



Article

Zero-Shot SAR Target Recognition Based on a Conditional Generative Network with Category Features from Simulated Images

Guo Chen , Siqian Zhang *, Qishan He, Zhongzhen Sun , Xianghui Zhang and Lingjun Zhao

College of Electronic Science and Technology, National University of Defense Technology, Changsha 410073, China; chenguo@nudt.edu.cn (G.C.); heqishan15@nudt.edu.cn (Q.H.); sunzhongzhen14@nudt.edu.cn (Z.S.); zhangxianghui18@nudt.edu.cn (X.Z.)

* Correspondence: zhangsiqian@nudt.edu.cn

Abstract: SAR image target recognition relies heavily on a large number of annotated samples, making it difficult to classify the unseen class targets. Due to the lack of effective category auxiliary information, the current zero-shot target recognition methods for SAR images are limited to inferring only one unseen class rather than classifying multiple unseen classes. To address this issue, a conditional generative network with the category features from the simulated images for zero-shot SAR target recognition is proposed in this paper. Firstly, the deep features are extracted from the simulated images and fused into the category features that characterize the entire class. Then, a conditional VAE-GAN network is constructed to generate the feature instances of the unseen classes. The high-level semantic information shared in the category features aids in generalizing the mapping learned from the seen classes to the unseen classes. Finally, the generated features of the unseen classes are used to train a classifier that can classify the real unseen images. The classification accuracies for the targets of the three unseen classes based on the proposed method can reach $99.80 \pm 1.22\%$ and $71.57 \pm 2.28\%$ with the SAMPLE dataset and the MSTAR dataset, respectively. The advantage and validity of the proposed architecture are indicated with a small number of the seen classes and a small amount of the training data. Furthermore, the proposed method can be extended to generalized zero-shot recognition.



Citation: Chen, G.; Zhang, S.; He, Q.; Sun, Z.; Zhang, X.; Zhao, L. Zero-Shot SAR Target Recognition Based on a Conditional Generative Network with Category Features from Simulated Images. *Remote Sens.* **2024**, *16*, 1930. <https://doi.org/10.3390/rs16111930>

Academic Editor: Dusan Gleich

Received: 22 March 2024

Revised: 17 May 2024

Accepted: 24 May 2024

Published: 27 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: synthetic aperture radar (SAR); zero-shot recognition; category features; feature learning; conditional generative network

1. Introduction

Synthetic Aperture Radar (SAR) is a remote sensing technology that uses microwave signals to observe the Earth's surface. Unlike optical remote sensing techniques, SAR has the advantage of being able to operate in all weather conditions and at all times of day [1]. SAR images are widely used in various fields due to their unique advantages. SAR Automatic Target Recognition (SAR-ATR) is a crucial component of SAR image processing techniques and plays a significant role in both civil and military applications. A standard SAR ATR system consists of three main components: detection, discrimination, and classification [2]. The classification of targets in SAR has become a challenging research issue.

SAR target classification methods can be categorized into the traditional and the deep learning-based methods. The traditional method comprises the template matching method [3,4], the model-based method [5], and the machine learning-based method [6]. Among these, the machine learning-based method exhibits a fast processing speed, a high recognition capability and a robust performance. However, it relies on a manually designed feature extractor, which requires specialized knowledge and experience. Compared to the traditional method, the deep learning-based approach has the advantage of automatically extracting features, which provides stronger generalization capabilities and enhances

recognition performance. The use of neural networks for image classification has rapidly developed since the first application of the AlexNet network [7]. The convolutional neural network (CNN) models, such as VggNet [8] and ResNet [9], have achieved excellent performance in this field. Additionally, network architectures like recurrent neural networks (RNN) [10] and graph neural networks (GNN) [11] can also play a unique role in the image recognition task. The deep learning methods have demonstrated an outstanding performance in SAR image recognition [12–16]. However, deep learning methods are data-driven and typically require a large amount of data for effective feature extraction. When the data are scarce, the recognition capability of the network significantly decreases. It is difficult to obtain large quantities of SAR images due to various limitations such as technological and policy constraints. Thus, the issue of the small sample size is a prominent concern in the field of SAR image target recognition. There are various methods to address the issue of the small sample size. Expanding the training dataset is a common approach [17–21]. On the other hand, the transfer learning-based methods [22–24] and meta-learning methods [25,26] are also effective strategies. While the methods mentioned above can address the issue of scarce data, they are unable to recognize the unseen class targets that lack any training data.

In real-world scenarios, it is often impossible to acquire SAR images of non-cooperative military facilities and other specific targets in advance for network training. Therefore, there is an urgent need for the ability to recognize the targets appearing for the first time. The issue of identifying the unseen classes with no training data is referred to as zero-shot recognition, which is an extreme case of the few-shot recognition problem. Figure 1 illustrates the learning approaches under the conditions of sufficient samples, limited samples, and zero samples. Four types of vehicle targets from the MSTAR SAR image dataset [27], 2S1, BMP2, BTR70, and T72, are used as examples. In cases where there are sufficient samples, a large amount of the labeled data for the 2S1 and BMP2 target classes can be used to train the network. However, in few-shot learning scenarios, the network can only be trained with a small number of samples. In zero-shot learning scenarios, it is impossible to obtain the data for the BTR70 and T72 target classes in advance for training. The network trained with the 2S1 and BMP2 requires to recognize the unseen classes such as BTR70 and T72. The concept of zero-shot learning was first proposed by Larochelle et al. [28], typically referring to training learning models based on the data from the seen classes and the prior auxiliary information, then the training enables the model to recognize the data from the unseen classes. The methods for solving zero-shot recognition include the direct semantic prediction methods [28,29], embedding model-based methods [30–33], and visual sample generation methods [34–37].

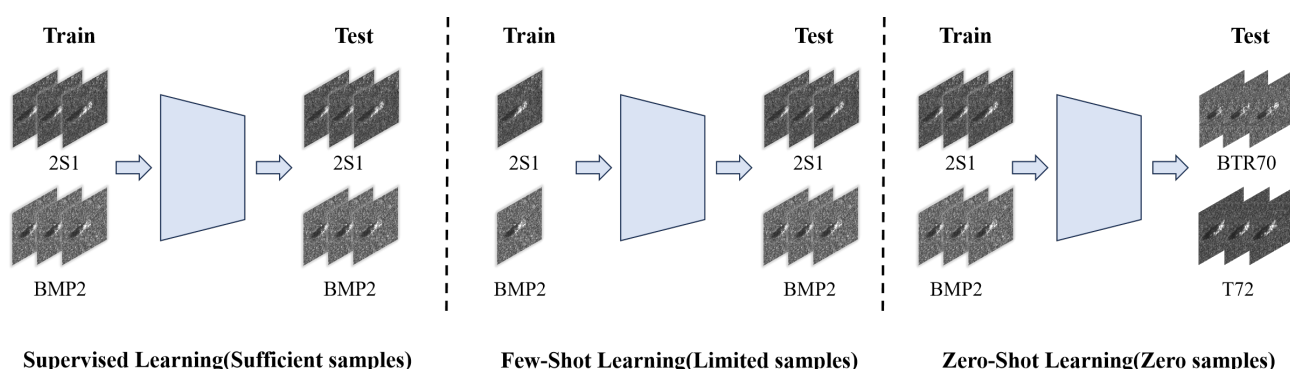


Figure 1. Different types of network learning methods.

Although zero-shot recognition has been widely studied in the natural image domain, the research on zero-shot target recognition in SAR images is quite limited. Some studies [38–40] draw inspiration from the zero-shot recognition methods in the optical domain. The semantic information such as label one-hot encoding or attribute information is constructed to assist in zero-shot recognition for SAR images. During the training phase,

the seen class data are used to construct an embedding space. During testing, the images from unseen classes are mapped to this embedding space, and their relationships are inferred by measuring the distances between embedding points. Reference [38] proposed a SAR image zero-shot recognition method to explore the relationships between the unseen class target T72 and the seven seen class targets based on a two-dimensional embedding space. Reference [39] proposed an architecture consisting of two autoencoders that utilize the reflection information of SAR images to assist in constructing the embedding space. Reference [40] utilized data mining techniques to annotate binary 10-dimensional attribute information for SAR targets and employed a classifier to assist in constructing the embedding space. The main challenge faced by the aforementioned methods is that the manually designed semantic information is too simple to effectively characterize SAR targets. The names of SAR targets, such as 2S1 and T72, are merely symbols composed of characters without actual semantic meaning. Defining effective class attribute information is also a challenging task. Because of the difficulty in assuring the quality of semantic information and the absence of supervised learning, the embeddings of unseen classes frequently encounter significant domain shift issues and become discrete. The embedding model-based method can only infer one unseen class at a time, rather than completing the recognition of multiple unseen class targets.

SAR electromagnetic simulated images generated from CAD models of targets have been used for SAR target recognition due to their provision of many target details similar to real images (the real images from the real measurement of a radar system) [23,41,42]. In certain studies, the networks are trained on fully simulated images and tested on real images [43–46]. However, the distributional differences between the simulated and real images limit the transferability of the model between these two types of images. The generative networks such as generative adversarial networks (GANs) [47] have been used to convert simulated data into pseudo-real data (the generated data approximating real images) that more closely resemble the distribution of real images in the case of sufficient or small samples [48,49]. The generated pseudo-real data can serve as more effective training data for training the classification network. However, these generation networks cannot generate data for unseen classes in a zero-shot setting because the real images of unseen classes cannot participate in training [46].

To classify the unseen targets with multiple classes, a conditional generative network with category features from simulated images for zero-shot SAR target recognition is proposed for generating pseudo-real data of unseen classes for supervised learning. The feature extraction network called CANet was first designed to extract deep features from both the simulated and real SAR images. The features from the simulated images are fused into the category features which serve as auxiliary information to represent the characteristics of the entire category. A conditional VAE-GAN network is then constructed. The network utilizes the category features of the seen classes as the conditions to learn the mapping relationship between these features and the deep features of the real SAR images. This mapping relationship can be extended to generate the features for the unseen classes. Ultimately, the generated features of the unseen classes are utilized to train a classifier for the classification of the real SAR images. The main contributions of the article are as follows:

- The category features constructed from the simulated images are proposed. The feasibility of these category features utilized as the category auxiliary information for SAR zero-shot learning is verified.
- A framework for zero-shot generation of SAR data based on the conditional VAE-GAN is proposed. The network establishes a connection between the seen and the unseen class data through the category features. By learning the mapping from the category features to the real data using the seen class data, it can generate the unseen class data.
- Compared to the embedding model-based methods assisted by the semantic information, our architecture can recognize multiple unseen class SAR targets instead of inferring a single one.

The rest of this article is organized as follows. Section 2 provides a detailed overview of the method, including the complete architecture and the entire training and testing processes. Section 3 presents the experimental results of our method and further analyzes the impact of various factors on our approach, besides, a generalized zero-shot recognition extension experiment is conducted. In Section 4, the limitations of the proposed method and directions for future improvements are discussed. Finally, Section 5 summarizes the paper.

2. Methodology

Figure 2 shows the overall architecture of the proposed method, which comprises three parts: the feature extraction module (the red dashed box), the feature generation module (the green dashed box), and the classification module (the yellow dashed box). The feature extraction module is the first stage of the overall architecture. The real features and the category features are extracted using the pre-trained CANet in this stage. Then, the feature generation module is the second stage. The real features and the category features of seen classes extracted in the first stage are used to train the feature extraction network. The network learns the mapping from the category features to the real features. The last stage is the classification recognition module. The category features of unseen classes, extracted in the first stage, are used to generate the pseudo-real features of unseen classes using the learned mapping from the second stage. The generated features are used to train a classifier to recognize the real images. The specific modules of the proposed method and the detailed training and testing procedures are explained in this section. Section 2.1 provides the symbolic representation of zero-shot target recognition. Sections 2.2–2.4 introduce the feature extraction module, the feature generation model, and the classification module, respectively. Section 2.5 presents the detailed processes of the training and the testing.

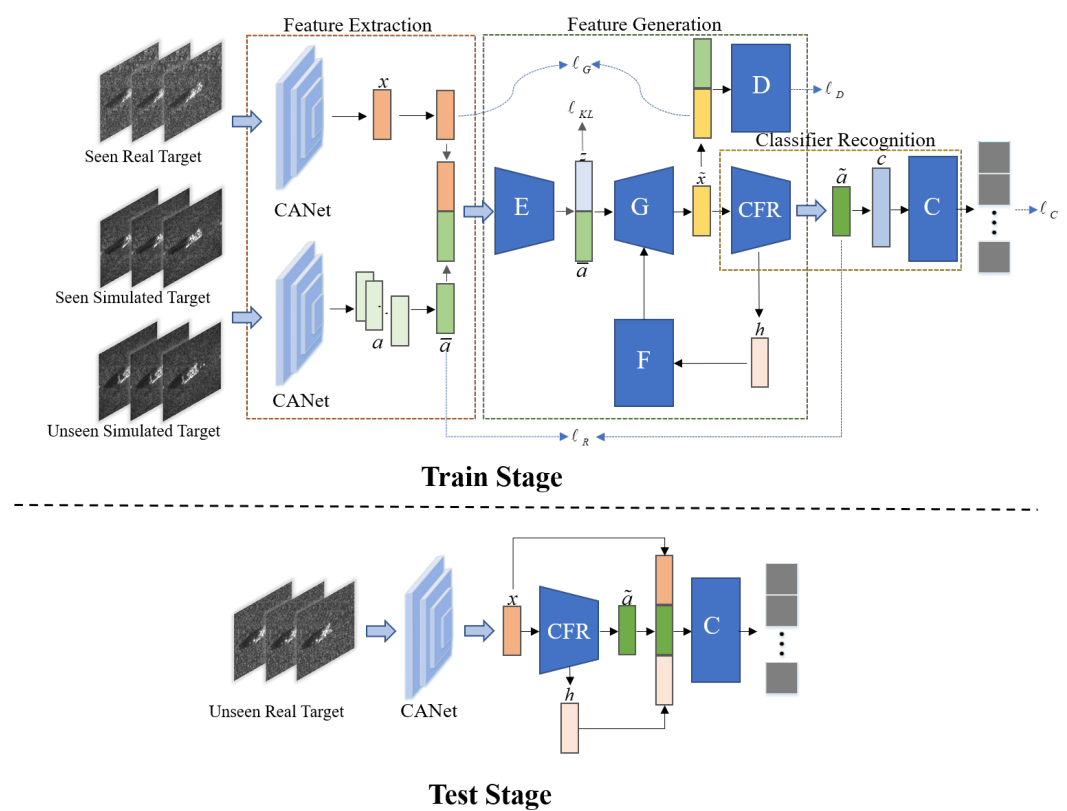


Figure 2. The overall architecture of the proposed method.

2.1. Symbolic Representation of Zero-Shot Target Recognition

Suppose there are a total of k target categories, with m categories as the seen classes, and n categories as the unseen classes. The set of the seen class samples can be represented as $S = \{x^s, y^s, a^s \mid x^s \in X^s, y^s \in Y^s, a^s \in A^s\}$, where x^s represents the samples from the seen classes, $Y^s = \{y_1^s, y_2^s, \dots, y_m^s\}$ represents the labels for the seen classes, and $A^s = \{a_1^s, a_2^s, \dots, a_m^s\}$ represents the auxiliary information. The set of the unseen class samples can be represented as $U = \{y^u, a^u \mid y^u \in Y^u, a^u \in A^u\}$, where $Y^u = \{y_1^u, y_2^u, \dots, y_n^u\}$ represents the labels for the unseen classes, and $A^u = \{a_1^u, a_2^u, \dots, a_n^u\}$ represents the auxiliary information for the unseen classes. $m + n = k$, $Y^u \cap Y^s = \emptyset$. The zero-shot learning task in this paper involves training a network using S to learn the mapping $A^s \rightarrow \tilde{X}^s$, with $\tilde{X}^s$ being the generated pseudo-real data. The network can then achieve $A^u \rightarrow \tilde{X}^u$, with $\tilde{X}^u$ being the generated pseudo-real data of the unseen classes. Finally, a classifier $f_{zsl} : \{\tilde{X}^u \rightarrow Y^u\}$ can be trained using $\tilde{X}^u$ to classify the unseen class real images.

2.2. The Feature Extraction Module

Feature extraction is the first stage of the overall architecture, consisting of extracting the real features from the real images and the category features from the simulated images. The extracted features are then used to support the subsequent learning of the feature mappings.

2.2.1. Extraction of the Real Features

The feature extraction module called CANet designed in this paper represents an improvement over A-ConvNet [18] which replaces fully connected layers with sparsely connected convolutional structures. First, the CANet kept the first 4 layers of the A-ConvNet to preserve the feature extraction capability. Second, the additional fifth and sixth layer are added to extract features with a shape of $1 \times 1 \times 256$ to facilitate the training of the subsequent feature generation network. The specific structure of CANet is illustrated in Figure 3. The network is composed of six layers. The first three layers consist of a combination of a convolution module, ReLU activation function and maxpooling module. After the fourth layer, a Dropout operation is applied. The design of the first four layers of the network is based on A-ConvNet. On this basis, the last two separate convolution layers have been added to extract effective features. The input image size is initially 128×128 , and it is resized to $88 \times 88 \times 3$ before being fed into the network. The feature size after the fourth layer is $3 \times 3 \times 128$. The convolutional kernel size of the fifth layer is designed as $3 \times 3 \times 256$, resulting in a $1 \times 1 \times 256$ feature output of the network. The 256-dimensional feature reduces the complexity of subsequent model training while maintaining the representation capability for the SAR image target. The kernel size of the sixth layer is designed as $1 \times 1 \times n$, enabling it to output the probabilities of n categories for the loss calculation and the gradient backpropagation.

The CANet network is pre-trained to extract the real features from the real images. Only the real images of the seen classes are used to train the network, as the real images of the unseen classes cannot participate in training. The real features of the seen classes are extracted for training in the next stage.

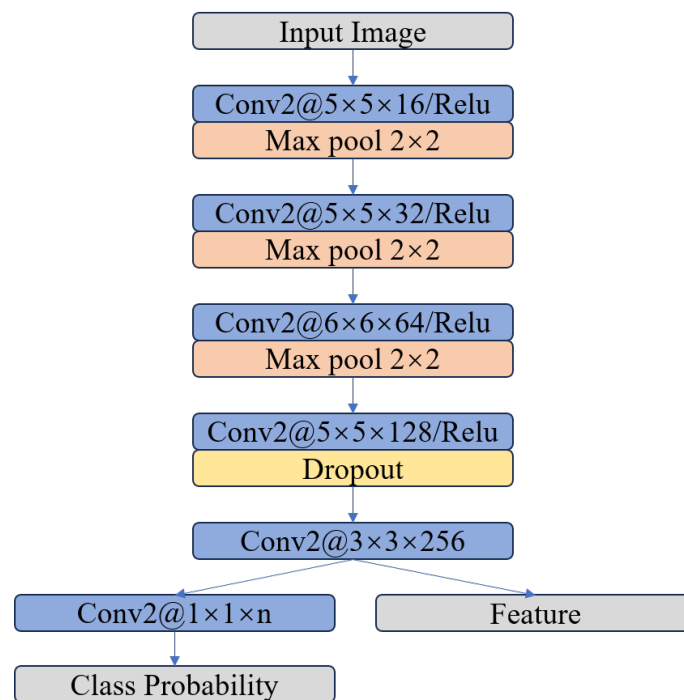


Figure 3. The specific structure of CANet.

2.2.2. Extraction of Category Features

The existing research on generating real images from simulated images has primarily focused on one-to-one mapping [48,49]. In these works, the generative network maps each simulated image to a unique pseudo-real image corresponding to it, which limits the generation of the pseudo-real data. A conditional generative network [50,51] is used in this paper to achieve a one-to-many generation. Therefore, a category feature is constructed from the simulated images for each class and used as the condition to generate the multiple pseudo-real features.

The simulated images are used to extract the category feature. The extraction process is illustrated in Figure 4. First, CANet is pre-trained using real images of the seen classes. Then, for all simulated images in the same class, the pre-trained CANet is used to extract simulated features a_{ij} . Finally, these simulated features are fused into the category feature $\bar{a}_i$. To ensure that a category feature accurately reflects the distinctive features of a given class, we define it as the average of all simulated features of that class. The approach is similar to the prototype network used in the few-shot learning [52]:

$$\bar{a}_i = \frac{1}{k_i} \sum_{j=1}^{k_i} a_{ij} \quad (1)$$

where k_i represents the number of the simulated images for the class i . The category features of the seen classes are extracted for training in the next stage. Meanwhile, the category features of the unseen classes are extracted to generate pseudo-real features for the unseen classes.

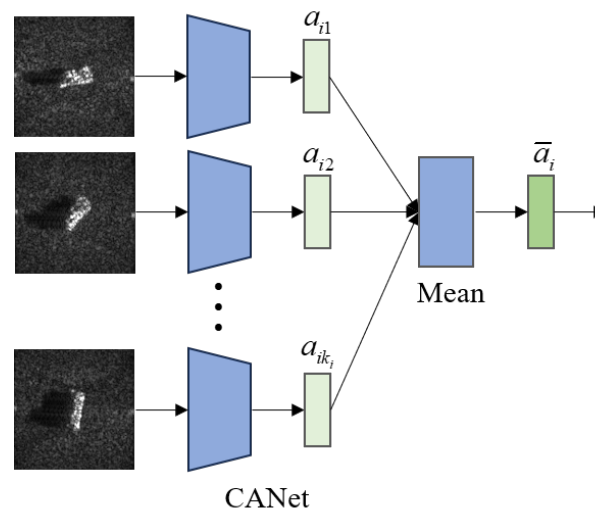


Figure 4. The process of extracting category features.

2.3. The Feature Generation Module

The learning of the feature generation is the second stage of the overall architecture. The goal is to learn the mapping relationship between the category features and the real features of the seen classes, allowing for better generalization to the generation of the unseen classes. A conditional generation network is created by integrating a VAE network and a GAN network. Additionally, a category feature reconstructor and a feedback module are integrated to enhance the generation capabilities of the network. The generation module learns the mappings at the deep feature level without focusing on the image reconstruction. Therefore, it is constructed only with shallow fully connected layers.

Figure 5 shows the composition of the feature generation module, where x is the real features and a represents the category features. The conditional variational autoencoder (CVAE) is formed by conditioning the encoder $E(x, a)$ and generator $G(z, a)$ on a , while the conditional generative adversarial network (CGAN) is formed by conditioning $G(z, a)$ and the discriminator $D(x, \tilde{x}, a)$ on a . The CVAE and CGAN share a generator, forming a conditional VAE-GAN (CVAE-GAN) [53]. The VAE-GAN combines the advantages of the latent space encoding of the VAE with the high-quality feature generation of the GAN, resulting in more stable training and the ability to generate more smoothly varying pseudo-real features.

For the feature generation part of the CVAE, we use the category feature a as a condition and concatenate the real feature x as the input for $E(x, a)$. Firstly, the encoder $E(x, a)$ encodes the real feature into a low-dimensional vector z in the continuous latent space. Assuming z follows an isotropic Gaussian distribution, the output of E is the mean vector and variance vector of the Gaussian distribution, denoted as (μ_x, σ_x) . Then, the generator G decodes z and produces the fake feature $\tilde{x}$. Finally, ensuring the cyclic consistency of the generated features with the original features is achieved by minimizing the difference between $\tilde{x}$ and x . CVAE is optimized through the following loss function:

$$\begin{aligned}\ell_{VAE} &= \ell_{KL} + \ell_G \\ \ell_{KL} &= KL(q(z|x, a) \parallel p(z|a)) \\ \ell_G &= -E_{q(z|x, a)}[\log p(\tilde{x} | z, a)]\end{aligned}\quad (2)$$

where the conditional distribution $q(z | x, a)$ represents the probability distribution modeled by $E(x, a)$, while $p(z|a)$ is the prior distribution following $N(0, 1)$. ℓ_{KL} refers to the Kullback–Leibler divergence between $q(z | x, a)$ and $p(z | a)$. $p(\tilde{x} | z, a)$ is equal to $G(z, a)$. ℓ_G is the generation reconstruction loss of VAE, which is set as the cross-entropy loss between the generated feature and the original feature.

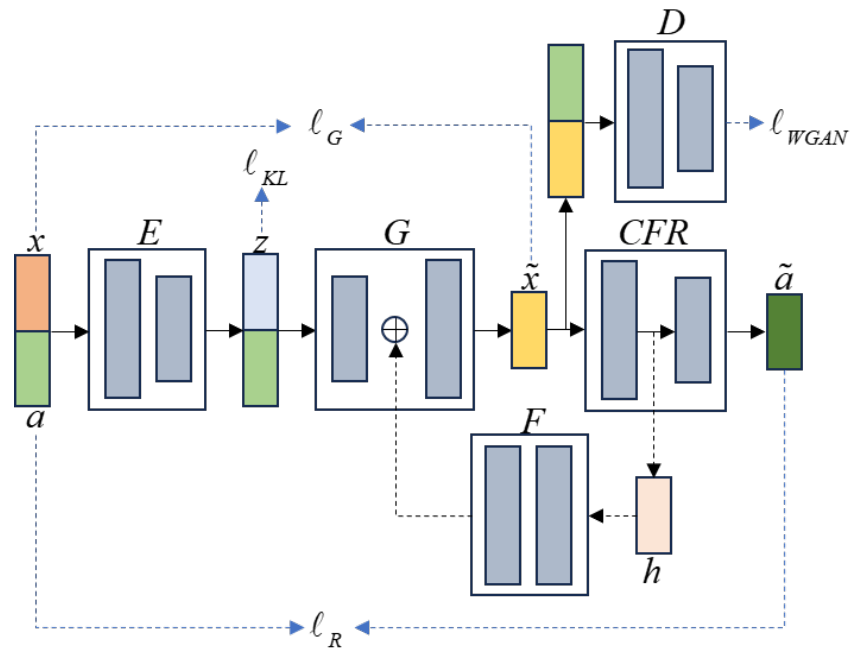


Figure 5. The specific structure of the feature generation module.

For the feature generation part of CGAN, the generator is identical to the generator G of CVAE. The discriminator $D(x, \tilde{x}, a)$ takes the real feature, the fake feature and condition a as input, and outputs a real number indicating the authenticity of the input feature. CGAN is optimized through the improved WGAN loss [54]:

$$\ell_{WGAN} = E[D(x, a)] - E[D(\tilde{x}, a)] - \lambda E[(\|\nabla_{\tilde{x}} D(\hat{x}, a)\|_2 - 1)^2] \quad (3)$$

where $\tilde{x} = G(x, a)$, $\hat{x} = ax + (1 - a)\tilde{x}$, here $a \sim U(0, 1)$. The penalty term coefficient is represented by λ . The first two terms in the formula approximate the Wasserstein distance between the distribution of the generated features and the real features, and the last term is the gradient penalty term that forces the gradient at any point to be close to the unit norm. The improved WGAN loss enhances training smoothness and stability, mitigating model collapse issues during traditional GAN network training. The CVAE-GAN optimization objective combines the losses from both the CVAE and CGAN parts:

$$\ell_{VAE-GAN} = \min_{E, G} \max_D \ell_{VAE} + \alpha \ell_{WGAN} \quad (4)$$

where α is a hyperparameter, for specific details please refer to the literature [36]. The generation module introduces two new modules, the category feature reconstruction module (CFR) and the feedback module (F), based on VAE-GAN. These modules enhance the generative capabilities of the network. CFR reconstructs the generated fake features into the category features, learning the inverse mapping from category features to real features to ensure the semantic consistency of the generated features. The network is optimized through cycle consistency loss.

$$\ell_R = E[\|\tilde{a} - a\|_1] \quad (5)$$

where $\tilde{a} = CFR(\tilde{x})$. F provides feedback by feeding the intermediate layer embedding h from CFR into G . This allows G to iteratively improve the feature generation and obtain improved feature representations. In summary, the overall loss function of the zero-shot feature generation module is as follows:

$$\ell_{Total} = \ell_{VAE-GAN} + \beta \ell_R \quad (6)$$

where β is the weighting coefficient, with specific details please refer to in the literature [37]. After training the feature generation module, the network can generate pseudo-real features from category features of unseen classes. These features are then fed into the classification module to train the classifier.

2.4. The Classification Module

The third stage of the model involves the classification module training. The objective is to train the classifier using generated features of unseen classes, allowing it to classify real unseen class images. Figure 6 illustrates the structure of the classification module.

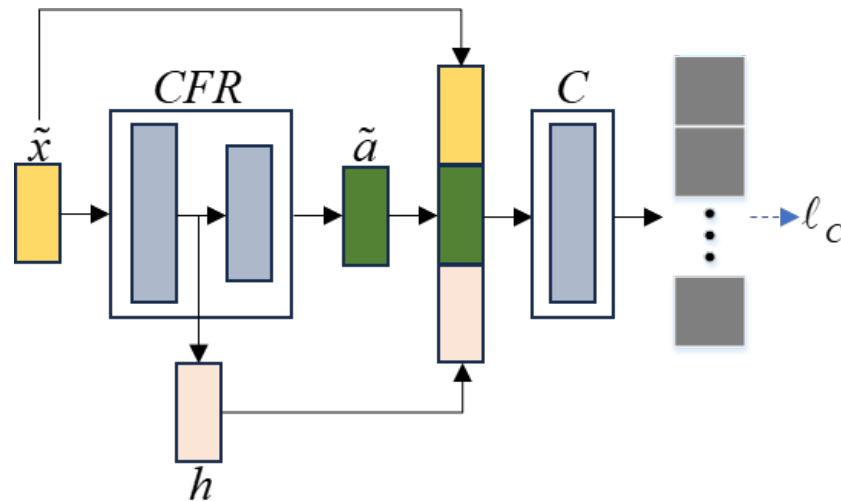


Figure 6. The specific structure of the classification module.

In the classification module, we retained *CFR*, rather than directly using the generated features $\tilde{x}$ from unseen classes to train the classifier. *CFR* is the inverse mapping of $G(z, a)$. The intermediate layer embedding h and the reconstructed category feature $\tilde{a}$ encode the complementary category information to the generated feature instances, serving as an auxiliary information source to assist in training the classifier. We concatenate $\tilde{x}$, $\tilde{a}$, and h as classification features c to train the classifier, which outputs the probability of unseen classes and is optimized through the cross-entropy loss:

$$\ell_c = -\frac{1}{|U|} \sum_{(c,y) \in U} \log p(y|c; \theta) \quad (7)$$

here, θ represents the parameters of the classifier, and $P(y|c; \theta) = \frac{\exp(\theta_y^T c)}{\sum_i^n \exp(\theta_i^T c)}$ denotes the probability of the output class. Once the classifier training is complete, the entire module is trained.

2.5. Training and Test Process

Figure 2 illustrates the complete training process of the proposed method. In the feature extraction stage, as shown in Section 2.2, the seen class features x^s are extracted. Simultaneously, the category features $\bar{a}^s$ and $\bar{a}^u$ are extracted for both the seen and the unseen classes. x^s , $\bar{a}^s$, and $\bar{a}^u$ are all 256-dimensional features. During the feature generation training stage, x^s is concatenated with the corresponding category feature $\bar{a}^s$ and inputted into the encoder E . The encoder outputs a 256-dimensional mean and variance for the latent code z . Then, z is concatenated with $\bar{a}^s$ and inputted into G , which produces the reconstructed pseudo-real feature $\tilde{x}^s$. G is composed of two fully connected layers, with the first layer having 2048 neurons and the second layer having 256 neurons as the dimension of the real feature. Next, $\tilde{x}^s$ and $\bar{a}^s$ are concatenated to train the discriminator D . The optimization of E , G , and D is conducted preliminarily through the loss $\ell_{VAE-GAN}$.

Afterward, $\tilde{x}^s$ is input into the *CFR* module to reconstruct the category features $\tilde{a}^s$. The *CFR* module comprises two fully connected layers, with the first layer having 2048 neurons. The second layer has the same number of neurons as the dimension of the category feature. The first layer of *CFR* produces a 2048-dimensional vector known as the intermediate layer feature h , which is then sent to the feedback layer F . The feature h , after passing through the feedback layer F , is weighted and fused with the feature after the first fully connected layer of the generator G . F is also a two-layer fully connected layer, with both layers consisting of 2048 neurons, the same dimension as the input intermediate layer feature. *CFR* and F are optimized through the loss ℓ_R , and ℓ_R further constraining the learning parameters of E , G , and D . During the classification training phase, the pseudo-real feature $\tilde{x}^u$ for the unseen class is generated by the feature concatenating random noise with $\tilde{a}^u$. Next, $\tilde{x}^u$ is input into *CFR* and concatenated with the output fake category feature $\tilde{a}^u$ and the intermediate feature h to train the final classifier. The parameters of the classifier are optimized through the loss function ℓ_C . During the testing phase, the pre-trained CANet should be used to extract the real features x^u of the unseen class images. These features should then be input into the trained classification module for classification recognition. The training and testing processes in this paper are integrated, with testing being conducted after each training iteration.

3. Experiments

To verify the performance of the proposed method, we first introduce the dataset settings in Section 3.1. In Section 3.2, we present the experimental results. Section 3.3 examines how the intermediate layer dimension size affects the effectiveness and efficiency of the method. In Section 3.4 we investigate the impact of different settings for the category feature and the seen class real features. In Section 3.5, we further discuss the impact of the seen class number on the model. Finally, we present extended experiments on the generalized zero-shot recognition in Section 3.6.

All the experiments are carried out with both the Matlab platform and the Pytorch framework on a NVIDIA GeForce RTX 3090 GPU card manufactured by NVIDIA Corporation, which is based in Santa Clara, California, USA.

3.1. Datasets

The experiments are conducted on the SAMPLE public dataset [55] and the MSTAR datasets. The SAMPLE public dataset is a subset of the SAMPLE dataset, consisting of SAR real images and simulated images of 10 vehicle targets. The image size is 128×128 pixels with a resolution of $0.3 \text{ m} \times 0.3 \text{ m}$, and there is a one-to-one correspondence between the simulated and the real images. The data cover a range of depression angles from 14° to 17° and a range of aspect angles from 10° to 80° . The MSTAR dataset was developed with funding from the Defense Advanced Research Projects Agency (DARPA) of the United States Department of Defense. It consists of 10 classes of vehicle targets with a resolution of $0.3 \text{ m} \times 0.3 \text{ m}$. The dataset taken at multiple depression angles (15° , 17° , 30° , 45°) over a 0° – 360° range of aspect angles. In this paper, the data from the SAMPLE public dataset with the depression angles of 16° and 17° , as well as the data from the MSTAR dataset with a depression angle of 15° , are utilized.

Two experimental groups are established: SAMPLE-MSTAR and SAMPLE-MSTAR. The category features are extracted from the simulated images of the SAMPLE public dataset, while the real images from both the SAMPLE and MSATAR datasets are used for testing. The SAMPLE-SAMPLE experimental group includes all 10 categories from the SAMPLE public dataset, while the SAMPLE-MSTAR experimental group comprises only five categories that are shared between the two datasets. Figure 7 displays the simulated and the real images for each class in the two experimental groups. The unseen classes in both experimental groups are BMP2, BTR70, and T72, which are used for testing. The seen class targets comprise the remaining seven and two classes in the respective groups, which are used for network training. Tables 1 and 2 show the specific composition.

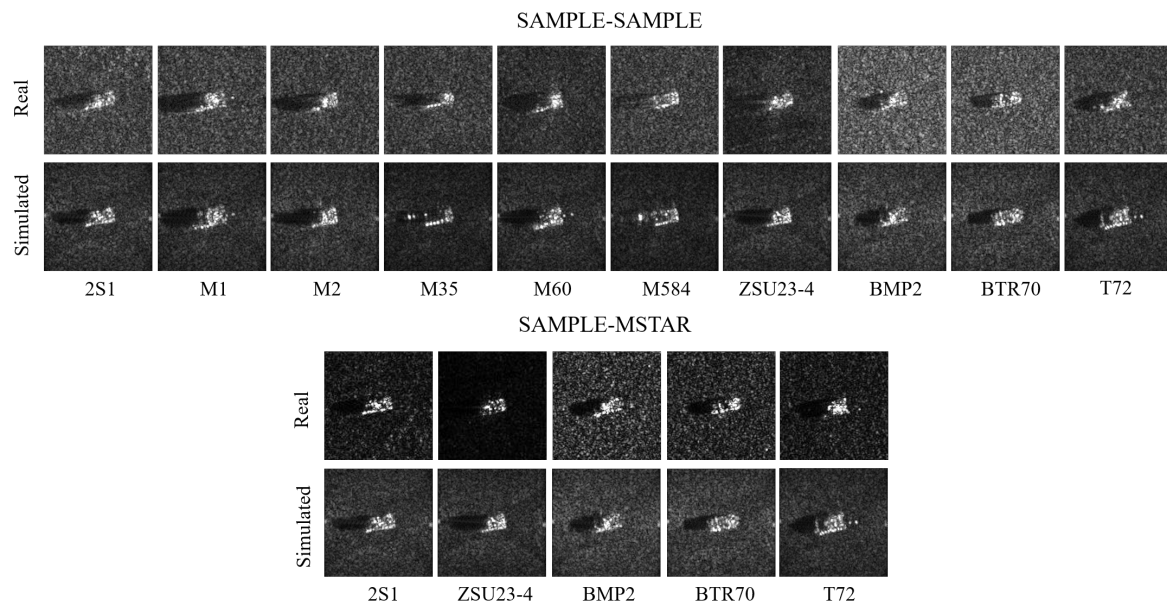


Figure 7. Examples of image pairs in the two experimental groups.

Table 1. The SAMPLE-SAMPLE experimental group.

Type	Seen Classes						Unseen Classes			
	2S1	M1	M2	M35	M60	M584	ZSU23-4	BMP2	BTR70	T72
Simulated	108	103	105	105	111	105	108	107	92	108
Real	108	103	105	105	111	105	108	107	92	108

Table 2. The SAMPLE-MSTAR experimental group.

Type	Seen Classes			Unseen Classes	
	2S1	ZSU23-4	BMP2	BTR70	T72
Simulated	108	108	107	92	108
Real	274	274	195	196	196

3.2. Effectiveness of the Method

The effectiveness of the proposed method has been tested on the two experimental groups. Ten independent training and testing runs were conducted. Tables 3 and 4 present the minimum, the maximum, and the average recognition accuracies (with standard deviations) across 10 training–testing cycles, along with the recognition rates for each of the three unseen target classes. Figures 8 and 9 depict the confusion matrix of the experiment with the highest recognition rate during the 10 training–testing cycles.

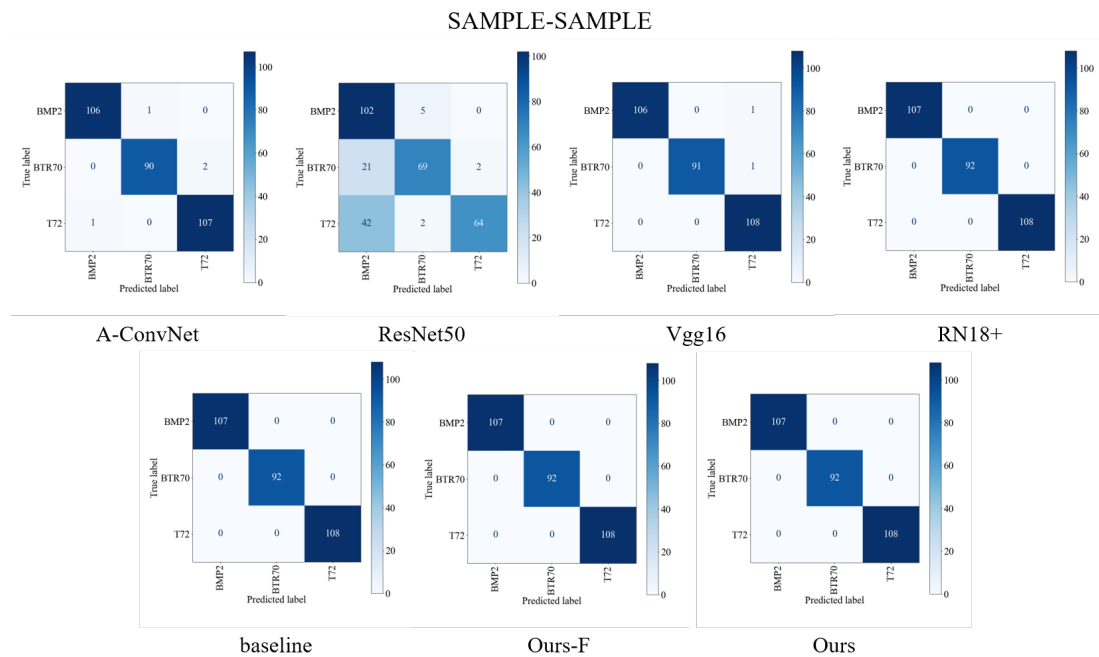
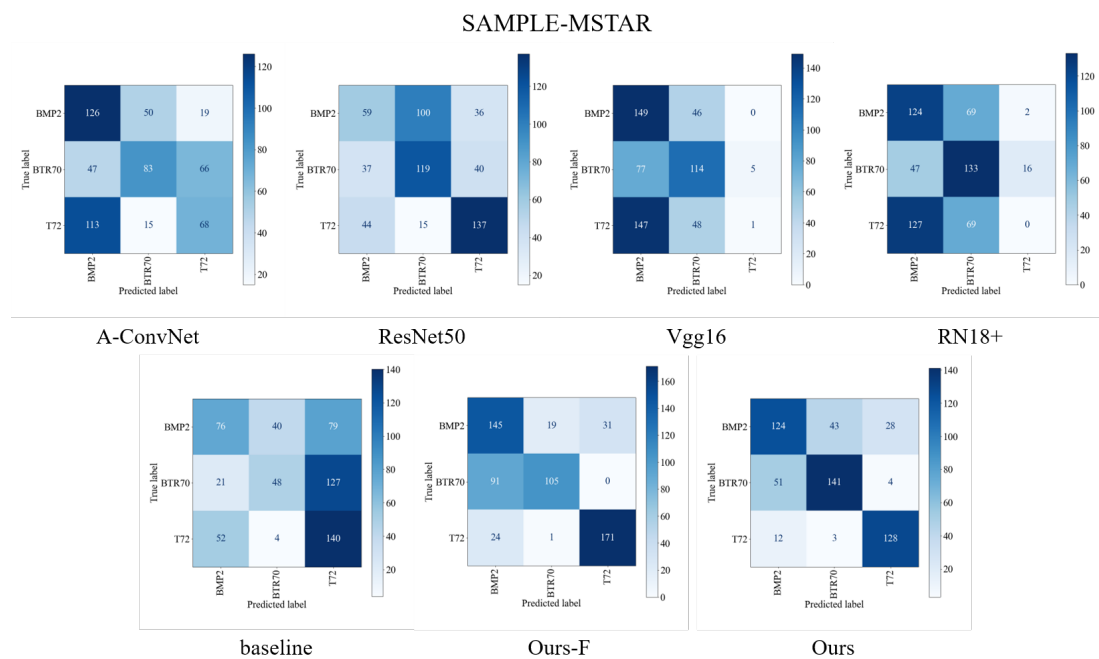
Table 3. Experimental results of SAMPLE-SAMPLE.

Methods	BMP2 (%)	BTR70 (%)	T72 (%)	Min (%)	Max (%)	Avg (%)
A-ConvNet	99.44	98.59	95.74	96.74	98.70	97.88 ± 0.69
ResNet50	76.64	55.98	75.65	66.45	76.55	70.10 ± 3.20
Vgg16	99.07	98.91	96.39	94.46	99.35	98.07 ± 1.29
RN18+	100	99.78	97.87	98.05	100	99.19 ± 0.51
baseline	98.69	98.80	98.15	97.07	100	98.53 ± 0.92
Ours -F	99.91	100	99.26	99.35	100	99.71 ± 0.23
Ours	99.81	99.89	99.72	99.35	100	99.80 ± 0.22

The highest recognition rate achieved for each column are marked in bold.

Table 4. Experimental results of SAMPLE-MSTAR.

Methods	BMP2 (%)	BTR70 (%)	T72 (%)	Min (%)	Max (%)	Avg (%)
A-ConvNet	60.98	42.50	19.39	35.95	47.19	40.92 ± 2.87
ResNet50	43.95	33.01	67.55	43.10	53.66	48.18 ± 3.67
Vgg16	60.77	63.67	2.14	38.33	44.97	42.67 ± 1.90
RN18+	69.18	54.34	0.26	38.67	42.16	41.21 ± 1.9
baseline	53.13	49.49	24.24	37.48	44.97	42.27 ± 2.13
Ours -F	47.39	65.20	90.15	64.91	71.72	67.62 ± 2.27
Ours	59.59	68.42	86.83	66.78	75.98	71.57 ± 2.28

**Figure 8.** The confusion matrix of SAMPLE-SAMPLE.**Figure 9.** The confusion matrix of SAMPLE-MSTAR.

3.2.1. The Analysis of the Experimental Results

The classic networks A-ConvNet, ResNet18, and Vgg16 were trained directly using simulated images and then tested on the real images for comparison. RN18+ [46] is a network trained exclusively on simulated data, with its training and classification strategy specifically designed based on the SAMPLE dataset. In the SAMPLE-SAMPLE experimental group, the A-ConvNet and Vgg16 networks were able to recognize the unseen class real images with an average recognition rate above 97%. RN18+ achieves a recognition rate of 99.19%, which is higher than the classic networks. The proposed method has an average recognition rate of $99.80 \pm 0.22\%$, with a minimum of 99.35%, outperforming all compared methods. For RN18+, the recognition rate for BMP2 is 100%, which is better than our method. However, it exhibits more misclassifications for the T72. Although the maximum recognition rate of RN18+ can reach 100%, the average recognition rate of the proposed method over ten 10 training–testing cycles is slightly higher than that of RN18+. In the SAMPLE-MSTAR experimental group, the classic networks achieved recognition rates distributed around 40–50%. The confusion matrix shows that the comparative methods tend to identify the three classes' targets as two of them. For example, Vgg16 and RN18+ almost exclusively identify all T72 targets as either BMP2 or BTR70. This indicates that the networks trained on the simulated images are prone to confusion when directly tested on the real images. The proposed method achieved a final recognition rate of $71.57 \pm 2.28\%$, with a maximum recognition rate of 75.68%. This represents an improvement of nearly 20–30% compared to the classic networks. The confusion matrix of the proposed method demonstrates that the proposed method correctly identifies the majority of targets in each class, indicating that it significantly reduces class confusion.

Furthermore, the ablation experiments are conducted on the overall feature generation module and the feedback module (F) within the feature generation module. The baseline represents the results of training CANet using only simulated images through cross-entropy loss and directly testing it on real images without training the subsequent feature generation network. Ours - F represents the results when the feature generation module is involved in training but the feedback module (F) is discarded. Firstly, in both experimental groups, the recognition results of the baseline are higher than those of A-ConvNet, indicating that the modification of CANet from A-ConvNet retains the feature extraction capability when adapting to the subsequent feature generation network training. Both the baseline and the proposed method achieve a maximum recognition rate of 100% in the SAMPLE-SAMPLE experimental group, while the proposed method maintains the smallest standard deviation. Secondly, the average recognition rates of Ours - F and Ours are both higher than that of the baseline, indicating that the classifier trained using the generated pseudo-real data has better generalization to real data. This validates the effectiveness of the feature generation module. Finally, the average recognition rate of Ours is higher than that of Ours - F . This indicates that the feedback module (F) effectively improves the capability of feature generation during training, resulting in enhanced feature representations.

The experimental results demonstrate that the proposed method can recognize three unseen class targets for both experimental groups. The recognition performance surpasses that of classic networks trained directly using the simulated images, demonstrating the effectiveness of the proposed method. The reason is that the feature generation architecture can create pseudo-real features for the unseen class targets, which aids in classifier training. The generated data have a stronger resemblance to the real data than the simulated data, resulting in a network with better generalization ability.

3.2.2. The Significance Test of the Experimental Results

The Wilcoxon signed-rank test is a non-parametric statistical test employed to assess the dissimilarities between two sets of related samples. In this context, the Wilcoxon signed-rank test is utilized to ascertain the statistical significance of the discrepancy between the proposed method and the comparison method in two experimental groups, as it cannot be assumed that the recognition rates of each method satisfy the normal distribution

assumption. Tables 5 and 6 show the p -values of the paired Wilcoxon test at the 95% significance level for SAMPLE-SAMPLE and SAMPLE-MSTAR. The null hypothesis of the Wilcoxon test is that there is no significant difference between the proposed method and the comparison method. When the p -value is less than 0.05, the null hypothesis is rejected, indicating that the proposed method is significantly superior to the comparison method.

Table 5. The p -value of the paired Wilcoxon test at the 95% significance level for SAMPLE-SAMPLE.

Method	A-ConvNet	ResNet50	Vgg16	RN18+	Baseline	Ours -F
p -value	9.8×10^{-4}	9.8×10^{-4}	9.8×10^{-4}	7.0×10^{-3}	3.8×10^{-3}	0.13

Table 6. The p -value of the paired Wilcoxon test at the 95% significance level for SAMPLE-MSTAR.

Method	A-ConvNet	ResNet50	Vgg16	RN18+	Baseline	Ours -F
p -value	9.8×10^{-4}	9.8×10^{-4}	9.8×10^{-4}	9.8×10^{-4}	9.8×10^{-4}	2.9×10^{-3}

The results demonstrate that for the SAMPLE-SAMPLE experimental group, the p -values for A-ConvNet, ResNet50, Vgg16, RN18+ and baseline are all significantly less than 0.05, indicating that the proposed method outperforms them. The p -value for Ours -F is 0.13 which is greater than 0.05, indicating that the absence of the feedback module (F) in the feature generation module does not significantly affect the final recognition results for the SAMPLE-SAMPLE experimental group. For the SAMPLE-MSTAR experimental group, the p -values for all comparison methods are less than 0.05, indicating that the proposed method is significantly superior to the comparison methods. In this case, the recognition rate of Ours is superior to that of Ours -F, demonstrating that when there is a significant difference between the simulated and measured images, the feedback module (F) demonstrates its advantage.

3.2.3. The Analysis of the Differences between the Two Experimental Groups

Upon further analysis of the experimental results, a significant difference in the recognition rate of the unseen classes is observed between the SAMPLE-SAMPLE and the SAMPLE-MSTAR experimental groups. The difference is attributed to the varying disparities between the simulated and the real data in the two experimental groups. The T-SNE visualization plots Figure 10 show that the simulated images of the unseen classes in the SAMPLE are mostly distributed near their corresponding real images, indicating a strong similarity between them. However, the distribution of the real images in the MSTAR shows weaker correlations compared to the corresponding simulated images in the SAMPLE. When there is a significant difference between the simulated and the real images, the transferability of the networks trained on simulated images is greatly reduced. This is particularly noticeable for the RN18+ network, which enhances performance in the SAMPLE-SAMPLE experimental group but has no impact in the SAMPLE-MSTAR experimental group. The experimental results indicate that the proposed method is effective in the SAMPLE-MSTAR experimental group. It indicates that even when the quality of the simulated images is not high, the extracted category features can still assist the proposed framework in achieving zero-shot recognition of the targets of the multiple classes

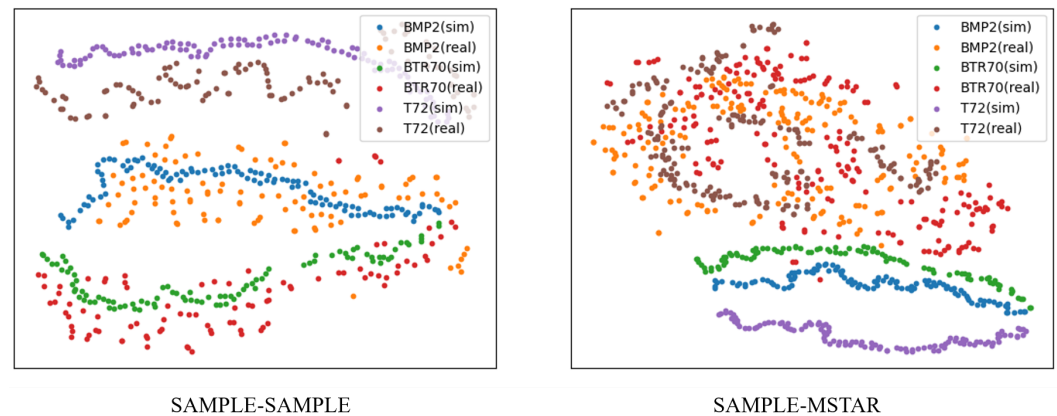


Figure 10. The T-SNE visualization plots of three unseen targets.

3.3. Impact of the Intermediate Layer Dimension Size

The intermediate layer dimension size of G and CFR is equal to the dimension of the feedback feature F . It is an important parameter that impacts the training effectiveness and complexity of the model. Tables 7 and 8 display the average recognition rates with different intermediate layer dimension sizes.

Table 7. Experimental results for different intermediate dimension sizes in SAMPLE-SAMPLE.

Dimensions	Min (%)	Max (%)	Avg (%)	Time (s)
512	99.67	100	99.77 ± 0.15	128.75
1024	99.35	100	99.80 ± 0.22	134.52
2048	99.67	100	99.84 ± 0.16	158.19
4096	99.35	100	99.71 ± 0.23	226.76

Table 8. Experimental results for different intermediate dimension sizes in SAMPLE-MSTAR.

Dimensions	Min (%)	Max (%)	Avg (%)	Time (s)
512	63.54	73.25	67.56 ± 2.56	130.54
1024	64.05	75.13	69.30 ± 3.24	140.40
2048	66.78	75.98	71.57 ± 2.28	164.37
4096	66.61	75.98	71.14 ± 2.72	237.83

The highest recognition rate achieved for each column are marked in bold.

Four different intermediate layer dimension sizes: 512, 1024, 2048, and 4096 are tested. The results show that for the intermediate layer dimension size less than 2048, the average recognition rate increases with the increase in the intermediate layer dimension size. However, when the dimension size further increases to 4096, the average recognition rate decreases. The analysis indicates that expanding the intermediate layer dimension size can enhance the representational and learning capabilities of the model, thereby improving the recognition performance. However, enlarging the intermediate layer dimension size also increases the parameter count. If the capacity of the model is too large, it may overfit irrelevant features, resulting in a decrease in the final recognition rate. Analyzing the training efficiency of the model, it was observed that the training time increases as the dimension of the intermediate layer increases. This is mainly due to the increase in the model complexity. However, the increase in the training time was relatively slow compared to the change in the intermediate layer dimension. For example, the training time for the 2048 dimension only increased by less than 20 s compared to the 1024 dimension. Even when the intermediate dimension size is set to 4096, the training time for 1000 epochs does not exceed 4 min. This is due to the fact that the generation module in this paper consists entirely of simple, shallow fully connected layers. In the feature generation stage, there

is no need to update the weights of the feature extraction module, which significantly improves the training efficiency. For the experimental dataset in this paper, a network with an intermediate layer dimension size of 2048 achieves the best performance without significantly increasing the training time and is considered an appropriate size setting.

3.4. Impact of the Number of Simulated and Real Images

The impact of the quantity of the data on the method is further explored in this section. It is important to note that the final performance of the model may be influenced by the simulated and the real images used by the network. Firstly, The representational capability of the categories may vary depending on the number of simulated images used to extract the category features. Secondly, the mapping between the real features and the simulated category features is learned using the real features of the seen classes, which means that the amount of real data can affect the model training. We focus on the SAMPLE-SAMPLE experimental group. Different numbers of the simulated images are used to extract the category features, and various numbers of the real images are used for the model training. The number of the images is determined by the range of the aspect angle. The aspect angle range is represented by 1°, 5°, 10°, 20°, 30°, 40°, 50°, 60°, 70°, which correspond to 10°–11°, 10°–15°, 10°–20°, 10°–30°, 10°–40°, 10°–50°, 10°–60°, 10°–70°, and 10°–80°, respectively. The specific relationship between the aspect angle range and the number of images used is shown in Table 9, where the quantity represents the total number of images across all categories. The number of simulated or real images decreases as the range decreases.

Table 9. The correspondence between the aspect angle range and the number of images.

The Aspect Angle Range	1°	5°	10°	20°	30°	40°	50°	60°	70°
The Simulated Image Number	7	35	147	268	395	504	603	670	745
The Real Image Number	7	35	147	268	395	504	603	670	745

Table 10 shows the average recognition rate for unseen classes in different combinations. According to the table, the recognition performance reaches its maximum at 99.80% when all simulated and real images are used. The lowest recognition rate of 97.31% occurs when the aspect angle range of the simulated images is 10° and only one real image. Figure 11 presents a visualization of the changes in recognition rates. The high recognition rates are mostly achieved when the number of simulated and real images is greater than 40. The recognition rate generally decreases as the quantity of both types of images is reduced, but the decreasing trend is not significant, with a difference of 3% between the minimum and maximum recognition rates.

Table 10. Experimental results for different number configurations (%).

		Real								
		1°	5°	10°	20°	30°	40°	50°	60°	70°
Simulated	1°	97.92	97.79	98.14	98.79	98.86	98.70	98.99	98.63	98.79
	5°	97.62	97.65	98.53	98.50	98.08	98.40	98.37	98.53	98.60
	10°	97.31	97.82	98.40	98.31	98.31	98.40	98.34	98.24	98.21
	20°	98.37	98.31	98.50	98.50	98.53	98.40	98.79	98.96	98.70
	30°	98.91	98.70	99.06	99.06	99.06	99.25	99.32	98.93	99.35
	40°	99.19	99.58	99.35	99.54	99.48	99.54	99.51	99.67	99.35
	50°	99.41	99.51	99.61	99.48	99.64	99.54	99.67	99.58	99.67
	60°	99.48	99.48	99.77	99.67	99.45	99.61	99.84	99.67	99.67
	70°	99.51	99.67	99.64	99.71	99.58	99.51	99.77	99.77	99.80

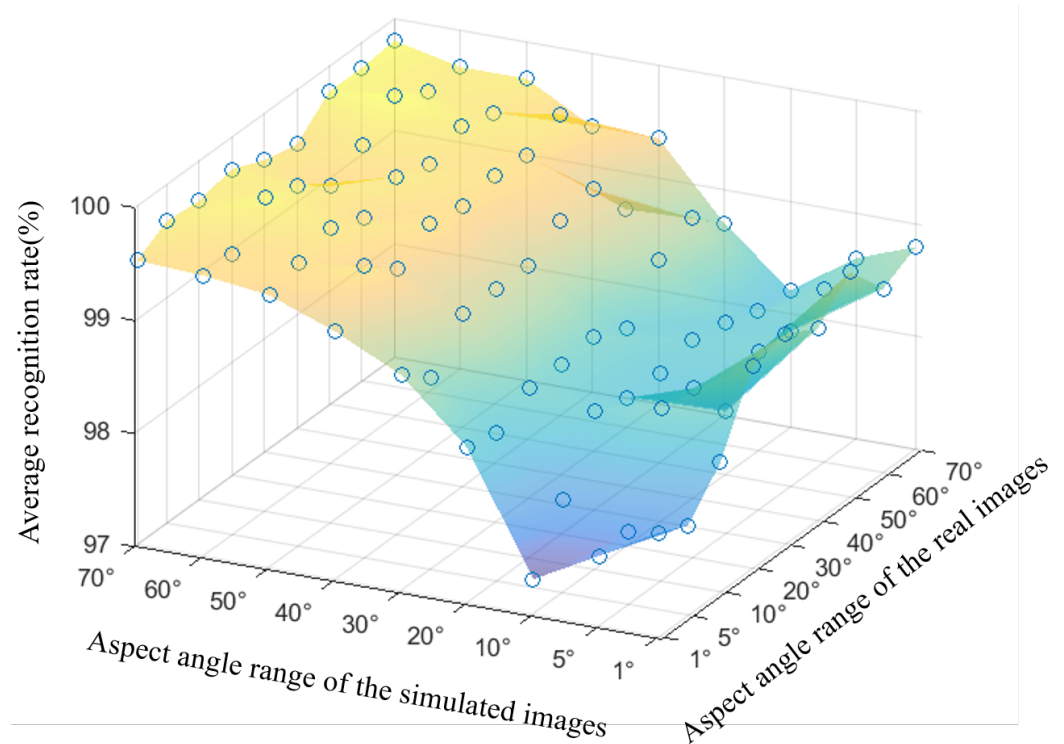


Figure 11. The visualization of the changes in recognition rates.

First, the impact of the number of simulated images is analyzed. When there are enough simulated images, the coverage of the aspect angle is broad encompassing more information. As a result, the category features possess stronger representational capabilities. A decrease in the number of simulated images results in a reduction in the introduced aspect angle information, which weakens the representational capacity of the category features. This, in turn, affects the generation of pseudo-real features and further impacts the final classification performance. However, the model maintains its effectiveness with a recognition rate above 97% even when using a small number of simulated images. It may be because the deep features still include scatter characteristics and other category information, which to an extent represent the entire category. Second, concerning the impact of the number of real images. When the training data volume is large, the model can learn rich mappings as the aspect angle range of the seen class can cover 0–80°. The features generated for the unseen class incorporate information from multiple aspect angles, resulting in the effective recognition of true samples with varying aspect angles. As the number of real images decreases, the acquisition of aspect angle information also decreases, leading to a slight decline in the recognition performance of the model. However, the model can still achieve a relatively high recognition rate because the model can focus on learning the relatively simple mappings and other features unrelated to the aspect angle.

The results suggest that the proposed method can maintain a relatively high recognition rate when training on a sparse set of simulated and real images. The requirement for the number of simulated and real images is reduced.

3.5. Impact of the Seen Class Number

The ability to generalize to unseen classes of the model may be affected by the number of seen classes in addition to the number of images used by the network. This is because the number of seen classes is related to the coverage of the feature space. More seen classes may provide additional features shared with unseen classes, which can improve the ability to generate unseen class data. In this section, the number of the seen classes will be reduced to explore their impact on the final recognition rate. The unseen classes remain BMP2, BTR70, and T72. The initial number of the seen classes was seven in the SAMPLE-SAMPLE

and two in the SAMPLE-MSTAR. Both experimental groups ultimately ended up with a single commonly seen class, which is 2S1. Tables 11 and 12 present the experimental results.

Table 11. SAMPLE-SAMPLE experimental results for different seen class number (%).

The Seen Class Number	7	6	5	4	3	2	1
Average Accuracy	99.80	99.74	99.90	99.32	99.48	92.41	88.83

Table 12. SAMPLE-MSTAR experimental results for different seen class number (%).

The Seen Class Number	2	1
Average Accuracy	71.57	69.23

The results show a decrease in accuracy when the number of seen classes is less than three for the SAMPLE-SAMPLE experimental group. Specifically, the final accuracy for the two and one seen classes is 92.41% and 88.83%, respectively. Similarly, the recognition rate for one seen class in the SAMPLE-MSTAR experimental group is 2.34% lower than the recognition rate for two seen classes. However, the performance of the model is not strictly proportional to the number of the seen classes. In the experimental group SAMPLE-SAMPLE, the recognition rate of the method can reach 99% or higher when the number of seen classes is three or more. This may be due to the fact that when the number of seen classes is more than three, the features selected from the seen classes can already cover most of the common information with the unseen classes. As a result, the mapping learned from the seen classes can generalize well to the unseen classes.

The recognition performance of the model decreases as the number of seen classes decreases, but there is not necessarily a linear relationship between them. Moreover, the model can maintain a certain level of performance even with fewer seen classes.

3.6. Extended Experiments on the Generalized Zero-Shot Recognition

In real-world scenarios, both the seen class samples and the unseen class samples often coexist. Therefore, it is necessary to be able to recognize both types of samples simultaneously. This recognition problem is known as the generalized zero-shot recognition problem, which is an extension of the zero-shot problem. The training process for generalized zero-shot recognition remains consistent with that of zero-shot recognition, allowing only the seen class samples to be used for training. During the testing phase, the network must be able to recognize both seen and unseen class targets simultaneously. To achieve this, the final classifier training includes both the real features of the seen classes and the generated features of the unseen classes. That is, a softmax classifier $f_{gzs} : \{X^s \cup \tilde{X}^u \rightarrow Y^s \cup Y^u\}$ is trained with $X^s \cup \tilde{X}^u$, and the classifier can recognize both the seen and the unseen class samples. Compared to the zero-shot recognition, the generalized zero-shot recognition is more challenging.

This section presents the results of an extension experiment on the generalized zero-shot SAR target recognition with two experimental groups. In the SAMPLE-SAMPLE experimental group, the separation of the seen and the unseen classes aligns with zero-shot target recognition. In the SAMPLE-MSTAR experimental group, the unseen classes are BMP2 and BMP2, with T72 designated as the seen class. In the comparison experiment, the network is trained with a combination of simulated images of the unseen class and real images of the seen class. Tables 13 and 14 display the experimental results.

Table 13. Experimental results of the generalized zero-shot recognition in SAMPLE-SAMPLE (%).

	Seen Classes					Unseen Classes					Accuracy
	2S1	M1	M2	M35	M548	ZSU23-4	M60	BMP2	BTR70	T72	
A-ConvNet	99.43	98.45	96.09	100	100	100	100	0	0	12.04	77.55
ResNet50	100	100	100	100	100	100	98.85	1.87	0	0.93	77.25
Vgg16	100	100	100	100	100	100	100	0	0	3.70	77.47
RN18+	97.70	100	96.88	95.35	100	100	100	3.74	60.87	64.81	85.80
baseline	100	100	99.22	100	100	98.86	100	31.78	45.65	25.93	84.68
Ours	93.10	95.35	95.31	100	100	100	100	95.33	51.09	84.26	91.45

Table 14. Experimental results of the generalized zero-shot recognition in SAMPLE-MSTAR (%).

	Seen Classes			Unseen Classes		
	T72	ZSU23-4	2S1	BMP2	BTR70	Accuracy
A-ConvNet	100	100	100	0	0	65.55
ResNet50	99.49	100	100	2.05	0	65.81
Vgg16	100	100	100	1.54	0	65.81
RN18+	73.98	100	79.20	0	0	56.04
baseline	100	100	100	0	0.51	65.64
Ours	91.33	100	86.13	51.28	67.35	81.15

In the SAMPLE-SAMPLE experimental group, the classic networks A-Convnets, ResNet50 and Vgg16 almost misclassify all unseen category targets as seen class targets. While RN18+ and baseline have some recognition ability for the unseen classes, the recognition rate does not exceed 45%. In the SAMPLE-MSTAR experimental group, the comparison methods trained directly with supplemented simulated images of unseen classes have an almost 0% recognition rate for unseen class targets. The reason for the results is that there are statistical distribution differences between the real and the simulated images and the model more easily captures features from real images that dominate the training set. In the SAMPLE-SAMPLE experimental group, the recognition rate of the best comparison method, RN18+, reached 85%. At the same time, our method achieved an overall recognition rate of 91.45%. Although the recognition rate for seen class targets has slightly decreased, the recognition rate for unseen category targets has significantly increased, with an average recognition rate of over 70%. For the SAMPLE-MSTAR experimental group, our method achieved an overall recognition rate of 81.15%, which is approximately 15% higher than that of the comparison methods. Our method generates pseudo-real features for unseen classes that have a feature distribution closer to the real features. When mixing training with real samples of the seen classes, the model preference for the seen classes is reduced and achieves a balance between the seen and unseen classes. The results indicate that our method contributes to the generalized zero-shot recognition.

4. Discussion

The experiments in Section 3 demonstrate that the proposed method can achieve zero-shot recognition for multiple classes of targets, rather than inferring only a single class. It represents a breakthrough compared to existing embedding model-based methods for SAR target zero-shot recognition. Furthermore, the proposed feature generation architecture outperforms classical networks trained directly on simulated images in terms of recognition performance. The effectiveness of the architecture is highlighted. Several factors influencing the method were examined, as discussed in detail in Section 3. Additionally, the proposed method has the following limitations.

1. The deep features utilized for network learning in this paper are extracted solely from a model pre-trained on seen class real images. It may limit their representational capacity thereby influencing the learning of the network.

2. The proposed method relies on the similarity of the features between the seen and the unseen class targets, allowing for the transfer of the mapping learned from the seen classes to the unseen classes. If there is a significant difference between the seen and the unseen classes, the proposed method may have limitations.

Based on the analysis above, the proposed method presents a new approach to address zero-shot recognition in SAR images. However, it still has limitations that require further exploration and improvement. In the future, the following directions will be mainly researched. Firstly, multiple feature extraction methods will be adopted. For example, initially using a large-scale SAR image pre-trained feature extraction network, and then transferring it to specific tasks. Alternatively, we will use self-supervised networks like VAE to learn more effective feature representations. Furthermore, the vector embedding of large models can also be utilized. Secondly, the situations where there is a notable contrast between the seen and the unseen classes will be examined. Domain adaptation methods will be introduced into zero-shot recognition tasks to address this issue.

5. Conclusions

The traditional classification networks cannot classify unseen SAR targets. To generate pseudo-real samples of unseen classes for supervised learning and achieve classification of the targets for the multiple unseen classes, a conditional generative network with category features from simulated images for zero-shot SAR target recognition is proposed. Specifically, the process begins with the extraction of the category features from the simulated images. Next, a conditional VAE-GAN network is trained using samples of seen classes. Pseudo-real samples of the unseen classes are then generated using the category features of unseen classes as conditions. Finally, the classification network is trained using the generated samples to achieve supervised learning. The proposed method can recognize three unseen category targets in both the SAMPLE and MSTAR datasets, achieving recognition rates of $99.80 \pm 1.22\%$ and $71.57 \pm 2.28\%$, respectively. The recognition performance of the method decreases slightly when fewer simulated images are used to extract category features and when fewer seen class real images are used for training. However, the decrease is not significant, with a difference of no more than 3% between the lowest and highest recognition rates. The proposed method remains effective even with only a few seen classes. The recognition rates for the three unseen class targets in the SAMPLE and MSTAR datasets exceed 90% and 70%, respectively, when only two seen classes are used. Additionally, the proposed method can be expanded to the tasks of the generalized zero-shot recognition tasks.

Author Contributions: Conceptualization, G.C.; methodology, G.C.; software G.C.; validation, G.C.; writing—original draft preparation, G.C.; supervision, S.Z.; project administration, S.Z.; data processing, Q.H.; visualization, Z.S. and X.Z.; Literature collection, L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The MSTAR dataset can be obtained at: <https://www.sdms.afrl.af.mil/index.php?collection=mstar>. The SAMPLE dataset can be obtained at: https://github.com/benjaminlewis-afrl/SAMPLE_dataset_public?tab=readme-ov-file

Acknowledgments: The authors would like to thank the reviewers and editors who provided valuable comments and suggestions for this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Khoshnevis, S.A.; Ghorshi, S. A tutorial on tomographic synthetic aperture radar methods. *SN Appl. Sci.* **2020**, *2*, 1504. [\[CrossRef\]](#)
2. Dudgeon, D.E.; Lacoss, R.T. An overview of automatic target recognition. *Linc. Lab. J.* **1993**, *6*, 3–10.
3. Popova, M.; Shvets, M.; Oliva, J.; Isayev, O. MolecularRNN: Generating realistic molecular graphs with optimized properties. *arXiv* **2019**, arXiv:1905.13372.
4. Shaulskiy, D.V.; Evtikhiev, N.N.; Starikov, R.S.; Starikov, S.N.; Zlokazov, E.Y. MINACE filter: Variants of realization in 4-f correlator. In Proceedings of the Optical Pattern Recognition XXV, Baltimore, MD, USA, 5–9 May 2014; SPIE: Bellingham, WA, USA, 2014; Volume 9094, pp. 135–142.
5. Diemunsch, J.R.; Wissinger, J. Moving and stationary target acquisition and recognition (MSTAR) model-based automatic target recognition: Search technology for a robust ATR. In Proceedings of the Algorithms for Synthetic Aperture Radar Imagery V, Orlando, FL, USA, 13–17 April 1998; SPIE: Bellingham, WA, USA, 1998; Volume 3370, pp. 481–492.
6. Li, J.; Yu, Z.; Yu, L.; Cheng, P.; Chen, J.; Chi, C. A Comprehensive Survey on SAR ATR in Deep-Learning Era. *Remote Sens.* **2023**, *15*, 1454. [\[CrossRef\]](#)
7. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1–9. [\[CrossRef\]](#)
8. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
9. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
10. Pearlmutter. Learning state space trajectories in recurrent neural networks. In Proceedings of the International 1989 Joint Conference on Neural Networks, Washington, DC, USA, 18–22 June 1989; IEEE: Piscataway, NJ, USA, 1989; pp. 365–372.
11. Scarselli, F.; Tsoi, A.C.; Gori, M.; Hagenbuchner, M. Graphical-based learning environments for pattern recognition. In Proceedings of the Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshops, SSPR 2004 and SPR 2004, Lisbon, Portugal, 18–20 August 2004; Proceedings; Springer: Berlin/Heidelberg, Germany, 2004; pp. 42–56.
12. Morgan, D.A. Deep convolutional neural networks for ATR from SAR imagery. In Proceedings of the Algorithms for Synthetic Aperture Radar Imagery XXII, Baltimore, MD, USA, 20–24 April 2015; SPIE: Bellingham, WA, USA, 2015; Volume 9475, pp. 116–128.
13. Chen, S.; Wang, H.; Xu, F.; Jin, Y.Q. Target classification using the deep convolutional networks for SAR images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 4806–4817. [\[CrossRef\]](#)
14. Zhang, F.; Hu, C.; Yin, Q.; Li, W.; Li, H.C.; Hong, W. Multi-aspect-aware bidirectional LSTM networks for synthetic aperture radar target recognition. *IEEE Access* **2017**, *5*, 26880–26891. [\[CrossRef\]](#)
15. Zhao, C.; Zhang, S.; Luo, R.; Feng, S.; Kuang, G. Scattering features spatial-structural association network for aircraft recognition in SAR images. *IEEE Geosci. Remote. Sens. Lett.* **2023**, *20*, 4006505. [\[CrossRef\]](#)
16. Zhang, X.; Feng, S.; Zhao, C.; Sun, Z.; Zhang, S.; Ji, K. MGSFA-Net: Multi-Scale Global Scattering Feature Association Network for SAR Ship Target Recognition. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2024**, *17*, 4611–4625. [\[CrossRef\]](#)
17. Ding, J.; Chen, B.; Liu, H.; Huang, M. Convolutional neural network with data augmentation for SAR target recognition. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 364–368. [\[CrossRef\]](#)
18. Ding, B.; Wen, G.; Huang, X.; Ma, C.; Yang, X. Data augmentation by multilevel reconstruction using attributed scattering center for SAR target recognition. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 979–983. [\[CrossRef\]](#)
19. Guo, J.; Lei, B.; Ding, C.; Zhang, Y. Synthetic aperture radar image synthesis by using generative adversarial nets. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1111–1115. [\[CrossRef\]](#)
20. Cui, Z.; Zhang, M.; Cao, Z.; Cao, C. Image data augmentation for SAR sensor via generative adversarial nets. *IEEE Access* **2019**, *7*, 42255–42268. [\[CrossRef\]](#)
21. Niu, S.; Qiu, X.; Peng, L.; Lei, B. Parameter prediction method of SAR target simulation based on convolutional neural networks. In Proceedings of the EUSAR 2018; 12th European Conference on Synthetic Aperture Radar, Aachen, Germany, 4–7 June 2018; VDE: Frankfurt am Main, Germany, 2018; pp. 1–5.
22. Zhai, Y.; Deng, W.; Xu, Y.; Ke, Q.; Gan, J.; Sun, B.; Zeng, J.; Piuri, V. Robust SAR automatic target recognition based on transferred MS-CNN with L 2-regularization. *Comput. Intell. Neurosci.* **2019**, *2019*, 9140167. [\[CrossRef\]](#) [\[PubMed\]](#)
23. Malmgren-Hansen, D.; Kusk, A.; Dall, J.; Nielsen, A.A.; Engholm, R.; Skriver, H. Improving SAR automatic target recognition models with transfer learning from simulated data. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1484–1488. [\[CrossRef\]](#)
24. Zelong, W.; Xianghui, X.; Lei, Z. Study of deep transfer learning for SAR ATR based on simulated SAR images. *J. Univ. Chin. Acad. Sci.* **2020**, *37*, 516.
25. Wang, K.; Zhang, G. SAR target recognition via meta-learning and amortized variational inference. *Sensors* **2020**, *20*, 5966. [\[CrossRef\]](#)
26. Wang, K.; Zhang, G.; Xu, Y.; Leung, H. SAR target recognition based on probabilistic meta-learning. *IEEE Geosci. Remote Sens. Lett.* **2020**, *18*, 682–686. [\[CrossRef\]](#)
27. Keydel, E.R.; Lee, S.W.; Moore, J.T. MSTAR extended operating conditions: A tutorial. *Algorithms Synth. Aperture Radar Imag. III* **1996**, *2757*, 228–242.
28. Larochelle, H.; Erhan, D.; Bengio, Y. Zero-data learning of new tasks. In Proceedings of the AAAI, Chicago, IL, USA, 13–17 July 2008; Volume 1, p. 3.

29. Lampert, C.H.; Nickisch, H.; Harmeling, S. Attribute-based classification for zero-shot visual object categorization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *36*, 453–465. [\[CrossRef\]](#)
30. Chen, L.; Zhang, H.; Xiao, J.; Liu, W.; Chang, S.F. Zero-shot visual recognition using semantics-preserving adversarial embedding networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1043–1052.
31. Shigeto, Y.; Suzuki, I.; Hara, K.; Shimbo, M.; Matsumoto, Y. Ridge regression, hubness, and zero-shot learning. In Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, 7–11 September 2015; Proceedings, Part I 15; Springer: Berlin/Heidelberg, Germany, 2015; pp. 135–151.
32. Yang, Y.; Hospedales, T.M. A unified perspective on multi-domain and multi-task learning. *arXiv* **2014**, arXiv:1412.7489.
33. Liu, Y.; Zhou, L.; Bai, X.; Huang, Y.; Gu, L.; Zhou, J.; Harada, T. Goal-oriented gaze estimation for zero-shot learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 3794–3803.
34. Mishra, A.; Krishna Reddy, S.; Mittal, A.; Murthy, H.A. A generative model for zero shot learning using conditional variational autoencoders. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2188–2196.
35. Xian, Y.; Lorenz, T.; Schiele, B.; Akata, Z. Feature generating networks for zero-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5542–5551.
36. Xian, Y.; Sharma, S.; Schiele, B.; Akata, Z. f-vaegan-d2: A feature generating framework for any-shot learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 10275–10284.
37. Narayan, S.; Gupta, A.; Khan, F.S.; Snoek, C.G.; Shao, L. Latent embedding feedback and discriminative features for zero-shot classification. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XXII 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 479–495.
38. Song, Q.; Xu, F. Zero-shot learning of SAR target feature space with deep generative neural networks. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 2245–2249. [\[CrossRef\]](#)
39. Wei, Q.R.; He, H.; Zhao, Y.; Li, J.A. Learn to recognize unknown SAR targets from reflection similarity. *IEEE Geosci. Remote Sens. Lett.* **2020**, *19*, 4002205. [\[CrossRef\]](#)
40. Wei, Q.R.; Chen, C.Y.; He, M.; He, H.M. Zero-Shot SAR Target Recognition Based on Classification Assistance. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 4003705. [\[CrossRef\]](#)
41. Cha, M.; Majumdar, A.; Kung, H.; Barber, J. Improving SAR automatic target recognition using simulated images under deep residual refinements. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 2606–2610.
42. Wang, K.; Zhang, G.; Leung, H. SAR target recognition based on cross-domain and cross-task transfer learning. *IEEE Access* **2019**, *7*, 153391–153399. [\[CrossRef\]](#)
43. Liping, H.; Chunzhu, D.; Jinfan, L.; Hongcheng, Y.; Chao, W.; Chao, N. Non-homologous target recognition of ground vehicles based on SAR simulation image. *Syst. Eng. Electron.* **2021**, *43*, 3518–3525.
44. Zhang, C.; Wang, Y.; Liu, H.; Sun, Y.; Hu, L. SAR target recognition using only simulated data for training by hierarchically combining CNN and image similarity. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 4503505. [\[CrossRef\]](#)
45. Song, Q.; Chen, H.; Xu, F.; Cui, T.J. EM simulation-aided zero-shot learning for SAR automatic target recognition. *IEEE Geosci. Remote Sens. Lett.* **2019**, *17*, 1092–1096. [\[CrossRef\]](#)
46. Inkawhich, N.; Inkawhich, M.J.; Davis, E.K.; Majumder, U.K.; Tripp, E.; Capraro, C.; Chen, Y. Bridging a gap in SAR-ATR: Training on fully synthetic and testing on measured data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 2942–2955. [\[CrossRef\]](#)
47. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
48. Liu, L.; Pan, Z.; Qiu, X.; Peng, L. SAR target classification with CycleGAN transferred simulated samples. In Proceedings of the IGARSS 2018–2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 4411–4414.
49. Lewis, B.; Liu, J.; Wong, A. Generative adversarial networks for SAR image realism. In Proceedings of the Algorithms for Synthetic Aperture Radar Imagery XXV, Orlando, FL, USA, 15–19 April 2018; SPIE: Bellingham, WA, USA, 2018; Volume 10647, pp. 37–47.
50. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv* **2014**, arXiv:1411.1784.
51. Sohn, K.; Lee, H.; Yan, X. Learning structured output representation using deep conditional generative models. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; Volume 28.
52. Snell, J.; Swersky, K.; Zemel, R. Prototypical networks for few-shot learning. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
53. Larsen, A.B.L.; Sønderby, S.K.; Larochelle, H.; Winther, O. Autoencoding beyond pixels using a learned similarity metric. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; PMLR: Cambridge, MA, USA, 2016; pp. 1558–1566.

54. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved training of wasserstein gans. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
55. Lewis, B.; Scarnati, T.; Sudkamp, E.; Nehrbass, J.; Rosencrantz, S.; Zelnio, E. A SAR dataset for ATR development: The Synthetic and Measured Paired Labeled Experiment (SAMPLE). In Proceedings of the Algorithms for Synthetic Aperture Radar Imagery XXVI, Baltimore, MD, USA, 14–18 April 2019; SPIE: Bellingham, WA, USA, 2019; Volume 10987, pp. 39–54.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.