Few-Shot SAR Target Recognition Based on Deep Kernel Learning

KE WANG1, QI QIAO1,2, GONG ZHANG3, (Member, IEEE), AND YIHAN XU1

1School of Computer and Communication, Jiangsu Vocational College of Electronics and Information, Huai’an 223001, China
2School of Computer and Mathematical Science, Universiti Teknologi MARA, Shah Alam 40450, Malaysia
3School of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211100, China

Corresponding author: Yihan Xu (xuyh@jsei.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61871218, in part by the Jiangsu Software Testing Engineering Technology Research and Development Center under Grant JSES2018.No10, in part by the Huai’an Key Laboratory of Software Testing under Grant HAP201904, in part by the Science Foundation of Jiangsu Vocational College of Electronics and Information under Grant JSEIYY2020003, and in part by the Huai’an Natural Science Research Project under Grant HAB201912.

ABSTRACT Deep learning methods have achieved state-of-the-art performance on synthetic aperture radar (SAR) target recognition tasks in recent years. However, obtaining sufficient SAR images for training these deep learning methods is costly in time and labor. This paper focuses on recognizing targets with a few training samples, that is, few-shot target recognition. We combine deep neural networks’ powerful feature representation capabilities with the nonparametric flexibility of Gaussian processes (GPs) and propose a few-shot recognition model based on deep kernel learning. Deep neural networks map input samples into a low-dimensional embedding space. GPs employ a family of kernel functions to measure the similarity between embedded samples and classify them. During training, the model builds diverse related tasks to learn kernel functions with parameters shared across few-shot tasks. These learned kernel functions define common prior knowledge that can be transferred to unseen tasks. During testing, the model can recognize novel tasks with a few samples based on learned kernel functions. We conducted extensive experiments on a widely-used real SAR dataset to evaluate the model’s effectiveness. The test results demonstrate that our model is superior to several recently proposed few-shot recognition methods.

INDEX TERMS Synthetic aperture radar (SAR), target recognition, deep kernel learning, few-shot classification.

I. INTRODUCTION

Synthetic aperture radar (SAR) is a sensor system capable of acquiring ground or sea target images. Compared with optical sensors, SAR works in the microwave band and has all-day, all-weather, and long-distance operating capabilities, so it plays a vital role in both military and civilian fields. By interpreting SAR images, we can obtain various intelligence information about the target that is usually not available in optical images. The unique imaging mechanism of SAR means SAR images are greatly affected by the physical and electrical properties of the imaging targets. Compared with optical images, SAR images have lower resolution and are disturbed by speckle noise. These defects make it difficult to interpret and classify SAR images manually, which leads to the generation of synthetic aperture radar automatic target recognition (SAR-ATR) [1]. Currently, SAR-ATR is one of the essential tasks in SAR image applications.

Over the past decade, data-driven methods have been popularized in SAR-ATR and achieved excellent performance. Traditional SAR-ATR methods [2], [3] require manual design and extraction of discriminative image features, which rely heavily on experts with rich domain knowledge. Data-driven methods can automatically learn hierarchies of features from images, thus improving the accuracy and efficiency of recognition. Among various data-driven methods, the convolutional neural network (CNN) and its variants are widely used to extract image features. Researchers have designed
many novel CNN architectures to extract SAR image features effectively. To reduce the number of parameters of CNN, Chen et al. [4] proposed an all-convolutional network that removes fully connected (FC) layers. Wagner replaced FC layers with a support vector machine (SVM), improving the network’s robustness against imaging errors and target variations [5]. In practice, target images are acquired at different view angles, such as azimuth and depression angles. Training CNN with multi-view images can improve its robustness to angle variations. Multi-stream CNN [6] and multi-view CNN [7] use parallel network structures to extract features of a set of multi-view images simultaneously and then fuse these features via fusion modules. The intrinsic speckle noises of SAR images interfere with the image feature extraction and degrade the recognition performance. The feature aggregation [8] and training regularization [9] are proposed to extract noise-robust features and reduce the influence of speckle noise.

The performance of data-driven methods relies heavily on the quality and quantity of the training data. However, the acquisition and annotation of SAR data are complex and expensive. The resulting scarcity of training data degrades the performance of data-driven SAR-ATR algorithms. Data augmentation, transfer learning, self-supervised learning, and few-shot learning are used to address target recognition with limited training data. In [10], synthetic SAR images are generated by translation, angular rotation, adding noise, and mixed with real SAR images as training data. Instead of generated by translation, angular rotation, adding noise, and limited training data. In [10], synthetic SAR images are few-shot learning are used to address target recognition with augmentation, transfer learning, self-supervised learning, and the performance of data-driven SAR-ATR algorithms. Data expensive. The resulting scarcity of training data degrades the recognition performance. The feature aggregation [8] and training regularization [9] are proposed to extract noise-robust features and reduce the influence of speckle noise.

Few-shot classification (FSC) [20]–[22], the algorithms capable of building classifiers to novel categories given only a few data samples, is a rapidly growing area of SAR-ATR. These FSC-based methods first utilized embedding networks based on CNNs to map the image samples into embedding spaces. Then, the categories of samples are inferred by the hybrid inference network [20], the relation network [21], and the graph attention network [22]. These methods aim to increase samples’ intra-class similarity and inter-class divergence, thus enhancing their discrimination abilities in few-shot scenarios. The characteristics of SAR targets can be used as domain-related knowledge to assist the training of the SAR-ATR model, thus improving the model’s accuracy in FSC scenarios. Zhang et al. [23] designed a dual-stream CNN powered by domain knowledge such as azimuth angles and phase information of SAR images. To exploit electromagnetic scattering characteristics of SAR targets, Wang et al. [24] performed sub-band decomposition on complex-valued SAR images. Feng et al. [25] decomposed a SAR target into multiple components according to its attribute scattering characteristics. The decomposed components are fed into a bidirectional network to extract the target’s local features.

Deep kernel learning (DKL) [26] provides a scalable Bayesian framework that combines the representational power of deep learning with the nonparametric flexibility of kernel methods. DKL employs deep neural networks to map data samples into embedding vectors, which are then used as the inputs for kernel methods such as kernel ridge regression and Gaussian processes (GPs) [27]. The network weights and kernel’s hyperparameters are updated by maximizing the likelihood of training data, building an end-to-end training scheme. More recently, Patacchiola et al. [28] proposed a deep kernel transfer method in which neural networks estimate the hyperparameters of kernel functions. Tossou et al. [29] developed a kernel family parameterized by neural networks for various few-shot regression tasks. As a Bayesian approach, DKL can deal with model uncertainty due to the scarcity of training data, so it is well-suited to address FSC problems.

This paper focuses on solving few-shot SAR-ATR problems, which need to recognize tasks of unseen categories given only a few labeled samples. Inspired by previous...
studies [28], [29], we propose a few-shot recognition model based on DKL, combining deep CNN's powerful feature representation capabilities with the nonparametric flexibility of GPs. The motivation is twofold. First, our model selects nonparametric GPs as classifiers. This selection reduces the model’s parameters and helps alleviate overfitting due to insufficient training samples. Second, the key to FSC is to learn common priors from various tasks and then transfer them to new tasks with a few samples. Our model builds many small but related tasks to learn kernel functions with task-common parameters. These kernel functions define common priors that can be transferred to unseen tasks. Therefore, the model can recognize new tasks with a few samples based on learned kernel functions. The framework of our model is similar to that of deep metric learning (DML). DML aims to learn a distance metric to measure the data similarity in embedding space. The kernel functions used in our model can be regarded as a parametric distance metric.

The specific contributions of this paper are as follows.

1) We frame few-shot SAR recognition as a deep kernel learning problem and propose a DKL-based model combining deep neural networks and GPs.

2) The proposed model can recognize new target categories with a few samples. Our model obtains improved accuracy compared with other recently proposed few-shot SAR-ATR methods.

II. PRELIMINARIES

A. FEW-SHOT CLASSIFICATION

By learning transferable knowledge from various training data, FSC aims to build a classifier capable of recognizing novel categories using a few test samples. In traditional supervised learning, the categories of training data and test data are identical. However, the training and test data used in FSC are disjoint and contain different target categories. The classifier trained by traditional supervised learning cannot generalize well to novel categories. Therefore, FSC usually trains the classifier in the episodic training mode [30]. The samples used in episodic training are organized as few-shot tasks, each containing a support set and a query set. To build a C-way K-shot task, we first select C categories from a dataset, randomly sample K samples from each category, and make a support set of C × K samples. Next, we build a query set by sampling N samples from each category. The support set and query set have the same target category but different samples. Finally, the few-shot task contains C × (K + N) samples. In our model, K is set to 1 or 5, and the corresponding N is set to 5 or 15.

In each iteration of the training stage, the FSC model randomly selects a batch of samples from the training dataset to build a C-way K-shot task. The model trains a classifier on the support set and evaluates its classification loss on the query set. The samples of tasks in each iteration are randomly selected so that a considerable number of different tasks can be built for training, alleviating the problem of limited data. In the testing stage, the model samples a C-way K-shot task from the testing dataset, fine-tunes the parameters with support samples, then predicts the query samples' categories. Fig. 1 shows an example of tasks used in the 3-way 1-shot FSC problem.

B. DEEP KERNEL LEARNING

In DKL, kernels are used to describe the similarity or relation of data samples. Specifically, given any two samples \( x \) and \( \tilde{x} \) in the input space \( \Omega \), the kernel \( k(x, \tilde{x}) \) on \( \Omega \) is a function \( k : \Omega \times \Omega \rightarrow R, \forall x, \tilde{x} \in \Omega \) that quantifies the similarity of \( x \) and \( \tilde{x} \). The simplest kernel measures the similarity between a pair of samples by their inner product, and we call it the linear
kernel. It is defined as:

\[ k_v(x, \tilde{x}) = v \cdot x^T \tilde{x}, \]  

(1)

where \( v \) is a scaling parameter adjusted according to the dimensions of the input samples. In addition, squared exponential kernels, polynomial kernels, and cosine similarity kernels are widely used in kernel methods.

DKL integrates kernel functions with embedding networks (e.g., deep neural networks). Specifically, DKL first employs an embedding network \( F_\theta(\Omega) \rightarrow H \) to map the data samples from the input space \( \Omega \) to the embedding space \( H \), where \( \theta \) denotes the weights of \( F_\theta (\cdot) \). The kernel function is then responsible for measuring the similarity between embedded samples. Overall, DKL aims to build a kernel function that quantifies the similarity of input pairs in the embedding space:

\[ k(x, \tilde{x}|\theta, \phi) = k_\phi(F_\theta(x), F_\theta(\tilde{x})|\theta). \]  

(2)

The kernel function’s hyperparameters \( \phi \) and the network weights \( \theta \) are optimized by maximizing the predictive likelihood of training data.

### III. METHODOLOGY

This section introduces the proposed few-shot SAR target recognition model, containing a data augmentation module, an embedding network, and a set of GPs. Fig.2 illustrates the overall framework of the model in the case of 3-way recognition. The model randomly samples a batch of samples in a training iteration to build a few-shot task. To increase the diversity of the training samples, the augmentation module expands the samples via random flipping (RF), color jittering (CJ), random rotation (RR), and Gaussian noising (GN) [17]. The embedding network maps both the support and query samples into a low-dimensional embedding space. GPs utilize kernel functions to measure the similarity of embedded samples of support and query sets. The model’s parameters are learned by maximizing the predictive likelihood of the query set. During testing, the weights of the embedding network and hyperparameters of the kernels are fixed. The model requires only a small number of test samples to recognize new target categories.

### A. FEW-SHOT DEEP KERNEL LEARNING

In this section, we apply deep kernel learning to FSC problems. According to the episodic training mode, the training and testing data of the model are organized as few-shot tasks, each of which refers to a classification problem given a few data samples. The model samples a few-shot task \( \tau \) from the task distribution \( P(\tau) \) in each training iteration. The task \( \tau \) contains two disjoint subsets, a support set \( \tau_s = \{X, Y\} = \{(x_i, y_i), i=1, 2, \ldots m\} \) and a query set \( \tau_q = \{X_s, Y_s\} = \{(x_j, y_j), j=1, 2, \ldots n\} \). The pair \((x_i, y_i)\) denotes the image...
and label, respectively. Our model assumes that different tasks drawn from the same distribution \( P(\tau) \) share common structures or features, which is crucial for fast adaptation to new tasks. These shared structures or features are extracted and transferred by task-common parameters \( \{\theta, \phi\} \). \( \theta \) denotes the weights of embedding networks, and \( \phi \) denotes the hyper-parameters of the GPs. The optimized objective of the model is to maximize the predictive likelihood of each task. For a single task \( \tau \), the predictive log-likelihood \( \mathcal{L} \) of a query example \( x_s \) given a support set \( \{X, Y\} \) is defined as:

\[
\mathcal{L} = \log P(y_s|x_s, X, Y; \theta, \phi). \tag{3}
\]

Typical hierarchical FSC methods [31], [32] use task-specific parameters to deal with differences between tasks. The training process of these methods includes an inner loop and an outer loop. The inner loop is used to update task-specific parameters. The outer loop is used to update task-common parameters. As the number of tasks increases, computing and maintaining different task-specific parameters is challenging. Inspired by [28], our model instead maintains the distribution over latent functions of GPs rather than the distribution over task-specific parameters. GPs take samples of a task as input and use task-common parameters to produce specific latent functions for this task. These latent functions are task-specific and can be used to deal with task variations. The updated log-likelihood \( L \) is defined as:

\[
L = \log \int P(y_s|f_s)P(f_s|x_s, X, Y; \theta, \phi) df_s, \tag{4}
\]

where \( f_s \) represents the task-specific latent function of \( x_s \). \( P(y_s|f_s) \) denotes predictive distribution over \( y_s \), conditioned on inferred \( f_s \). The model executes an integral of the latent function \( f_s \) and optimizes only for the task-common parameters \( \{\theta, \phi\} \).

To produce a categorical likelihood with probabilistic components, \( f_s \) needs to be further squashed by a nonlinear mapping function. Each category label is a Bernoulli random variable in the binary classification case. The sigmoid function \( \sigma(\cdot) \) is a common choice for nonlinear mapping. \( P(y_s|f_s) \) is formulated as:

\[
P(y_s|f_s) = \sigma(f_s), \quad \text{where } f_s \sim P(f_s|x_s, X, Y; \theta, \phi). \tag{5}
\]

According to the definition of GPs [33], the latent function \( f_s \) of a single sample \( x_s \) is a Gaussian variable with mean and variance specified as:

\[
P(f_s|x_s, X, Y; \theta, \phi) = N(\mu, \sigma^2), \quad \text{where}
\]

\[
\mu = K(x_s, X)K(X, X)^{-1}y
\]

\[
\sigma^2 = K(x_s, x_s) - K(x_s, X)K(X, X)^{-1}K(X, x_s),
\]

where the covariance matrix \( K(X, X) = (k_\phi(x_i, x_j))_{i,j} \) encodes the properties of the desirable kernel function. The entry \( k_\phi(x_i, x_j) \) specifies the similarity between \( i \)-th sample \( x_i \) and \( j \)-th sample \( x_j \) in \( X \). The other matrices \( K(x_s, x_s), K(x_s, X) \) and \( K(X, x_s) \) are defined similarly.

Finally, the training objective for a task \( \tau = \{X, Y, X_s, Y_s\} \) is to maximize the predictive log-likelihood \( \mathcal{L} \) of all the query samples, as:

\[
\max_{\theta, \phi} \mathcal{L} = \frac{1}{|X_s|} \sum_{x_s, y_s \in X_s, Y_s} \log \int \sigma(f_s)P(f_s|x_s, X, Y; \theta, \phi) df_s, \tag{7}
\]

where the latent posterior \( P(f_s|x_s, X, Y; \theta, \phi) \) is estimated by (6) and transformed by \( \sigma(\cdot) \) to produce a predictive distribution over category probabilities. Due to the non-conjugacy between \( P(f_s|x_s, X, Y; \theta, \phi) \) and \( \sigma(f_s) \), the integral in (7) is analytically intractable. However, in the case of binary classification, this integral is one-dimensional and can be computed numerically. The maximization of log-likelihood \( \mathcal{L} \) is treated as a least square classification (LSC) problem [28], [33]. The model takes the predicted and true category labels as continuous values and uses the gradient of mean square error (MSE) between them to update the parameters \( \theta \) and \( \phi \).

**B. EMBEDDING NETWORKS**

Similar to previous studies [20]–[22], our model employs a simple convolutional network to map SAR images of a few-shot task into embedded vectors. Using shallow and simple backbones has been verified to highlight differences between FSC methods [34]. As shown in Fig.3, the embedding network contains four convolutional blocks. Each block sequentially performs convolution, batch normalization (BN), nonlinear activation, and max-pooling operations on the input. Conv@3 \( \times 3 \times 1 \times 64 \) indicates that the kernel size of the convolutional layer is \( 3 \times 3 \), the number of input channels is 1, and the number of output channels is 64. MaxPool@2 \( \times 2 \) means downsampling the input with a max-pooling operation of stride 2. The first convolutional block transforms an input image of size \( 64 \times 64 \) pixels into a collection of feature maps of size \( 64 \times 32 \times 32 \). The subsequent three blocks perform a similar refining process on the feature maps to eliminate redundant information. The last block outputs feature maps of size \( 64 \times 4 \times 4 \), which are flattened into a 1024-dimensional vector for further processing by the GPs.

**C. GAUSSIAN PROCESSES**

1) KERNELS

Our model learns prior knowledge via diverse training tasks and then transfers it to unseen test tasks with a few samples. The prior knowledge accumulates in kernel functions combined with simple embedding networks to make predictions. Therefore, the choice of kernel function is critical for recognition performance. Our model experimentally selects several kernels (including the squared exponential kernel, polynomial kernel, and cosine similarity kernel) to measure the similarity between a pair of inputs \( x \) and \( \tilde{x} \) in the embedding space. The polynomial kernel is defined as follows:

\[
k_p(x, \tilde{x}) = (x^T \tilde{x} + c)^d, \tag{8}
\]
where $c$ is an offset parameter and $d$ is the degree parameter. The squared exponential kernel takes the exponential of a quadratic form:

$$k_l(x, \tilde{x}) = \exp\left(-\frac{\|x - \tilde{x}\|^2}{2l^2}\right)$$  \hspace{1cm} (9)

where the hyperparameter $l$ defines the characteristic length-scale. Similar in spirit to the linear kernel, the cosine similarity kernel computes the inner product between the unit-normalized input vectors:

$$k(x, \tilde{x}) = \frac{x^T \tilde{x}}{\|x\| \|\tilde{x}\|}.$$  \hspace{1cm} (10)

We found that the cosine similarity kernel outperforms other kernels in most few-shot test settings.

2) MULTI-CATEGORY CLASSIFICATION

The original GPs can only address binary classification tasks. However, our model must address C-way K-shot few-shot tasks ($C > 2$). One solution is to design a composite likelihood function (e.g., a modified Softmax likelihood) suitable for multi-category problems [35]. An alternative solution is to split the multi-category problem into multiple binary classification problems. Adopting the one-versus-the-rest scheme, our model builds a C-category classifier by combining multiple GPs. Specifically, the model trains C independent GPs, where the $i^{th}$ GP is trained using positive samples from class $C_i$ and negative samples from the remaining $C-1$ categories. The labels of positive and negative samples are set to $+1$ and $-1$, respectively. During testing, the target category is provided by the GP of the highest confidence. Assuming that the inference process of each GP is independent, the predictive log-likelihood of the multi-category classifier is the sum of the likelihood of the individual GP. For a task $\tau = \{X, Y, X_a, Y_a\}$, the log-likelihood of a query example $x_a$ is defined as:

$$\log P(y_a|x_a, X, Y; \phi) = \sum_{c=1}^{C} \log P(y^c_a|x_a, X, Y; \theta, \phi)$$  \hspace{1cm} (11)

Given a query sample $x_a$ and C outputs of all the GPs, the category label $y_a$ is decided by selecting the output with the highest probability $c = \arg\max_c (y^c_a)$.

IV. EXPERIMENTAL RESULTS

A. DATASET

This paper’s training and testing images are collected from the moving and stationary target acquisition and recognition (MSTAR) dataset [36]. The MSTAR project, initiated by a U.S. military laboratory, aims to acquire SAR images of ground targets under various operating conditions. These SAR images are obtained by the airborne platform with an X-band SAR imaging system. The resolution of the SAR image is $0.3m \times 0.3m$, and the image size is $128 \times 128$ pixels. Only experts with domain knowledge can recognize the target category of SAR images, which leads to the requirement for SAR-ATR systems. We divided the MSTAR dataset into training and test data with disjoint categories according to the experimental setup used in previous studies [20], [22]. The testing dataset contains three categories of target images, 2S1, BRDM-2, and ZSU-234, with depression angles of $15^\circ$, $17^\circ$, and $30^\circ$. The training dataset contains the remaining seven categories of target images with a depression angle of $17^\circ$. The depression angle difference between these two datasets is helpful to evaluate the robustness of the model under depression variations.
The two datasets’ target category, depression angle, and size are summarized in TABLE 1 and TABLE 2.

The workflow of our model contains two stages: the training and testing stages. To effectively transfer the prior knowledge acquired during training to testing, both stages use the same C-way K-shot task settings. TABLE 3 lists three different experiment settings whose main difference lies in the different test depression angles. The model randomly samples 3-way 5-shot or 3-way 1-shot tasks from the training dataset in the training stage. The depression angles of the support set and query set are both 17°. During the testing stage, the model is evaluated in similar task settings but with different target categories and angles. The depression angle of the support set is 17°, and the depression angle of the query set varies according to different experimental settings.

The model is optimized with the adaptive momentum estimation (ADAM) algorithm [37], which sets a learning rate of 1e-4 for the GPs and a learning rate of 1e-3 for the embedding network. The model parameters learned in the training stage are directly applied to testing without any fine-tuning operations.

### TABLE 3. Experimental settings.

| Settings | Training | Testing |
|----------|----------|---------|
|          | Support  | Query   | Support | Query   |
| 17s/15q  | 17°      | 17°     | 17°     | 15°     |
| 17s/17q  | 17°      | 17°     | 17°     | 17°     |
| 17s/30q  | 17°      | 17°     | 17°     | 30°     |

### TABLE 4. Recognition results obtained in different task settings.

| Task Settings | Accuracy (%) | Standard deviation |
|---------------|--------------|--------------------|
|               | Min | Max | Mean |                  |
| 17s/15q 1-shot| 54.89 | 98.35 | 92.49 | 0.0785 |
| 5-shot      | 84.63 | 99.71 | 97.25 | 0.0264 |
| 17s/17q 1-shot| 55.66 | 99.38 | 92.94 | 0.0839 |
| 5-shot      | 87.57 | 99.91 | 98.54 | 0.0134 |
| 17s/30q 1-shot| 43.96 | 98.62 | 78.32 | 0.0901 |
| 5-shot      | 73.35 | 99.25 | 91.11 | 0.0563 |

3-way 5-shot or 3-way 1-shot tasks from the training dataset in the training stage. The depression angles of the support set and query set are both 17°. During the testing stage, the model is evaluated in similar task settings but with different target categories and angles. The depression angle of the support set is 17°, and the depression angle of the query set varies according to different experimental settings.

### B. RECOGNITION PERFORMANCE OF THE MODEL

Few-shot tasks are sampled from a large dataset, so the samples in each task may be completely different, resulting in a large variation in the recognition accuracy of different tasks. Therefore, the model randomly samples 1000 few-shot tasks to obtain their average accuracy. TABLE 4 shows the recognition results obtained in different task settings. The model achieved an average accuracy of over 90% in almost all task settings, demonstrating the model’s effectiveness in few-shot recognition scenarios. Compared with 1-shot tasks, 5-shot tasks have higher average accuracies and more minor standard deviations. In DKL, the category of the query sample is closely related to the category of its adjacent support samples. Therefore, the more support samples, the higher the model’s recognition accuracy. In addition, recognition results obtained in the 17s/17q setting are better than those obtained in the 17s/15q and 17s/30q settings. SAR images of the same target category vary significantly at different depression angles. The greater the gap in the depression angle between the support set and the query set, the more significant the difference between support images and query images, which leads to a decrease in the recognition accuracy.

For 1-shot tasks, the minimum accuracies obtained in three settings are only 54.89%, 55.66%, and 43.96%, which are
FIGURE 6. Histograms of recognition accuracies in various task settings. (a) 17s/15q (5-shot); (b) 17s/15q (1-shot); (c) 17s/17q (5-shot); (d) 17s/17q (1-shot); (e) 17s/30q (5-shot); (f) 17s/30q (1-shot).

43.46%, 43.72%, and 54.66% less than the maximum accuracies, respectively. The maximum and minimum accuracies differ significantly, indicating that the accuracies are spread in a wide range. At this time, there is only one sample per category in the support set. The selection of support samples has a significant impact on test accuracy. For 5-shot tasks, the minimum accuracy of three settings increases to 84.63%, 87.57%, and 73.35%, and the gap between maximum and minimum accuracy decreases. As the size of the support set increases, the recognition accuracy and robustness of the model are improved. We also use statistical histograms to visualize the distributions of accuracies for each group of 1000 independent test tasks. As shown in Fig.6, the accuracy distributions of 5-shot tasks are more concentrated than those of 1-shot tasks. This also verifies the conclusion shown in TABLE 4 that the standard deviations of the 5-shot tasks are generally smaller than those of the 1-shot tasks.

TABLE 5 uses a set of confusion matrices to show the average accuracy of each target category. The average accuracy obtained in 17s/17q is significantly better than in the other two settings. In 17s/17q, the 5-shot accuracy of all three targets exceeds 98.5%. The minor the gap in depression angle between the support set and the query set, the better model’s recognition performance. TABLE 5 also shows that the accuracy of 2S1 is lower than that of BRDM2 and ZSU234, especially in 17s/30q. The classifier misclassifies 2S1 as other targets with more than 50% probability in the 1-shot task. The target appearance of 2S1 is greatly affected by the variation in the depression angle, which causes the classifier to make a wrong judgment.

C. CONTRAST EXPERIMENTS

We also compare the model with several classical few-shot recognition methods. The Prototypical-Net [38], Relation-Net [39], and Matching-Net [40] are reproduced using the same backbone structure and optimizer. The results of hybrid inference network (HID) [20] and mixed loss graph attention network (MGA-Net) [22] are directly collected from their papers. Fig.7 compares the test results of different methods obtained in various task settings. Note that HID does not provide test accuracy of the 17s/17q setting. The HID, MGA-Net, and our model are explicitly designed for SAR few-shot recognition. Their accuracies surpass Prototypical-Net, Relation-Net, and Matching-Net by a large margin, which was proposed to classify optical images. Our model outperforms other comparative methods in 17s/15q and 17s/17q, demonstrating the superiority in few-shot recognition scenarios. Especially in 17s/17q, the 5-shot accuracy of our model is 98.5%, which is 5.7%, 5.9%, 4.4%, and 2.6% better than the competitors, respectively. When the test setting changes to 17s/30q, the recognition performance of all methods deteriorates. In 17s/30q, the depression angles of the support set and the query set differ by 13°, resulting in a significant difference in the corresponding images. It is challenging to improve the robustness of few-shot recognition under large depression angle variations.

D. ABLATION STUDY

This section evaluates the impact of the data augmentation and the kernel selection on the model.
TABLE 5. Confusion matrices of average recognition accuracies. Each row represents the real category of the target, and each column represents the predicted category. Diagonal entries of the matrix show the average accuracy of each category.

| Settings | 3-way 1-shot | 3-way 5-shot |
|----------|--------------|--------------|
|          | 2S1 | BRDM2 | ZSU234 | 2S1 | BRDM2 | ZSU234 |
| 17s/15q  | 0.8605 | 0.1087 | 0.0307 | 0.9538 | 0.0387 | 0.0075 |
|          | 0.0596 | 0.9245 | 0.0159 | 0.0227 | 0.9720 | 0.0053 |
|          | 0.0062 | 0.0041 | 0.9897 | 0.0052 | 0.0030 | 0.9918 |
| 17s/17q  | 0.9183 | 0.0311 | 0.0505 | 0.9856 | 0.0084 | 0.0060 |
|          | 0.0314 | 0.9363 | 0.0323 | 0.0085 | 0.9851 | 0.0065 |
|          | 0.0268 | 0.0396 | 0.9335 | 0.0067 | 0.0079 | 0.9854 |
| 17s/30q  | 0.4836 | 0.3765 | 0.1399 | 0.8527 | 0.0929 | 0.0544 |
|          | 0.0698 | 0.9209 | 0.0093 | 0.0849 | 0.9012 | 0.0139 |
|          | 0.0531 | 0.0023 | 0.9446 | 0.0194 | 0.0013 | 0.9793 |

1) DATA AUGMENTATION

Increasing the diversity of training samples can improve the recognition performance of few-shot SAR-ATR models [17], [22]. We found through extensive experiments that four image transformations can boost the performance of our model. These data augmentation methods include random flipping (RF), random rotation (RR), color jitter (CJ), and Gaussian noising (GN). RF randomly flips the image horizontally or vertically with equal probability. RR rotates the image clockwise or counterclockwise by any angle. In this experiment, the rotation angle range is set to $[-15^\circ, 15^\circ]$, which means that the image is rotated clockwise or counterclockwise at most $15^\circ$. CJ randomly jitters the brightness, contrast, and saturation of the image. Considering that the SAR image is a single-channel grayscale image, we set brightness $= 0.4$, contrast $= 0.4$, and saturation $= 0$. GN adds Gaussian noise to each pixel of the image. We set the mean of Gaussian noise to 0 and the standard deviation to 0.01. Fig.8 compares the 5-shot accuracies of different augmentation combinations in 17s/15q. Compared with no data augmentation, using RF, RR, CJ, and GN to augment the training samples can improve the accuracy by 0.6%, 0.8%, 1.1%, and 0.4%, respectively. For our model, CJ is the most effective means of data augmentation. Furthermore, combining different augmentation methods can effectively boost recognition performance. Therefore, the model adopts augmentation combinations including RF, RR, CJ, and GN and achieves 97.2% accuracy in 5-shot tasks.

2) EFFECT OF KERNEL CHOICE

The GPs used in our model can select different kernel functions to measure the similarity between samples. We compared six models with the same backbone network and optimizer but different kernels. The polynomial kernel is tested with two variants, where $d$ is the degree of the polynomial. The average accuracies are obtained from 1000 randomly generated tasks. As shown in TABLE 6, the accuracy obtained in 17s/17q is better than those obtained in 17s/15q and 17s/30q. The cosine similarity (CosSim) kernel outperforms other kernels in almost all task settings. Compared with other kernels, the CosSim kernel could provide nearly 1% and 2% boost in 1-shot and 5-shot tasks. The BNCosSim kernel centralizes and normalizes the output of the backbone network compared to the CosSim kernel. However, this operation does not improve the recognition performance.

E. CROSS-DOMAIN FEW-SHOT RECOGNITION

This section evaluates the model in cross-domain few-shot recognition scenarios, in which the training and testing tasks are from different datasets. We train the model on a simulated SAR dataset [41] and evaluate it on the MSTAR dataset. The computer-aided design (CAD) models of vehicles are fed into electromagnetic simulation software to produce simulated SAR images. The resolution of the simulated image is 0.3m × 0.3m, and the image size is 128 × 128 pixels, which are identical to those of real SAR images. The simulated...
dataset includes seven vehicles: cars, trucks, bulldozers, buses, Humvees, motorcycles, and tanks. Each vehicle's images are produced by two CAD models, so each vehicle contains two sub-categories. Simulated images are acquired at 7 depression angles (15°, 17°, 25°, 30°, 35°, 40°, and 45°) and 72 azimuth angles (0°–360°, 5° intervals). Compared with real images, simulated SAR images' categories and depression angles are more abundant. TABLE 7 describes the details of the simulated dataset.

The experimental setting used in this section is referenced from DKTS-N [23]. This method achieves state-of-the-art performance in cross-domain few-shot SAR-ATR. Our model randomly builds 10-way K-shot tasks (K=1, 5, 10, and 25) from the simulated dataset for training. It is then tested with 10-way K-shot tasks sampled from the MSTAR dataset. To achieve better performance in cross-domain few-shot tasks, our model adopts a more sophisticated embedding network [42] and fine-tunes the GPs during testing. The network structure and training details can be found in [42].

The results show that our model surpasses those of other comparison methods in 1-shot and 5-shot tasks.

Future work will focus on cross-domain few-shot recognition problems, in which the source tasks and target tasks are from distinct domains (e.g., simulated SAR data and real SAR data).

| Category ID | Target CAD Model | Size |
|-------------|------------------|------|
| 1           | Bulldozer 8020   | 72x7 |
| 2           | Bulldozer 13013  | 72x7 |
| 3           | Bus 30726        | 72x7 |
| 4           | Bus 55473        | 72x7 |
| 5           | Car Toyota       | 72x7 |
| 6           | Car Peugeot      | 72x7 |
| 7           | Humvee 3663      | 72x7 |
| 8           | Humvee 9657      | 72x7 |
| 9           | Motorbike 3651   | 72x7 |
| 10          | Motorbike 3972   | 72x7 |
| 11          | Tank 63047       | 72x7 |
| 12          | Tank 86347       | 72x7 |
| 13          | Truck 2096       | 72x7 |
| 14          | Truck 2107       | 72x7 |

V. CONCLUSION

Due to the cost and complexity of the acquisition of SAR data, few-shot recognition is a promising research direction in SAR-ATR. This paper proposes a few-shot SAR-ATR model based on deep kernel learning. The model provides a principal framework incorporating deep learning and kernel methods, realizing the knowledge transfer from known target categories to novel target categories. Specifically, a deep CNN maps samples into a low-dimensional embedding space. The GPs with various kernel functions are responsible for measuring the similarity of embedded samples. Experimental results show that our model’s average accuracies surpass those of other comparison methods in 1-shot and 5-shot tasks.

Future work will focus on cross-domain few-shot recognition problems, in which the source tasks and target tasks are from distinct domains (e.g., simulated SAR data and real SAR data).

TABLE 7. Details of The Simulated Dataset. The size of each category is the product of the number of azimuths and depressions it contains.

| Methods                  | Accuracy (%) |
|--------------------------|--------------|
|                          | 1-shot | 5-shot | 10-shot | 25-shot |
| Ours                     | 45.37   | 69.14  | 79.45   | 91.73   |
| DKTS-N                   | 49.26   | 72.32  | 84.59   | 96.15   |
| Prototypical-Net         | 37.48   | 55.76  | 63.10   | 70.96   |
| Relation-Net             | 34.12   | 49.67  | 56.11   | 57.58   |
| Matching-Net             | 44.26   | 64.45  | 75.74   | 86.20   |

REFERENCES

[1] K. El-Darymli, E. W. Gill, P. McGuire, D. Power, and C. Moloney, “Automatic target recognition in synthetic aperture radar imagery: A state-of-the-art review,” IEEE Access, vol. 4, pp. 6014–6058, 2016.
[2] C. Nilubol, R. M. Mersereau, and M. J. T. Smith, “A SAR target classifier using transforms and hidden Markov models,” Digital Signal Process., vol. 12, nos. 2–3, pp. 274–283, 2002.
[3] L. M. Novak, G. J. Owirka, W. S. Brower, and A. L. Weaver, “The automatic target recognition system in SAIP,” Lincoln Lab. J., vol. 10, no. 2, pp. 187–201, 1997.
[4] S. Chen, H. Wang, F. Xu, and Y-Q. Jin, “Target classification using the deep convolutional networks for SAR images,” IEEE Trans. Geosci. Remote Sens., vol. 54, no. 8, pp. 4806–4817, Aug. 2016.
[5] S. A. Wagner, “SAR ATR by a combination of convolutional neural network and support vector machines,” IEEE Trans. Aerosp. Electron. Syst., vol. 52, no. 6, pp. 2861–2872, Dec. 2016.
[6] P. Zhao, K. Liu, H. Zou, and X. Zhen, “Multi-stream convolutional neural network for SAR automatic target recognition,” Remote Sens., vol. 10, no. 9, pp. 1473–1495, Sep. 2018.
[7] J. Pei, Y. Huang, W. Huo, Y. Zhang, J. Yang, and T-S. Yeo, “SAR automatic target recognition based on multiview deep learning framework,” IEEE Trans. Geosci. Remote Sens., vol. 56, no. 4, pp. 2196–2210, Apr. 2018.
[8] J. H. Cho and C. G. Park, “Multiple feature aggregation using convolutional neural networks for SAR image-based automatic target recognition,” IEEE Geosci. Remote Sens. Lett., vol. 15, no. 12, pp. 1882–1886, Dec. 2018.
[9] Y. Kwak, W.-J. Song, and S.-E. Kim, “Speckle-noise-invariant convolutional neural network for SAR target recognition,” IEEE Geosci. Remote Sens. Lett., vol. 16, no. 4, pp. 549–553, Apr. 2019.
[10] J. Ding, B. Chen, H. Liu, and M. Huang, “Convolutional neural network with data augmentation for SAR target recognition,” IEEE Geosci. Remote Sens. Lett., vol. 13, no. 3, pp. 364–368, Mar. 2016.
[11] T. Jiang, Z. Cui, Z. Zhou, and Z. Cao, “Data augmentation with Gabor filter in deep convolutional neural networks for SAR target recognition,” in Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS), Valencia, Spain, Jul. 2018, pp. 689–692.
[12] Z. Cui, M. Zhang, Z. Cao, and C. Cao, “Image data augmentation for SAR sensor via generative adversarial nets,” IEEE Access, vol. 7, pp. 42255–42268, 2019.
[13] Y. Sun, Y. Wang, H. Liu, N. Wang, and J. Wang, “SAR target recognition with limited training data based on angular rotation generative network,” IEEE Geosci. Remote Sens. Lett., vol. 17, no. 11, pp. 1928–1932, Nov. 2020.
[14] C. Kang and C. He, “SAR image classification based on the multi-layer network and transfer learning of mid-level representations,” in Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS), Beijing, China, Jul. 2016, pp. 1146–1149.
J. Snell, K. Swersky, and R. S. Zemel, “Prototypical networks for few-shot learning,” in Proc. 19th Int. Conf. Artif. Intell. Statist. (AISTATS), Cambridge, MA, USA: MIT Press, 2016, pp. 34–51.

D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. Adv. Neural Inf. Process. Syst., Barcelona, Spain, 2016.

A. Kusk, A. Abualitijiang, and J. Dall, “Synthetic SAR image generation using sensor, terrain and target models,” in Proc. EUSAR, Hamburg, Germany, 2016, pp. 405–408.

K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

K. Wang et al.: Few-Shot SAR Target Recognition Based on Deep Kernel Learning