP.2017.2784180](https://doi.org/10.1109/JSTSP.2017.2784180).
- [27] F. Liang, C. Shen, and F. Wu, "An Iterative BP-CNN Architecture for Channel Decoding," *IEEE J. Sel. Top. Signal Process.*, vol. 12, no. 1, pp. 144–159, Feb. 2018, doi: [10.1109/JSTSP.2018.2794062](https://doi.org/10.1109/JSTSP.2018.2794062).
- [28] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, Dec. 2014, doi: [10.48550/arxiv.1412.6980](https://doi.org/10.48550/arxiv.1412.6980).
- [29] R. Sattiraju, A. Weinand, and H. D. Schotten, "Performance Analysis of Deep Learning based on Recurrent Neural Networks for Channel Coding," *Int. Symp. Adv. Networks Telecommun. Syst. ANTS*, vol. 2018-December, Jul. 2018, doi: [10.1109/ANTS.2018.8710159](https://doi.org/10.1109/ANTS.2018.8710159).
- [30] D. Wu, M. Nekovee, and Y. Wang, "An Adaptive Deep Learning Algorithm Based Autoencoder for Interference Channels," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12081 LNCS, pp. 342–354, 2020, doi: [10.1007/978-3-030-45778-5_23](https://doi.org/10.1007/978-3-030-45778-5_23).
- [31] A. Al-Baidhani and H. H. Fan, "Deep ensemble learning: A communications receiver over wireless fading channels," *Glob. 2019 - 7th IEEE Glob. Conf. Signal Inf. Process. Proc.*, Nov. 2019, doi: [10.1109/GLOBALSIP45357.2019.8969302](https://doi.org/10.1109/GLOBALSIP45357.2019.8969302).
- [32] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 4, pp. 2595–2621, Oct. 2018, doi: [10.1109/COMST.2018.2846401](https://doi.org/10.1109/COMST.2018.2846401).