UniQGAN: Unified Generative Adversarial Networks for Augmented Modulation Classification

Insup Lee, *Student Member, IEEE*, and Wonjun Lee, *Fellow, IEEE*

***Abstract*— Deep learning has been widely applied to automatic modulation classification (AMC), and there have been many studies on data augmentation techniques using deep generative models to improve performance. However, existing solutions need to train different models independently for each SNR, which leads to undeniable overhead. This letter presents *UniQGAN*, Unified Generative Adversarial Networks for IQ constellations of various SNRs, requiring a single model training. The proposed method introduces *multi-conditions embedding* and *multi-domains classification* to leverage both conditions, i.e., modulation type and SNR. Experimental results show that UniQGAN effectively improves the AMC performance, while the training time is reduced.**

***Index Terms*— Automatic modulation classification, generative adversarial networks, single model training, IQ constellations.**

I. Introduction

AUTOMATIC modulation classification (AMC) serves a vital role in communication systems, e.g., cognitive radio and non-cooperative systems, which triggers lots of related studies such as security [1] and deep learning applications [2]–[6]. Especially, many deep learning methods are recently investigated for the AMC to automatically extract hidden features from received signals. Specifically, the convolutional neural network (CNN) has attracted considerable attention since it shows minimal errors even with noticeable speed improvements. Even though the CNN-based AMC methods [2], [3] need enough labeled training data to achieve high performance, preparing the data is a time-consuming and costly task. Without sufficient data, a trained model can experience *overfitting* and significant performance degradation. Especially in the noisy environment of low SNRs, it is a more critical issue since a performance at low SNRs is inferior to that of higher SNRs.

To handle the labeled data insufficiency of AMC, there are two representative fields: leveraging unlabeled data with semi-supervised learning and generating the labeled data. Since the latter generally show the better performance than the former, we focus on generating high-quality labeled data using deep generative models. The models such as generative adversarial networks (GAN) [7] and variational autoencoder (VAE) have an advantage in that they can approximate the original sample's probability distribution to *generate* realistic data, rather than *reproduce* it. For instance, conditional GAN (cGAN) has been employed for augmenting IQ signals [4], auxiliary classifier GAN (ACGAN) for constellation diagrams [5], while conditional VAE for constellation images is introduced in [6]. Nevertheless, these existing methods only consider generating data for specific SNR. Note that abundant data from various SNRs should be obtained because the model-based AMC only works on the SNRs from which training data is sampled. Conventional generation approaches have limited scalability in that they need to train generative models for every desired SNR, followed by a long training time.

In this context, we propose UniQGAN to design a unified generative architecture for IQ data of various SNRs. UniQGAN is based on ACGAN [8] since although GAN-related models accompany challenging training processes, the models generally achieve higher quality than VAE variants. The primary intuition behind our model is to leverage SNR in addition to modulation type as generation instructions, with the proposed *multi-conditions embedding*. In the optimization process, both SNR and modulation type are reflected in the objective function by *multi-domains classification*. Experimental results show that UniQGAN successfully enhances CNN-based AMC while it reduces GAN training time dramatically.

II. Background

We consider single-input single-output communication system, and the received signal $r(t)$ can be expressed as:

$$r(t) = ae^{j(2\pi f_c t+\phi)}s(t) + n(t), \tag{1}$$

where $s(t)$ is the transmitted signal after modulation, a is the amplitude, f_c is the carrier frequency offset, ϕ is the phase offset, and $n(t)$ is the additive white Gaussian noise, respectively. Based on the signal model, real part In-phase (I) and imaginary part Quadrature (Q) are extracted from the historical data $r(t)$ of specific SNR, and converted into a constellation diagram on a two-dimensional IQ plane. Then the diagram x is obtained with two class labels c_m (modulation type) and c_s (SNR). Although accurate information within the $r(t)$ cannot be restored from x, it is inconsequential since our focus is only to decide c_m ahead of demodulation.

As shown in Fig.1, the problem scope consists of two parts: deep learning-based AMC and data augmentation to complement the classifier. The c_m is used in the both problems, while the c_s is leveraged only in the augmentation problem. As depicted in Fig.1a, AMC is a kind of typical classification problem in which given x, a classifier recognizes c_m.

Manuscript received October 3, 2021; accepted November 16, 2021. Date of publication December 1, 2021; date of current version February 12, 2022. This work was supported by the National Research Foundation (NRF) of Korea grant funded by the Korea Government (MSIT) under Grant 2019R1A2C2088812. The associate editor coordinating the review of this letter and approving it for publication was M. Oner. *(Corresponding author: Wonjun Lee.)*

The authors are with the Network and Security Research Laboratory, School of Cybersecurity, Korea University, Seoul 02841, Republic of Korea (e-mail: wlee@korea.ac.kr).

Digital Object Identifier 10.1109/LCOMM.2021.3131476

Assuming that the training data is limited, Fig.1b shows data enlargement to train the classifier with sufficient data. In traditional approaches, each generator should be trained on x and c_m of each c_s. Then the trained generator can generate fake data of corresponding c_s, while the proposed method can address the problem with only a single trained model, which is detailed in the next section.

III. Design of UniQGAN

This section describes the core of UniQGAN including preliminaries, proposed structure, and optimization process.

A. Preliminaries

As a representative deep learning-based generative model, generative adversarial networks (GAN) [7] is composed of two competitive networks, generator G and discriminator D. G generates fake data $G(z)$ where z is a random noise from Gaussian distribution, while D outputs a single scalar 0 to 1, which means a validity of given data. G is trained to deceive the D, while D learns to differentiate successfully. This competitive training is modeled as a minimax game about one loss function between two networks, i.e., G tries to minimize the loss while D trains to maximize. Adversarial loss of GAN is expressed as follows:

$$\mathcal{L}_{adv} = \mathbb{E}_x[\log D_{src}(x)] + \mathbb{E}_z[\log(1 - D_{src}(G(z)))], \quad (2)$$

where D_{src} predicts the probability that given input is from real data x, i.e., probability distribution over sources. After G and D reach the converged point of the competitive optimization, D is unable to distinguish the real data x and fake data $G(z)$, theoretically. However, original GAN has limits due to the low quality and impossibility of conditional generation, i.e., synthesizing data corresponding to the intended category.

Auxiliary classifier GAN (ACGAN) [8] is one of the improved GAN variants and addresses the aforementioned problems effectively. In the ACGAN, G leverages class label c to generate fake data $G(c, z)$ and every generated data has a corresponding label c. In particular, D has two parts; one is D_{src} same as in Eq.(2), and the other is D_{aux} called auxiliary classifier, which outputs domain classification probability to improve generation quality. Both losses of ACGAN are expressed as follows:

$$\mathcal{L}_{adv} = \mathbb{E}_x[\log D_{src}(x)] + \mathbb{E}_{z,c}[\log(1 - D_{src}(G(z, c)))], \quad (3)$$

$$\mathcal{L}_{aux} = -\mathbb{E}_{x,c}[\log D_{aux}(c|x)] - \mathbb{E}_{z,c}[\log D_{aux}(c|G(z, c))]. \quad (4)$$

The objective function (to minimize) of D is $-\mathcal{L}_{adv} + \mathcal{L}_{aux}$, and G is $\mathcal{L}_{adv} + \mathcal{L}_{aux}$.

B. UniQGAN Architecture

As shown in Fig.2, the architecture of UniQGAN deviates from the ACGAN in that both generator G and discriminator D consider SNR as well as modulation type. Embedding techniques for multiple inputs used so far are not appropriate in this problem since the methods are excessively complex for

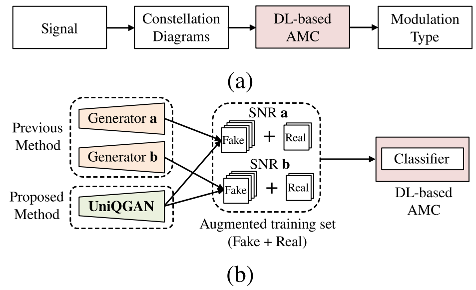

Fig. 1. Problem scope. (a) Deep learning-based AMC. (b) Data augmentation to improve classification performance.

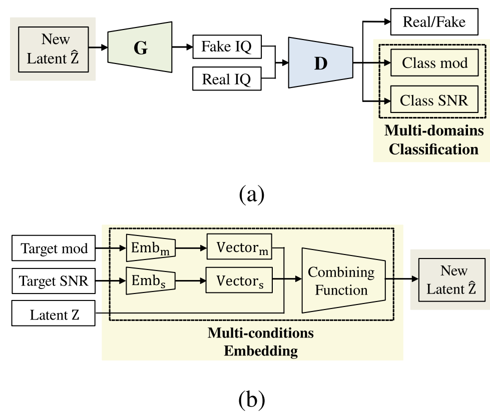

Fig. 2. UniQGAN Architecture. (a) Overview of UniQGAN components. (b) Details of multi-conditions embedding.

IQ constellations generation. We propose a simple but effective embedding method called *multi-conditions embedding* for G and a *multi-domains classification* for D. Through the proposed methods, IQ data of various SNRs can be generated with a single trained UniQGAN model. In our method, we use some techniques for stable training with low complexity.

1) Multi-Conditions Embedding: As shown in Fig.2b, both conditions c_s (SNR) and c_m (modulation type) are embedded separately by each *embedding* layer. The layer maps inputs to *dense* vectors and learns optimized embedding weights via the training process. Note that some conditions show higher correlations in the AMC problem, e.g., similar SNRs. To represent relationships between embedded vectors, we choose neural embedding instead of widely used one-hot encoding which adopts *sparse* vectors. The embedded vectors are concatenated and then multiplied with random noise z to form a new latent vector $\hat{z}$, which will be fed into the G. In our experiments, multiplication (not concatenation) with random noise relieved the *mode collapse*, i.e., the representative GAN failure in which G produces limited varieties of samples. The overall derivation process of $\hat{z}$ is defined as follows:

$$\hat{z} = z \cdot (\mathrm{Emb_s}(c_s) \parallel \mathrm{Emb_m}(c_m)), \quad (5)$$

where $\mathrm{Emb_s}$ and $\mathrm{Emb_m}$ are embedding layers for each case.

2) Multi-Domains Classification: the simple but effective idea of D is expanding the ability of the auxiliary classifier D_{aux} in Eq.(4) to modulation type and SNR. That is, D_{aux} is divided into D^s_{aux} and D^m_{aux}, where the D^s_{aux} gives a probability distribution over the c_s and the D^m_{aux} over c_m. It enables D to deal with three problems,

TABLE I

PERFORMANCE OF DIFFERENT AUGMENTATION METHODS ON RADIOML2018.01A BENCHMARK

Augmentation method	SNR (dB) -2	0	2	4	6	8	10	Low SNRs (-2~4) Average	All SNRs (-2~10) Average
Original (No augmentation)	0.234	0.325	0.555	0.841	0.956	0.976	0.985	0.489	0.696
CGAN	0.252	0.328	0.548	0.820	0.937	0.951	0.953	0.487	0.684
ACGAN	0.256	0.341	0.527	0.839	0.945	0.968	0.976	0.491	0.693
UniQGAN (One-hot encoding)	**0.272**	0.341	0.574	0.854	0.959	0.979	0.984	0.510	0.709
UniQGAN (Multi-conditions embedding)	0.267	**0.356**	**0.599**	**0.861**	**0.962**	**0.982**	**0.987**	**0.521**	**0.716**

$D : x \rightarrow \{D_{src}(x), D^s_{aux}(x), D^m_{aux}(x)\}$. The three components are employed to construct losses detailed in the next subsection.

3) Model Improvement: we adopt several techniques to stabilize the training procedure since GAN architecture is susceptible to slight parameter tuning. First, to prevent overfitting, we use a dropout rate of 0.25 for each layer of D. Second, we adopt label smoothing, which replaces correct labels for real data x with 0.9 and fake data $G(\hat{z})$ with 0.1, from 1.0 and 0.0 respectively. Lastly, spectral normalization [9] is employed to constrain the Lipschitz constant of D to enhance convergence speed. All these methods contribute to UniQGAN's stability and faster loss convergence.

C. Training Algorithm

The optimization process of the proposed method builds upon previously defined terms, including $\hat{z}$, D_{src}, D^m_{aux}, and D^s_{aux}. After parameters initialization, we prepare training data, i.e., constellation diagram x and corresponding labels such as c_m and c_s. To make generated data realistic, adversarial loss is expressed as follows:

$$\mathcal{L}_{adv} = \mathbb{E}_x[\log D_{src}(x)] + \mathbb{E}_{z,c_m,c_s}[\log(1 - D_{src}(G(\hat{z})))]. \tag{6}$$

Since we design the D to have two auxiliary classifiers D^m_{aux} and D^s_{aux}, two auxiliary losses are defined as follows:

$$\begin{aligned}\mathcal{L}^m_{aux} &= -\mathbb{E}_{x,c_m}[\log D^m_{aux}(c_m|x)] \\ &\quad -\mathbb{E}_{z,c_m,c_s}[\log D^m_{aux}(c_m|G(\hat{z})), \qquad (7)\\ \mathcal{L}^s_{aux} &= -\mathbb{E}_{x,c_s}[\log D^s_{aux}(c_s|x)] \\ &\quad -\mathbb{E}_{z,c_m,c_s}[\log D^s_{aux}(c_s|G(\hat{z})). \qquad (8)\end{aligned}$$

Consequently, our full objective functions derived from equations above (Eq.(6), Eq.(7), and Eq.(8)) to optimize D and G are expressed as follows:

$$\mathcal{L}_D = -\mathcal{L}_{adv} + \lambda_m \mathcal{L}^m_{aux} + \lambda_s \mathcal{L}^s_{aux}, \tag{9}$$

$$\mathcal{L}_G = \mathcal{L}_{adv} + \lambda_m \mathcal{L}^m_{aux} + \lambda_s \mathcal{L}^s_{aux}, \tag{10}$$

where λ_m and λ_s are weight scalars, respectively. Although optimized λ_m and λ_s may vary under experimental conditions, we set $\lambda_m = 0.7$ and $\lambda_s = 0.3$ according to the experimental results, which are detailed in the next section. Both D and G optimize their parameters using stochastic gradient descent to minimize $\mathcal{L}_D$ and $\mathcal{L}_G$, respectively.

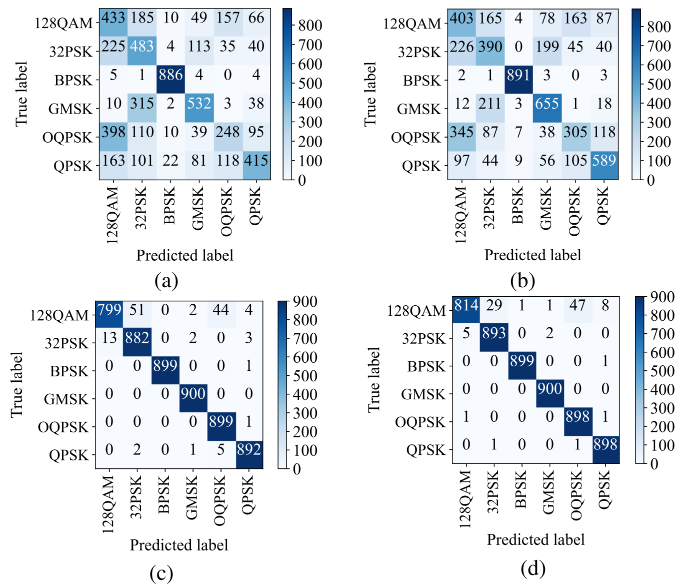

Fig. 3. Confusion matrix of (a) Original at 2 dB, (b) UniQGAN at 2 dB, (c) Original at 8 dB, and (d) UniQGAN at 8 dB.

IV. PERFORMANCE EVALUATION

For the sake of the reproducibility, we adopt a widely used benchmark RadioML2018.01a [10]. The benchmark is composed of signals with 24 modulation types and 26 SNRs, from -20 dB to 30 dB with 2 dB as an interval. Therefore, there are 624 cases where each case has 4096 signal samples, and each sample has a length of 1024. We select six modulation types (BPSK, QPSK, 32PSK, 128QAM, GMSK, and OQPSK) and seven SNRs (from -2 dB to 10 dB with an interval of 2 dB) in experiments. We use 1000 samples per case, which is divided into training and test sets by $1:9$, and each sample is converted into a 64×64 constellation diagram.

To implement UniQGAN, we adopt LeakyReLU as an activation function, Adam optimizer with a learning rate of 0.001, and 64 for batch size. Then we use L2 loss as a criterion for $\mathcal{L}_{adv}$, cross-entropy loss for $\mathcal{L}^m_{aux}$ and $\mathcal{L}^s_{aux}$, and normalization techniques, i.e., batch normalization for G and spectral normalization [9] for D. For performance comparison, we consider AMC whose training data is augmented by cGAN [4], ACGAN [5], and a variant of UniQGAN that uses the traditional one-hot encoding with concatenation. As a baseline classifier, we adopt a CNN consisting of four convolutional layers followed by one fully-connected layer. We set models on a desktop platform configured with an NVIDIA GeForce RTX 3070 GPU to use the PyTorch framework.

As shown in Table I and Fig.3, we measure the impact of training set augmentation on CNN-based AMC. To train

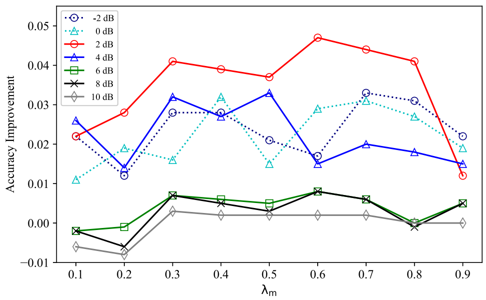

Fig. 4. Classification accuracy improvement at different SNRs, according to λ_m of UniQGAN.

TABLE II
COMPLEXITY ANALYSIS

Augmentation Method	Training time (seconds/epoch)	Converged point (epochs)	Generation time (seconds)
ACGAN	25.9 (3.7×7)	2500	52.4
UniQGAN	33.5	**500**	57.2

generative networks, cGAN and ACGAN models are trained on real data of each SNR, while UniQGAN only trains a single model. The trained models synthesize 500 constellation images for each 42 (6×7) case, where the numbers of the original training set, test set, and generated training data are 100, 900, and 500, respectively. As shown in Table I, data augmentation by the proposed UniQGAN outperforms all other methods in terms of average accuracy of both low SNRs and all SNRs. Especially at low SNRs (from -2 dB to 4 dB), UniQGAN leads to the classification performance enhancement by 6.54% compared to the original. Meanwhile, note that comparisons experience performance degradation in some cases. Specifically, relatively low performance at high SNR cases is due to the limited room for accuracy improvement, i.e., reduced performance for one category may have a greater impact on overall performance. Furthermore, Fig.3 illustrates confusion matrices that analyze the impact of UniQGAN at 2 dB and 8 dB. From Fig.3a and Fig.3b, in complex modulations such as 128QAM and 32PSK, classifier's performance decreased with UniQGAN augmentation. It means the generative model has difficulty learning sophisticated and fine modulation types in a noisy environment. However, as shown in Fig.3c and Fig.3d, UniQGAN successfully models complex modulations at a high SNR, which leads to performance improvement.

We further show the process to decide weight values, including λ_m and λ_s of UniQGAN. As shown in Fig.4, we set λ_m varying from 0.1 to 0.9 and $\lambda_s = 1 - \lambda_m$, to train UniQGAN and analyze accuracy improvement compared to the original. Experimental results show that lower SNR tends to achieve better accuracy improvement, except for the too low SNRs such as -2 dB and 0 dB. Considering the average accuracy improvement at both low SNRs and all SNRs, we determine λ_m to be 0.7 and λ_s to be 0.3.

To assess the complexity and the feasibility, we measure the training and generation time of ACGAN and UniQGAN as shown in Table II. To train an epoch, ACGAN needs 3.7 seconds for each SNR, a total of 25.9 seconds, while UniQGAN needs 33.5 seconds for training a single generative model. However, while ACGAN needs 2500 epochs for equilibrium,

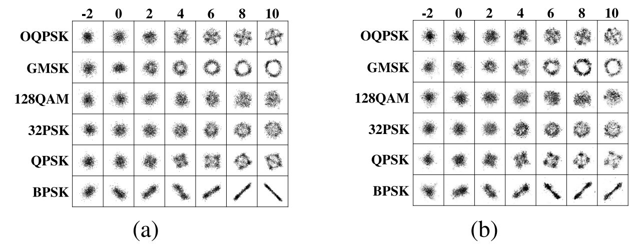

Fig. 5. Constellation diagrams from (a) RadioML2018.01a and (b) 1000-epochs trained single UniQGAN model.

UniQGAN converges around 500 epochs. Not only UniQGAN minimizes the number of trained generators, but it also reduces training time by a quarter. It means that some common hidden information among each SNR is exploited better in UniQGAN. Generation time to augment 500 images for 42 cases is almost the same as less than 60 seconds, which is negligibly short compared to the training time.

Finally, we present constellation diagrams in Fig.5. Fig.5a displays images converted from the RadioML2018.01a, while fake images generated by UniQGAN are shown in Fig.5b. Visualized results show that a single UniQGAN model trained with 1000 epochs learns probability distribution of real data successfully to synthesize realistic data of various SNRs and modulation types.

V. CONCLUSION

In this letter, we have proposed UniQGAN, which handles the data insufficiency of deep learning-based AMC. To generate various IQ constellation diagrams with a single trained model, we suggested multi-conditions embedding and multi-domains classification methods. Experimental results have proved that UniQGAN improves performance by 6.54% on the average accuracy of low SNRs and reduces training time by 75%, compared to ACGAN.

REFERENCES

[1] W. Jang and W. Lee, "Detecting wireless steganography with wavelet analysis," *IEEE Wireless Commun. Lett.*, vol. 10, no. 2, pp. 383–386, Feb. 2021.

[2] H. Zhang, M. Huang, J. Yang, and W. Sun, "A data preprocessing method for automatic modulation classification based on CNN," *IEEE Commun. Lett.*, vol. 25, no. 4, pp. 1206–1210, Apr. 2021.

[3] S. Peng *et al.*, "Modulation classification based on signal constellation diagrams and deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 718–727, Mar. 2019.

[4] M. Patel, X. Wang, and S. Mao, "Data augmentation with conditional GAN for automatic modulation classification," in *Proc. 2nd ACM Workshop Wireless Secur. Mach. Learn.*, Jul. 2020, pp. 31–36.

[5] B. Tang, Y. Tu, Z. Zhang, and Y. Lin, "Digital signal modulation classification with data augmentation using generative adversarial nets in cognitive radio networks," *IEEE Access*, vol. 6, pp. 15713–15722, 2018.

[6] X. Ji, J. Wang, Y. Li, Q. Sun, S. Jin, and T. Q. S. Quek, "Data-limited modulation classification with a CVAE-enhanced learning model," *IEEE Commun. Lett.*, vol. 24, no. 10, pp. 2191–2195, Oct. 2020.

[7] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 1–9.

[8] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. ICML*, 2017, pp. 2642–2651.

[9] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," 2018, *arXiv:1802.05957*.

[10] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 168–179, Feb. 2018.