remote sensing

MDPI

Article

Semantics- and Physics-Guided Generative Network for Radar HRRP Generalized Zero-Shot Recognition

Jiaqi Zhou, Tao Zhang, Siyuan Mu, Yuze Gao, Feiming Wei and Wenxian Yu *

Shanghai Key Laboratory of Intelligent Sensing and Recognition, Shanghai Jiao Tong University, Shanghai 200240, China; qi131460@sjtu.edu.cn (J.Z.)
* Correspondence: wxyu@sjtu.edu.cn

Highlights

What are the main findings?

- A semantics- and physics-guided generative network (SPGGN) was proposed for radar HRRP generalized zero-shot recognition.
- A strong scattering point reconstruction module based on contrastive learning (CSPR module) introduced physical constraints.

What are the implications of the main findings?

- The combination of semantic knowledge and physical priors enhances the robustness and interpretability of HRRP recognition models in open-set scenarios.
- The proposed SPGGN achieved a superior performance on measured aircraft HRRP datasets, demonstrating strong generalization to unseen classes under varying aspects.

Abstract

High-resolution range profile (HRRP) target recognition has garnered significant attention in radar automatic target recognition (RATR) research for its rich structural information and low computational costs. With the rapid advancements in deep learning, methods for HRRP target recognition that leverage deep neural networks have emerged as the dominant approaches. Nevertheless, these traditional closed-set recognition methods require labeled data for every class in training, while in reality, seen classes and unseen classes coexist. Therefore, it is necessary to explore methods that can identify both seen and unseen classes simultaneously. To this end, a semantic- and physical-guided generative network (SPGGN) was innovatively proposed for HRRP generalized zero-shot recognition; it combines a constructed knowledge graph with attribute vectors to comprehensively represent semantics and reconstructs strong scattering points to introduce physical constraints. Specifically, to boost the robustness, we reconstructed the strong scattering points from deep features of HRRPs, where class-aware contrastive learning in the middle layer effectively mitigates the influence of target-aspect variations. In the classification stage, discriminative features are produced through attention-based feature fusion to capture multi-faceted information, while the design of balancing loss abates the bias towards seen classes. Experiments on two measured aircraft HRRP datasets validated the superior recognition performance of our method.

Keywords: high-resolution range profile (HRRP); radar automatic target recognition (RATR); generalized zero-shot learning (GZSL); knowledge graph (KG); strong scattering points

Check for updates

Academic Editor: Piotr Samczynski

Received: 29 October 2025
Revised: 14 December 2025
Accepted: 16 December 2025
Published: 19 December 2025

Copyright: © 2025 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.

1. Introduction

Radar automatic target recognition (RATR) plays a crucial role in surveillance, defense, and remote sensing applications, typically relying on either synthetic aperture radar (SAR) images [1–3] or high-resolution range profiles (HRRPs) [4–6]. Compared to SAR images, HRRPs offer one-dimensional profiles with a higher range resolution and a lower data dimensionality, making them well suited for real-time and resource-constrained RATR tasks. HRRPs represent the projection of the target scattering centers along the radar's line of sight (RLOS), offering detailed structural information that is very valuable for target recognition.

In general, HRRP target recognition has progressed through three main stages: from manually designed methods to shallow models, and then to deep learning approaches. Although manual feature design offers simplicity and interpretability, it yields a limited performance. Shallow models improve by automating feature learning, but struggle with handling complex patterns. Deep learning has brought a significant leap in accuracy and robustness [7–10], but at the expense of a large dataset demand and interpretability challenges. To solve these issues, recent research has focused on hybrid methods that integrate domain knowledge with deep learning. For example, ref. [11] integrated the scattering centers and target aspects of HRRPs to enhance the recognition performance, and ref. [12] presented a hierarchical classification framework guided by both semantic knowledge and data-driven information.

Overall, the methods mentioned above assume that all target categories are known during training, relying on labeled data from each class. However, in reality, seen (in-database) classes and new unseen (out-of-database) classes usually coexist. In this case, traditional closed-set recognition systems forcibly classify samples from unseen classes as seen classes, causing decision errors and limiting practical applications. Approaches for HRRP open-set recognition [13,14] can only reject unseen classes without an identification capability. Therefore, it is essential to explore HRRP generalized zero-shot recognition (GZSR) methods that can simultaneously identify seen and unseen classes.

Generalized zero-shot learning (GZSL) seeks to transfer knowledge from seen classes to unseen classes through semantic information [15], enabling the simultaneous classification of both. It is more practical and difficult than zero-shot learning (ZSL), which is dedicated to classifying exclusively unseen classes. In particular, semantic knowledge is the key to realizing ZSL and GZSL, as it helps establish relations between seen and unseen classes, enabling the implementation of GZSR. The most widely used semantic knowledge includes manually annotated attributes [16], textual descriptions (e.g., words and sentences [17]), and knowledge graphs [18,19]. Knowledge graphs belong to the category of structured knowledge, including class hierarchies, common sense knowledge, category attributes, etc. They can usually express richer semantics than attributes and text.

The existing GZSL approaches can be broadly divided into two main categories: embedding-based methods and generative-based methods [15]. The latter typically relies on models like variational auto-encoders (VAEs) [20] or generative adversarial networks (GANs) [21] to generate samples for unseen classes. For this reason, compared with the former, the latter usually perform better in alleviating the seen class bias and improving the generalization ability [22], and they are suitable for complex scenarios that require a high accuracy and can handle large inter-class differences.

To the best of our knowledge, there are few GZSL methods specifically designed for HRRP GZSR, and few studies have constructed a knowledge graph tailored to aircraft as semantics. To address these issues, firstly, we constructed a knowledge graph for aircraft targets (ATKG) and integrated it with manually annotated attribute vectors to form the dedicated semantics. Secondly, inspired by the aforementioned methods, we

innovatively proposed a semantics- and physics-guided generative network (SPGGN), which generates features for unseen classes based on the semantics for HRRP GZSR. Specifically, in the feature-generation stage, we reconstructed the strong scattering points (i.e., shallow features) of an HRRP from its deep features, obtained by a pre-trained CNN, to introduce physical constraints. In the classification stage, we adopted an attention-based feature fusion strategy to capture multiple sets of complementary information, thereby enhancing the inter-class feature discriminability. Additionally, the introduction of balancing loss effectively mitigates the bias toward seen classes.

The main contributions of this paper are summarized below:

(1) To develop semantic knowledge, we constructed a knowledge graph and manually annotated attributes for our aircraft HRRP dataset. The knowledge graph reflects the relations between classes through the graph structure, while the attribute vectors accurately describe various parameters. Both of these are combined to comprehensively represent semantics, enabling the network to better understand seen and unseen classes.

(2) SPGGN was proposed for HRRP GZSR; it generates deep features conditioned on semantics as the basic module and employs a strong scattering point reconstruction module based on contrastive learning (CSPR module) to introduce physical constraints, thereby boosting the model's interpretability and robustness. Class-aware contrastive learning (CACL) was adopted in the middle layer of the CSPR module to mitigate the influence of target aspects and facilitate the reconstruction.

(3) In the classification stage, to obtain more discriminative features, we leveraged an attention-based feature fusion strategy to utilize rich information. Furthermore, to alleviate the bias towards seen classes, we used the balancing loss to adjust the prediction probability on seen and unseen classes during training.

(4) Comprehensive experiments on the two measured HRRP datasets validated the superior recognition performance of our method for both seen and unseen classes, despite varying target aspects.

The rest of this paper is structured as follows. Section 2 outlines the preliminary work. Section 3 offers a detailed explanation of the proposed method, SPGGN. Section 4 presents comprehensive experiments on two measured aircraft HRRP datasets. Section 5 concludes this paper.

2. Preliminary Work

2.1. HRRP Dataset Construction

A fixed radar system operating in the X-band with a bandwidth of 600 MHz measures the HRRP of moving aircraft, as shown in Figure 1. The HRRP varies with changes in the azimuth θ and elevation ϕ, which is the target-aspect sensitivity. We acquired eight types of aircraft HRRP data, including airliner, airlifter, helicopter, etc. Compared with the three-class HRRP used in [8,10], we incorporated a greater diversity of aircraft types and accounted for the effect of the target aspect, thereby providing more compelling validation for the proposed method. An overview of the measured HRRP is shown in Table 1. Examples of an HRRP with 512 range cells obtained after preprocessing (including noise reduction and sensitivity elimination) are shown in Table 2, where only five airliners are presented. To provide a more convincing evaluation of our method, we formed two datasets: S5U3 (five seen classes and three unseen classes) and S6U2 (six seen classes and two unseen classes). Taking into account the number of HRRPs for each class, we treated B787 and ZXX as unseen classes. In addition to the two, we randomly selected one class as an unseen class to form the S5U3 dataset. For the seen classes, we randomly divided them

into a training set and a testing set in a 4:1 ratio. Samples of the unseen classes were only used for the testing set, with approximately 650 samples per class.

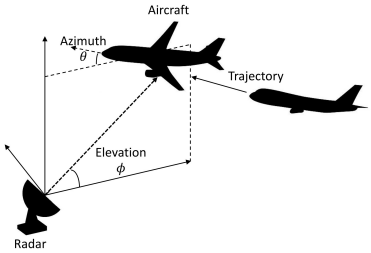
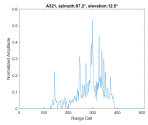
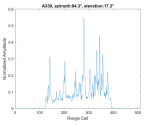
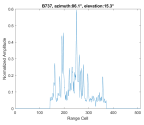
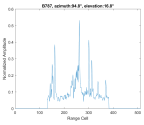
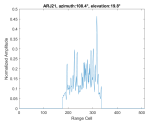


Figure 1. Schematic of target HRRP measurement with azimuth and elevation.

Table 1. Radar parameters and overview of aircraft HRRP data.

Radar Parameters	Frequency	X-Band	
	Bandwidth	600 MHz	
Aircraft Type	**Azimuth Range (°)**	**Elevation Range (°)**	**Number of HRRPs**
A321	40–160	10–40	2517
A330	75–105	16–19	2500
B737	60–120	10–18	2557
KXX	150–180	1–4	2588
YXX	150–180	0.5–1.5	2592
ARJ21	0–200	0–45	2531
B787	60–120	15–19	674
ZXX	150–180	6.5–8.5	654
Sum	-	-	16,613

Table 2. Examples of measured HRRPs for 5 types of airliners after preprocessing.

Type	A321	A330	B737	B787	ARJ21
Image					
HRRP					

2.2. Scattering Center Model

The scattering center model serves as a physical model to describe the mechanism of radar echo formation [23]. The extraction of scattering centers identifies the position, intensity, and phase information of these key points from complex HRRP signals to construct a sparse representation model for the target. Taking pulse-based radar as an example, assume it transmits a linear frequency modulation (LFM) signal. After pulse compression, the frequency response $r(n)$ at the n-th frequency point of the target echo is given by the following:

$$r(n) = \sum_{s=1}^{S} \sigma(s) \cdot \exp\left(-j2\pi \frac{2D_s(n-1)\Delta f}{c}\right), \quad (1)$$

where $n = 1, \ldots, N$, N represents the number of frequency points, S signifies the number of target scattering centers (SCs), $\sigma(s)$ signifies the scattering intensity of the s-th SC, D_s

signifies the radial distance from the s-th SC to the radar, Δf means the spacing between adjacent frequency points, and c denotes the propagation speed of the radar signal. Due to the limitation of the radar distance resolution, assuming that the scattering center of the target in Equation (1) is located at an integer multiple of the distance unit, the frequency response of the target echo containing I distance units intercepted by the distance window at the n-th frequency point can be rewritten as follows:

$$r(n) = \sum_{i=1}^{I} \sigma(i) \cdot \exp\left(-j2\pi \frac{2p_i(n-1)\Delta f}{c}\right), \tag{2}$$

where $i = 1, \ldots, I$ represents the distance unit index and p_i is the position of the i-th distance unit. Given a matrix $\Theta = [\alpha(p_1), \cdots, \alpha(p_i), \cdots, \alpha(p_I)]$, then

$$\alpha(p_i) = \left[1, \exp\left(-j\frac{4\pi p_i \Delta f}{c}\right), \exp\left(-j\frac{4\pi p_i 2\Delta f}{c}\right), \cdots, \exp\left(-j\frac{4\pi p_i (N-1)\Delta f}{c}\right)\right]^{\mathrm{T}}. \tag{3}$$

Considering the noise impact, the frequency response $\boldsymbol{r}$ of the target signal can be expressed as $\boldsymbol{r} = \boldsymbol{\Theta}\boldsymbol{\sigma} + \boldsymbol{\varphi}$, where $\boldsymbol{r} = [r(1), \cdots, r(N)]^{\mathrm{T}}$, $\boldsymbol{\varphi} = [\varphi(1), \cdots, \varphi(N)]^{\mathrm{T}}$ signifies the noise; $\boldsymbol{\sigma} = [\sigma(1), \cdots, \sigma(N)]^{\mathrm{T}}$ represents the complex HRRP sample; and $|\sigma|$ signifies the respective real HRRP. This is the radar echo scattering center model.

Based on the scattering point model, the amplitude and position of strong scattering points are estimated using the orthogonal matching pursuit algorithm to achieve noise-robust HRRP recognition [23]. Ref. [24] combined the amplitude and position of scattering centers to form low-dimensional features; however, dimensionality reduction operations inevitably lose a large amount of useful information, which we solved through the use of features with the same dimension as the HRRP.

2.3. Construction of Semantic Embeddings

Semantic knowledge is a necessary prerequisite for HRRP GZSR. Regarding its construction, we combined the knowledge graph embeddings of classes with the annotated attribute vectors as semantic embeddings. Knowledge graph embeddings focus on reflecting the relations between different categories (Figure 2a), while attribute vectors accurately describe the various parameters of different category targets (Table 3). For the acquisition of knowledge graph embeddings, the following two steps were performed:

(1) Construction of Knowledge Graph: The ATKG consisted of 88 triples with 43 entities and 11 types of relations, including the target, aircraft type, length, height, wingspan, weight, presence of disk radar, tail type, wing type, engine type and position, and number of engines. We selected attributes that can provide information to assist in recognition. To reflect the similarity among different aircraft, continuous attributes such as the length, height, weight, and wingspan were discretized into three levels (short, medium, and long) based on the empirical distribution of aircraft specifications from publicly available databases. The thresholds for each attribute were determined by clustering the numerical values into three groups rather than relying on arbitrary fixed ranges. The graph can be incrementally extended when new aircraft types emerge and the same discretization and relation-definition principles can be directly applied to other domains.

(2) Knowledge Representation Learning: Knowledge representation learning (KRL), also known as knowledge graph embedding (KGE), projects the entities and relations of knowledge graphs, typically represented as triples, into a low-dimensional continuous vector space. Inspired by [25], we adopted a rotation mechanism to obtain knowledge graph embeddings of the ATKG, expressed as triples $(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$. This models relation $\boldsymbol{r}$ as a rotational transformation from head entity $\boldsymbol{h}$ to tail entity $\boldsymbol{t}$, embedding entities

and relations from the knowledge graph into a complex vector space. For a given triple $(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$, the objective was to satisfy the following as much as possible:

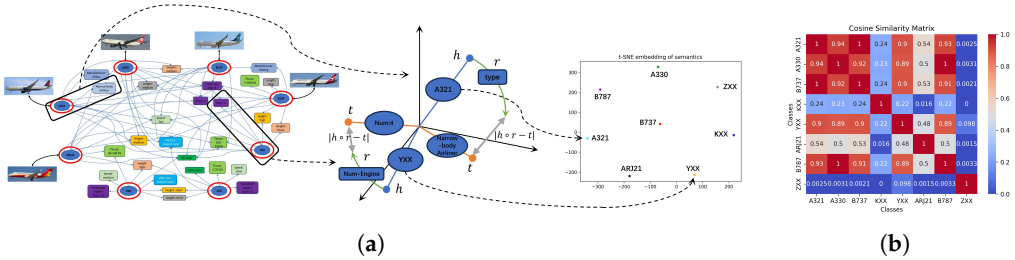
$$\boldsymbol{t} = \boldsymbol{h} \circ \boldsymbol{r}, \boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t} \in \mathbb{C}^k, \tag{4}$$

where $|r_i| = 1$, $\circ$ represents the element-wise (Hadamard) product (i.e., rotation operation) and $\mathbb{C}^k$ represents the complex vector space with k dimension. This approach flexibly represents different types of relations. The score function was formulated as follows:

$$s_r(\boldsymbol{h}, \boldsymbol{t}) = \|\boldsymbol{h} \circ \boldsymbol{r} - \boldsymbol{t}\|. \tag{5}$$

The score should be minimized as much as possible. Then, the trained model is leveraged to extract the knowledge graph embeddings of the class nodes, as shown in Figure 2a. Compared with other KRL models, such as translation models [26] and bilinear models [27], the rotation mechanism can more effectively capture complex relation structures. In addition, the interactions between entities and relations can be intuitively explained through the angles and amplitudes of rotation, which is easier to explain than addition or multiplication. Due to the modest scale of our ATKG, we opted against graph neural networks (e.g., QuatGAT [28]) to mitigate potential overfitting and training instability.

Attribute vectors were generated through manual annotation, and their similarity matrix is shown in Figure 2b. Some of the important attributes included are shown in Table 3. Unlike structured relations in knowledge graphs, attribute vectors provide a refined description of each attribute. By combining the knowledge graph with attribute vectors, semantic embeddings can be expressed more comprehensively, avoiding the limitations of a single representation. Consequently, the model can better understand the semantics of unseen classes and improve its generalization in ZSL and GZSL scenarios.



(**a**) (**b**)

Figure 2. (**a**) The ATKG and the class nodes, from which we extracted knowledge graph embeddings using a rotation mechanism. (**b**) Similarity matrix of attribute vectors for different aircraft.

Table 3. Some important attributes of aircrafts, forming a part of the attribute vectors.

Target	Aircraft Type	Disk Radar	Tail Type	Wing Type	Engine Type	Engine Position	Number of Engines	Length (m)	Wingspan (m)	Operating Empty Weight (t)
A321	Narrow-body airliner	No	Single vertical	Swept	Turbofan	Underwing	2	44.51	35.8	48.5
B737							2	39.47	35.8	41.413
ARJ21			T-tail			Rear-mounted	2	33.46	27.28	24.955
A330	Wide-body airliner		Single vertical			Underwing	2	58.82	60.3	119.6
B787							2	63	60	138
YXX	Airlifter						4	47	45	100
KXX	AEW	Yes	Twin-tail	Straight	Turboprop		2	18.14	25	25.401
ZXX	Helicopter	No	No tail	Rotor	Turboshaft	Top	3	19	18.9	7.095

3. Methods

3.1. Problem Definition

Let $x \in \mathcal{X} = \mathcal{X}^s \cup \mathcal{X}^u$ represent the deep features, where $\mathcal{X}^s$ and $\mathcal{X}^u$ correspond to the sets of features from seen and unseen classes, respectively, obtained from a pre-trained CNN. The corresponding labels are represented by $y \in \mathcal{Y} = \mathcal{Y}^s \cup \mathcal{Y}^u$, with $\mathcal{Y}^s = \{y_1, \ldots, y_K\}$ being the labels for K seen classes and $\mathcal{Y}^u = \{y_{K+1}, \ldots, y_{K+L}\}$ being the labels for L unseen classes, where $\mathcal{Y}^s \cap \mathcal{Y}^u = \emptyset$. Each class, whether seen or unseen, is described by their respective semantic embeddings $a_i \in \mathcal{A}, \forall i \in \mathcal{Y}^s \cup \mathcal{Y}^u$. The objective in GZSL is to learn the classifier $f_{gzsl} : \mathcal{X} \rightarrow \mathcal{Y}^s \cup \mathcal{Y}^u$. To achieve this, we began by training a feature generator using the seen class features $x_s \in \mathcal{X}^s$ and their associated semantic embeddings $a_i \in \mathcal{A}^s, \forall i \in \mathcal{Y}^s$. Then, the trained generator was employed to generate synthetic features x'_u for the unseen classes according to their semantic embeddings $a_i \in \mathcal{A}^u, \forall i \in \mathcal{Y}^u$. Subsequently, both the real seen class features x_s and the synthesized unseen class features x'_u were simultaneously fed into the model to train the GZSL classifier f_{gzsl}.

3.2. Overall Architecture

The overall architecture of the proposed SPGGN is shown in Figures 3 and 4, which display the feature-generation stage and the classification stage, respectively. The two-stage process is as follows:

- **Feature-generation stage.** As shown in Figure 3, first the HRRP features of the seen classes and the corresponding semantic embeddings are sent into the deep feature generation module (DFG module). Then, both the synthesized features and real features of the seen classes are input into the semantic embedding reconstruction module (SER module) and CSPR module for semantic embedding reconstruction and strong scattering point reconstruction, respectively, to enhance the robustness of the synthesizing features. Three modules are trained simultaneously to learn a high-performance feature generator.

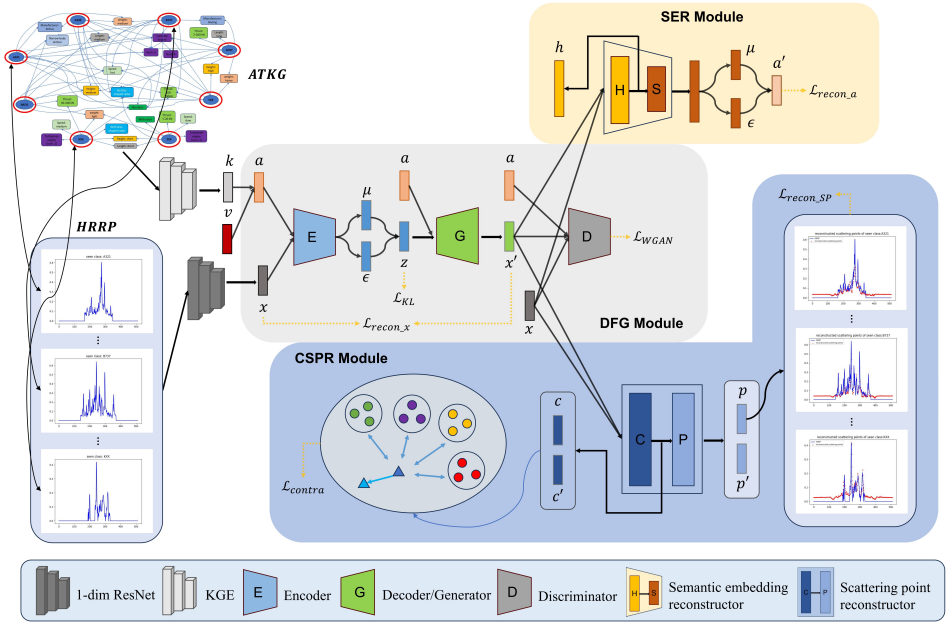


Figure 3. Illustration of feature-generation stage for SPGGN. For semantic embedding construction module, knowledge graph embeddings k and attribute vectors v were combined as semantic embeddings a. For CSPR module, strong scattering points of HRRP p were reconstructed from its deep features x to utilize physical properties.

- **Classification stage.** As shown in Figure 4, the learned generator is applied to synthesize features for unseen classes using the respective semantic embeddings. Subsequently, the synthesized unseen class features and real seen class features are sent into the SER module and CSPR module to generate richer features. The new discriminative features formed by feature fusion are sent to the classifier for training with the designed balancing loss. Finally, the trained classifier is used for inference on the test set.

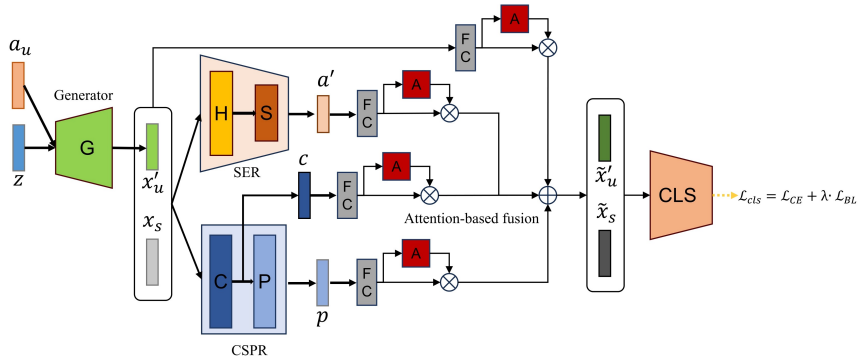


Figure 4. Illustration of classification stage.

3.3. Feature-Generation Stage

In the feature-generation stage, for the semantic embedding construction, we provide a detailed introduction in Section 2.3; for the HRRP deep feature extraction, the widely used ResNet [29] with one-dimensional convolution was adopted to extract the deep features x shown in Figure 3 after the preprocessing of the HRRP. The details of the other modules are as follows.

3.3.1. DFG Module

The generative-based method [30] was utilized as the baseline, integrating VAE and GAN by sharing the decoder and generator, with the aim of overcoming the problems of fuzzy reconstruction and unstable training [31].

A feature-generating VAE with class semantic embedding a as the condition was leveraged, comprising an encoder $E(x, a)$ (signified as E) and a decoder $G(z, a)$ (denoted as G, and also as the conditional generator of GAN). E encodes x to a latent embedding z, while G reconstructs x' from z. The optimization of the conditional VAE was performed as follows:

$$\mathcal{L}_{VAE} = \mathcal{L}_{KL} + \mathcal{L}_{recon_x} = D_{KL}(E(x,a)\|p(z \mid a)) - \mathbb{E}_{E(x,a)}[\log G(z,a)]. \tag{6}$$

Here, $\mathcal{L}_{KL}$ signifies the Kullback–Leibler divergence, $\mathcal{L}_{recon_x}$ means the reconstruction loss expressed as $-\log G(z, a)$, and $p(z \mid a) \sim \mathcal{N}(0, 1)$.

For GAN, f-WGAN [22] with class semantic embedding a as the condition was adopted. Leveraging the Wasserstein distance as the objective function mitigated the issue of mode collapse [21]. It included a generator $G(z, a)$ generating x' from noise z and a discriminator $D(x, a)$ (signified as D), indicating the level of authenticity or falsity. The f-WGAN is optimized by the following:

$$\mathcal{L}_{WGAN} = \mathbb{E}[D(x,a)] - \mathbb{E}[D(x',a)] - \gamma\mathbb{E}\left[\left(\|\nabla D(\hat{x},a)\|_2 - 1\right)^2\right], \tag{7}$$

where x' is synthesized by G, the last term signifies the gradient penalty to ensure Lipschitz continuity [32], and $\hat{x} = \tau x + (1 - \tau)x'$ with $0 \le \tau \le 1$.

3.3.2. SER Module

Although the DFG module leverages the strengths of VAE and GAN for generation, the lack of constraints in the synthesized HRRP features tends to cause significant differences between the synthesized features and real features. To alleviate this issue, we adopted the SER module shown in Figure 3 to reconstruct semantic embeddings. Because the synthesized features are generated through semantics, the model should be capable of reconstructing semantic embeddings [31,33].

The semantic reconstruction loss $\mathcal{L}_{rec_a}$ can be calculated by the ℓ_1 reconstruction error, expressed as follows:

$$\mathcal{L}_{rec_a} = \mathbb{E}\left[\left\|a'_{\text{real}} - a\right\|_1\right] + \mathbb{E}\left[\left\|a'_{\text{syn}} - a\right\|_1\right], \tag{8}$$

where a'_{real} represents the reconstructed semantic embeddings from the real features x, a'_{syn} is from the synthesized features x', and $a' = a'_{\text{real}} \cup a'_{\text{syn}}$. a denotes the true semantic embeddings. The semantic loss ensures the alignment between the real features and synthesized features in the semantic space, improving the quality and stability of the generated samples. Therefore, the generative model can ensure that the generated results for unseen classes are consistent with their category-specific semantic descriptions.

3.3.3. CSPR Module

After solving the semantic consistency problem, we needed to consider the limitations of HRRP features themselves. Using the deep network ResNet to extract the HRRP features resulted in an insufficient interpretability and robustness; hence, strong scattering points of the HRRP were introduced to increase the physical property constraints. As shown in Figure 5, the strong scattering points of the HRRP, namely shallow features (p or p'), were reconstructed from its deep features (x or x'). To reduce the impact of target aspects and better reconstruct the scattering points, class-aware contrastive learning (CACL) was applied to increase the inter-class separability and intra-class clustering [34].

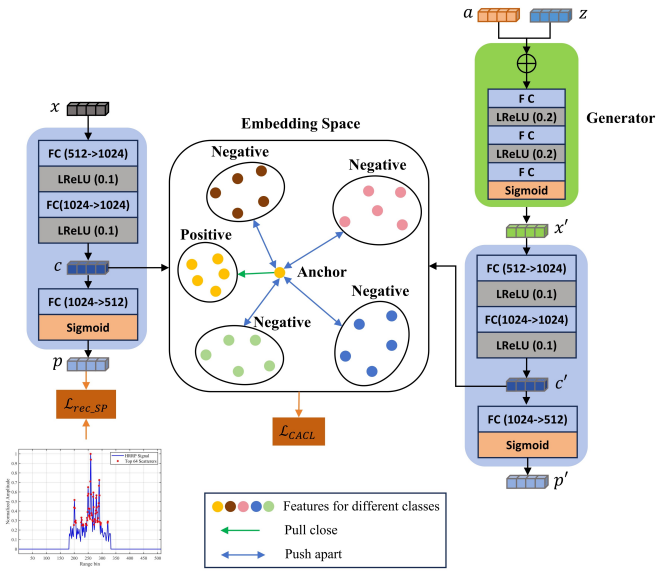


Figure 5. Illustration of the CSPR module.

CACL was employed in the middle layer, which defines positive and negative sample pairs by utilizing labels, aligning samples within the same category more closely while separating samples from different categories further in the feature space. In the contrastive embedding space, $c = C(x)$ and $c' = C(x')$ were achieved from real HRRP features x

and synthesized HRRP features x', respectively. For $c_i, c_i' \in \mathcal{C}$, consisting of both real and synthesized features, a $(K+1)$-way task was designed to identify the single positive example c^+ from K negative examples $\{c_1^-, \ldots, c_K^-\}$. The positive example c^+ was randomly chosen and shared the same class label as c_i (or c_i'), while all K negative samples had labels that were different from the label of c_i (or c_i'). The contrastive loss for c_i achieved from the real HRRP features x was calculated as follows:

$$\mathcal{L}_{cl}(c_i, c^+) = -\log \frac{\exp(c_i^\top c^+ / \tau_c)}{\exp(c_i^\top c^+ / \tau_c) + \sum_{k=1}^{K} \exp(c_i^\top c_k^- / \tau_c)}, \tag{9}$$

where τ_c is the temperature parameter and K signifies the number of negative examples. Although a higher K raises the computational complexity, it helps the model learn stronger discriminative features for different classes.

The contrastive loss for c_i' achieved from the synthesized HRRP features x' was similar to that shown in Equation (9). Therefore, the total loss CACL was calculated over the randomly selected pairs c_i (or c_i') and c^+ for both the real and synthesized examples:

$$\begin{aligned} \mathcal{L}_{CACL} &= \mathbb{E}_{c_i, c_i', c^+}\left[\mathcal{L}_{cl}(c_i, c^+) + \mathcal{L}_{cl}(c_i', c^+)\right] \\ &= \mathbb{E}_{c_i, c^+}\left[\mathcal{L}_{cl}(c_i, c^+)\right] + \mathbb{E}_{c_i', c^+}\left[\mathcal{L}_{cl}(c_i', c^+)\right], \end{aligned} \tag{10}$$

where $c_i \neq c^+$ and they pertain to the same seen class. This loss ensures the consistency between real and synthesized HRRP representations in the embedding space. It effectively mitigates the domain shift brought by generation while encouraging class-specific compactness. By optimizing such objectives, the learned feature space becomes more structured and robust to aspect variations, benefiting GZSL tasks.

To calculate the reconstruction error of strong scattering points, we adopted the following formula:

$$\mathcal{L}_{rec_SP} = \mathbb{E}^{weighted}\left[\|p - p^t\|_1\right] = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{(p_i - p_i^t)^2}{\sqrt{\sum_{j=1}^{N}\left(p_j - p_j^t\right)^2}}\right) \cdot |p_i - p_i^t|, \tag{11}$$

where p represents the strong scattering points reconstructed from the real HRRP features x, p^t represents the true values of strong scattering points, and N represents the batch size. The weighted reconstruction loss was adopted to emphasize large reconstruction errors that typically correspond to strong scattering points with a higher physical significance. Unlike standard L1 or L2 losses that treat all dimensions equally, the proposed weighting term normalizes the squared error within each sample and assigns larger weights to components with greater deviations. This adaptive normalization makes the loss focus on accurately restoring salient scattering responses, which are crucial for representing the HRRP's structural characteristics.

3.3.4. Total Loss for Feature Generation

The total loss function in the feature-generation stage is the sum of the five types of losses mentioned above, as shown in the following equation:

$$\mathcal{L}_{FG} = \mathcal{L}_{VAE} + \mathcal{L}_{WGAN} + \mathcal{L}_{rec_a} + \mathcal{L}_{CACL} + \mathcal{L}_{rec_SP}. \tag{12}$$

The training process of the feature-generation stage is conducted in two steps:

- The first step focuses on training the discriminator D in the DFG module and training the SER and CSPR modules on real HRRP features x:

$$\mathcal{L}_{fg1} = \mathcal{L}_{WGAN} + \alpha_1 \mathcal{L}_{rec_SP} + \alpha_2 \mathbb{E}\left[\left\|a'_{\text{real}} - a\right\|_1\right] + \alpha_3 \mathbb{E}_{c_i,c^+}\left[\mathcal{L}_{cl}\left(c_i, c^+\right)\right], \quad (13)$$

  where $\mathbb{E}\left[\left\|a'_{\text{real}} - a\right\|_1\right]$ represents the semantic reconstruction error for the real features x, part of $\mathcal{L}_{rec_a}$ in Equation (8). $\mathbb{E}_{c_i,c^+}[\mathcal{L}_{cl}(c_i, c^+)]$ represents the contrastive loss for the real features x, part of $\mathcal{L}_{CACL}$ shown in Equation (10). α_1, α_2, and α_3 are the weighting factors.
- The second step focuses on training the encoder E and generator/decoder G in the DFG module and training the SER and CSPR modules using the HRRP features x' synthesized by the generator:

$$\mathcal{L}_{fg2} = \mathcal{L}_{VAE} - \beta_1 \mathbb{E}[D(x', a)] + \beta_2 \mathbb{E}\left[\left\|a'_{\text{syn}} - a\right\|_1\right] + \beta_3 \mathbb{E}_{c'_i,c^+}\left[\mathcal{L}_{cl}\left(c'_i, c^+\right)\right], \quad (14)$$

  where $\mathbb{E}[D(x', a)]$ denotes the score achieved by D on x' to indicate the quality of the fake samples synthesized by G. $\mathbb{E}\left[\left\|a'_{\text{syn}} - a\right\|_1\right]$ denotes the semantic reconstruction error for x', another part of $\mathcal{L}_{rec_a}$. $\mathbb{E}_{c'_i,c^+}\left[\mathcal{L}_{cl}\left(c'_i, c^+\right)\right]$ denotes the contrastive loss for x', another part of $\mathcal{L}_{CACL}$.

Before training the generator G, we first trained the discriminator D several times according to [32]. This approach ensures that the discriminator can accurately estimate the Wasserstein distance, thereby effectively guiding the update of the generator and the SER module and CSPR module through the HRRP features synthesized by the generator.

3.4. Classification Stage

In the classification stage, the trained generator G is leveraged to synthesize the features for the unseen classes, which are then fused with the embeddings obtained by the SER module and CSPR module to obtain more distinctive features. When using both the fused real features of the seen classes and the fused synthesized features of the unseen classes to train the classifier, a balancing loss is added to mitigate the bias towards the seen classes. Finally, the trained classifier is applied to perform inference on the test set, including seen and unseen class samples.

3.4.1. Feature Fusion and Classifier Training

After the generator G is trained using x_s, it is employed to synthesize HRRP features x'_u for unseen classes based on their corresponding semantic embeddings a_u. To construct more informative and discriminative representations, we adopted an attention-based fusion strategy to integrate multiple heterogeneous embeddings, as shown in Figure 4.

Let $x \in \mathbb{R}^{d_x}$, $a' \in \mathbb{R}^{d_a}$, $c \in \mathbb{R}^{d_c}$, and $p \in \mathbb{R}^{d_p}$ represent the deep feature, semantic embedding, contrastive embedding, and scattering embedding, respectively. These components are first projected into a shared latent space $\mathbb{R}^d$ via individual fully connected (FC) layers:

$$\hat{f}_x = W_x x + b_x, \hat{f}_a = W_a a' + b_a, \hat{f}_c = W_c c + b_c, \hat{f}_p = W_p p + b_p, \quad (15)$$

where $W_{\{\cdot\}} \in \mathbb{R}^{d \times d_\cdot}$ and $b_{\{\cdot\}} \in \mathbb{R}^d$ are trainable parameters. Next, a shared attention mechanism is applied to adaptively weight each projected feature:

$$\alpha_i = \frac{\exp(w^\top \tanh(\hat{f}_i))}{\sum_{j=1}^{4} \exp(w^\top \tanh(\hat{f}_j))}, \quad i \in \{x, a', c, p\} \quad (16)$$

where $w \in \mathbb{R}^d$ is a learnable vector. The final fused feature is obtained by a weighted sum:

$$\widetilde{x} = \alpha_x \hat{f}_x + \alpha_{a'} \hat{f}_{a'} + \alpha_c \hat{f}_c + \alpha_p \hat{f}_p, \quad \hat{f}_x, \hat{f}_{a'}, \hat{f}_c, \hat{f}_p \in \mathbb{R}^d. \tag{17}$$

Accordingly, we obtained $\widetilde{x}_s$ for the seen classes and $\widetilde{x}'_u$ for the unseen classes through this attention-based fusion.

Then, classifier CLS needs to be optimized using the new features $\widetilde{x}_s$ and $\widetilde{x}'_u$ based on cross-entropy loss:

$$\mathcal{L}_{CE} = \mathbb{E}\left[-\sum_{c=1}^{K+L} y_i^c \log(\hat{y}_i^c) \right] = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{K+L} y_i^c \log\left(\frac{\exp(s_i^c)}{\sum_{c' \in \mathcal{Y}} \exp\left(s_i^{c'}\right)} \right), \tag{18}$$

where $c \in \mathcal{Y}$ represents the classes, including K seen classes and L unseen classes. y_i^c represents the label of the ith sample on class c, with a value of 1 on the true class and 0 on the other classes. $\hat{y}_i^c$ is the predicted probability, derived by applying the softmax function to the class scores s_i^c.

Although our method addresses the GZSL problem by generating samples for unseen classes, effectively converting it into a traditional supervised learning task and thus partially abating the bias towards seen classes, the bias problem still exists due to limitations in the semantic representation, the quality of the generated samples [22], the optimization objectives of the model [33], etc. After combining the knowledge graph embeddings with the attribute vectors to form a more comprehensive semantic representation and adopting the DFG module, SER module, and CSPR module to enhance the quality of the generated samples, we needed to consider improving the classifier objective function. Inspired by [35,36], we designed a balancing loss (BL) to adjust the predicted probability for the seen and unseen classes and mitigate the bias problem:

$$\mathcal{L}_{BL} = \mathbb{E}\left[-\log\left(\sum_{c \in \mathcal{Y}^u} \tilde{y}_i^c \right) \right] = -\frac{1}{N} \sum_{i=1}^{N} \log\left(\sum_{c \in \mathcal{Y}^u} \frac{\exp(\tilde{s}_i^c)}{\sum_{c' \in \mathcal{Y}^u} \exp\left(\tilde{s}_i^{c'}\right)} \right), \tag{19}$$

where $\tilde{y}_i^c$ is the adjusted prediction probability that sample i falls under unseen class $c \in \mathcal{Y}^u$, achieved by employing softmax on the adjusted class scores:

$$\tilde{s}_i^c = s_i^c + \eta f_{sig}(c), \tag{20}$$

where $f_{sig}(c)$ is a sign function that equals 1 for $c \in \mathcal{Y}^u$ and -1 for $c \in \mathcal{Y}^s$, and $\eta > 0$ is a margin hyper-parameter (e.g., 0.5). For $c \in \mathcal{Y}^u$, $\tilde{s}_i^c = s_i^c + \eta$. The reason for using adjusted class scores $\tilde{s}_i^c$ in the BL shown in Equation (19) was to increase the probability of the samples being predicted as unseen classes $\tilde{y}_i^c (c \in \mathcal{Y}^u)$, while not excessively increasing the scores of the samples being classified as unseen classes $s_i^c (c \in \mathcal{Y}^u)$, in order to avoid the misclassification of the seen class samples.

Therefore, the overall classification loss is as follows:

$$\mathcal{L}_{cls} = \mathcal{L}_{CE} + \lambda \mathcal{L}_{BL}, \tag{21}$$

where λ is the weighting factors. Of course, we optimized Equation (21) on both the real seen class features $\widetilde{x}_s$ and the synthesized unseen class features $\widetilde{x}'_u$ simultaneously.

3.4.2. Inference on Test Set

Finally, for inference on the test set, including both seen and unseen class samples, $\widetilde{x}$ was first obtained through attention-based feature fusion according to Equation (17);

then, $\widetilde{x}$ was input into the trained classifier CLS and the class with the highest calibrated score $\tilde{s}_i^c$ was considered the predicted label for the test sample:

$$c^* = \text{argmax}_{c \in \mathcal{Y}} \tilde{s}_i^c = \text{argmax}_{c \in \mathcal{Y}} (s_i^c + \eta f_{sig}(c)). \quad (22)$$

By appropriately increasing the scores of samples classified into unseen classes and decreasing the scores of samples classified into seen classes, it balances the predicted outputs of the seen and unseen classes.

4. Results and Discussion

4.1. Experimental Setting

4.1.1. HRRP Features and Semantic Embeddings

ResNet-18 [29] trained only on the seen classes was used to extract the deep features of the HRRP, e.g., it was only trained on five seen classes for the S5U3 dataset. A total of 64 strong scattering points were extracted as the ground truth for the CSPR module. To avoid losing useful information due to the dimensionality reduction in [24], we still used 512-dimensional vectors for the HRRP scattering features, retaining strong scattering points and setting the remaining values to zero. In order to compare the impact of different forms of semantic embeddings, in addition to the knowledge graph embeddings and attribute vectors introduced in Section 2.3, we also obtained sentence embeddings through the pre-trained model Sentence-BERT [37].

4.1.2. Evaluation Metrics

In line with the GZSL evaluation protocol, we calculated the average accuracy for both the seen classes $\mathcal{Y}^s$ and the unseen classes $\mathcal{Y}^u$, signified as S and U, respectively. More importantly, to capture the trade-off between both accuracies, their harmonic mean $H = 2 \times S \times U / (S + U)$ was computed to assess the GZSL performance. A higher harmonic mean reflected a better performance across both the seen and unseen classes.

4.1.3. Implementation Details

In the DFG module, the encoder E, generator/decoder G, and discriminator D are multi-layer perceptrons (MLPs) that incorporate a hidden layer of 1024 units with LeakyReLU activation. The latent embedding z and semantic embedding a are assigned equal dimensions. The semantic embedding reconstructor is also a two-layer MLP with 1024 hidden units and LeakyReLU activation to match the structure of E and G. It generates a' via the reparametrization trick [20] on μ and ϵ, just like the encoder E does. The scattering point reconstructor is an MLP that incorporates a 512-unit hidden layer with LeakyReLU activation and an output layer with Sigmoid activation to generate strong scattering points after normalization. The network for the feature-generation stage was trained with the Adam optimizer. The gradient penalty coefficient γ in the WGAN loss was fixed at 10. The weighting factors were $\beta_1 = 10$ and $\alpha_2 = \beta_2 = 10^{-3}$. $\alpha_1 = 10^{-2}$ and $\alpha_3 = \beta_3 = 10^{-2}$ for the S5U3 dataset, while $\alpha_1 = 10^{-3}$ and $\alpha_3 = \beta_3 = 10^{-3}$ for the S6U2 dataset. The classifier CLS was also trained with the Adam optimizer. The balancing loss weight λ was set to 10^{-4} and the margin η was set to 0.5.

4.2. Comparison with State-of-the-Art Methods

4.2.1. Average Top-1 Accuracy of SOTA Methods

We compared our inductive GZSL method SPGGN with some state-of-the-art (SOTA) methods, also under inductive settings, on both the S5U3 and S6U2 datasets, as shown in Table 4. Our method achieved the highest U and the highest H on both datasets. On the S5U3 dataset, H obtained by our method reached 72.51%, outperforming the second-best

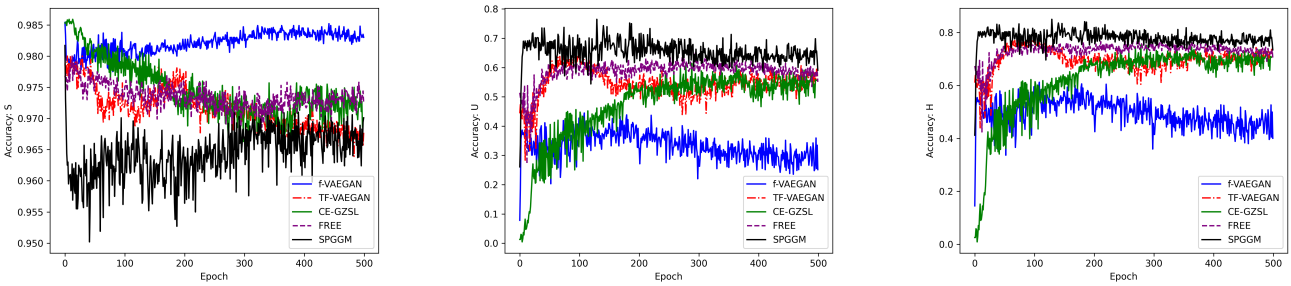
SDGZSL [38] by 9.54%. Similarly, on the S6U2 dataset, our method reached an H value of 84.98%, exceeding SDGZSL by 5.52%. Compared with our baseline [30], on the S5U3 dataset, U and H increased by 30.29% and 28.44%, respectively, while S only decreased by 2.81%. Similarly, on the S6U2 dataset, U and H increased by 31.99% and 23.77%, respectively, while S only decreased by 2.47%. The reason for the slight decrease in S is that we adopted the BL to mitigate the bias towards seen classes, which greatly improves the accuracy of unseen classes while sacrificing a little bit of the seen class accuracy.

Table 4. Comparison with SOTA GZSL methods on both S5U3 and S6U2. The best results are in bold and the second best results are underlined.

Method	S5U3			S6U2		
	S	**U**	**H**	**S**	**U**	**H**
LisGAN [39]	**98.63%**	12.46%	22.13%	**98.50%**	28.16%	43.80%
TCN [40]	95.67%	21.81%	35.52%	97.13%	29.60%	45.37%
f-VAEGAN [30]	<u>97.45%</u>	28.48%	44.07%	<u>98.11%</u>	44.48%	61.21%
TF-VAEGAN [31]	96.68%	37.66%	54.21%	97.31%	64.92%	77.88%
CE-GZSL [41]	96.50%	41.48%	58.02%	96.85%	61.38%	75.14%
$(AR)^2$ [42]	96.82%	40.15%	56.76%	97.04%	62.37%	75.93%
FREE [43]	96.31%	45.71%	62.00%	96.87%	64.93%	77.75%
SDGZSL [38]	95.67%	<u>46.93%</u>	<u>62.97%</u>	95.94%	<u>67.82%</u>	<u>79.46%</u>
Ours (SPGGN)	94.64%	**58.77%**	**72.51%**	95.64%	**76.47%**	**84.98%**

4.2.2. Accuracy Curves Varying with the Number of Training Epochs

In Figure 6a, the black curve representing our method sharply drops at the beginning and then steadily increases; the decrease in accuracy S from epoch 0 to epoch 1 was due to the balancing loss. Although our accuracy S was slightly lower than that of the other methods, the difference was minimal. Figure 6b shows the accuracy U trend over epochs; our method's curve increases significantly at first, fluctuates greatly in the middle, and gradually stabilizes after epoch 200. This is similar to the trend of our baseline shown in the blue curve, but with a 30% higher accuracy. Figure 6c shows the variation in the accuracy H with the epoch, which is similar to Figure 6b, but with smaller fluctuations, indicating that H is more affected by U due to the smaller variation in the accuracy S. In summary, our model had the highest harmonic mean accuracy, fast convergence, and the best overall performance.



(**a**) (**b**) (**c**)

Figure 6. The accuracy curves of different methods, varying with the number of training epochs, on the validation set of the S6U2 dataset. (**a**) S; (**b**) U; (**c**) H.

4.2.3. Efficiency and Model Complexity Analysis

We analyzed the computational efficiency and model complexity of different methods on the S6U2 dataset. Table 5 reports the number of parameters, the training cost, and the inference efficiency of each method. All of the models were trained under identical settings with a fixed batch size of 128 for a fair comparison. The model complexity was measured by

the number of learnable parameters (Params), while the training efficiency was evaluated by the total time required to reach the best validation performance (train time to best). The parameters and training time of the classifier were excluded, with the analysis focusing on the representation learning and generation components. The inference efficiency was measured as the average latency per 1000 samples (inference time per 1k samples).

Table 5. Comparison of model size and computational efficiency on the S6U2 dataset.

Method	Params (M)	Train Time to Best (s)	Inference Time per 1k Samples (ms)
LisGAN [39]	1.700	4457.60	0.8847
TCN [40]	1.268	15.03	7.9077
f-VAEGAN [30]	3.053	671.22	1.8078
TF-VAEGAN [31]	5.894	1543.68	4.5759
CE-GZSL [41]	1.364	9263.80	1.3352
$(AR)^2$ [42]	1.836	418.73	1.6429
FREE [43]	4.012	3466.26	5.2951
SDGZSL [38]	8.108	61.48	2.8135
Ours (SPGGN)	4.536	539.99	6.8306

As shown in Table 5, lightweight discriminative models generally converge faster with a lower training cost, whereas generative or adversarial-based methods require a substantially longer training time due to their complex optimization processes. With a moderate number of parameters, our method achieved a competitive training and inference efficiency, indicating a favorable trade-off between model complexity and computational cost. Importantly, the inference time of SPGGN remains at the millisecond level for 1k samples, indicating that our method is still applicable to practical HRRP recognition tasks where efficiency is a concern.

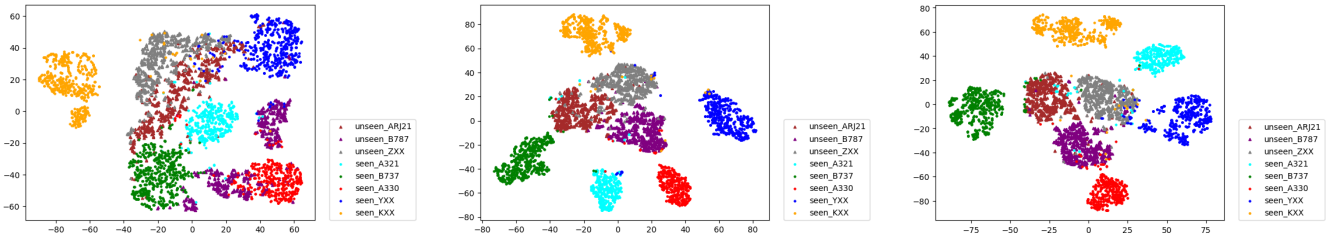
4.3. Comparison of Different Semantic Embeddings

Firstly, we compared the impact of different forms of semantics on the accuracy. Table 6 shows the results on the S5U3 and S6U2 datasets. It can be seen that the combination of knowledge graph embeddings and attribute vectors as semantics yielded the best performance on both datasets, confirming the effectiveness of the semantic construction method introduced in Section 2.3.

Table 6. Comparison of different forms of semantic embeddings on S5U3 and S6U2 datasets. Best results are in bold.

Semantic Type	S5U3			S6U2		
	S	**U**	**H**	**S**	**U**	**H**
Attribute Vectors	78.29%	55.89%	65.22%	91.56%	62.71%	74.44%
Sentence Embeddings	**97.06%**	29.23%	44.93%	96.39%	49.62%	65.52%
Knowledge Graph Embeddings	94.70%	55.04%	69.62%	95.93%	70.70%	81.41%
Sentence Embeddings + Attribute Vectors	96.52%	29.67%	45.39%	**96.47%**	52.21%	67.76%
Knowledge Graph Embeddings + Attribute Vectors	94.64%	**58.77%**	**72.51%**	95.64%	**76.47%**	**84.98%**

Secondly, to further examine how different semantic forms influence the feature space, we compared the distributions produced by attribute vectors, knowledge graph embeddings, and their combination. To avoid the interference of the CSPR module, we used a simplified model consisting only of the DFG and SER modules. As shown in Figure 7, the attribute vectors led to dispersed class clusters with large intra-class variation, while knowledge graph embeddings produced more compact structures, but still showed overlaps between unseen classes. In contrast, the combined semantics yielded both tighter intra-class distributions and clearer inter-class margins, indicating that the complementary information from the attributes and knowledge graphs jointly improved the separability of the generated features.



(**a**) (**b**) (**c**)

Figure 7. The t-SNE visualization of the synthesized unseen class features and real seen class features of the S5U3 dataset using different forms of semantic embeddings. (**a**) Using attribute vectors; (**b**) using knowledge graph embeddings; and (**c**) Using the combination of knowledge graph embeddings and attribute vectors.

Finally, we compared the effect of different semantic compositions, namely the importance of various attributes. To ensure the reliability of the experiments, we adopted two forms of semantics: knowledge graph embeddings and the combination of knowledge graph embeddings and attribute vectors. The results on the S5U3 dataset are shown in Table 7. Using all the selected attributes yielded the best results on both forms of semantics, validating the rationality of our semantic construction. In addition, attributes like the disk radar and engine type have less impact on HRRP recognition, while the weight, wingspan, and aircraft type are crucial, followed by the engine count and position. This is also in line with reality, proving the effectiveness of our constructed semantics.

Table 7. Comparison of the impact of different semantic compositions on the accuracy on the S5U3 dataset, including the form of knowledge graph embeddings and the combination of knowledge graph embeddings and attribute vectors. The best results are in bold.

Semantic Composition	KGE			KGE + Attribute Vectors		
	S	U	H	S	U	H
No weight	95.15%	25.68%	40.45%	94.79%	28.86%	44.24%
No wingspan	**96.50%**	30.32%	46.14%	**96.17%**	33.92%	50.15%
No aircraft type	95.09%	35.20%	51.38%	94.65%	36.54%	52.72%
No number of engines	96.20%	34.07%	50.32%	95.13%	38.05%	54.36%
No engine position	95.68%	41.81%	58.19%	94.91%	41.59%	57.84%
No tail type	94.09%	37.41%	53.54%	94.97%	41.86%	58.10%
No length	93.74%	42.97%	58.93%	94.92%	45.69%	61.69%
No wing type	95.56%	43.47%	59.76%	95.16%	46.02%	62.03%
No engine type	95.50%	41.39%	57.75%	95.42%	46.54%	62.56%
No presence of disk radar	94.20%	47.04%	62.74%	92.98%	49.54%	64.64%
All attributes	94.70%	**55.04%**	**69.62%**	94.64%	**58.77%**	**72.51%**

4.4. Ablation Study

To verify the contribution of each component in our method, we conducted ablation studies, and the overall results are shown in Table 8.

Table 8. Ablation study for different components of our model SPGGN on both S5U3 dataset and S6U2 dataset. Best results are in bold.

ID	E	G	D	SER	CSPR		$\mathcal{L}_{BL}$	S5U3			S6U2		
					$\mathcal{L}_{CACL}$	$\mathcal{L}_{rec_SP}$		S	U	H	S	U	H
1	✓	✓	✓					97.45%	28.48%	44.07%	**98.11%**	44.48%	61.21%
2	✓	✓	✓		✓	✓		**97.50%**	32.02%	48.21%	97.90%	50.99%	67.05%
3		✓	✓	✓	✓	✓		97.40%	33.39%	49.73%	97.58%	47.62%	64.01%
4	✓	✓		✓	✓	✓		96.95%	40.88%	57.51%	97.54%	53.81%	69.36%
5	✓	✓	✓	✓				96.80%	39.95%	56.56%	97.39%	59.48%	73.86%
6	✓	✓	✓	✓	✓			96.54%	42.10%	58.63%	97.55%	60.53%	74.70%
7	✓	✓	✓	✓		✓		96.37%	43.01%	59.47%	97.32%	60.29%	74.46%
8	✓	✓	✓	✓	✓	✓		96.86%	**44.11%**	**60.61%**	97.30%	**63.50%**	**76.85%**
9	✓	✓	✓				✓	94.56%	53.77%	68.55%	96.19%	69.93%	80.99%
10	✓	✓	✓		✓	✓	✓	94.48%	55.03%	69.55%	96.17%	71.68%	82.14%
11		✓	✓	✓	✓	✓	✓	**95.71%**	50.26%	65.91%	**96.72%**	67.29%	79.37%
12	✓	✓		✓	✓	✓	✓	95.01%	55.80%	70.31%	95.53%	73.08%	82.81%
13	✓	✓	✓	✓			✓	94.48%	56.24%	70.51%	95.58%	71.45%	81.77%
14	✓	✓	✓	✓	✓		✓	95.00%	57.24%	71.43%	95.98%	74.13%	83.65%
15	✓	✓	✓	✓		✓	✓	94.74%	57.22%	71.35%	95.31%	74.46%	83.61%
16	✓	✓	✓	✓	✓	✓	✓	94.64%	**58.77%**	**72.51%**	95.64%	**76.47%**	**84.98%**

4.4.1. DFG Module

To demonstrate the necessity of the combination, we conducted experiments using only *G* and *D*, and only *E* and *G*. The third and fourth rows of Table 8 show the results without the BL. Similarly, the 11th and 12th rows show the results using the BL. It is evident that using only VAE yielded better results than using only GAN; nevertheless, both failed to yield good experimental results compared to using both.

4.4.2. SER Module

We prove the validity of the SER module in the second and eighth rows of Table 8, without using the BL, and similarly in the tenth and sixteenth rows with the BL. The adoption of the SER module resulted in an improvement in *H* by 2.96% on the S5U3 dataset and 2.84% on the S6U2 dataset when using the BL. It can be observed that, regardless of the dataset and whether or not the BL was used, using the SER module yielded better results, proving the necessity of semantic reconstruction.

4.4.3. CSPR Module

Firstly, we present the reconstruction results of the HRRP's strong scattering points from deep features for the S6U2 dataset. Figures 8 and 9 show the results of randomly selected seen and unseen class samples in the test set, respectively. We found that the scattering points of the seen classes were well reconstructed, both in terms of their structure and size. The structure of unseen classes was also well reflected, even without participating in the training. Overall, the CSPR module effectively reconstructed strong scattering points, regardless of changes in the target aspects.

Secondly, we present feature visualizations on both datasets to compare the representations before and after applying the CSPR module, as shown in Figure 10. Without CSPR, the features exhibited intra-class dispersion due to variations in the target aspects. After training with the CSPR module and performing feature fusion, the intra-class distributions became more compact. This effect was especially clear on the S6U2 dataset: when comparing Figure 10b,d, the unseen classes (shown in purple and gray) were dispersed when CSPR was not used, whereas with CSPR, they formed much tighter clusters, despite the lack of training samples and the presence of aspect variations. These results demonstrate that the CACL-based CSPR module effectively mitigates the influence of target aspects.

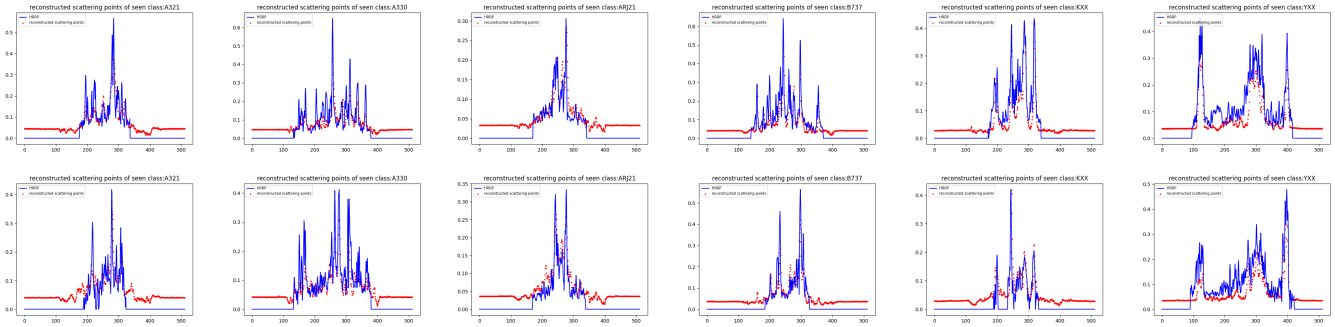


Figure 8. Reconstruction results of the HRRP's strong scattering points for the seen classes of the S6U2 dataset. Each column shows a class, with samples from varying aspects.

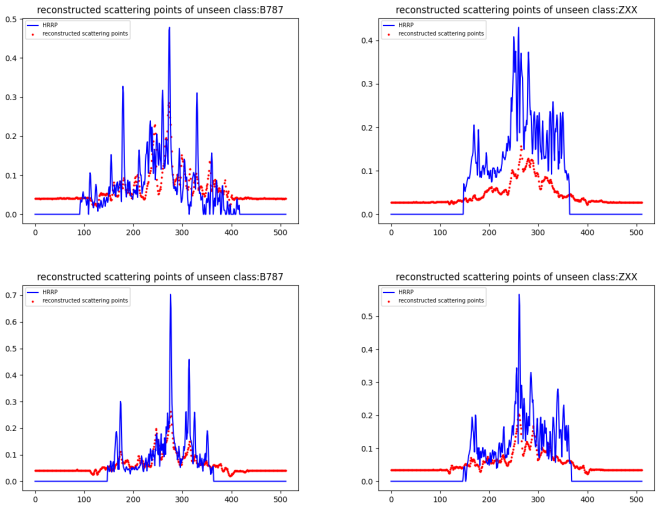


Figure 9. Reconstruction results of the HRRP's strong scattering points for the unseen classes of the S6U2 dataset. Each column shows a class, with samples from varying aspects.

Finally, according to the 13th and 16th rows of Table 8, the CSPR module improved H by 2% on the S5U3 dataset and by 3.21% on the S6U2 dataset when using the BL. In these cases, S remained nearly unchanged, while U improved by 2.53% on the S5U3 dataset and by 5.02% on the S6U2 dataset. These results confirm the effectiveness of the CSPR module, particularly in enhancing the accuracy for unseen classes. A comparison between Rows 5 and 8 in Table 8 demonstrates similar improvement effects. Apart from comparing the inclusion or exclusion of the entire CSPR module, we also conducted experiments on only using the contrastive loss $\mathcal{L}_{CACL}$ and only using the reconstruction loss $\mathcal{L}_{rec_SP}$. The results reveal the effectiveness of CACL and strong scattering point reconstruction, both of which are indispensable. The two losses operate at different, but mutually reinforcing, levels. $\mathcal{L}_{CACL}$ acts in the contrastive embedding space to enforce class-aware alignment to enhance the intra-class similarity and inter-class discriminability. It encourages the learned representations to be more structured and robust to target aspect variations. In contrast, $\mathcal{L}_{rec_SP}$ focuses on the reconstruction of strong scattering points to inject physical priors of the HRRP into the learning process. When jointly optimized, $\mathcal{L}_{CACL}$ provides global and class-consistent discrimination, while $\mathcal{L}_{rec_SP}$ enforces local and physical constraints, which explains why their combination leads to larger performance gains.

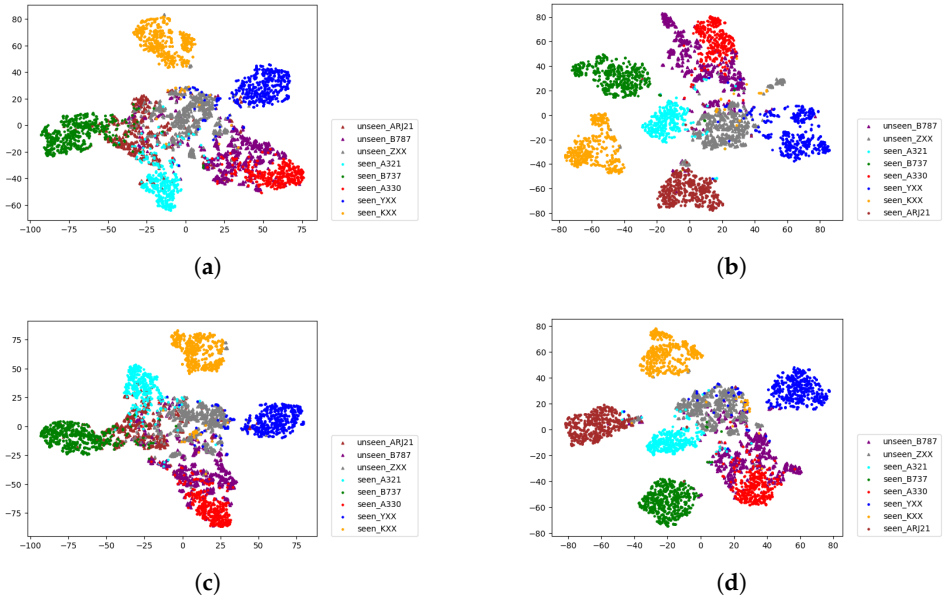


Figure 10. The t-SNE visualization of features before and after using the CSPR module on both the S5U3 and S6U2 datasets. (**a**) Before using the CSPR module on the S5U3 dataset; (**b**) before using the CSPR module on the S6U2 dataset; (**c**) after using the CSPR module on the S5U3 dataset; and (**d**) after using the CSPR module on the S6U2 dataset.

4.4.4. Classifier Module

Feature fusion: Figure 11 compares the feature visualization on the S6U2 dataset before and after feature fusion. After feature fusion, the degree of clustering for each class was higher and the distinguishability among classes was enhanced, demonstrating the effectiveness of our approach in fusing deep features, shallow features, and semantic features of the HRRP. In addition, we compared the accuracies before and after feature fusion with and without the BL, as shown in Figure 12. The results show that feature fusion significantly improved both U and H across both datasets.

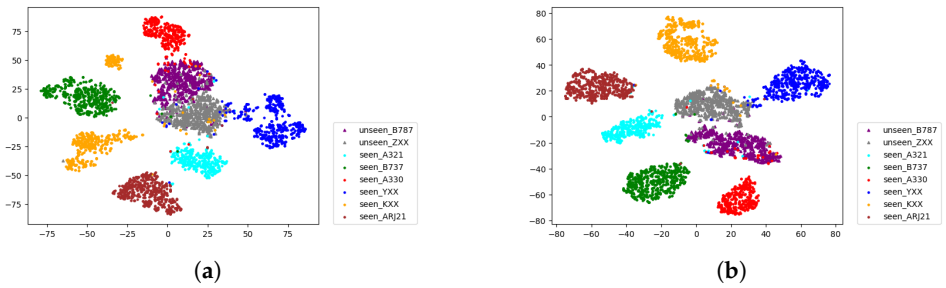
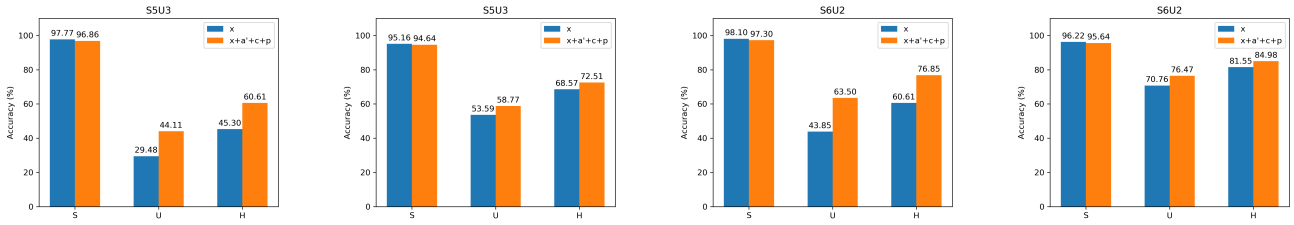


Figure 11. The t-SNE visualization of the synthesized unseen class features and the real seen class features of the S6U2 dataset before and after feature fusion. (**a**) Before feature fusion; (**b**) after feature fusion.

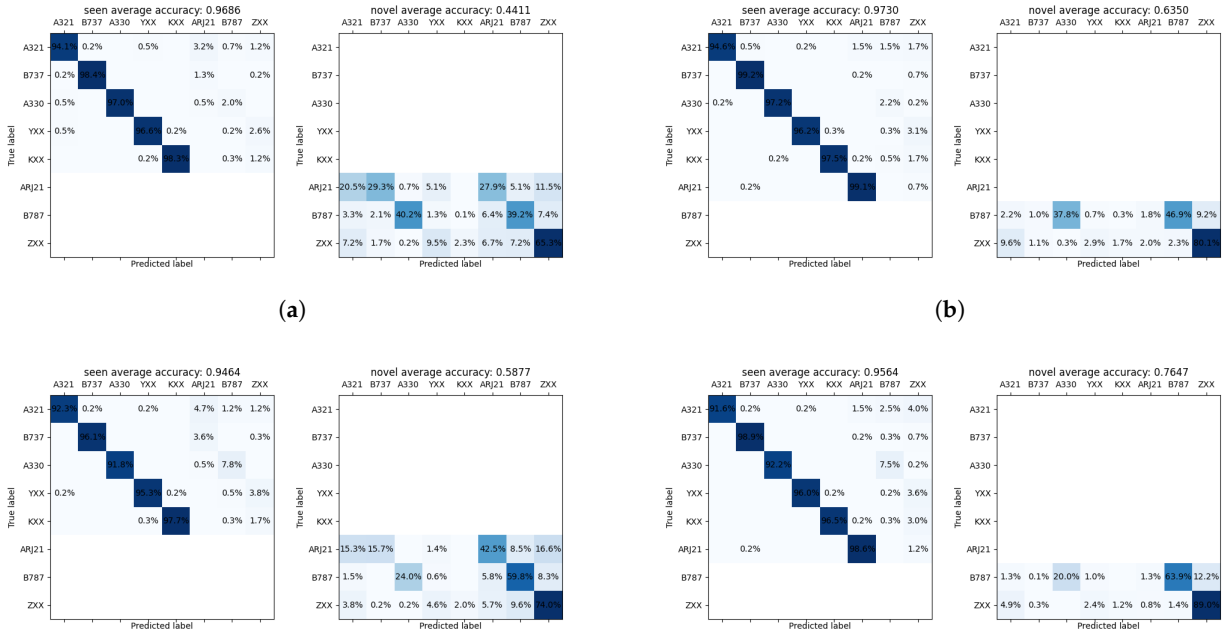
Balancing loss: For the same model composition, using the BL greatly improved the classification accuracy on both datasets. Among them, the H of our method improved by 11.9% and 8.13%, respectively, on the S5U3 dataset and S6U2 dataset, as shown by the 8th and 16th rows of Table 8. We further plotted the confusion matrices, as shown in Figure 13. By comparing Figure 13a,c, it can be seen that S decreased by 2.22%, but still remained above 94%, while U increased by 14.66%. In this case, the accuracy of the ARJ21, B787, and ZXX classes increased by 14.6%, 20.6%, and 8.7%, respectively. For each unseen class, the BL reduced the probability of being recognized as a seen class, especially for the B787,

which is very similar to the seen class A330. From the right sub-figures of Figure 13a,c, it can be seen that, before the BL, the probability of predicting B787 as A330 was higher than predicting it correctly as B787, while using the BL resulted in a significant increase. The results on the S6U2 dataset (Figure 13b,d) are similar to those on the S5U3 dataset.



(**a**) (**b**) (**c**) (**d**)

Figure 12. Comparison before and after feature fusion with and without BL on both datasets. (**a**) Accuracy on S5U3 dataset without BL; (**b**) accuracy on S5U3 dataset with BL; (**c**) accuracy on S6U2 dataset without BL; and (**d**) accuracy on S6U2 dataset with BL.



(**a**) (**b**)

(**c**) (**d**)

Figure 13. Confusion matrices before and after using the BL on both the S5U3 and S6U2 datasets. For each figure, the classification results of the seen classes are on the left and the results of the unseen classes are on the right. (**a**) Confusion matrix of the S5U3 dataset without the BL; (**b**) confusion matrix of the S6U2 dataset without the BL; (**c**) confusion matrix of the S5U3 dataset with the BL; and (**d**) confusion matrix of the S6U2 dataset with the BL.

4.4.5. Overall Analysis

In addition to Table 8, Figure 14 presents the feature visualization using models composed of different modules, taking the S5U3 dataset as an example. It can be observed that the addition of the SER module enhanced the separability between the synthesized unseen classes and the real seen classes, and the addition of the CSPR module further strengthened the inter-class separability and intra-class compactness of the synthesized features of unseen classes, indicating the good performance of SPGGN. In addition, Figure 14d and Figure 14e respectively show the distribution of real features before and after using SPGGN, including the seen and unseen class samples. After using our model, the distance between seen classes increases, and both the seen and unseen classes exhibited a significant

increase in intra-class clustering. Furthermore, even if the target aspects of the same class are different, our model can still cluster these features together, indicating that our model is relatively insensitive to target aspects, proving the effectiveness of CSPR and feature fusion.

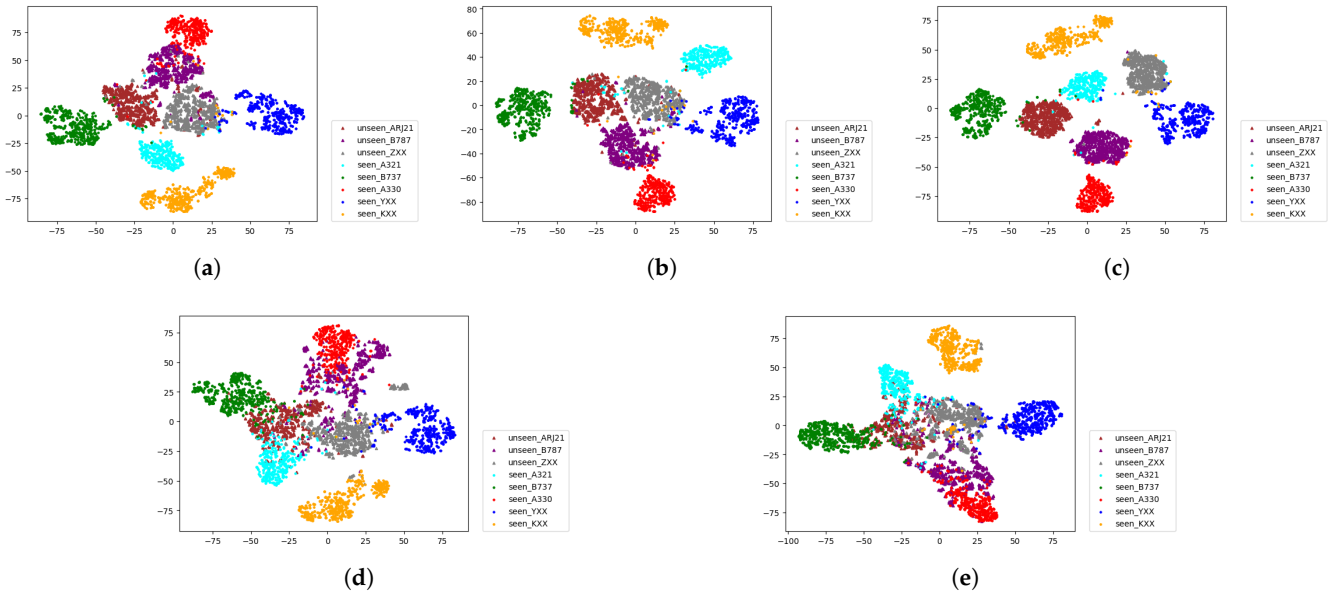


Figure 14. t-SNE visualization of features of S5U3 test set. (**a**–**c**) show real seen class features and synthesized unseen class features using different models. (**d**,**e**) show real features before and after using our model, SPGGN. (**a**) Only using DFG module; (**b**) using DFG module and SER module; and (**c**) using DFG module, SER module, and CSPR module (i.e., SPGGN).

5. Conclusions

This paper proposes SPGGN for HRRP GZSR, which generate deep features conditioned on semantics as the basic module, and develops CSPR module to reconstruct strong scattering points based on CACL to enhance physical constraints. In terms of semantic construction, we combined a knowledge graph with attribute vectors to characterize different categories. In the feature-generation stage, we reconstructed scattering features from the deep features to improve the interpretability and generalization, and applied CACL to reduce the impact of the variation in the target aspects. In the classification stage, we utilized rich information through feature fusion to improve category separability. To mitigate the bias towards seen classes, we designed the BL to adjust the prediction probability. Experiments on two measured HRRP datasets demonstrated the superior performance of our method for both seen and unseen classes, regardless of variations in the target aspects.

Author Contributions: Conceptualization, J.Z.; methodology, J.Z. and S.M.; software, J.Z. and S.M.; validation, T.Z. and Y.G.; formal analysis, T.Z. and Y.G.; investigation, J.Z. and Y.G.; resources, F.W. and W.Y.; data curation, S.M.; writing—original draft preparation, J.Z.; writing—review and editing, J.Z. and T.Z.; visualization, J.Z.; supervision, F.W. and W.Y.; project administration, F.W.; funding acquisition, T.Z. and W.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the National Natural Science Foundation of China under Grant 62201343 and the United Fund of National Key Laboratory of Automatic Target Recognition (Shanghai) under Grant ATR(S)2025-006. This project was funded by the Shanghai Municipal Commission of Economy and Informatization (Grant No. 2024-GZL-RGZN-01029).

Data Availability Statement: The data presented in this study are available from the authors upon request.

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

References

1. Cho, J.H.; Park, C.G. Multiple Feature Aggregation Using Convolutional Neural Networks for SAR Image-Based Automatic Target Recognition. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 1882–1886. [CrossRef]
2. Xie, N.; Zhang, T.; Zhang, L.; Chen, J.; Wei, F.; Yu, W. VLF-SAR: A Novel Vision-Language Framework for Few-shot SAR Target Recognition. *IEEE Trans. Circuits Syst. Video Technol.* **2025**, *35*, 9530–9544. [CrossRef]
3. Zhang, T.; Xie, N.; Quan, S.; Wang, W.; Wei, F.; Yu, W. Polarimeric SAR Ship Detection based on the Sub-look Decomposition Technology. *IEEE Trans. Radar Syst.* **2026**, *4*, 35–49. [CrossRef]
4. Li, H.J.; Wang, Y.D.; Wang, L.H. Matching score properties between range profiles of high-resolution radar targets. *IEEE Trans. Antennas Propag.* **1996**, *44*, 444–452. [CrossRef]
5. Du, L.; Liu, H.; Bao, Z. Radar HRRP statistical recognition based on hypersphere model. *Signal Process.* **2008**, *88*, 1176–1190. [CrossRef]
6. Du, C.; Chen, B.; Xu, B.; Guo, D.; Liu, H. Factorized discriminative conditional variational auto-encoder for radar HRRP target recognition. *Signal Process.* **2019**, *158*, 176–189. [CrossRef]
7. Xu, B.; Chen, B.; Wan, J.; Liu, H.; Jin, L. Target-aware recurrent attentional network for radar HRRP target recognition. *Signal Process.* **2019**, *155*, 268–280. [CrossRef]
8. Pan, M.; Liu, A.; Yu, Y.; Wang, P.; Li, J.; Liu, Y.; Lv, S.; Zhu, H. Radar HRRP target recognition model based on a stacked CNN-Bi-RNN with attention mechanism. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5100814. [CrossRef]
9. Chen, J.; Du, L.; Guo, G.; Yin, L.; Wei, D. Target-attentional CNN for radar automatic target recognition with HRRP. *Signal Process.* **2022**, *196*, 108497. [CrossRef]
10. Gao, S.; Bie, B.; Wang, H.; Xing, M.; Quan, Y. Radar High-Resolution Range Profile Target Recognition Method Based on Structure-Aware Network. *IEEE Sens. J.* **2024**, *24*, 32660–32672. [CrossRef]
11. Liu, Q.; Zhang, X.; Liu, Y. A Prior-Knowledge-Guided Neural Network Based on Supervised Contrastive Learning for Radar HRRP Recognition. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *60*, 2854–2873. [CrossRef]
12. Liu, Y.; Long, T.; Zhang, L.; Wang, Y.; Zhang, X.; Li, Y. SDHC: Joint Semantic-Data Guided Hierarchical Classification for Fine-Grained HRRP Target Recognition. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *60*, 3993–4009. [CrossRef]
13. Xia, Z.; Wang, P.; Dong, G.; Liu, H. Radar HRRP Open Set Recognition Based on Extreme Value Distribution. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5102416. [CrossRef]
14. Li, W.; Li, S.; Tian, B.; Huang, P.; Zheng, M.; Xu, S. Regularization-Based Region Learning for Radar HRRP Open Set Recognition. *IEEE Trans. Instrum. Meas.* **2025**, *74*, 2512615. [CrossRef]
15. Pourpanah, F.; Abdar, M.; Luo, Y.; Zhou, X.; Wang, R.; Lim, C.P.; Wang, X.Z.; Wu, Q.J. A review of generalized zero-shot learning methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 4051–4070. [CrossRef]
16. 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]
17. Reed, S.; Akata, Z.; Lee, H.; Schiele, B. Learning deep representations of fine-grained visual descriptions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 49–58. [CrossRef]
18. Kampffmeyer, M.; Chen, Y.; Liang, X.; Wang, H.; Zhang, Y.; Xing, E.P. Rethinking knowledge graph propagation for zero-shot learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 11487–11496. [CrossRef]
19. Roy, A.; Ghosal, D.; Cambria, E.; Majumder, N.; Mihalcea, R.; Poria, S. Improving zero-shot learning baselines with commonsense knowledge. *Cogn. Comput.* **2022**, *14*, 2212–2222. [CrossRef]
20. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
21. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems 27 (NIPS 2014), Montreal, QC, Canada, 8–13 December 2014.
22. 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, 18–22 June 2018; pp. 5542–5551.
23. Du, L.; He, H.; Zhao, L.; Wang, P. Noise robust radar HRRP target recognition based on scatterer matching algorithm. *IEEE Sens. J.* **2015**, *16*, 1743–1753. [CrossRef]
24. Zhu, F.; Zhang, X.D.; Hu, Y.F.; Xie, D. Nonstationary hidden Markov models for multiaspect discriminative feature extraction from radar targets. *IEEE Trans. Signal Process.* **2007**, *55*, 2203–2214. [CrossRef]
25. Sun, Z.; Deng, Z.H.; Nie, J.Y.; Tang, J. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv* **2019**, arXiv:1902.10197. [CrossRef]

26. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014; Volume 28. [CrossRef]
27. Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; Bouchard, G. Complex embeddings for simple link prediction. In Proceedings of the International Conference on Machine Learning, New York City, NY, USA, 20–22 June 2016; pp. 2071–2080.
28. Zhou, J.; Yu, W.; Zhang, J.; Mu, S.; Li, Y. A Graph Neural Network for Ship Link Prediction Based on Graph Attention Mechanism and Quaternion Embedding. *IEEE Geosci. Remote Sens. Lett.* **2024**, *21*, 6001205. [CrossRef]
29. 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.
30. 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. [CrossRef]
31. 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 European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 479–495. [CrossRef]
32. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved training of wasserstein gans. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
33. Felix, R.; Reid, I.; Carneiro, G. Multi-modal cycle-consistent generalized zero-shot learning. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 21–37.
34. Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; Krishnan, D. Supervised contrastive learning. In Proceedings of the Advances in Neural Information Processing Systems 33 (NeurIPS 2020), Virtual, 6–12 December 2020; pp. 18661–18673.
35. Huynh, D.; Elhamifar, E. Fine-grained generalized zero-shot learning via dense attribute-based attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 4483–4493. [CrossRef]
36. Chen, S.; Hong, Z.; Liu, Y.; Xie, G.S.; Sun, B.; Li, H.; Peng, Q.; Lu, K.; You, X. Transzero: Attribute-guided transformer for zero-shot learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 22 February–1 March 2022; Volume 36, pp. 330–338. [CrossRef]
37. Reimers, N.; Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv* **2019**, arXiv:1908.10084. [CrossRef]
38. Chen, Z.; Luo, Y.; Qiu, R.; Wang, S.; Huang, Z.; Li, J.; Zhang, Z. Semantics disentangling for generalized zero-shot learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11–17 October 2021; pp. 8712–8720. [CrossRef]
39. Li, J.; Jing, M.; Lu, K.; Ding, Z.; Zhu, L.; Huang, Z. Leveraging the invariant side of generative zero-shot learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7402–7411. [CrossRef]
40. Jiang, H.; Wang, R.; Shan, S.; Chen, X. Transferable contrastive network for generalized zero-shot learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9765–9774. [CrossRef]
41. Han, Z.; Fu, Z.; Chen, S.; Yang, J. Contrastive embedding for generalized zero-shot learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 2371–2381. [CrossRef]
42. Rao, Z.; Guo, J.; Lu, X.; Zhou, Q.; Zhang, J.; Wei, K.; Li, C.; Guo, S. Attribute-aware representation rectification for generalized zero-shot learning. *arXiv* **2023**, arXiv:2311.14750. [CrossRef]
43. Chen, S.; Wang, W.; Xia, B.; Peng, Q.; You, X.; Zheng, F.; Shao, L. FREE: Feature refinement for generalized zero-shot learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11–17 October 2021; pp. 122–131. [CrossRef]

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.