UniQGAN: Towards Improved Modulation Classification with Adversarial Robustness Using Scalable Generator Design

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

Abstract—Automatic modulation classification (AMC) has been envisioned as a significant element for security issues at the physical layer due to its indispensable role in accurate communications. Recent attention to deep learning has impacted the AMC, which exhibits exceptional performance without manual feature engineering. To guarantee the accuracy and robustness of deep learning-based AMC, data augmentation is a critical issue. While existing studies have used several deep generative models to handle the data insufficiency, these studies face three challenges including low scalability, lengthy training time, and limited accuracy improvement. To this end, this paper presents UniQGAN, a novel unified generative architecture that models I/Q constellation diagrams from various signal-to-noise ratios (SNRs) using a single model. The proposed method enables the generation of high-quality data with a scalable generator, while requiring reduced training time. At the core of UniQGAN are *multi-conditions embedding* and *multi-domains classification* techniques that leverage both SNR and modulation type during the optimization process to enable unified modeling. Using abundant high-quality training data, UniQGAN accelerates the enhanced AMC with high performance and adversarial robustness. Experimental results demonstrate that the data generation by UniQGAN achieves superiority in terms of scalability, training time, and accuracy.

Index Terms—Automatic modulation classification, data augmentation, adversarial robustness, deep learning, GAN

1 Introduction

Automatic modulation classification (AMC) enables legitimate communications by synchronizing modulation schemes between a transmitter and a receiver. It makes AMC essential in ensuring accurate communications and further communication security [2]. Although AMC has started to draw attention in the military domain (e.g., electronic warfare), AMC is also adopted in civilian scenarios and various security problems such as physical layer authentication [3], jamming, and spoofing [4]. Specifically, since complex connections between wireless devices complicate the radio environments, AMC serves key roles in spectrum monitoring and physical-layer authentication to detect attackers. Therefore, we can consider AMC as a starting point for addressing wireless physical-layer threats. Conventional AMC studies include maximum likelihood-based [5] and feature-based [6], [7] approaches. However, the former has a limitation due to the high computational complexity, while the latter accompanies overheads from feature engineering by domain knowledge [8].

One possible solution for the feature engineering issue is using deep learning. Since AMC is expected to be a key component of future 6G communications whose most distinguishing feature is the intelligent communications enabled by deep learning [9], it has triggered numerous related studies applying deep learning to AMC [8], [10], [11], [12], [13], [14]. Unlike the earlier AMC methods that relied on feature-based or likelihood-based approaches, deep learning-based AMC automatically extracts hidden features from received signals without manual feature engineering. There have been many examples for AMC using Recurrent Neural Network (RNN) [10], Long Short Term Memory (LSTM) [11], and Convolutional Neural Network (CNN) [8], [12], [13], [14]. RNN and LSTM perform well on I/Q signals due to their superior ability to process time-series signals. Especially, CNN shows excellent accuracy even with speed improvements [15] on I/Q constellation diagrams, which are typical image data that represent signals. In this paper, we focus on the case that employs the CNN-based AMC for classifying the I/Q diagrams.

The most critical factor affecting classification performance is availability of sufficient high-quality training data. Note that abundant training data is required also for robustness against adversarial attacks [16]. Neural networks are inherently vulnerable to adversarial examples; even a minor perturbation can cause misclassification [17], which motivates studies on adversarial attacks for deep learning-based AMC [18], [19], [20]. Since deep learning-based AMC has deployed in various fields, it would pose fatal impacts if the dependability of the deep learning-based AMC violated. As shown in Fig.1, if adversarial attacks occurred on deep learning-based AMC, the attacks would deteriorate diverse scenarios such as electronic warfare [2], physical layer authentication [3], jamming detection [4], and the internet of things [21]. Addressing adversarial attacks is a crucial issue, and expanding the training data is a significant challenge for deep learning-based AMC. Therefore, this paper focuses on **I/Q data augmentation** for two reasons: high performance and adversarial robustness for AMC.

• *The authors are with the Network and Security Research Laboratory, School of Cybersecurity, Korea University, Seoul 02841, Republic of Korea. E-mail: {islee94, wlee}@korea.ac.kr*

The preliminary version of this paper has been presented in [1] as a letter. Manuscript received 20 May 2022; revised 17 December 2022. (Corresponding author: Wonjun Lee.)

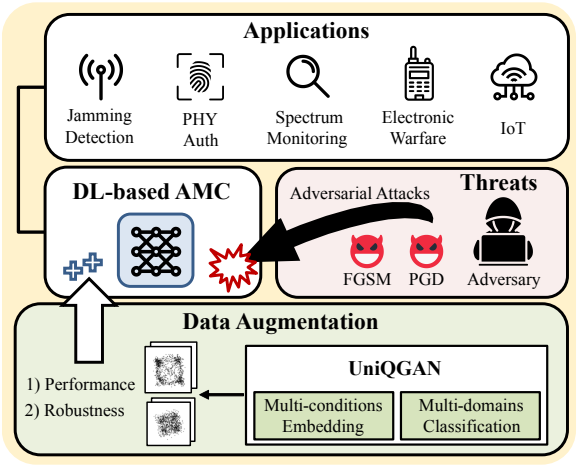


Fig. 1. System overview. An adversary may threaten deep learning-based AMC with adversarial attacks such as FGSM and PGD, resulting in damages to various applications. Training data augmentation is an effective solution to improve the performance and adversarial robustness of deep learning-based AMC.

Since manual data collection may cause overheads and face privacy concerns [22], data augmentation using **deep generative models** may be an effective solution. The generative models, e.g., Generative Adversarial Networks (GAN) [23] and Variational Autoencoder (VAE), can approximate the original sample's probability distribution rather than simply reproduce it. Although some traditional augmentation methods have been used to augment I/Q data such as signals and constellation diagrams [24], there have been numerous attempts to use GAN and VAE for modeling I/Q data [25], [26], [27], [28].

These existing approaches should be reviewed in three aspects: (i) scalability, (ii) training time, and (iii) AMC accuracy improvement. First, keep in mind that we prepare training data from various signal-to-noise ratios (SNRs), as deep learning-based AMC operates only at the SNRs where training data is sampled. The traditional studies [25], [26], [27], [28] are less scalable for a range of SNRs because they focus merely on data generation for a single SNR value using a single generator. Second, the scalability issue results in a long training time. Total GAN training time, i.e., a sum of training times for each SNR case, increases proportionally to the number of SNRs to be modeled. Lastly, accuracy improvements in higher SNRs are frequently underestimated to focus on accuracy in lower SNRs. In this context, we consider the following three challenges derived from the existing literature.

Challenge 1) How can we design a scalable generative architecture over SNRs? To minimize management overheads associated with independent generators, we need to develop a model capable of modeling I/Q data at multiple SNRs. The scalable generator is also expected to uncover some hidden information among the SNRs, which will aid in modeling.

Challenge 2) How can we reduce the generative model's lengthy training time? Deep generative models are accompanied by long training time even though they exhibit excellent performance. After solving the time issue, it is easier to train the generators as needed.

Challenge 3) How can we improve the quality of the generated data? The primary goal of I/Q data augmentation is to enhance the performance of deep learning-based AMC. While accuracy improvement is more important at lower SNRs due to its poor performance, it would be preferable if we also improved accuracy at higher SNRs.

To this end, we propose UniQGAN, a Unified GAN architecture for I/Q constellation diagrams at diverse SNRs. UniQGAN enables modeling high-quality data at different SNRs using a single generator, while significantly reducing training time. The proposed method is based on Auxiliary Classifier GAN (ACGAN) [29] since ACGAN allows generating data of the intended category with good quality. The main contributions of this paper are listed as follows.

- We propose a scalable generator design, UniQGAN, and analyze its scalability for different SNR ranges. UniQGAN shows the potential of the hidden information among the different SNRs, which affects the performance in terms of AMC accuracy and GAN training time.
- We present lightweight and effective ways to reflect both SNR and modulation types simultaneously in an integrated architecture via *multi-conditions embedding* and *multi-domains classification*.
- We provide extensive experimental results demonstrating that data augmentation by UniQGAN successfully improves AMC accuracy at both low and high SNRs, even with reduced GAN training time by a quarter.
- We consider the sophisticated attackers that perform adversarial attacks for CNN-based AMC, providing the detailed security model and experimental results on adversarial robustness.

The remainder of this paper is organized as follows. Section 2 summarizes an overview of the existing literature. Section 3 provides background including system and security models. Section 4 and Section 5 describe the preliminaries and details of UniQGAN. Section 6 discusses experimental results in terms of accuracy, training time, scalability, and adversarial robustness. Section 7 concludes the paper.

2 Related Work

In this section, we briefly review the previous literature on deep learning-based AMC and efforts to handle the data insufficiency issue.

2.1 Deep Learning for AMC

Deep learning has proved superior performance in a variety of communication fields. Deep learning is particularly useful in intelligent communications [10], [11], [30], [31], [32], [33], [34], [35], [36] that serve a critical role in the next-generation communications. Specifically, we examine AMC studies associated with CNN [8], [12], [13], [14], [15], [37], [38], [39], [40] and adversarial learning [18], [19], [20], [41], [42] that has gained a great deal of popularity due to security concerns in neural networks.

Intelligent communications. Deep learning has been widely adopted for intelligent communications in vehicular

networks [30] and general wireless communications. For example, many related studies have used semi-supervised learning [31], transfer learning [32], [33], GAN [34], multi-task learning [35], RNN [10], [36], and LSTM [11]. Perenda et al. [33] applied transfer learning to design a robust AMC classifier against unknown environment information, including channel and signal parameters. The authors also showed that deep learning-based AMC outperforms feature-based methods. Due to the ability to process time-series data (e.g., I/Q signals), RNN and LSTM have been employed for AMC. Zhou et al. [10] proposed RCNet, which enables online learning-based signal detection by converting the structure of RNN. RCNet successfully handles the interference in MIMO-OFDM signals. Rajendran et al. [11] used LSTM-based AMC to learn from the time domain amplitude and phase information. The proposed method showed good performance for time-domain sequences with variable lengths.

CNN-based AMC. Because CNN achieved excellent performance in classification, many researchers have applied CNN to AMC to process both sequence and image data (e.g., spectrograms [13] and I/Q constellation diagrams). For instance, studies using CNN have been conducted to explore weak signal detection [14], multi-task learning [37], classifier complexity [38], and robustness [39]. O'Shea et al. [12] studied CNN in the complex-valued radio signal domain. In [15], several modified constellation diagrams have been proposed to analyze AMC accuracy given different formats of the diagrams. Meng et al. [8] proposed an end-to-end CNN-based AMC, supported by transfer learning to improve the retraining efficiency. However, while many deep learning-based AMC approaches are attracting attention, they can experience overfitting and significant performance degradation without sufficient training data.

Adversarial learning. As deep learning is deployed for many domains such as human activity recognition [41] and network intrusion detection systems [42], adversarial attacks that cause misclassification have also drawn lots of attention. Sadeghi et al. [18] studied the vulnerability of deep learning-based AMC to adversarial attacks. They employed computationally efficient algorithms for black-box and white-box attacks, targeted to CNN-based classifier. Lin et al. [19] conducted extensive simulations for adversarial attacks in terms of accuracy, feasibility, and robustness. They showed that signals with a classifier's low confidence levels may lead to a high risk for attacks. Kim et al. [20] proposed realistic wireless attack methods with adversarial learning, considering channel effects for perturbation design. To address those attacks, deep learning-based classifiers require more training data for robustness against adversarial examples [43].

2.2 Handling Data Insufficiency

To guarantee satisfactory performance of deep learning-based classification, it requires preparing enough amount of labeled training data with high quality. Data augmentation methods have been utilized in many problems, including fraud review detection [44], long range communications [45], and AMC [24], [25], [26], [27], [28]. Huang et al. [24] used traditional augmentation methods in the image domain such as cropping and rotation to augment constellation diagrams. Since the methods merely reproduced the data without modeling it, the quantity has increased but the diversity has not. Ji et al. [25] generated constellation diagrams using a deep generative model called conditional variational autoencoder (CVAE). The authors proposed a feedback unit to link a classifier to CVAE generative network, demonstrating that classification results help train the generator. Patel et al. [26] solved the I/Q data insufficiency problem by employing conditional GAN. They also presented a visualization of generated data, proving that the synthesized data is very similar to the original data. Tang et al. [27] and Chen et al. [28] proposed smart approaches using ACGAN to model constellation diagrams. The generated data contributed to AMC accuracy improvements. However, the traditional AMC studies using generative models [25], [26], [27], [28] have to train each generator corresponding to each SNR independently, resulting in undeniable overheads.

We reviewed existing studies on deep learning-based AMC and learned that data augmentation is a critical issue to achieve high accuracy and robustness. Although there have been several studies using deep generative models for data enlargement, no work has considered generative models' scalability over SNRs. This motivates our work to design a scalable generator, whose architecture and training algorithms will be detailed in Section 5.

3 System Model and Security Model

To clarify the problem scope, we discuss the system and security models considered in this paper. The system model outlines the following three preliminaries for understanding our method: (i) signal model and data preparation, (ii) deep learning-based AMC, and (iii) data augmentation to improve classification performance. The security model describes the considered attack scenarios and assumptions, also giving backgrounds for each attack algorithm.

3.1 System Model

We consider a 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. $r(t)$ consists of a real part (In-phase; I) and an imaginary part (Quadrature; Q). The I/Q components are extracted from the historical data $r(t)$ at certain SNR, and the components are converted into a constellation diagram on a two-dimensional I/Q plane. Then the diagram x is produced with two class labels c_m (modulation type) and c_s (SNR). Even though accurate information within the $r(t)$ cannot be retrieved from x, it is irrelevant since our scope is only to determine c_m prior to demodulation.

As shown in Fig.2, received I/Q signals are converted into an I/Q constellation diagram, which is then passed to the deep learning-based classifier as an input. The classifier

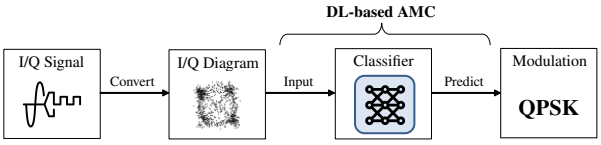


Fig. 2. Deep learning-based AMC. A classifier predicts the modulation type which corresponds to a given I/Q constellation diagram.

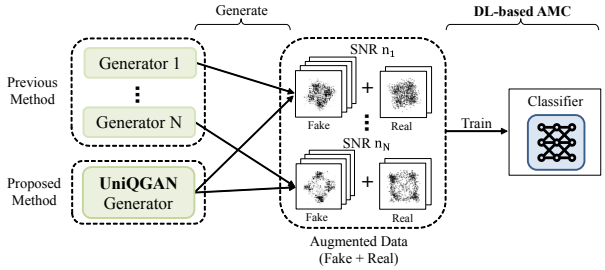


Fig. 3. Data augmentation to improve classification performance. To prepare sufficient training data for the classifier, we assume generative model-based data enlargement where UniQGAN requires a single model for all SNRs, whereas conventional methods train each generator for a corresponding SNR. After the generation, synthesized fake data is combined with the original real data to augment the training dataset.

automatically extracts features from the diagram and identifies the corresponding modulation type. Deep learning-based AMC is a kind of typical classification problem and this supervised learning process can be expressed as $H(x) \rightarrow c_m$, where $H(\cdot)$ represents the classifier.

While enough training data is required to achieve acceptable classification performance, data collection is a costly task. As shown in Fig.3, we concentrate on data augmentation by generative models to train the AMC classifier with the augmented training set. This paper assumes using a unique classifier that trains constellation diagrams from various SNRs, which excludes c_s in both training and prediction. Training data should be prepared with a broad range of SNRs for expected operation at diverse SNRs. Previous studies using generative models [25], [26], [27], [28] have trained and managed each generator independently since the generative models have only one condition c, using the condition for c_m in training. Each generator is trained on the (x, c_m) of each c_s, and the trained generator can generate fake data of corresponding c_s. This low scalability over SNRs causes overheads in managing a number of models. Therefore, we aim to design a scalable generator for different SNRs to improve AMC, and the proposed UniQGAN can address the problem with a unified model.

3.2 Security Model

This paper assumes intelligent attackers who leverage adversarial examples to exploit the dependability of deep learning-based AMC. The security model consists of three parts: (i) the security goal (what the proposed model aims to achieve from a security perspective), (ii) attacker capabilities (knowledge and attacker's goal), and (iii) the considered threat model (background and detailed attack process).

3.2.1 Security goal

We define *adversarial robustness* as a **low drop in AMC accuracy** after adversarial attacks. This paper aims to present a scalable generator design for I/Q data augmentation, also improving the adversarial robustness of deep learning-based AMC and assuring the dependability of AMC. Specifically, our security goal is defined as follows:

- *Availability Goal.* In security context, the data augmentation's purpose is to mitigate the effects of adversarial attacks, preventing adversaries from interfering with legitimate wireless communications.

Since we assume the powerful attackers with full knowledge about the target AMC classifier, the *integrity goal* (i.e., preventing attackers from accessing and modifying the target system) is out of our scope.

3.2.2 Attacker capabilities

There are several categories for adversarial attacks with different criteria. From the attacker's knowledge, we can classify attacks as white-box attacks and black-box attacks.

- *White-box attacks.* An adversary knows all about the target classifier, e.g., model weights, and uses them to calculate gradients for adversarial attacks.
- *Black-box attacks.* An adversary has limited knowledge such as the output of the target classifier.

From the specificity of attacker, we can classify as targeted attacks and non-targeted attacks.

- *Targeted attacks.* The success of the attack is defined as the adversary making the target system misclassify the input to the intended category.
- *Non-targeted attacks.* The success of the attack needs the adversary to let the target system classify the input into any kind of wrong categories.

To consider advanced attackers, we assume attackers with the ability to conduct white-box and non-targeted attacks. In other words, the attackers have full knowledge of the victim AMC classifier, exploiting the model's information (structure and weights) for adversarial attacks. Since our security goal includes only the availability, it does not matter whether the misclassification has become the intended modulation type.

3.2.3 Threat model

This paper envisages the white-box *evasion* attacks, which deteriorate the trained deep learning-based AMC classifiers by adding imperceptibly small noise to the input. We choose the following two representative evasion attack algorithms: fast gradient sign method (FGSM) [46] and projected gradient descent (PGD) [47].

1) *Fast gradient sign method (FGSM).* FGSM starts from the hypothesis that **neural networks are too linear to resist linear adversarial perturbation [17]**. The FGSM finds indistinct noise which maximizes the target model's loss, causing misclassification for the target classifier. The attack generates an adversarial example via an element-wise sum of the noise for the original data as follows:

$$x_{adv} = x + \delta, \tag{2}$$

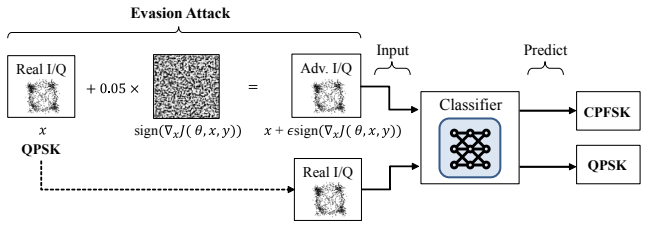


Fig. 4. Generating adversarial examples for I/Q constellation diagrams via evasion attack. An adversary makes an adversarial I/Q diagram by adding indistinguishable noise to an original I/Q diagram. A trained deep learning-based AMC classifier may predict the wrong modulation type as CPFSK on the adversarial examples from the diagram of QPSK.

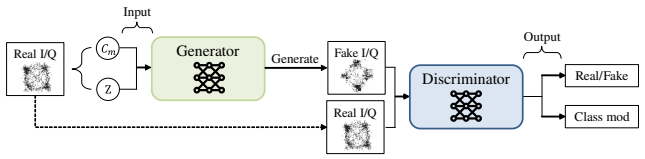


Fig. 5. Modeling I/Q constellation diagrams with ACGAN. The generator synthesizes a fake I/Q diagram using modulation category c_m and random noise z in order to deceive the discriminator. Next, the discriminator attempts to classify differences between fake and real data, and the predicted results affect both neural networks via backpropagation. Following the conclusion of the competitive training phase, the generator theoretically trains the probability distribution of the original samples conditioned by modulation information.

where x_{adv} is the adversarial example, x is the original sample, and δ is the added noise to construct adversarial examples. The noise δ is defined employing a sign function for the gradient ∇_x as follows:

$$\delta = \epsilon sign(\nabla_x J(\theta, x, y)), \tag{3}$$

where $J(\theta, x, y)$ is the loss of the trained model with full information (parameters θ, data x and corresponding labels y), and ϵ constrains the scope of δ as $\|\delta\| \leq \epsilon$. A large ϵ allows higher attack success rate, also increasing the perceivability.

2) *Projected gradient descent (PGD)*. As a variant of FGSM, PGD uses multiple steps to generate adversarial samples x_{adv} while the FGSM adversary calculates a one-step gradient to find the optimal δ. Specifically, PGD uses a learning rate in each step to modify input data x. The attack generates adversarial examples at each step t as follows:

$$x^{t+1} = \prod_{x+S} (x^t + \alpha sign(\nabla_x J(\theta, x, y))), \tag{4}$$

where α is the learning rate and S is the set of allowed perturbations, constrained by ϵ.

Fig.4 explains the evasion attack process that generates adversarial I/Q diagram examples. Since we assume that an adversary has complete knowledge about the target classifier, the adversary can calculate the gradient to decide a proper perturbation that is imperceivable but critical for the classifier. The victim classifier may predict the generated adversarial diagrams as CPFSK, while the original category is QPSK. To achieve adversarial robustness against these threats, I/Q data augmentation is a possible solution [16], and the process for modeling I/Q diagrams will be detailed in the next section.

4 Modeling I/Q Diagrams with GAN

Before examining the design details of UniQGAN, we describe the modeling of I/Q constellation diagrams with Generative Adversarial Networks (GAN) [23]. GAN is a representative deep learning-based generative model composed of two competitive networks, a generator G and a discriminator D. G generates synthetic data $G(z)$ where z is random noise from Gaussian distribution, whereas D outputs a single scalar between 0 and 1, indicating the validity of the input data. We train G to deceive D, while D learns to differentiate successfully. This competitive training is described as a two-player minimax game about one loss function between two networks, with G attempting to minimize the loss and D trying to maximize it. The adversarial loss associated with GAN is expressed as follows:

$$\mathcal{L}_{adv} = \mathbb{E}_x[\log D_{src}(x)] + \mathbb{E}_z[\log(1 - D_{src}(G(z)))], \tag{5}$$

where D_{src} predicts the probability distribution over sources, the probability that a given input comes from real data x. After G and D reach the convergence point (Nash equilibrium) of the competitive optimization, D is theoretically unable to distinguish between the original data x and generated data $G(z)$. However, basic GAN has limitations owing to its low quality and inability of conditional generation, i.e., generating data with an intended category.

Auxiliary Classifier GAN (ACGAN) [29] is one of the enhanced GAN variants that successfully overcomes the aforementioned limitations. The generator G in ACGAN uses the class label c to generate fake data $G(c, z)$, each of which has a corresponding label c. The discriminator D is divided into two components such as D_{src} and D_{aux}, where D_{aux} means the auxiliary classifier. D_{src} in ACGAN is identical to D_{src} in Eq. (5), while D_{aux} outputs domain classification probability. Adversarial loss and auxiliary loss in ACGAN are expressed as follows:

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

$$\mathcal{L}_{aux} = -\mathbb{E}_{x,c}[\log D_{aux}(c|x)] - \mathbb{E}_{z,c}[\log D_{aux}(c|G(z, c))]. \tag{7}$$

The auxiliary loss is beneficial since it allows stability in ACGAN's training by providing additional information about the correct class. Objective functions to be minimized for D and G are $-\mathcal{L}_{adv} + \mathcal{L}_{aux}$ and $\mathcal{L}_{adv} + \mathcal{L}_{aux}$, respectively.

As depicted in Fig.5, we explain the modeling process for I/Q constellation diagrams using ACGAN. To prepare the real constellation diagrams, we convert I/Q sequences of 2×1024 into a two-dimensional I/Q plane. Each diagram has two category labels including c_s (SNR) and c_m (modulation type). Existing studies have used c_m for the only condition since ACGAN is designed to receive one condition in addition to latent vector z. The goal of G is to generate realistic constellation diagrams from c_m and z to deceive D, while D attempts to distinguish between fake and real diagrams. Both outputs from D, i.e., D_{src} and D_{aux}, are used for optimization by backpropagation. Specifically, D_{src} predicts the probability that a given I/Q diagram comes from the original data, experiencing various channel effects on data collection. After training is finished,

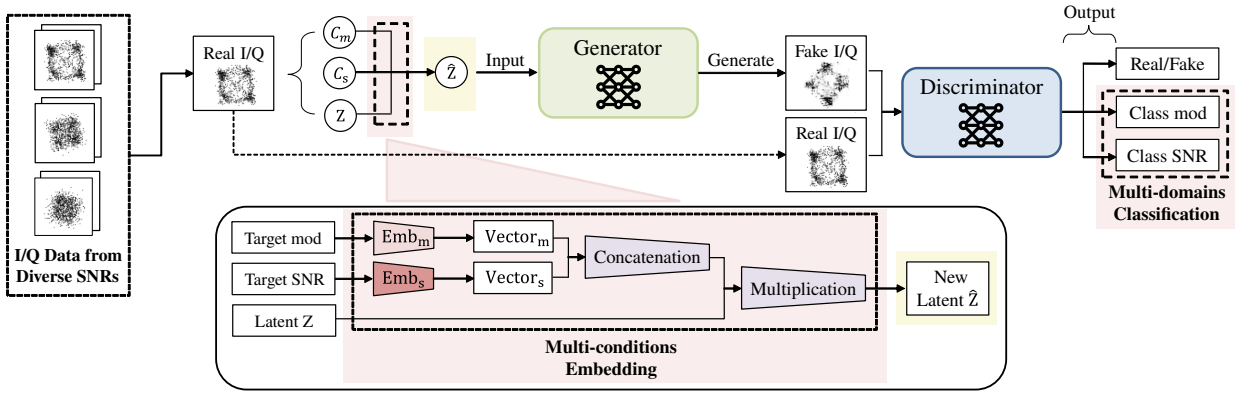


Fig. 6. UniQGAN Architecture. Its key components consist of *multi-conditions embedding* and *multi-domains classification*. In multi-conditions embedding, after concatenating the independently embedded vectors Vector_m and Vector_s, we multiply them with the latent Z to create a new latent $\hat{\mathrm{Z}}$. Then, we extend the capability of ACGAN's auxiliary classifier to two domains (modulation type and SNR) in multi-domains classification. As a result, both conditions c_m and c_s are successfully reflected in the optimization process of UniQGAN, allowing for the scalable architecture to be trained using a single model over diverse SNRs.

the trained generator can produce constellation diagrams for a given SNR. To prepare the diagrams at various SNRs, it is necessary to train and manage generators as many as the number of SNRs, which sparks our research.

5 UniQGAN Design

We propose UniQGAN, a scalable GAN design to generate constellation diagrams at various SNRs with a single generator. This section discusses design objectives, suggested structure, and training algorithm.

5.1 Design Goals

UniQGAN sets three design objectives including scalability, training time, and data quality. We summarize our insights for addressing each design requirement.

- *Scalable design.* We define *scalability* as the capability for generating data at multiple SNRs with a single model. The simple but most intuitive approach is to include an additional condition (SNR) in modeling, as well as a modulation type that is the only information used in traditional methods. We would introduce a new embedding technique to reflect both conditions and modify losses. In Section 6.5, we analyze scalability in more detail.
- *Reduced training time.* The loss convergence point between the generator and discriminator should be advanced to reduce training time. Since considered constellation diagram is a low-resolution image, we would design the lightweight generative architecture. In Section 6.4, we measure how much UniQGAN saves training time.
- *High-quality of generated data.* To increase the generated data quality, reasonable criteria for assessment are necessary. We evaluate two aspects including visualization (Section 6.6) and accuracy improvement (Section 6.1, Section 6.2, and Section 6.3). Although visualization analysis itself may not be a strict standard, it is widely used in the majority of machine learning research and we can also refer to the results. In addition, we quantify the changes in AMC accuracy caused by data augmentation and reflect them in determining UniQGAN's weights (Section 5.3).

5.2 UniQGAN Architecture

As illustrated in Fig.5, the architecture of UniQGAN deviates from ACGAN in that the generator and discriminator consider modulation type and SNR simultaneously. In this section, we describe the main parts of UniQGAN and several techniques to achieve faster convergence with reduced training time.

5.2.1 Multi-conditions embedding

Traditional deep learning-based embedding methods for multiple inputs are inappropriate for our problem domain due to the excessive complexity of the methods. We present a simple but effective embedding method called *multi-conditions embedding* to handle both modulation type and SNR. The proposed embedding allows the generation of constellation diagrams from various SNRs using a single trained generator.

In multi-conditions embedding as shown in Fig.5, each embedding layer embeds both conditions c_m and c_s individually. The embedding layers convert inputs to *dense* vectors and train on optimized embedding weights. Note that certain conditions exhibit stronger correlations in the AMC problem, e.g., similar SNRs. To represent relationships between embedded vectors, we employ neural embedding rather than the commonly adopted one-hot encoding which uses *sparse* vectors. The embedded vectors are concatenated and then multiplied with random noise z to form a new latent vector $\hat{z}$, which is fed into the G. In our experiments, multiplication (instead of concatenation) with random noise alleviated the mode collapse, the representative GAN failure in which G produces a limited variety of samples. The overall derivation process to form a new latent vector $\hat{z}$ is defined as follows:

$$\hat{z} = z \cdot (\mathrm{Emb_s}(c_s) \parallel \mathrm{Emb_m}(c_m)), \tag{8}$$

where $\mathrm{Emb_s}$ and $\mathrm{Emb_m}$ are embedding layers for SNR and modulation type, respectively. We implement output vectors

of $\mathrm{Emb_s}(\cdot)$ and $\mathrm{Emb_m}(\cdot)$ having a size of 1×64, while z and $\hat{z}$ having a size of 1×128 in the experiments.

5.2.2 Multi-domains classification

Since the discriminator D in ACGAN is designed to consider only one condition, we need to modify D for scalable design. The capability of the auxiliary classifier D_{aux} in Eq.(7) is extended to both conditions c_s and c_m for multi-domains classification. D_{aux} is divided into D^s_{aux} and D^m_{aux}, where D^s_{aux} represents a probability distribution over the c_s and the D^m_{aux} over c_m. Multi-domains classification enables discriminator D in UniQGAN to output three scalars simultaneously: $D : x \rightarrow \{D_{src}(x), D^s_{aux}(x), D^m_{aux}(x)\}$. The three outputs of D are employed to construct losses in UniQGAN (Section 5.3). The structure of D is composed of repeated four convolutional blocks, followed by one linear layer that has three styles. In $D_{src}(x)$, the last layer adopts one output node and Sigmoid activation function. In two auxiliary classifiers D^s_{aux} and D^m_{aux}, however, employ seven (the number of SNRs) output nodes and Softmax activation function.

5.2.3 Model enhancement

Stabilized convergence between the generator and discriminator is important to guarantee the quality of generated data. UniQGAN employs several techniques to stabilize the training process. We adopt a dropout [48] rate of 0.25 for each layer in D. Dropout is one of the representative strategies to handle overfitting by partially omitting neurons in deep neural networks. We use label smoothing [49], replacing correct labels for real data x with 0.9 and for fake data $G(\hat{z})$ with 0.1, from 1.0 and 0.0 respectively. Label smoothing is a commonly used approach that substitutes smoothed values for the target values to prevent a classifier (i.e., discriminator in GANs) from overfitting [50]. Finally, we employ spectral normalization [51] to restrict the Lipschitz constant of D, resulting in accelerated convergence. Even though the most essential components of UniQGAN architecture are multi-conditions embedding and multi-domains classification, all of these enhancement techniques also contribute to the stable convergence of UniQGAN.

5.3 Training Algorithm

The optimization process of UniQGAN relies on previously defined terms ($\hat{z}$, D_{src}, D^m_{aux}, and D^s_{aux}), and Algorithm 1 shows the pseudocode on training UniQGAN. We initialize generator parameters θ_G and discriminator parameters θ_D as presented in [52]: all model weights are initialized from a Normal distribution with a mean of 0 and a standard deviation of 0.02. After initializing the parameters θ_G and θ_D, we prepare training data x with corresponding labels (c_m, c_s). To guarantee that the generated data preserves high quality, adversarial loss in UniQGAN is expressed as follows:

$$\mathcal{L}_{adv} = \mathbb{E}_x[\log D_{src}(x)] + \mathbb{E}_{z,c_m,c_s}[\log(1 - D_{src}(G(\hat{z})))]. \quad (9)$$

Generator G synthesizes fake data $G(\hat{z})$ corresponding to the intended categories (c_m, c_s), using $\hat{z}$ derived from multi-conditions embedding. The discriminator D tries to distinguish whether the given data is from x or $G(\hat{z})$, and the predicted results are used to compute losses.

Algorithm 1: UniQGAN Training Algorithm

Require: Training epochs N; Learning rate α;
Batch size n; Weights for auxiliary losses λ_m, λ_s.

Initialize θ_G, θ_D.
for $i = 1$ to N **do**
  for $j = 1$ to n **do**
    Sample real I/Q data $x \sim \mathbb{P}_r$ with labels c_s, c_m.
    Sample latent variable $z \sim p(z)$.
    $\hat{z} \leftarrow z \cdot (\mathrm{Emb_s}(c_s) \parallel \mathrm{Emb_m}(c_m))$
    $\hat{x} \leftarrow G(\hat{z})$
    Compute $\mathcal{L}_{adv}, \mathcal{L}^m_{aux}, \mathcal{L}^s_{aux}$ with $D(x), D(\hat{x})$.
    $\mathcal{L}^j_D \leftarrow -\mathcal{L}_{adv} + \lambda_m \mathcal{L}^m_{aux} + \lambda_s \mathcal{L}^s_{aux}$
    $\mathcal{L}^j_G \leftarrow \mathcal{L}_{adv} + \lambda_m \mathcal{L}^m_{aux} + \lambda_s \mathcal{L}^s_{aux}$
  end for
  $grad_{\theta_D} \leftarrow -\nabla_{\theta_D}[\frac{1}{n}\sum_{j=1}^{n} \mathcal{L}^j_D]$
  $\theta_D \xleftarrow{update} \theta_D - \alpha \cdot \mathrm{Adam}(\theta_D, grad_{\theta_D})$
  $grad_{\theta_G} \leftarrow -\nabla_{\theta_G}[\frac{1}{n}\sum_{j=1}^{n} \mathcal{L}^j_G]$
  $\theta_G \xleftarrow{update} \theta_G - \alpha \cdot \mathrm{Adam}(\theta_G, grad_{\theta_G})$
end for

Since we have designed D to have two auxiliary classifiers D^m_{aux} and D^s_{aux}, we define two auxiliary losses as follows:

$$\begin{aligned}\mathcal{L}^m_{aux} = &-\mathbb{E}_{x,c_m}[\log D^m_{aux}(c_m|x)] \\ &- \mathbb{E}_{z,c_m,c_s}[\log D^m_{aux}(c_m|G(\hat{z}))],\end{aligned} \quad (10)$$

$$\begin{aligned}\mathcal{L}^s_{aux} = &-\mathbb{E}_{x,c_s}[\log D^s_{aux}(c_s|x)] \\ &- \mathbb{E}_{z,c_m,c_s}[\log D^s_{aux}(c_s|G(\hat{z}))].\end{aligned} \quad (11)$$

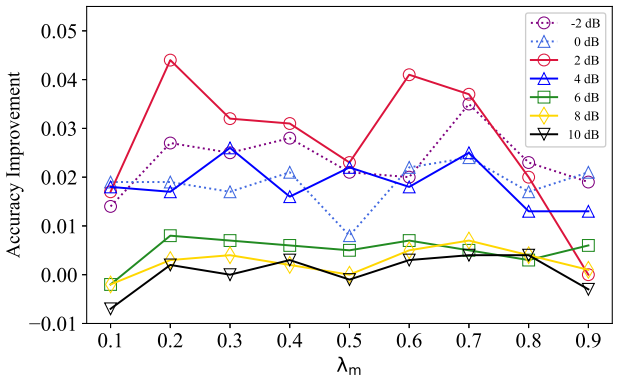
Each auxiliary loss contributes to auxiliary classifiers' prediction capabilities for modulation type and SNR, respectively. Consequently, our full objective functions derived from equations above (Eq.(9), Eq.(10), and Eq.(11)) to train D and G are expressed as follows

$$\mathcal{L}_D = -\mathcal{L}_{adv} + \lambda_m \mathcal{L}^m_{aux} + \lambda_s \mathcal{L}^s_{aux}, \quad (12)$$

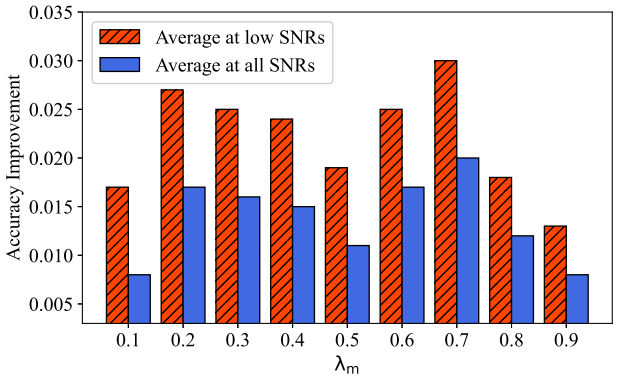
$$\mathcal{L}_G = \mathcal{L}_{adv} + \lambda_m \mathcal{L}^m_{aux} + \lambda_s \mathcal{L}^s_{aux}, \quad (13)$$

where $\lambda_m \in (0, 1)$ and $\lambda_s \in (0, 1)$ are weight values, respectively. Although optimized λ_m and λ_s may fluctuate under experimental conditions, we set $\lambda_m = 0.7$ and $\lambda_s = 0.3$ on the basis of experimental results. Both D and G optimize their parameters using stochastic gradient descent to minimize $\mathcal{L}_D$ and $\mathcal{L}_G$, respectively.

To improve the generated data quality, we consider AMC accuracy when determining the weights of UniQGAN, including λ_m and λ_s. As shown in Fig.7, we measure the AMC accuracy enhancement by data augmentation according to λ_m of UniQGAN. We set λ_m varying from 0.1 to 0.9 and $\lambda_s = 1 - \lambda_m$. Fig.7(a) shows results at each SNR and the accuracy improvement is more apparent at lower SNRs except for too low SNRs (-2 dB and 0 dB). We also observe that if the λ_m is set too high or too low, the accuracy improvement is limited. Fig.7(b) presents the average of accuracy improvement at both low SNRs (from -2 dB to 4 dB) and all SNRs (from -2 dB to 10 dB). We can achieve the highest accuracy improvement with λ_m of 0.7 and λ_s of 0.3. Specifically, with λ_m of 0.7, UniQGAN leads to accuracy enhancement by 6.13% (from 0.489 to 0.519) in low SNRs and 2.87% (from 0.696 to 0.716) in all SNRs, respectively.



(a)



(b)

Fig. 7. Accuracy improvement according to λ_m of UniQGAN. Considering (a) detailed results at different SNRs and (b) averages of accuracy improvements at low and all SNRs, we set λ_m to be 0.7.

6 Performance Evaluation

For reproducibility, we employ a widely used benchmark called RadioML2018.01a [53] collected by software-defined radios (two USRP B210s). The benchmark consists of signals with 24 different modulation types and 26 different SNRs, from -20 dB to 30 dB with a 2 dB interval. There are 624 (24×26) cases where each case contains 4096 signal samples, with a length of 1024. In experiments, we use the representative six modulation types (BPSK, QPSK, 32PSK, 128QAM, GMSK, and OQPSK) and seven SNRs, ranging from -2 dB to 10 dB with 2 dB of an interval. The selected modulation types contain four simple modulations (BPSK, QPSK, GMSK, and OQPSK) and two complex modulations (128QAM and 32PSK). We exclude cases where the SNR is less than -2 or greater than 10, as the data in the two SNR ranges are very similar visualization at both ends; we judge the similarity via accuracy and visualized results. In the selected 42 (6×7) cases, we use 1000 samples per case and convert each sample into a 64×64 constellation diagram. Experiments generally consist of two phases: data augmentation and classification. Since the primary scope of this research is data augmentation, we do not use the K fold cross-validation for classification to decrease overwhelming complexity. We divide the original samples into training and test sets by $1:9$ to assume the data insufficiency.

To implement UniQGAN, we use LeakyReLU as an activation function, Adam optimizer with a learning rate of 0.001, and a batch size of 64. We adopt L2 loss as a criterion for $\mathcal{L}_{adv}$, cross-entropy loss for $\mathcal{L}^m_{aux}$ and $\mathcal{L}^s_{aux}$. Normalization methods such as batch normalization and spectral normalization [51] are employed for G and D, respectively. To compare performance, we analyze AMC whose training data is augmented by cGAN [26], ACGAN [27], and UniQGAN*, a variant of UniQGAN that employs the traditional one-hot encoding with concatenation. As a baseline classifier, we use CNN consisting of four convolutional layers followed by one fully-connected layer, which predicts the modulation type of the highest probability. The classifier's detailed structure is summarized in Table 1. Each convolutional layer employs a kernel size of 5×5, a ReLU activation function, batch normalization, and a dropout mechanism. The classifier trains repetitively on a 64×64 image with a label (modulation type) throughout the training phase. We set models on a desktop platform configured with an NVIDIA GeForce RTX 3070 GPU to implement the PyTorch framework.

TABLE 1
The Structure of CNN-based Classifier

Type	Structure
-	**Input** (constellation diagram)
Convolutional layer	Conv2D (64, 5×5) + ReLU + BN + Dropout
Convolutional layer	Conv2D (32, 5×5) + ReLU + BN + Dropout
Convolutional layer	Conv2D (16, 5×5) + ReLU + BN + Dropout
Convolutional layer	Conv2D (8, 5×5) + ReLU + BN + Dropout
-	Flatten
Fully-connected layer	Dense (2048) + Softmax
-	**Output** (predicted modulation type)

To demonstrate the ability of UniQGAN, we analyze how the data augmentation methods affect the performance of CNN-based AMC. Although there are other metrics for classification such as precision, recall, and f1-score, we only consider accuracy since the experiments assume a class-balanced dataset. The employed metric is defined as follows:

- Accuracy = (TP + TN) / (TP + TN + FP + FN),

where True Positive (TP) and True Negative (TN) are the number of correctly predicted ones, while False Positive (FP) and False Negative (FN) mean the opposite.

6.1 Effect of Data Augmentation on Accuracy

We examine the effect of training data augmentation on CNN-based AMC accuracy. Traditional generative methods like cGAN and ACGAN need to train seven distinct generators for each SNR, whereas UniQGAN requires only a single generator. Trained generators produce 500 constellation diagrams for each 42 case, and the numbers of the original training set, test set, and generated data are 100, 900, and 500, respectively. The generated data is only used for training set enlargement, while not exposed to the test.

As shown in Table 2, data augmentation using the proposed UniQGAN surpasses all other approaches in terms of average accuracy at low and all SNRs. UniQGAN improves classification accuracy compared to the original at low SNRs (from -2 dB to 4 dB) by 6.13% (from 0.489 to 0.519) and at all SNRs (from -2 dB to 10 dB) by 2.87% (from 0.696 to

TABLE 2
Accuracy on RadioML2018.01a Benchmark Augmented by Different GANs

Augmentation Method	SNR (dB)							Low SNRs (-2∼4)	All SNRs (-2∼10)
	-2	0	2	4	6	8	10	Average	Average
Original (No augmentation)	0.234	0.325	0.555	0.841	0.956	0.976	0.985	0.489	0.696
cGAN	0.253	0.336	0.530	0.824	0.946	0.966	0.976	0.486	0.690
ACGAN	0.250	0.340	0.518	0.847	0.950	0.970	0.978	0.489	0.693
UniQGAN* (One-hot encoding)	0.267	**0.352**	0.573	0.851	0.958	0.976	0.988	0.511	0.709
UniQGAN (Multi-conditions embedding)	**0.269**	0.349	**0.592**	**0.866**	**0.961**	**0.983**	**0.989**	**0.519**	**0.716**

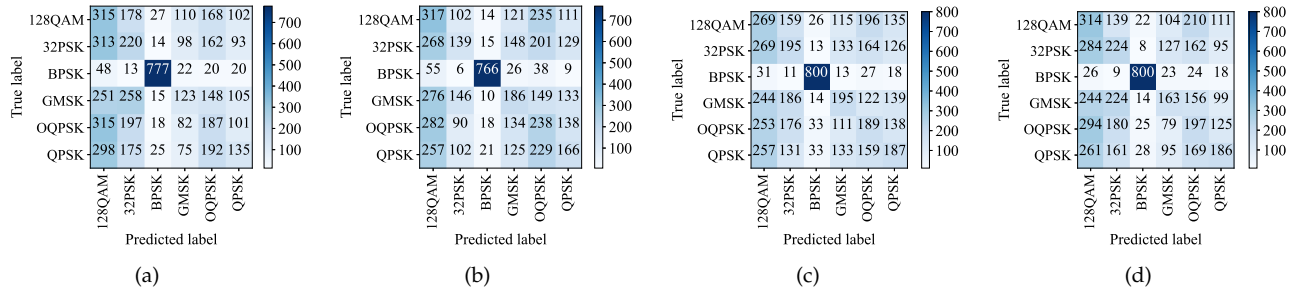


Fig. 8. Confusion matrix of 0 dB for (a) Original, (b) cGAN, (c) ACGAN, and (d) UniQGAN.

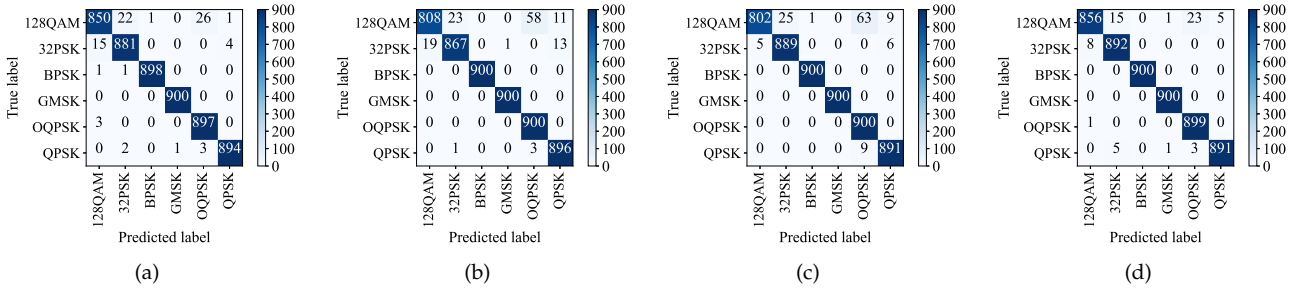


Fig. 9. Confusion matrix of 10 dB for (a) Original, (b) cGAN, (c) ACGAN, and (d) UniQGAN.

0.716). Note that cGAN and ACGAN experience accuracy decreases after augmentation in some cases. Specifically, they exhibit a drop in overall accuracy when SNR is 2 dB or above. The reason for the low enhancement at higher SNRs is that there is limited space for accuracy improvement, i.e., decreased accuracy in one category may have a significant influence on overall performance. Although UniQGAN also achieves better accuracy improvement at lower SNRs than at higher SNRs, at least it does not suffer from accuracy degradation at higher SNRs.

We further assess the accuracy of each category via a confusion matrix. Fig.8 and Fig.9 illustrate confusion matrices that analyze the impact of three generative models at 0 dB and 10 dB, respectively. As shown in Fig.8, all the generative models successfully model simple modulation types at lower SNR, resulting in accuracy enhancement after the data enlargement. However, modeling complex modulation schemes (128QAM and 32PSK) is challenging under noisy conditions. Both cGAN and ACGAN fail to model at least either of the schemes, resulting in an accuracy decrease on sophisticated modulation types. Nevertheless, the overall accuracy at 0 dB increases with data augmentation by cGAN and ACGAN, from 0.325 to 0.336 and 0.340, respectively. The reason is that the rise in accuracy for simple modulations exceeds the drop in complex modulations.

The performance difference between baselines and UniQGAN becomes more apparent at higher SNR as illustrated in Fig.9. While cGAN and ACGAN still struggle to learn sophisticated modulation types, UniQGAN improves performance beyond maintaining it. Since the accuracy improvement on simple modulations fades away, cGAN and ACGAN experience a decrease in overall accuracy. However, as shown in Fig.9(a) and Fig.9(d), UniQGAN successfully models complex modulations at high SNR, enhancing overall accuracy.

6.2 Effect of Multi-conditions Embedding

To analyze the effect of multi-conditions embedding, we compare UniQGAN with UniQGAN* which is a variant that adopts traditional one-hot encoding. The embedding of UniQGAN* to derive a $\hat{z}$ is expressed as follows:

$$\hat{z} = z \parallel \mathrm{Emb}^*_{\mathrm{s}}(c_s) \parallel \mathrm{Emb}^*_{\mathrm{m}}(c_m), \tag{14}$$

where $\mathrm{Emb}^*_{\mathrm{s}}$ and $\mathrm{Emb}^*_{\mathrm{m}}$ indicate one-hot encoding for SNR and modulations, respectively. The formed $\hat{z}$ is passed to G same as illustrated in Fig.5. The embedding in UniQGAN* also differs from the multi-conditions embedding in that

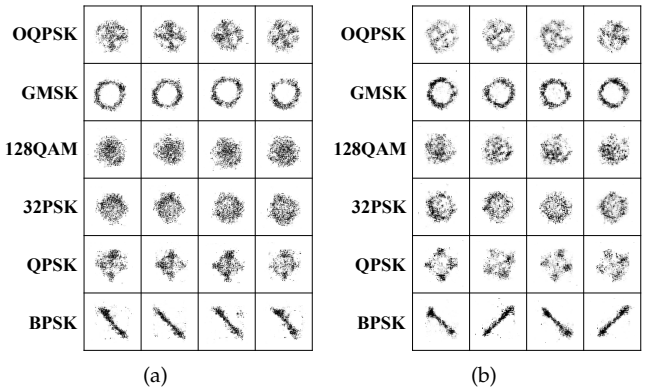


Fig. 10. Constellation diagrams at 10 dB generated by (a) UniQGAN* (a variant of UniQGAN with one-hot encoding) and (b) UniQGAN (with multi-conditions embedding). The multi-conditions embedding mitigates mode collapse, i.e., the representative GAN failure associated with the lack of diversity.

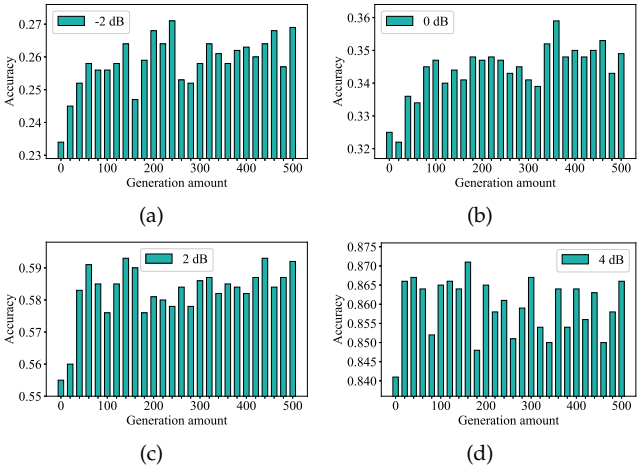


Fig. 11. Accuracy according to data generation amount of UniQGAN at (a) -2 dB, (b) 0 dB, (c) 2 dB, and (d) 4 dB. In X-axis, 0 means no augmentation.

UniQGAN* concatenates the embedded vectors rather than multiplying them.

As shown in Table 2 (Section 6.1), both UniQGAN and UniQGAN* show better results compared with the other augmentation methods such as cGAN and ACGAN. Additionally, UniQGAN achieves slightly better accuracy than UniQGAN*, except for the case of 0 dB. Note that the structures of both models are the same excepting embeddings. This indicates that multi-conditions embedding contributes a little more in terms of accuracy improvement compared to one-hot encoding.

The performance gap between these embeddings becomes distinguished in the aspects of generated data diversity. Fig.10 shows the results of generating four constellation diagrams for six modulation types at 10 dB with UniQGAN* and UniQGAN. As illustrated in Fig.10(a), while data from UniQGAN* look similar to original I/Q constellation diagrams, the diversity is extremely limited since mode collapse occurred. Meanwhile, Fig.10(b) shows that UniQGAN alleviates the diversity issue. Experimental results show that UniQGAN has strengths in accuracy enhancement and output diversity via multi-conditions embedding.

6.3 Effect of Generation Amount on Accuracy

We investigate how generation amount affects AMC accuracy, assuming that the amount correlates with accuracy when high-quality data is available. As shown in Fig.11, we measure the accuracy changes according to data augmentation by UniQGAN. We focus on low SNRs, from -2 dB to 4 dB, where the effect is more noticeable than at higher SNRs. Original training data consists of 42 cases (6×7) and each case contains 100 samples. We measure the accuracy by increasing the generation amount for each case from 0 to 500 at intervals of 20. Generating 500 samples per case implies a sixfold increase in the size of the training data, from 100 to 600.

As illustrated in Fig.11(a), Fig.11(b), and Fig.11(c), accuracy tends to gradually improve as generation amount increases at very low SNRs (from -2 dB to 2 dB). While the relationship between the amount and accuracy improvement is not always directly proportional due to the inherent randomness in deep learning, performance generally improves until the training data is doubled, i.e., where the generated amount is 100. However, Fig.11(d) shows the tendency disappears at higher SNRs, even at 2 dB. This shows that while there is no strong correlation between the generation amount and accuracy improvement, we can observe a certain tendency at lower SNRs.

TABLE 3
Feasibility Analysis

Augmentation Method	Training Secs/Epoch	Training Epochs	Training Time (Secs)	Generation Time (Secs)
cGAN	19.88 (2.84×7)	4500	89460	46.84
ACGAN	27.09 (3.87×7)	2500	67725	67.82
UniQGAN	33.71	**500**	**16855**	72.38

6.4 Feasibility Analysis

To analyze the feasibility and the complexity, we measure the training and generation time of cGAN, ACGAN, and UniQGAN as shown in Table 3. While cGAN and ACGAN need training of seven generators individually, UniQGAN requires training of a single model.

The total training time for each augmentation method is calculated by multiplying the training epochs by the training time needed for an epoch. For instance, to train an epoch, ACGAN requires 27.09 seconds (i.e., 3.87 seconds for each SNR) whereas UniQGAN takes 33.71 seconds to train a single generative model. While ACGAN needs 2500 epochs to reach equilibrium, UniQGAN converges around 500 epochs. Total training times for cGAN, ACGAN, and UniQGAN are 89460 (19.88×4500), 67725 (27.09×2500), and 16855 (33.71×500) seconds, respectively. Not only does UniQGAN minimize the number of trained generators, but it also decreases training time by at least a quarter compared to the baselines. After training is complete, we produce 500 constellation diagrams for each of the 42 cases (6×7) to measure generation time, a total of 21000 images. Similar to the training process, cGAN and ACGAN generate data

TABLE 4
Scalability Analysis for Different SNR Ranges

Case	Augmentation Method	SNR Ranges & # of Generators	Max Training Epochs	SNR (dB) -2	0	2	4	6	8	10	Low SNRs Average	All SNRs Average
(1)	Original	-	-	0.234	0.325	0.555	0.841	0.956	0.976	0.985	0.489	0.696
(2)	ACGAN	-2 / 0 / 2 / 4 / 6 / 8 / 10 # 7	2500	0.250	0.340	0.518	0.847	0.950	0.970	0.978	0.489	0.693
(3)	UniQGAN	-2 / 0 / 2 / 4 / 6 / 8 / 10 # 7	1200	0.253	0.331	0.549	0.847	0.960	0.976	0.980	0.495	0.699
(4)	UniQGAN	-2, 0 / 2, 4 / 6, 8 / 10 # 4	1200	0.263	0.346	0.559	0.844	0.964	0.980	0.987	0.503	0.706
(5)	UniQGAN	-2, 0, 2 / 4, 6 / 8, 10 # 3	1000	**0.271**	0.342	0.564	**0.872**	**0.966**	0.979	0.985	0.512	0.711
(6)	UniQGAN	**-2, 0, 2, 4, 6, 8, 10 # 1**	**500**	0.269	**0.349**	**0.592**	0.866	0.961	**0.983**	**0.989**	**0.519**	**0.716**

using seven trained generators while UniQGAN employs only one generator. The measured generation times are approximately 50 and 70 seconds, which is negligibly short compared to the training time.

We observed that UniQGAN diminishes GAN training time by advancing the convergence point between the generator and discriminator. This means that UniQGAN effectively exploits some shared hidden information between SNRs owing to its scalable design, which models data from multiple SNRs simultaneously with a unified model.

6.5 Scalability Analysis

Furthermore, we validate the scalability of UniQGAN by analyzing accuracy and GAN training time with varying SNR ranges as shown in Table 4. In cases (1), (2), and (6), they show the performance of the Original, ACGAN, and UniQGAN same as mentioned in Table 1 and Table 3. Case (3)∼(6) show the cases of UniQGAN trained on different SNR ranges. In case (5), for example, three UniQGAN generators are trained on {-2, 0, 2}, {4, 6}, and {8, 10} dB. Other experimental settings such as dataset and CNN-based classifier are the same as in Section 6.1.

As shown in case (3)∼(6), the training epochs required for UniQGAN decrease as the SNR range covered by one generator is expanded. Specifically, 1200, 1000, 800, and 500 epochs are required to train UniQGAN with data of one, two, three, and seven SNRs, respectively. Since the required training epochs vary from the covered SNR ranges of each generator, Table 4 displays only the max training epochs of the generator in each case. The results are somewhat intuitive because UniQGAN utilizes an unified model that can exploit helpful information from different SNRs. More surprisingly, the unified modeling also improves AMC accuracy, from 0.495 to 0.519 at low SNRs and from 0.699 to 0.716 at all SNRs as shown in cases (3) and (6). This means that the scalable design of UniQGAN improves the quality of generated data, even with the reduced training time.

UniQGAN enables not only unified modeling over multiple SNRs but also individual modeling for each SNR as shown in case (3). From cases (2) and (3), we observe that using UniQGAN instead of ACGAN can reduce training epochs needed for convergence by more than twice. Since training times for an epoch in cases (2) and (3) are 27.09 (3.87×7) and 33.81 (4.83×7) seconds, total training times for each case are 67725 (27.09×2500) and 40572 (33.81×1200) seconds. UniQGAN (in case (3)) also outperforms ACGAN in terms of average accuracy at both low and high SNRs.

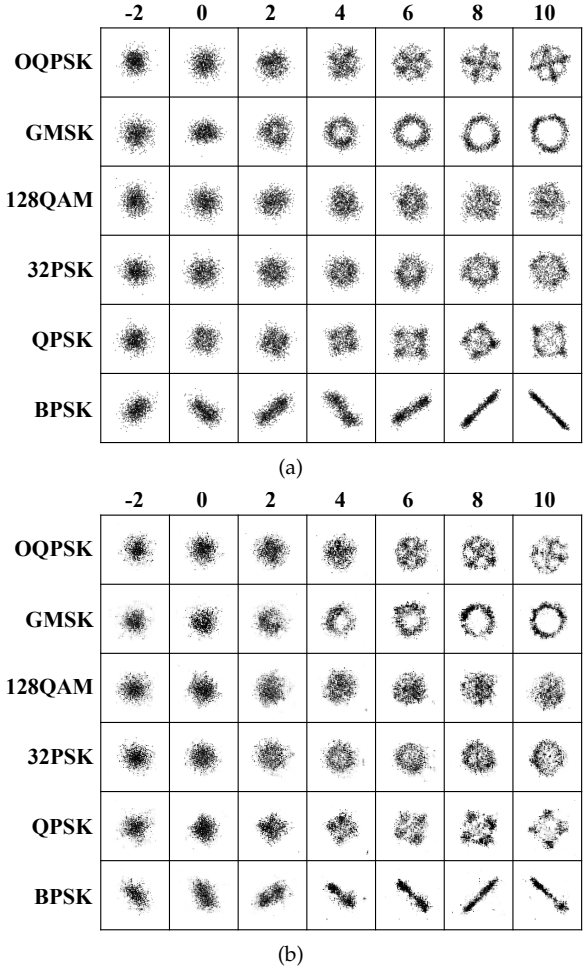


Fig. 12. Constellation diagrams (a) converted from RadioML2018.01a dataset and (b) generated by the 1000-epochs trained single UniQGAN model.

The results show that we can replace previous augmentation methods like ACGAN with UniQGAN, even for modeling data at each SNR independently.

6.6 Visualization of Generated Data

As shown in Fig.12, we present visualized constellation diagrams for each modulation type and SNR used in our experiments. Fig.12(a) displays original diagrams converted from the RadioML2018.01a. Distinguishing between modulation types is difficult at lower SNRs, which explains why

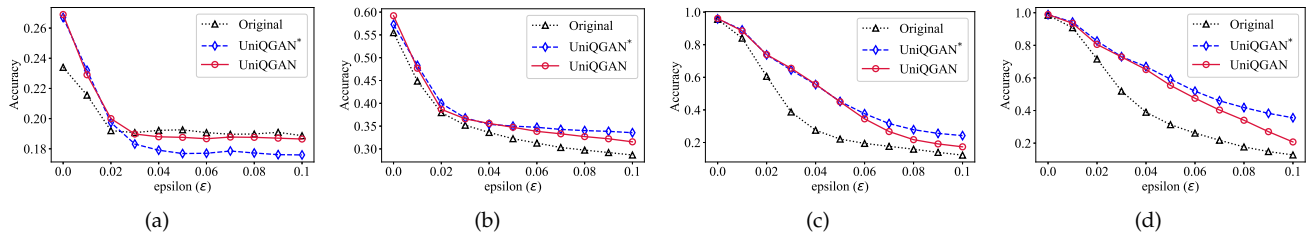


Fig. 13. Accuracy under FGSM attack with varying epsilons at (a) -2 dB, (b) 2 dB, (c) 6 dB, and (d) 10 dB.

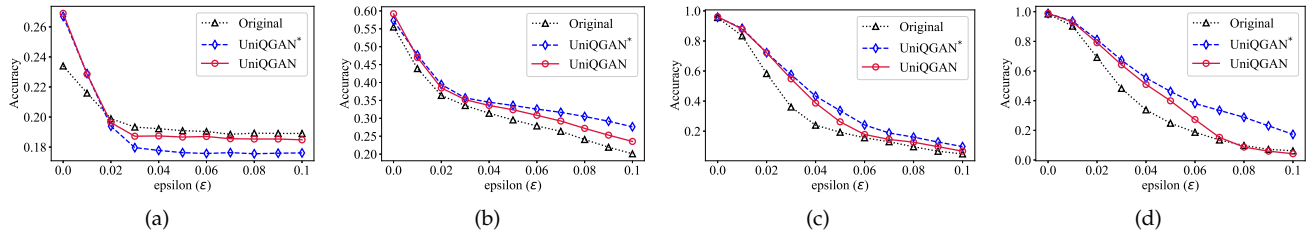


Fig. 14. Accuracy under PGD attack with varying epsilons at (a) -2 dB, (b) 2 dB, (c) 6 dB, and (d) 10 dB.

deep learning-based AMC shows low accuracy at those SNRs. Fig.12(b) exhibits diagrams generated by a single UniQGAN generator trained with 1000 epochs. Visualized results demonstrate that the generator successfully models real data to augment reasonable data of diverse SNRs and modulation types. Without duplicating the original data, the trained generator produces nonexistent data that follow the probability distribution of the original data.

6.7 Adversarial Robustness Analysis

Finally, we investigate the effect of training data augmentation on the adversarial robustness of deep learning-based AMC. We choose both UniQGAN and UniQGAN* as the augmentation methods to show the influence of multi-conditions embedding. We use an open-source framework IBM Robustness Toolbox [54] to implement two representative white-box evasion attacks: FGSM [46] and PGD [47]. The targeted system of the attacks is the same classifier as described in Section 6.1: the 4-layered CNN classifier, using a training set, test set, and generated data of 100, 900, and 500, respectively for each case (modulation type, SNR).

Robustness Under Different Perturbations. We analyze the impact of attacks' perturbation on adversarial robustness with three classifiers: one baseline classifier (Original, i.e., no augmentation) and two data-augmented classifiers (UniQGAN and UniQGAN*). We measure adversarial robustness by investigating a change in accuracy after the adversarial attacks; more robust classifier would experience less amount of drop in accuracy. As shown in Fig.13 and Fig.14, we observe AMC accuracy at different SNRs under FGSM and PGD attacks, with varying ϵ from 0.0 to 0.1 with an interval of 0.01, where 0.0 means no attack.

Accuracy overall decreases as epsilon increases since a larger perturbation means allowing more modifications from original data for adversarial examples. Accuracy drop is more remarkable at higher SNRs, which shows better accuracy without attacks than at lower SNRs. As shown in Fig.13(a) and Fig.13(d), FGSM with a perturbation of 0.05 drops the accuracy in Original at -2 dB by 17.5% (from 0.234 to 0.193), whereas at 10 dB by 68.2% (from 0.985 to 0.313). It indicates that even an adversarial attack with a small perturbation may lead to drastic degradation for deep learning-based AMC; the adversarial robustness becomes a critical issue. Data augmentation mitigates the decrease in accuracy except for too low SNR (-2 dB), and the degree stands out as SNR increases. For instance, Fig.13(d) shows that the UniQGAN-augmented classifier's accuracy drop is 44.0% (from 0.989 to 0.554) under FGSM with a perturbation of 0.05 at 10 dB, which means an improvement in adversarial robustness of 35.48% compared to the Original whose drop is 68.2%. Since the PGD is an improved version of FGSM, adversarial robustness enhancement is less noticeable under PGD than under FGSM. As shown in Fig.14(d), the UniQGAN-augmented classifier's accuracy drop is 59.7% (from 0.989 to 0.399) under PGD with perturbation of 0.05 at 10 dB, while Original's accuracy drop is 74.8% (from 0.985 to 0.248), indicating UniQGAN improves adversarial robustness by 20.19% against PGD.

We also observe that except for -2 dB, UniQGAN* tends to outperform UniQGAN in terms of adversarial robustness as epsilon increases. As shown in Fig.13(c), for example, the performance gap emerges since the epsilon of 0.05. Recall that the generated data from UniQGAN* lacks diversity due to mode collapse as described in 6.2. Although the diversity-limited training data may cause overfitting (i.e., reducing generalization ability and accuracy) to the classifier, the overfitting can improve the adversarial robustness [55]. Since there is a trade-off between generalization ability and adversarial robustness, we can choose the proper model based on the given scenario.

Visualization of Adversarial Examples. Fig.15 presents visualized adversarial examples of I/Q diagrams about six modulation types at 10 dB, generated by FGSM with varying epsilons from 0.0 to 0.3 with a 0.05 interval. Although we can perceive the added noise visually at the right column (0.3), a small perturbation of 0.05 is difficult to distinguish the difference from the left column (0.0). Remind that even the epsilon of 0.05 causes performance degradation by

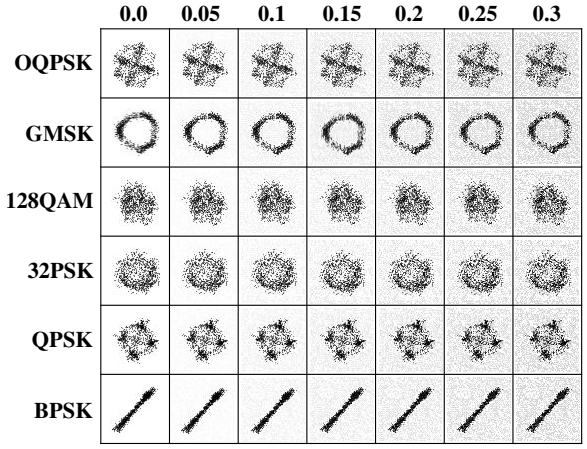


Fig. 15. Constellation diagrams at 10 dB, attacked by FGSM with varying epsilons. In the left column, 0.0 means no attack.

68.2%. The adversarial robustness and better AMC accuracy are crucial issues, and UniQGAN may be an effective option for them.

7 Conclusion

We have proposed UniQGAN, which enhances AMC in terms of accuracy and adversarial robustness through data augmentation. We have suggested multi-condition embedding and multi-domains classification techniques for scalable generator design at diverse SNRs. Experimental results showed that UniQGAN improves AMC average accuracy by 6.13% for low SNRs and 2.87% for all SNRs while reducing GAN training time by at least 75%. Through scalability analysis for various SNR ranges, we observed that shared information between SNRs can contribute to classification accuracy and GAN training time. We have also observed UniQGAN improves adversarial robustness by 35.48% after data augmentation, at 10 dB under FGSM attack with a perturbation of 0.05. In future work, we plan to analyze the shared information between SNRs theoretically for a more thorough investigation.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (Ministry of Science & Information and Communication Technology) (No. 2019R1A2C2088812).

References

[1] I. Lee and W. Lee, "Uniqgan: Unified generative adversarial networks for augmented modulation classification," *IEEE Communications Letters*, vol. 26, no. 2, pp. 355–358, 2022.

[2] Z. Zhu and A. K. Nandi, *Automatic modulation classification: principles, algorithms and applications*. John Wiley & Sons, 2015.

[3] S. Huang, C. Lin, W. Xu, Y. Gao, Z. Feng, and F. Zhu, "Identification of active attacks in internet of things: joint model-and data-driven automatic modulation classification approach," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 2051–2065, 2020.

[4] S. Huang, R. Dai, J. Huang, Y. Yao, Y. Gao, F. Ning, and Z. Feng, "Automatic modulation classification using gated recurrent residual network," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7795–7807, 2020.

[5] J. L. Xu, W. Su, and M. Zhou, "Likelihood-ratio approaches to automatic modulation classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 4, pp. 455–469, 2010.

[6] A. Swami and B. M. Sadler, "Hierarchical digital modulation classification using cumulants," *IEEE Transactions on communications*, vol. 48, no. 3, pp. 416–429, 2000.

[7] S. U. Pawar and J. F. Doherty, "Modulation recognition in continuous phase modulation using approximate entropy," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 843–852, 2011.

[8] F. Meng, P. Chen, L. Wu, and X. Wang, "Automatic modulation classification: A deep learning enabled approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10 760–10 772, 2018.

[9] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The roadmap to 6g: Ai empowered wireless networks," *IEEE communications magazine*, vol. 57, no. 8, pp. 84–90, 2019.

[10] Z. Zhou, L. Liu, S. Jere, J. Zhang, and Y. Yi, "Rcnet: Incorporating structural information into deep rnn for online mimo-ofdm symbol detection with limited training," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 3524–3537, 2021.

[11] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 3, pp. 433–445, 2018.

[12] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Proc. of International Conference on Engineering Applications of Neural Networks (EANN)*, 2016, pp. 213–226.

[13] S. Kojima, K. Maruta, Y. Feng, C.-J. Ahn, and V. Tarokh, "Cnn-based joint snr and doppler shift classification using spectrogram images for adaptive modulation and coding," *IEEE Transactions on Communications*, vol. 69, no. 8, pp. 5152–5167, 2021.

[14] Y. Guo, L. Yu, Q. Wang, T. Ji, Y. Fang, J. Wei-Kocsis, and P. Li, "Weak signal detection in 5g+ systems: A distributed deep learning framework," in *Proc. of ACM MobiHoc*, 2021.

[15] S. Peng, H. Jiang, H. Wang, H. Alwageed, Y. Zhou, M. M. Sebdani, and Y.-D. Yao, "Modulation classification based on signal constellation diagrams and deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 3, pp. 718–727, 2018.

[16] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Madry, "Adversarially robust generalization requires more data," in *Proc. of NeurIPS*, 2018.

[17] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. of ICLR*, 2015.

[18] M. Sadeghi and E. G. Larsson, "Adversarial attacks on deep-learning based radio signal classification," *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 213–216, 2018.

[19] Y. Lin, H. Zhao, Y. Tu, S. Mao, and Z. Dou, "Threats of adversarial attacks in dnn-based modulation recognition," in *Proc. of IEEE INFOCOM*, 2020.

[20] B. Kim, Y. E. Sagduyu, K. Davaslioglu, T. Erpek, and S. Ulukus, "Channel-aware adversarial attacks against deep learning-based wireless signal classifiers," *IEEE Transactions on Wireless Communications*, 2021.

[21] R. Zhang, S. Chang, Z. Wei, Y. Zhang, S. Huang, and Z. Feng, "Modulation classification of active attacks in internet of things: Lightweight mcbldn with spatial transformer network," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 19 132–19 146, 2022.

[22] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[23] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. of NeurIPS*, 2014.

[24] L. Huang, W. Pan, Y. Zhang, L. Qian, N. Gao, and Y. Wu, "Data augmentation for deep learning-based radio modulation classification," *IEEE Access*, vol. 8, pp. 1498–1506, 2019.

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

[26] M. Patel, X. Wang, and S. Mao, "Data augmentation with conditional gan for automatic modulation classification," in *Proc. of ACM Workshop on Wireless Security and Machine Learning (WiseML)*, 2020.

[27] 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. 15 713–15 722, 2018.
[28] S. Chen, Y. Zhang, Z. He, J. Nie, and W. Zhang, "A novel attention cooperative framework for automatic modulation recognition," *IEEE Access*, vol. 8, pp. 15 673–15 686, 2020.
[29] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *Proc. of ICML*, 2017.
[30] S. Moon, H. Kim, and I. Hwang, "Deep learning-based channel estimation and tracking for millimeter-wave vehicular communications," *Journal of Communications and Networks*, vol. 22, no. 3, pp. 177–184, 2020.
[31] T. J. O'Shea, N. West, M. Vondal, and T. C. Clancy, "Semi-supervised radio signal identification," in *Proc. of IEEE International Conference on Advanced Communication Technology (ICACT)*, 2017.
[32] Y. Wang, G. Gui, H. Gacanin, T. Ohtsuki, H. Sari, and F. Adachi, "Transfer learning for semi-supervised automatic modulation classification in zf-mimo systems," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, no. 2, pp. 231–239, 2020.
[33] E. Perenda, S. Rajendran, G. Bovet, S. Pollin, and M. Zheleva, "Learning the unknown: Improving modulation classification performance in unseen scenarios," in *Proc. of IEEE INFOCOM*, 2021.
[34] 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 Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3133–3143, 2020.
[35] A. Jagannath and J. Jagannath, "Multi-task learning approach for modulation and wireless signal classification for 5g and beyond: Edge deployment via model compression," *arXiv preprint arXiv:2203.00517*, 2022.
[36] D. Hong, Z. Zhang, and X. Xu, "Automatic modulation classification using recurrent neural networks," in *Proc. of IEEE International Conference on Computer and Communications (ICCC)*, 2017.
[37] Y. Wang, G. Gui, T. Ohtsuki, and F. Adachi, "Multi-task learning for generalized automatic modulation classification under non-gaussian noise with varying snr conditions," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 3587–3596, 2021.
[38] Y. Wang, J. Yang, M. Liu, and G. Gui, "Lightamc: Lightweight automatic modulation classification via deep learning and compressive sensing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3491–3495, 2020.
[39] S. Hu, Y. Pei, P. P. Liang, and Y.-C. Liang, "Deep neural network for robust modulation classification under uncertain noise conditions," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 564–577, 2019.
[40] Y. Wang, M. Liu, J. Yang, and G. Gui, "Data-driven deep learning for automatic modulation recognition in cognitive radios," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 4074–4077, 2019.
[41] P. Huang, X. Zhang, S. Yu, and L. Guo, "Is-wars: Intelligent and stealthy adversarial attack to wi-fi-based human activity recognition systems," *IEEE Transactions on Dependable and Secure Computing*, 2021.
[42] N. Wang, Y. Chen, Y. Xiao, Y. Hu, W. Lou, and T. Hou, "Manda: On adversarial example detection for network intrusion detection system," *IEEE Transactions on Dependable and Secure Computing*, 2022.
[43] C. Chen, J. Zhang, X. Xu, L. Lyu, C. Chen, T. Hu, and G. Chen, "Decision boundary-aware data augmentation for adversarial training," *IEEE Transactions on Dependable and Secure Computing*, 2022.
[44] S. Shehnepoor, R. Togneri, W. Liu, and M. Bennamoun, "Scoregan: A fraud review detector based on regulated gan with data augmentation," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 280–291, 2021.
[45] A. Al-Shawabka, P. Pietraski, S. B. Pattar, F. Restuccia, and T. Melodia, "Deeplora: Fingerprinting lora devices at scale through deep learning and data augmentation," in *Proc. of ACM MobiHoc*, 2021.
[46] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. of ICLR*, 2015, pp. 189–199.
[47] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. of ICLR*, 2018, pp. 1–23.
[48] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.
[49] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. of CVPR*, 2016, pp. 2818–2826.
[50] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Proc. of NeurIPS*, 2016.
[51] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.
[52] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.
[53] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, 2018.
[54] M.-I. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig *et al.*, "Adversarial robustness toolbox v1. 0.0," *arXiv preprint arXiv:1807.01069*, 2018.
[55] L. Rice, E. Wong, and Z. Kolter, "Overfitting in adversarially robust deep learning," in *Proc. of ICML*, 2020, pp. 8093–8104.

Insup Lee (S'20) received B.S degree in cyber defense from Korea University, Seoul, Republic of Korea, in 2018, where he is currently pursuing his Ph.D. degree in information security. He is a cybersecurity researcher at Cyber & Network Technology Center, Agency for Defense Development, Seoul, Republic of Korea. His research interests include deep learning, intelligent communication, generative adversarial networks, cybersecurity, and machine learning-based network security.

Wonjun Lee (M'00–SM'06–F'21) received the B.S. and M.S. degrees in computer engineering from Seoul National University, Seoul, Republic of Korea, in 1989 and 1991, respectively, the M.S. degree in computer science from the University of Maryland, College Park, MD, USA, in 1996, and the Ph.D. degree in computer science and engineering from the University of Minnesota, Minneapolis, MN, USA, in 1999. In 2002, he joined the faculty of Korea University, Seoul, Republic of Korea, where he is currently a professor in the School of Cybersecurity. His research interests include communication and network protocols, wireless communication and networking optimization techniques, security and privacy in mobile computing, and RF-powered computing and networking. He has authored 15 international patents, over 250 papers in refereed international journals and conferences, and a book "Optimal Coverage in Wireless Sensor Networks," Springer, 2020 (with. D.-Z. Du). He has served on program and organization committees of numerous leading wireless and networking conferences, including IEEE INFOCOM from 2008 to 2023, PC Track Chair of IEEE ICDCS 2019, Workshop Chair of IEEE ICDCS 2023, ACM MobiHoc from 2008 to 2009, and over 148 international conferences. He has received numerous awards, including IEEE Chester W. Sall Memorial Award (2018), KIISE Gaheon Research Award (2011), LG Yonam Foundation Overseas Faculty Member Award (2007), Best Teaching Award (2005, 2009, 2021) from Korea University, and the Best Paper Awards from IEEE ICOIN 2002, ICOIN 2008, and IEEE SocialCom 2016. In 2019, his project "BackPlugged: Wearable-optimized Ultra Low-Power Wi-Fi Networking with Plugged-in Backscatter Radio" was selected for 100 Outstanding National R&D Achievements by the Ministry of Science and ICT (MSIT) in Korea. He was a recipient of the Korean Government Overseas Scholarship between 1993 and 1996. He is the 2022 President-Elect (2023 President) of the Korean Institute of Information Scientists and Engineers (KIISE). He is a Fellow of the IEEE and KAST (Korean Academy of Science and Technology).