GAN-BASED JOINT ACTIVITY DETECTION AND CHANNEL ESTIMATION FOR GRANT-FREE RANDOM ACCESS

Shuang Liang[*†‡], *Yinan Zou*[*†‡], *and Yong Zhou*[*]

[*]School of Information Science and Technology, ShanghaiTech University, Shanghai201210, China
[†]Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, China
[‡]University of Chinese Academy of Sciences, Beijing 100049, China
E-mail:{liangshuang, zouyn, zhouyong}@shanghaitech.edu.cn

ABSTRACT

Joint activity detection and channel estimation (JADCE) for grant-free random access is a critical issue that needs to be addressed to support massive connectivity in IoT networks. However, the existing model-free learning method can only achieve either activity detection or channel estimation, but not both. In this paper, we propose a novel model-free learning method based on generative adversarial network (GAN) to tackle the JADCE problem. We adopt the U-net architecture to build the generator rather than the standard GAN architecture, where a pre-estimated value that contains the activity information is adopted as input to the generator. By leveraging the properties of the pseudoinverse, the generator is refined by using an affine projection and a skip connection to ensure the output of the generator is consistent with the measurement. Moreover, we build a two-layer fully-connected neural network to design pilot matrix for reducing the impact of receiver noise. Simulation results show that the proposed method outperforms the existing methods in high SNR regimes, as both data consistency projection and pilot matrix optimization improve the learning ability.

Index Terms— Massive connectivity, joint activity detection and channel estimation, deep generative adversarial network.

1. INTRODUCTION

Massive machine-type communications (mMTC), as an indispensable use case of the fifth generation (5G) wireless networks, is envisioned to provide massive connectivity for a large number of Internet of Things (IoT) devices [1, 2]. However, the conventional grant-based random access scheme [3] incurs excessive signaling overhead, and there are not enough orthogonal signature sequences to be allocated to all IoT devices. To address these issues, the 3rd generation partnership project (3GPP) for 5G new radio (NR) proposes the grant-free random access [4], where each device can directly send a unique non-orthogonal pilot sequence together with its data, without waiting for the permission from the base station (BS). Joint activity detection and channel estimation (JADCE) [5] is recognized as a critical issue for realizing grant-free random access [6].

Various methods have been proposed to tackle the JADCE problem and can generally be divided into two categories: optimization-based methods and learning-based methods. Despite the fact that the optimization-based methods, such as the block coordinate descent algorithm for Group Lasso [7] and iterative shrinkage thresholding algorithm (ISTA) [8], can solve the JADCE problem effectively, they require a large number of iterations. In order to accelerate the convergence and improve the estimation performance, learning-based methods have recently been proposed to tackle the JADCE problem. Learning-based methods can be further divided into model-based methods and model-free methods. For the model-based methods, the authors in [9] proposed a learned ISTA (LISTA). By integrating the unrolled deep neural network and ISTA, LISTA achieves a linear convergence rate. In addition, the authors in [10] proposed two model-based methods, which utilize data features to achieve better performance by combining the standard auto-encoder structure with an iterative optimization algorithm, such as Group Lasso and approximate message passing (AMP) method [4, 5].

Compared to the model-based methods, the model-free methods have stronger adaptability and learning ability. The existing studies all adopt the generative network, such as Auto-Encoder (AE), Variational AE (VAE) [11] and standard Generative Adversarial Network (GAN) [12]. In particular, the authors in [13] proposed a model-free method to jointly design the pilot matrix and detect active devices by using AE. The authors in [14] used GAN to achieve high dimensional wireless channel estimation by optimizing the input to a deep generative network. However, these model-free methods can only achieve either activity detection or channel estimation, but not both. In addition, the generative network in [14] learns the mapping from the latent variable with a specific distribution to the wireless channel distribution. As the gen-

erative network is fixed and the latent variable is optimized according to the measurement error. This generative network is unable to learn the feature of active devices. In addition, the optimization about the latent variable cannot ensure that its distribution does not change, which may conflict with the mapping of the generator and make the latent variable falls into local optimum. By combining the advantages of [13] and [14], a pre-estimated value containing active device information can be adopted as input to the generative network to tackle the JADCE problem, which motivates this work.

In this paper, we propose a novel model-free method to address the JADCE problem for grant-free random access in IoT networks. Instead of adopting the standard generative network architecture, we adopt the U-net [15–17] architecture to build generator which can be trained in an end-to-end manner based on a pre-estimated value. According to the properties of pseudoinverse, we project the output of the U-net to the nullspace of the pilot matrix to ensure the data consistency. Moreover, a two-layer fully-connected neural network was built to design the pilot matrix to reduce the impact of receiver noise. Simulation results show that the proposed method solves the problem of latent variable distribution variation and achieves a better performance than the existing methods for the JADCE problem in high signal-to-noise ratio (SNR) regimes.

2. SYSTEM MODEL

Consider the grant-free uplink transmission of a single-cell IoT network that consists of N single-antenna IoT devices and one M-antenna BS. We denote $\mathcal{N} = \{1, \ldots, N\}$ as the index set of IoT devices and assume that the transmission activity of each IoT device is independent in each transmission block. We denote a_n as the indicator of the activity of device n, where $a_n = 1$ if device n is active and $a_n = 0$ otherwise. We denote $\boldsymbol{y}(l) \in \mathbb{C}^M$ as the l-th symbol received by the BS, $\boldsymbol{h}_n \in \mathbb{C}^M$ as the channel coefficient vector of the link between IoT device n and the BS, $s_n(l) \in \mathbb{C}$ as the l-th signature symbol transmitted by device n, and $\boldsymbol{z}(l) \in \mathbb{C}^M$ as the additive white Gaussian noise (AWGN) vector at the BS. By assuming that the transmissions of active devices are synchronized, $\boldsymbol{y}(l) \in \mathbb{C}^M$ can be written as $\boldsymbol{y}(l) = \sum_{n=1}^{N} a_n \boldsymbol{h}_n s_n(l) + \boldsymbol{z}(l), l = 1, \ldots, L$, where L denotes the length of the signature sequence.

For notational ease, we denote the aggregated received pilot signal matrix $\boldsymbol{Y} \in \mathbb{C}^{L \times M}$ with $\boldsymbol{Y}[l,:] = \boldsymbol{y}(l)$, the channel matrix $\boldsymbol{H} \in \mathbb{C}^{N \times M}$ with $\boldsymbol{H}[n,:] = \boldsymbol{h}_n$, the additive noise matrix $\boldsymbol{Z} \in \mathbb{C}^{L \times M}$ with $\boldsymbol{Z}[l,:] = \boldsymbol{z}(l)$, and the pilot matrix $\boldsymbol{S} \in \mathbb{C}^{L \times N}$ with $\boldsymbol{S}[l,:] = [s_1(l), \ldots, s_N(l)]$. We have

$$\boldsymbol{Y} = \boldsymbol{S}\boldsymbol{X} + \boldsymbol{Z}, \tag{1}$$

where matrix $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{H} \in C^{N \times M}$ with $\boldsymbol{A} = \text{diag}(a_1, \ldots, a_N)$ being the diagonal activity matrix. By considering the sporadic traffic, matrix $\boldsymbol{X}$ endows with a group sparse structure

which means that all columns of matrix $\boldsymbol{X}$ have the same sparse structure. The goal of the JADCE problem is to detect the device activity matrix $\boldsymbol{A}$ and estimate the channel matrix $\boldsymbol{H}$ by recovering $\boldsymbol{X}$ from the noisy observation $\boldsymbol{Y}$.

As the standard neural networks cannot process complex-valued data, we reformulate (1) as

$$\begin{aligned}
\tilde{\boldsymbol{Y}} &= \tilde{\boldsymbol{S}}\tilde{\boldsymbol{X}} + \tilde{\boldsymbol{Z}} \\
&= \begin{bmatrix} \mathcal{R}\{\boldsymbol{S}\} & -\mathcal{I}\{\boldsymbol{S}\} \\ \mathcal{I}\{\boldsymbol{S}\} & \mathcal{R}\{\boldsymbol{S}\} \end{bmatrix} \begin{bmatrix} \mathcal{R}\{\boldsymbol{X}\} \\ \mathcal{I}\{\boldsymbol{X}\} \end{bmatrix} + \begin{bmatrix} \mathcal{R}\{\boldsymbol{Z}\} \\ \mathcal{I}\{\boldsymbol{Z}\} \end{bmatrix}
\end{aligned} \tag{2}$$

where $\mathcal{R}\{\cdot\}$ and $\mathcal{I}\{\cdot\}$ denote the real part and imaginary part of the input matrix, respectively.

3. ALGORITHM DESCRIPTION

3.1. Overview of the Proposed Method

Fig. 1 shows an overview of the proposed GAN-based method for the JADCE problem. For clarity, we represent a U-net with data consistency projection as a fundamental block. Our generator G is made up of many fundamental blocks. With $\tilde{\boldsymbol{Y}} = \tilde{\boldsymbol{S}}\tilde{\boldsymbol{X}} + \tilde{\boldsymbol{Z}}$, we use $\tilde{\boldsymbol{X}}_0 = \tilde{\boldsymbol{S}}^+\tilde{\boldsymbol{Y}}$ as an estimation of $\tilde{\boldsymbol{X}}$ to be the input of the generative network, where $(\cdot)^+$ is the pseudoinverse of a matrix. Inspired by the residual learning and the properties of the Moore–Penrose pseudoinverse, we build a shortcut connection from input to output in each fundamental block, which guarantees the data consistency. Our discriminator D attempts to differentiate the real group sparse instances from the fake instances generated by G. The whole system trains G and D adversarially until the loss of each neural network converges.

Training the network in Fig. 1 is equivalent to playing a game with conflicting objectives between the generator G and the discriminator D. The generator G aims to map the estimation $\tilde{\boldsymbol{S}}^+\tilde{\boldsymbol{Y}}$ to the fake data that fools the discriminator D. The discriminator D aims to distinguish between real data and data generated by the generator G. Various strategies have been devised to reach the balance. Motivated by LSGAN [18] and WGAN [19], we adopt a mixture of WGAN and l_2 costs to train the generator. We then formulate the overall problem to jointly minimize the generator cost $\mathbb{E}[||\tilde{\boldsymbol{X}} - G(\tilde{\boldsymbol{S}}^+\tilde{\boldsymbol{Y}})||_2] - \mathbb{E}[D(G(\tilde{\boldsymbol{S}}^+\tilde{\boldsymbol{Y}}))]$ and the discriminator cost $-\mathbb{E}[D(\tilde{\boldsymbol{X}})] + \mathbb{E}[D(G(\tilde{\boldsymbol{S}}^+\tilde{\boldsymbol{Y}}))]$.

3.2. U-net

The standard generative models, such as AE and VAE, map the input data to a low-dimensional latent variable to extract the main features of the input data. They feed the latent variables into decoder to recover data. Hence, these methods have to find the corresponding latent variables to recover the input data. However, finding the corresponding latent variables is not necessary for the JADCE problem and the dimensional reduction inevitably leads to performance loss. On the other
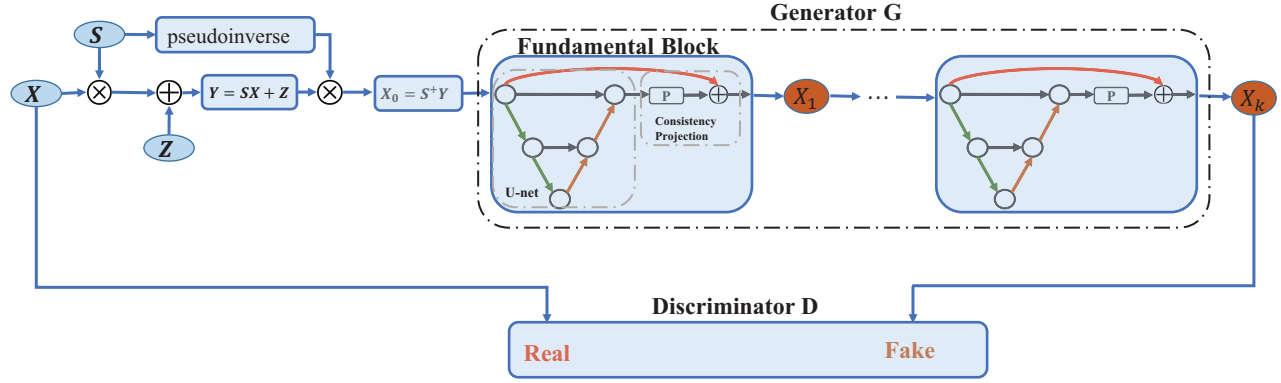
Fig. 1: Proposed model-driven approach.

hand, the standard GAN takes low-dimensional random latent variables that obey a specific distribution as input. A trained generator can generate many similar data to approximate the real data for different latent variables. The authors in [13] used $\tilde{Y}$ as input to AE to achieve a good performance on activity detection, which verifies that $\tilde{Y}$ contains the device activity information. The authors in [14] used standard GAN to achieve a good performance for channel estimation. Due to the independence of activity among devices, the method in [14] is unable to achieve activity detection.

By leveraging their advantages, we propose a novel GAN architecture which exploits the U-net for the generator. Under this structure, we use $\tilde{S}^+\tilde{Y}$ as the input to the generator, which contains the device activity information, to tackle the JADCE problem. By utilizing a shortcut connection of the U-net, it is not necessary to find the corresponding latent variables, which may introduce many additional iterations. In addition, the shortcut connections also enhance the feature reuse to reduce information loss.

3.3. Data Consistency

For a linear equations $\tilde{Y} = \tilde{S}\tilde{X}$, where $\tilde{S}$ is an underdetermined matrix, $\tilde{X}$ is variable and $\tilde{(\cdot)}$ is the real-valued form of matrix, all the solutions satisfy $\tilde{X} = \tilde{S}^+\tilde{Y} + (I - \tilde{S}^+\tilde{S})\tilde{W}$, where W is an arbitrary matrix. To ensure the data consistency, we project the output of U-net onto the nullspace of $\tilde{S}$ and add $\tilde{S}^+\tilde{Y}$ to arrive at $\tilde{X}_1 = \tilde{S}^+\tilde{Y} + (I - \tilde{S}^+\tilde{S})U_1(\tilde{S}^+\tilde{Y})$, where $U_k(\cdot)$ represents the output of the k-th U-net. Subsequently, we get $\tilde{X}_2 = \tilde{X}_1 + (I - \tilde{S}^+\tilde{S})U_2(\tilde{X}_1)$ which also satisfies data consistency. As a result, we can model it as a recurrent neural network (RNN). Moreover, we can express the unrolled RNN with k iterations as

$$\tilde{X}_{k+1} = \tilde{X}_k + (I - \tilde{S}^+\tilde{S})U_k(\tilde{X}_k), \qquad (3)$$

where $\tilde{X}_k$ is the k-th output of fundamental block and U_k is the k-th U-net of the fundamental block.

To account for the noise Z, we use $\tilde{X}_{k+1} = \tilde{X}_k + (I - \tilde{S}^+\tilde{S})U_k(\tilde{X}_k) - \tilde{S}^+\tilde{Z}$ to ensure the data consistency instead of (3). However, as the receiver noise is random, we cannot

completely eliminate the effect of noise. In order to reduce the impact of noise, the pilot matrix $\tilde{S}$ is designed to minimize $\mathbb{E}[||\tilde{S}^+\tilde{Z}||_F]$. Motivated by [13], we build a two-layer fully-connected neural network, which the input layer has $2N$ neurons and the output layer has $2L$ neurons. In addition, the weight of the connection from the n-th neuron in the input layer to the l-th neuron in the output layer represents the (l, n)-th element of the pilot matrix $\tilde{S}$. The $\mathbb{E}[||\tilde{S}^+\tilde{Z}||_F]$ is the loss function. Given training dataset $\{\tilde{Z}_i\}_{i=1}^n$ and a initial value of $\tilde{S}$, we use gradient descent algorithm minimize the loss function until the convergence.

3.4. Model Architecture

In this subsection, we introduce the details of our neural network architecture. Firstly, we introduce the network architecture of generator. The generator is made up of multiple fundamental blocks, which include two components: U-net block and projection block. By denoting the input of fundamental block as $\tilde{X}_0$, we can get the output of U-net block $U(\tilde{X}_0)$ and the output of projection block $\tilde{X}_1 = \tilde{X}_0 + (I - \tilde{S}^+\tilde{S})U(\tilde{X}_0)$. The input of U-net block is a 3D tensor. From the dimensional perspective, the dimensions of the input data first reduce and then increase during the feedforward of the input data. In the process of reducing dimension, we first use 1D convolution with a kernel size of 3, stride size of 2 and padding size of 1, so that it performs down-sampling without a separate max-pooling layer. Then, we use 1D convolution with a kernel size of 3 and stride size of 1 and ReLU activation function to extract features, and repeat this process twice. In the process of increasing dimension, we use 1D convtranspose with a kernel size of 2 and stride size of 2. Then we stack the tensor with the tensor which has the same size in the processing of reducing dimension. Next, we use 1D convolution with a kernel size of 3, stride size of 1 and ReLU activation function to extract features, and repeat this process twice.

Secondly, we design the network architecture of the discriminator as follows. We use 1D convolution with a kernel size of 3, stride size of 1 and ReLU activation function to extract features. Then we use 1D convolution with a kernel size

of 3 and stride size of 2, so that it performs down-sampling. After several dimensional reduction operations, we use 1D convolution with a kernel size of 3 to get final output.

4. SIMULATION RESULTS

4.1. Data Generation and Training Strategy

In this section, we present the simulation results of the proposed method, and compare the results with the classic optimization-based methods, ISTA [8] and Group Lasso [7], and the state-of-the-art model-based method LISTA [9]. In the simulations, the signature sequence is generated according to the complex Gaussian distribution, i.e, $S \sim \mathcal{CN}(0, 1)$, and the channels suffer from independent Rayleigh fading, i.e., $H \sim \mathcal{CN}(0, 1)$. We set the total number of devices N, the length of signature sequence L, and the number of antennas at the BS M to be 256, 128, and 8, respectively. Each entry of the activity sequence $\{a_1, ..., a_N\}$ follows the Bernoulli distribution with probability $p = 0.1$, i.e, $\mathcal{P}(a_n = 1) = p$ and $\mathcal{P}(a_n = 0) = 1 - p$. According to (2), we transform all the complex-valued matrix into real-valued matrix. And then, we obtain the data set $\{\tilde{X}_i, \tilde{Y}_i\}_{i=1}^n$, where n represents the size of the data set. We adopt normalized mean square error (NMSE) to evaluate the performance of those methods, defined as

$$NMSE = 10 \log_{10} \left(\frac{\mathbb{E}||\tilde{X} - \tilde{X}^*||_F^2}{\mathbb{E}||\tilde{X}^*||_F^2} \right), \quad (4)$$

where $\tilde{X}$ is the estimate solution and $\tilde{X}^*$ is the ground truth. In the the training stage, we set the batch size to 64 and the initial learning rate η_0 to $5*10^{-4}$. In the test stage, we generate 1000 samples to test the proposed model.

We train the our proposed model by adopting the block-wise training strategy. To stabilize the training process, we add two decayed learning rates η_1 and η_2, i.e., $\eta_1 = 0.2\eta_0$ and $\eta_2 = 0.02\eta_0$. We train the generator G block by block. The training process of each block is described as follows. First, we suppose that the previous blocks have been trained and train i-th block with learning rate η_0 until the convergence. And then, we train all blocks with learning rate η_1 and η_2 until the convergence. Finally, we add new block and repeat the process. The code is available at
https://github.com/deeeeeplearning/JADCE

4.2. Experiment

In Fig. 2(a), we compare the performance of our proposed method with the baseline methods over different SNR. When SNR is less than 30 dB, the performance of our proposed method is better than ISTA and Group Lasso, but not better than LISTA. However, the performance of the proposed method improves rapidly with the increase of SNR. When SNR is greater than 30 dB, the proposed method outperforms
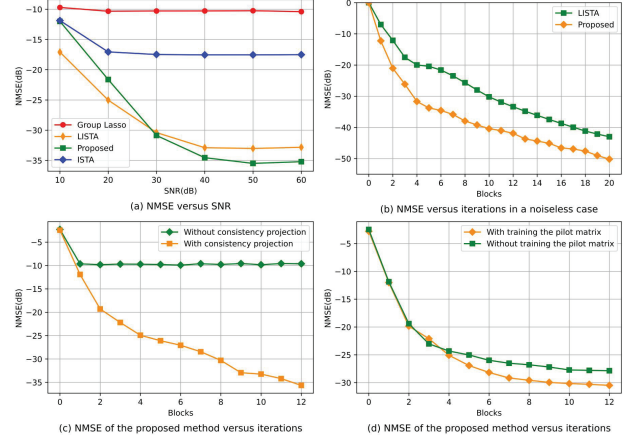
Fig. 2: Performance comparison in terms of NMSE when N=256, L=128, M=8, and p=0.1.

all baseline schemes, which means the performance upper bound of the proposed method is the best.

Considering that both the proposed method and LISTA use layer-wise or block-wise training strategy, we compare their NMSE over iterations in noiseless scenario. Fig. 2(b) shows that the proposed method outperforms the LISTA method when using the same number of blocks in this scenario.

We compare the performance of our proposed method with and without data consistency projection. Fig 2(c) indicates that data consistency brings a huge boost for performance. Without data consistency projection, the proposed method converges earlier and falls into a local solution. Under this circumstance, the generator is equivalent to AE and VAE with shortcut connection, which in turn implies that the performance of AE and VAE is not good for JADCE problem and data consistency projection makes the network learning ability stronger. Fig. 2(d) shows the performance difference of our method with and without training pilot matrix $\tilde{S}$. We can observe that the optimization about pilot matrix can alleviate the influence of noise to improve the performance about 3 dB.

5. CONCLUSION

In this paper, we proposed a novel model-free method to address the JADCE problem in IoT networks. We adopted U-net structure to build generator and used S^+Y as input to generator. Inspired by deep residual learning and the properties of the Moore–Penrose pseudoinverse, we used $X_1 = S^+Y + (I - S^+S)U_1(S^+Y)$ to ensure the data consistency. To reduce the impact of noise, we built a two-layer fully-connected neural network to design pilot matrix. Simulation results showed that the data consistency projection and designing pilot matrix can improve the performance of the proposed method. In high SNR regimes, the proposed method achieves a better performance for the JADCE problem.

6. REFERENCES

[1] S. Sharma and X. Wang, "Toward massive machine type communications in ultra-dense cellular iot networks: Current issues and machine learning-assisted solutions," *IEEE Commmun. Surv. & Tut.*, vol. 22, no. 1, pp. 426–471, Jan. 2019.

[2] Y. Shi, H. Choi, Y. Shi, and Y. Zhou, "Algorithm unrolling for massive access via deep neural networks with theoretical guarantee," *IEEE Trans. Wireless Commun.*, to appear.

[3] Y. Zou, Y. Zhou, Y. Shi, and X. Chen, "Learning proximal operator methods for massive connectivity in IoT networks," in *Proc. IEEE Globecom*, Madrid Madrid, Spain, Dec. 2021.

[4] L. Liu, E. Larsson, W. Yu, P. Popovski, C. Stefanovic, and E. de Carvalho, "Sparse signal processing for grant-free massive connectivity: A future paradigm for random access protocols in the internet of things," *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 88–99, 2018.

[5] Z. Chen, F. Sohrabi, and W. Yu, "Sparse activity detection for massive connectivity," *IEEE Trans. on Signal Process.*, vol. 66, no. 7, pp. 1890–1904, Jul. 2018.

[6] A. Mostafa, Y. Zhou, and V. W.S. Wong, "Connection density maximization of narrowband iot systems with noma," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4708–4722, Oct. 2019.

[7] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[8] Z. Qin, K. Scheinberg, and D. Goldfarb, "Efficient block-coordinate descent algorithms for the group lasso," *Math. Program. Comput.*, vol. 5, no. 2, pp. 143–169, 2013.

[9] Y. Shi, S. Xia, Y. Zhou, and Y. Shi, "Sparse signal processing for massive device connectivity via deep learning," in *2020 IEEE Int. Conf. on Commun. Workshops*, 2020, pp. 1–6.

[10] Y. Cui, S. Li, and W. Zhang, "Jointly sparse signal recovery and support recovery via deep learning with applications in mimo-based grant-free random access," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 3, pp. 788–803, 2021.

[11] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural inf. process. syst.*, vol. 27, 2014.

[13] S. Li, W. Zhang, Y. Cui, H. Cheng, and W. Yu, "Joint design of measurement matrix and sparse support recovery method via deep auto-encoder," *IEEE Signal Proces. Lett.*, vol. 26, no. 12, pp. 1778–1782, Dec. 2019.

[14] E. Balevi, A. Doshi, A. Jalal, A. Dimakis, and J. Andrews, "High dimensional channel estimation using deep generative networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 18–30, Jan. 2020.

[15] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Int. Conf. on Med. Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[16] T. Quan, T. Nguyen-Duc, and W. Jeong, "Compressed sensing MRI reconstruction using a generative adversarial network with a cyclic loss," *IEEE Trans. on Med. Imag.*, vol. 37, no. 6, pp. 1488–1497, Jun. 2018.

[17] M. Mardani, E. Gong, J. Cheng, S. Vasanawala, G. Zaharchuk, M. Alley, N. Thakur, S. Han, W. Dally, J. Pauly, et al., "Deep generative adversarial networks for compressed sensing automates mri," *arXiv preprint arXiv:1706.00051*, 2017.

[18] X. Mao, Q. Li, H. Xie, R. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proc. of IEEE Int. Conf. on Computer Vision*, 2017, pp. 2794–2802.

[19] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. of the 34th International Conference on Machine Learning*, Aug. 2017, vol. 70, pp. 214–223.