Cross-Domain Few-Shot Segmentation for Remote Sensing Image Based on Task Augmentation and Feature Disentanglement

Jiehu Chen, Xili Wang, Ling Hong, and Ming Liu

***Abstract*—Few-shot segmentation aims to segment a large number of unlabeled samples in the target domain, by leveraging the images and labels from the source domain as well as a few labeled samples from the target domain. This is pivotal in tackling the scarcity of labeled samples in remote sensing image segmentation tasks. However, prevalent few-shot segmentation methods overlook inter-domain discrepancies, do not model and leverage the relationship between samples, and often only implement binary classification but not multi-class classification directly. To address these problems, we propose a cross-domain few-shot segmentation method based on task augmentation and feature disentanglement for practical remote sensing segmentation tasks. On one hand, task augmentation, which involves increasing the diversity of the training set and generating more challenging training data, can improve the model's generalization. On the other hand, feature disentanglement, involving the extraction of domain-irrelevant features for segmentation, improves the transferability of the model. Furthermore, to flexibly capture the relationships between the segmented regions, a graph with regions as nodes and relationships between nodes as edges is constructed. Then, labels are propagated from the labeled nodes to the unlabeled nodes in the graph by label propagation algorithm to implement multi-class classification directly. We conducted experiments on two public datasets as well as a Tibetan Plateaudataset collected by our group.And the experimental results show that the proposed method leads to a significant improvement in accuracy compared to existing methods, demonstrating its effectiveness.**

***Index Terms*—Cross domain, feature disentanglement, few-shot segmentation, remote sensing image (RSI), task augmentation (TA).**

I. Introduction

BENEFITING from the rapid progress in remote sensing (RS) technology, remote sensing images (RSIs) have entered the era of Big Data, and the automatic analysis and understanding of these abundant RSIs have become an active research field in the RS and computer community [1]. Getting land cover information from RSIs by semantic segmentation is fundamental research in RSIs [2]. Semantic segmentation determines the land cover type on each image pixel and provides both semantic and location information for the earth observation and land use. Land cover information can be quickly obtained from RSIs by semantic segmentation and further applied in land planning, environmental protection, disaster monitoring [3], and other fields. Deep-learning-based methods are the mainstream in RSI land cover segmentation research. Traditional deep learning networks require a large amount of labeled training data, which, in reality, are time consuming and labor intensive to obtain. Compared with computers, humans are often able to utilize prior knowledge to quickly identify new things given only a few or even one sample. Enabling the machines to quickly segment the objects by leveraging existing information and a few labeled samples, like humans do, is the target of few-shot segmentation. Few-shot segmentation has become research cutting edge in the field of deep learning in recent years since it has more practical value though challenging [4].

In the existing few-shot segmentation tasks, two datasets are usually involved: 1) a source dataset D_S containing a large number of labeled samples and 2) a target dataset D_T containing only a few labeled samples and a majority of unlabeled samples. The corresponding label sets of the two datasets are Y_S and Y_T. The purpose of few-shot segmentation is to train a model using D_S and the few labeled samples in $D_T$that has good generalization for D_T. In D_T, the labeled samples form the support set and the unlabeled samples constitute the query set.D_S and D_T usually come from the same source but have disjoint category sets Y_S and Y_T, i.e., $Y_S \cap Y_T = \phi$.

Few-shot semantic segmentation is pioneered by Shaban et al. [5]. Later works mainly adopted the metric-based mainstream paradigm [6] with various improvements. The main idea of the metric-based method is to obtain the feature representation (i.e., prototype) of each category with the support set. Then, the categories of unlabeled pixels are predicted by the nearest distance between the category prototypes and unlabeled pixel features.

When the existing few-shot segmentation methods are directly applied to RSI land cover segmentation, three problems arise.

1) Most of the existing methods disregard the interdomain discrepancies between D_S and D_T, and assume that

Manuscript received 5 November 2023; revised 2 March 2024; accepted 17 April 2024. Date of publication 23 April 2024; date of current version 6 May 2024. This work was supported in part by the Second Tibetan Plateau Scientific Expedition and Research under Grant 2019QZKK0405 and by the National Natural Science Foundation of China under Grant 42361056. *(Corresponding author: Xili Wang.)*

The authors are with the School of Computer Science, Shaanxi Normal University, Xi'an 710119, China (e-mail: chenji_snnu@foxmail.com; wangxili@snnu.edu.cn; hongling@snnu.edu.cn; mliu@snnu.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2024.3392549

different datasets are derived from the same source. However, due to differences in sensors, shooting angles, geographical locations, etc., there are usually interdomain discrepancies between D_S and D_T in RSI segmentation tasks. At the same time, due to the limited diversity of the dataset, the model trained on D_S may not generalize well to D_T, which results in performance degradation when predicting samples inD_T.

2) The existing few-shot segmentation methods are mainly based on the convolutional neural network (CNN) structure, while the CNN only considers the local spatial relationships but cannot flexibly model the relationships among distant regions [7].
3) In the existing few-shot segmentation setting [5], only two categories need to be segmented (i.e., foreground and background) within a single image. Most methods are limited to binary classification with such a setting. But in RSIs, images often contain multiple categories where each of which needs to be segmented, which is a more challenging and realistic problem.

To solve the above problems, we propose a cross-domain few-shot multiclass segmentation method for the RSI based on task augmentation and feature disentanglement (TAFD). On the one hand, we introduce the idea of task augmentation [8] and feature disentanglement [9] to improve the transferability of the model from two levels. First, at the data level, we design a data augmentation method to improve the generalization ability of the model. Traditional data augmentation generally enlarges the training set through operations, such as rotation and random crop, which cannot provide more diverse training data for cross-domain tasks with large differences in data. Different from traditional data augmentation, we propose the method "task augmentation." This method expands data distribution space through transformed data using data transformation and benefits for cross-domain tasks. Second, at the feature level, domain-irrelevant features are extracted by feature disentanglement and used for label predictions; this mitigates the impact of interdomain discrepancies on segmentation. On the other hand, in order to flexibly capture relationships between local or long-range regions in an image or regions from different images, we first obtain superpixels by oversegmentation and then model the relationships among superpixels by the graph. With the graph, predictions can be made through label propagation. But for images, there are often unlabeled superpixels that do not have a path to any labeled superpixels; predictions are made by the metric distance in such cases.

In addition, the existing methods assume that the categories in D_S and D_T are disjoint [10], i.e., $Y_S \cap Y_T = \phi$. But in RSI few-shot segmentation tasks, the categories of D_S and D_T often partially overlap, as shown in Fig. 1. Therefore, in this article, we broaden the definition, and the cases $Y_S \cap Y_T \neq Y_T$ are all considered as few-shot segmentation.

The contributions of this article are threefold.

1) A cross-domain few-shot segmentation method TAFD is proposed to address practical problems in RS. TAFD significantly reduces the negative impact of domain discrepancies on segmentation. Moreover, by leveraging the relationships between data in the graph model, it provides additional information for label prediction. Compared to the existing methods, TAFD improves segmentation accuracy notably.
2) TAFD is proposed to improve the transferability of the model. At the data level, a task augmentation method based on gradient ascent is presented to transform training data in each iteration, which can indirectly extend data distribution and enhance the generalization ability of the model in a wider data distribution space; at the feature level, domain-irrelevant features extracted by feature disentanglement are utilized for segmentation; this reduces the impact of interdomain discrepancy and ultimately improves the model accuracy.
3) A graph network is utilized to model the relationships among regions from different or the same image, so that the structural information implicit in the data can be exploited, and the labels can be spread through label propagation on the graph.

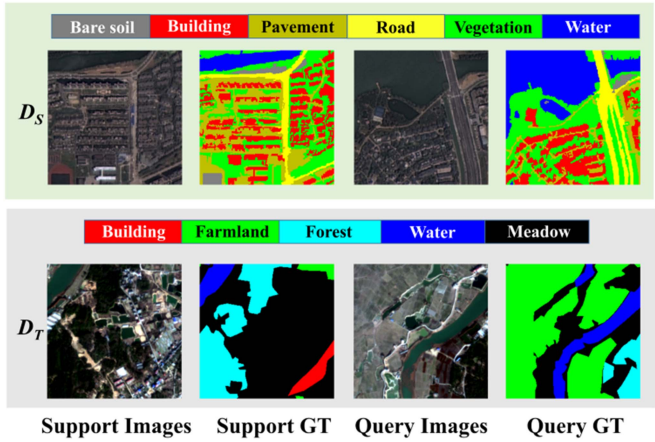


Fig. 1. Illustration of cross-domain few-shot segmentation for RS land cover, which aims to segment query images using D_S and support images in D_T.

The rest of this article is organized as follows. Section II gives a brief introduction to the related works. Section III describes our proposed method in detail. Section IV reports and analyzes the results on two public RS few-shot datasets and our collected Tibetan Plateau dataset, with comparisons to several comparable methods. Section V discusses the advantages and limitations of the proposed method. Finally, Section VI concludes this article.

II. Related Works

In this section, we will review the works related to few-shot segmentation, cross-domain few-shot learning, and graph neural networks (GNNs).

A. Few-Shot Segmentation

The purpose of few-shot segmentation is to segment a large number of unlabeled samples in the target domain D_T by leveraging the images and labels from domain D_S as well as a few labeled samples from domain D_T. Shaban et al. [5] first define the few-shot image segmentation problem and introduce

a method to address it. Their approach employs an encoder–decoder architecture, with decoder parameters adjusted based on the support set, enabling elementwise classification of the query set. Subsequently, more research has been done on few-shot segmentation.

Few-shot segmentation methods can be broadly classified into two types: metric-based methods [6], [11], [12], [13], [14], [15], [16] and weighted mask methods [17], [18], [19], [20], [21], [22] according to different decoders. Metric-based methods generally aim to obtain the prototype of each category by mask average pooling [11] with the support set's feature maps and then predict labels by measuring the metric distances between prototypes and pixel-level features in query images. Metric-based methods can be further divided into single-prototype methods [11], [12], [13], [14] and multiprototype methods [15], [16]. In the single-prototype approaches, each category is represented by one feature vector. Some methods compute prototypes only by a few support samples [11], [12], [13], which can easily fail to cover the underlying appearance discrepancy between the support set and the query set. To solve the problem and enhance prototype representation of each category, some methods compute the prototypes with both support images and high-confidence query predictions [14]. Single prototype is usually insufficient to represent a category with a complex appearance; therefore, the multiprototype methods are proposed. In the multiprototype methods, clusters are first obtained by clustering methods [15], [16] on the regions corresponding to each category in the support set, and then, each category is represented by multiple feature vectors (multiprototype) derived from clusters. Compared with single prototypes, multiprototype methods are often able to capture the diversity of objects but have higher computational complexity.

Besides the metric-based methods, some methods are implemented based on the weighted mask methods [17], [18], [19], [20], [21], [22]. Weighted mask methods adopt the prototype of the foreground from the support feature map to compute elementwise cosine distance with the query feature map to attain a weighted mask. The weighted mask that highlights the region of the foreground is multiplied or concatenated with the query feature map. The predictions are generated by deconvolution on the weighted query feature map. Restricted by the structure of the deconvolutional module, the weighted mask methods generally only solve the binary category segmentation task in one image and are difficult to generalize to RS segmentation tasks involving multiclassification.

The abovementioned methods assume that D_S and D_T come from the same domain, which often fails to hold in RSI segmentation tasks. Therefore, it is often difficult for them to obtain high segmentation accuracy in cross-domain segmentation.

B. Cross-Domain Few-Shot Learning

In recent years, some scholars have conducted research on the challenging cross-domain few-shot learning problem (CD-FSL) [23]. CD-FSL mainly solves the problem of few-shot classification or segmentation tasks when there are domain discrepancies between D_S and D_T, and the category sets are disjoint. In the early stage, research mainly focuses on the field of image classification [24]. Cross-domain few-shot segmentation research has been started since 2022 [25], [26], [27]. The methods mainly include domain-agnostic feature extraction [25], [26] and fine-tuning method with labeled samples in D_T [27].

The above cross-domain methods do not generalize well due to the limited diversity of training data. And they ignore the relationships between distant regions in the image. In addition, most of the abovementioned methods can only perform binary classification. Their applications are limited since we need to segment all the land cover categories in the RSI.

C. Graph Neural Networks

GNNs [28] are effective tools for modeling the relationships between nodes (data), which have been widely used in computer vision. Graph convolutional networks can model long-range spatial as well as local relations in the RSI naturally, while CNNs only consider local relations. In the field of image segmentation, a pixel or superpixel is treated as a node, and the edge represents the relationship between two nodes [29]. The graph provides more information implied in data, and this is beneficial for obtaining distinguishing features. Image segmentation can be implemented by sending the learned features on the graph to a classifier (such as softmax) or label propagation between nodes on the graph. Graph construction is the key in GNN-based image segmentation methods. Some methods directly construct a graph at the pixel level [30], [31]. However, when the image becomes large in size, such a fashion could lead to a tremendous amount of computation, which limits their applicability for general computers. To solve this problem, the methods that construct a graph at the superpixel level [32], [33] or in a mini-batch fashion [34] are proposed, which enables GNNs to model spatial structures of RSIs with acceptable computational overhead. However, the existing methods usually construct a graph on a single image, which cannot capture the relationship among superpixels of different images.

D. Domain Adaptation

Domain adaptation strives to address the performance degradation by the data distribution discrepancy between the source domain and the target domain. In semantic segmentation tasks, most studies endeavor to minimize domain gaps between the two domains through adversarial training. Some of these approaches focus on aligning distributions in the output space [35], while others seek to eliminate discrepancies in the input level or (and) feature level [36], [37]. Current domain adaptation methods for image segmentation predominantly target unsupervised domain adaptation tasks where the categories of the source and target domains are entirely identical. There are relatively fewer research works on unsupervised open-set domain adaptation [38] or few-shot domain adaptation [9], [39], [40], and most of them study image-level classification but not pixel-level semantic segmentation. Tavera et al. [40] first propose a domain adaptation method for the few-shot segmentation. However, this method requires an additional dataset apart from D_S and D_T during

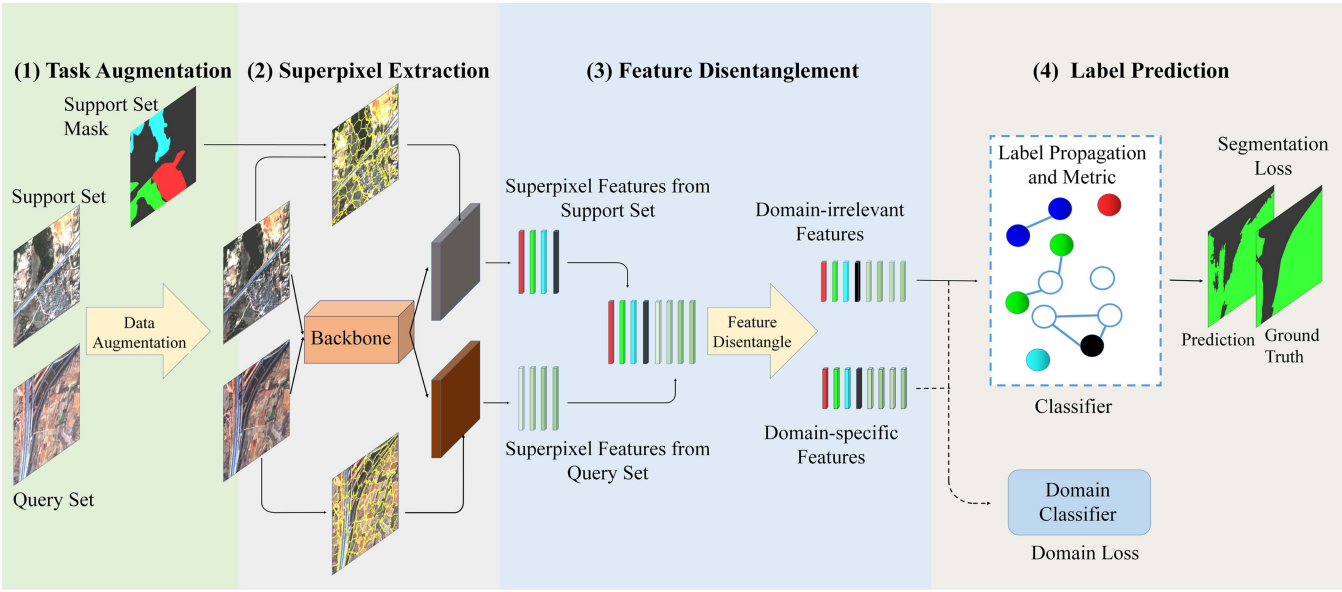


Fig. 2. Illustration of the proposed model. The proposed model can be divided into four parts: task augmentation, superpixel extraction, feature disentanglement, and label prediction.

training. This additional dataset needs to share the same label set with D_S and style properties with D_T, which poses significant difficulties in RS applications.

III. Methodology

The proposed method models the relationships among superpixels with the help of graph and spreads the labels from the labeled nodes to those unlabeled nodes by label propagation on the graph. Especially, feature disentanglement and task augmentation are proposed to improve the generalizability of the model. The model framework is shown in Fig. 2.

The model includes four parts: 1) task augmentation, which can improve the generalization ability of the model by augmenting the diversity of training data; 2) superpixel extraction, which extracts features and generates superpixels; 3) feature disentanglement, which extracts the domain-irrelevant features from the superpixels; and 4) label prediction, in which a graph network is first used to model relationships among superpixels with domain-irrelevant features and spread labels from labeled superpixels to unlabeled ones by label propagation. Then, the labels of the remaining unlabeled nodes are obtained through a metric method. Finally, the superpixels are mapped back to the size of the original image to obtain the segmentation result. We will explain each part and the training process of the model in the following subsections.

A. Task Augmentation

We propose a data augmentation method to increase the diversity of the training data (i.e., D_S and a few labeled samples in D_T) in the training phase. Data augmentation is a commonly employed technique in deep learning, especially when training data are scarce or a domain discrepancy exists between the training and test datasets. It often results in improved performance [41]. Traditional data augmentation (such as flipping, rotation, adding noise, and so on) can increase the quantity of training data, but it does not guarantee that the additional data will enhance the performance of the model [42]. Motivated by Wang and Deng [8], we propose a task augmentation method for cross-domain image segmentation. Compared with the traditional data augmentation method, the proposed method does not directly increase the amount of data, but iteratively transforms the training data during the training phase and produces new samples that are harder to segment. This implicitly expands the distribution space of samples and then improves the segmentation accuracy in the cross-domain task.

Let D_{train} denote the distribution of training data, and D_{test} denote the distribution of test data. The distribution distance between D_{train} and D_{test} is ρ. We expect that the model trained on D_{train} also achieves good performance on D_{test}; it is equivalent to finding the solution to the following problem:

$$\min_{\theta\in\Theta} \sup_{\text{Dist}(D_{\text{train}},D_{\text{test}})\le\rho} E[L((X,Y);\theta)] \tag{1}$$

where sup denotes supremum. The model parameter is θ, and Θ is the parameter space of the model. $(X,Y)\in D_{\text{test}}$ is a sample with its label come from D_{test}. $E[\cdot]$denotes the mathematical expectation and $L((X,Y);\theta)$ denotes the segmentation loss of predicting Y from X (see formula (19) in Section III-E). Aiming at finding the minimum value of the supremum of $E[L((X,Y);\theta)]$ in D_{test}, the solution to formula (1) can guarantee good performance over a wider distribution space with a distance of ρ from D_{train} compared to simply training on D_{train}. Since the labels in D_{test} are unknown, it is difficult to solve formula (1). The literature [8] proposes to transform the training data with task augmentation to generate fictitious test data outside the distribution of training data and uses these

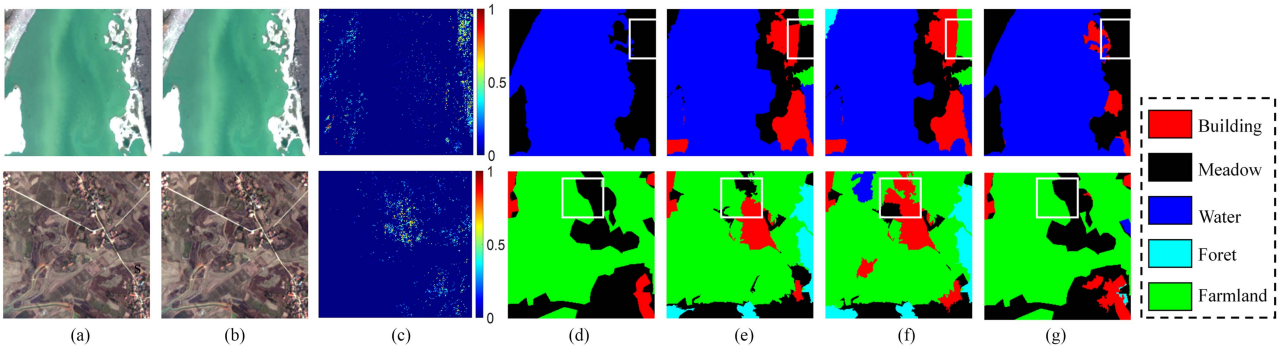


Fig. 3. Examples of images before and after task augmentation, heatmap of the changing pixels, and predicted results before and after training on task augmentation (TA) data. (a) Images before TA. (b) Images after TA. (c) Pixel change heatmap. (d) True labels. (e) Predicted results of images before TA (without training on TA data). (f) Predicted results of images after TA (without training on TA data). (g) Predicted results of images after TA (training on TA data).

fictitious test data to train the model. Thus, the model can achieve better generalization performance for cross-domain data.

Task augmentation is implemented iteratively. Assuming that the current sample from training set is X_0, the augmented data X_M (fictitious test data) are obtained through M iterations as follows:

$$X_i = X_{i-1} + \beta \cdot \nabla_{X_{i-1}} L((X_{i-1}, Y_0); \theta), \quad i = 1, 2, \ldots M \tag{2}$$

where Y_0 is the ground-truth mask of X_0(task augmentation does not change the mask; Y_0 is also the ground-truth mask of X_i). ∇ is the symbol for the gradient. β denotes the step size and $\beta > 0$. Too large β would lead to difficulties in model convergence during training. The difference between X_M and X_0 increases as M increases, bringing an expanded sample distribution space. However, a large M value will increase computation. In the experiment, it is found that $M = 10$ and $\beta = 0.0001$ can obtain good results without adding much time complexity.

To better understand formula (2), we substitute X_M generated by (2) into the loss function $L(\cdot)$and obtain

$$\begin{aligned} &L((X_M, Y_0); \theta) \\ &= L((X_{M-1} + \beta \cdot \nabla_{X_{M-1}} L((X_{M-1}, Y_0), \theta), Y_0); \theta). \end{aligned} \tag{3}$$

Let $\Delta X_{M-1} = \beta \cdot \nabla_{X_{M-1}} L((X_{M-1}, Y_0), \theta)$; then, we have

$$\begin{aligned} &L((X_M, Y_0); \theta) = L((X_{M-1} + \Delta X_{M-1}, Y_0); \theta) \\ &\approx L((X_{M-1}, Y_0); \theta) + \beta \cdot (\nabla_{X_{M-1}} L((X_{M-1}, Y_0); \theta))^2 \\ &\geq L((X_{M-1}, Y_0); \theta) \geq \cdots L((X_0, Y_0); \theta). \end{aligned} \tag{4}$$

From formula (4), it can be seen that the segmentation loss obtained by X_M is greater than or equal to that obtained by X_0. This means that we generate more "challenging" (i.e., difficult to train) data for model training. These harder data impel the model to achieve satisfactory results, thus improving the generalization of the model more effectively than the traditional data augmentation.

In the training phase, we sample a batch of labeled data in each iteration and perform data augmentation and then send them to the model for training. All of the training data in each epoch are transformed in this way. Since the value of the loss function changes continuously during training, the augmented data also change in each iteration, which enables the model to get more diverse data and improve the performance of the model.

The purpose of data augmentation is to generate more challenging data for model training, thus enhancing the performance of the model. Fig. 3 displays some examples before [see Fig. 3(a)] and after [see Fig. 3(b)] task augmentation. Due to the limited number of iterations of task augmentation, the visual difference between images before and after task augmentation is slight. To clearly demonstrate the differences before and after task augmentation, we calculate $\hat{X}^{(x,y)} = \|X_M^{(x,y)} - X_0^{(x,y)}\|_2 / \max_{i,j}(\|X_M^{(i,j)} - X_0^{(i,j)}\|_2)$ for the corresponding pixels before and after task augmentation, where $X_0^{(x,y)}$ and $X_M^{(x,y)}$ represent the pixel at position (x, y) before and after task augmentation, respectively. The value of $\hat{X}$ ranges from 0 to 1, where closer to 1 indicates that the corresponding pixel changes larger after task augmentation. We visualize $\hat{X}$ using a heatmap, as shown in Fig. 3(c). Fig. 3(e) and (f) represents the obtained segmentation results of X_0 and X_M using the model that trained on data without task augmentation. Fig. 3(g) illustrates the predicted results of X_M obtained by the model trained on data with task augmentation. Comparing Fig. 3(e) and (f), it can be seen that the misclassified regions are increased after task augmentation, indicating that task augmentation indeed generates more challenging training data. Comparing Fig. 3(f) and (g), it can be observed that after training on the augmented data, the model achieves better segmentation results on challenging data. Therefore, the model's performance is enhanced through this task augmentation approach.

B. Superpixel Extraction

A graph is used to model the relationships between nodes involving both adjacent and distant regions (i.e., nodes) in one image or from different images. To reduce the computational complexity, the graph is constructed at the superpixel level. As shown in Fig. 4(a), oversegment a batch of images by SLIC [43]. For the labeled images, labels (i.e., masks) can ensure that all the pixels in one region correspond to only one label. The obtained R oversegmented regions r_i, $i = 1, 2, \ldots, R$, are

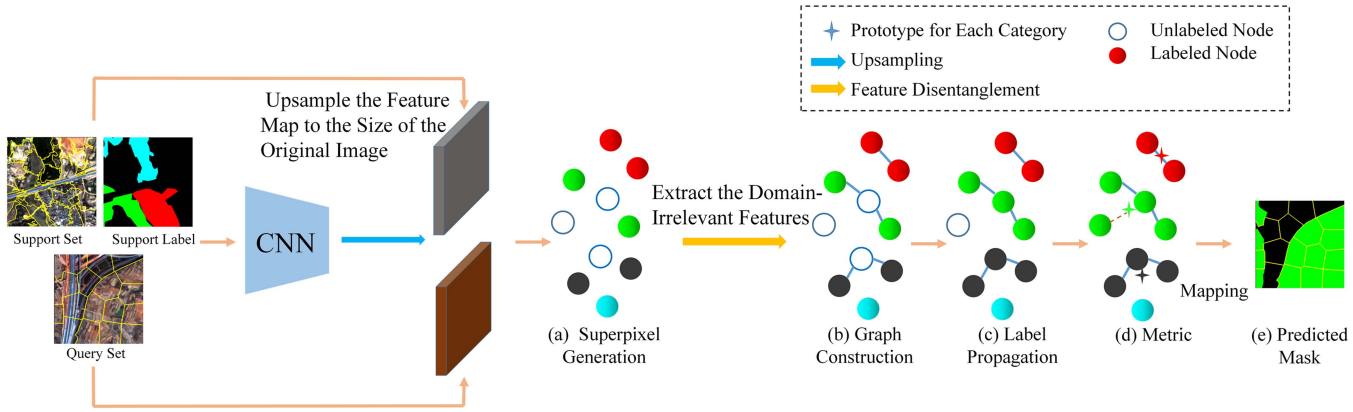


Fig. 4. Data processing flow of TAFD. (a) Extracting superpixels from the support set and the query set. (b) Constructing the graph with superpixels. (c) Spreading labels to the unlabeled nodes from labeled ones by label propagation. (d) Predict those unreachable unlabeled regions by metric distance. (e) Mapping the superpixels back to the original image.

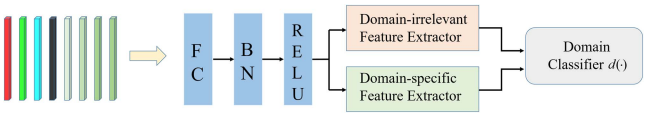


Fig. 5. Structure of the feature disentanglement network. Features from the labeled superpixels and unlabeled superpixels are concatenated together as the input; through feature disentanglement, the domain-irrelevant features and domain-specific features are obtained. The parameters of disentanglement are obtained by minimizing domain loss in the training phase.

regarded as R superpixels. A CNN (VGG16 or ResNet50 as in [13]) is used to extract the feature maps of the images and upsample the feature maps to the size of the original images. By averaging the corresponding feature maps on r_i, we can obtain the feature vectors $\boldsymbol{h}_i$ of r_i; then, $\boldsymbol{h}_i$ will be sent to the feature disentanglement part to extract domain-irrelevant features.

C. Feature Disentanglement

Feature disentanglement helps to find features that are invariant or specific to different data domains [9]. The domain-invariant feature is conducive to improving cross-domain generalization. Therefore, we use a feature disentanglement network $\phi(\cdot)$ to disentangle the superpixel's features $\boldsymbol{h}_i$ into domain-irrelevant features $\boldsymbol{h}_i^i$ and domain-specific features $\boldsymbol{h}_i^s$, namely:

$$(\boldsymbol{h}_i^i, \boldsymbol{h}_i^s) = \phi(\boldsymbol{h}_i). \tag{5}$$

The domain-irrelevant feature $\boldsymbol{h}_i^i$ is fed into the label prediction part to conduct the few-shot segmentation, while domain-specific feature $\boldsymbol{h}_i^s$ is abandoned since they contain inductive bias learned from a particular domain. The structure of the feature disentanglement network is the same as in [9], primarily comprising a batch normalization layer, a rectified linear unit (ReLU) activation layer, and several fully connected layers, as depicted in Fig. 5.

A domain classifier $d(\cdot)$ is set to help with feature disentanglement. For the domain-irrelevant feature $\boldsymbol{h}_i^i$, we cannot distinguish whether the feature derives from D_S or D_T. $d(\cdot)$ implements this discrimination, and its output is approximately 0.5, so the loss function is

$$L_{D1} = \sum_{n=1}^{N_S} \mathrm{KL}(d(\boldsymbol{h}_n^{i(S)}), 0.5) + \sum_{n=1}^{N_T} \mathrm{KL}(d(\boldsymbol{h}_n^{i(T)}), 0.5) \tag{6}$$

where $\mathrm{KL}(\cdot)$ is Kullback–Leibler divergence loss, N_S is the number of superpixels from D_S, and N_T is the number of superpixels from D_T. By minimizing (6), the output feature $\boldsymbol{h}_i^i$ will confuse $d(\cdot)$ and make the classification probabilities of both domains' data are around 0.5, which means that $\boldsymbol{h}_i^i$ does not contain domain-specific information. $\boldsymbol{h}_i^s$ contains much more domain-specific features, so $d(\cdot)$ may distinguish whether the input features come from D_S or D_T more easily. We set the output of $d(\cdot)$ to be 0 for samples from D_S and 1 for samples from D_T. Therefore, the domain loss function is

$$L_{D2} = \sum_{n=1}^{N_S} \mathrm{CE}(d(\boldsymbol{h}_n^{s(S)}), 0) + \sum_{n=1}^{N_T} \mathrm{CE}(d(\boldsymbol{h}_n^{s(T)}), 1) \tag{7}$$

where $\mathrm{CE}(\cdot)$ is the cross-entropy loss. By minimizing (7), the disentangled $\boldsymbol{h}_i^s$ contains specific information related to each domain.

The total domain loss is

$$L_D = L_{D1} + L_{D2}. \tag{8}$$

The training process of the feature disentanglement will be described in Section III-E.

D. Label Prediction

The label prediction process predicts labels using domain-irrelevant features. Label prediction consists of three steps: label propagation, metric prediction, and mapping superpixels to mask maps. Label propagation spreads labels from the labeled superpixels to unlabeled ones if there are paths between them, as shown in Fig. 4(b) and (c). There may be some unlabeled nodes that are not connected to any of the labeled samples (the graph is not fully connected, and this is common for images);

in this scenario, the metric-based method is used to predict the categories of such nodes.

The weight between two nodes in the graph is defined by the Gaussian function

$$\boldsymbol{A}_{ij} = \exp\left(-\frac{\left\|\boldsymbol{h}_i - \boldsymbol{h}_j\right\|_2^2}{2\sigma^2}\right) \tag{9}$$

where $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$ are the domain-irrelevant features corresponding to the superpixels *i* and *j*. σ is the scale parameter. The value of σhas an impact on the classification results. The weight of the edge between any two nodes is close to 0 when σ is too small, making it difficult for label propagation. If σ is too large, two nodes with low similarity will get a large weight, causing many nodes of different categories to interfere with each other. Therefore, we select σ adaptively based on a trained network $g(\cdot)$whose inputs are the node features [44]. The similarity function is

$$\boldsymbol{A}_{ij} = \exp\left(-\frac{1}{2}\left\|\frac{\boldsymbol{h}_i}{\sigma_i} - \frac{\boldsymbol{h}_j}{\sigma_j}\right\|_2^2\right) \tag{10}$$

where$\sigma_i = g(\boldsymbol{h}_i)$, $\sigma_j = g(\boldsymbol{h}_j)$, and $g(\cdot)$ consists of a two-layer fully connected network adding an activation layer. In the training phase, $g(\cdot)$ learns how to produce appropriate σaccording to the input feature to achieve high accuracy. Afterward, in the testing phase, $g(\cdot)$ automatically generates suitable σ for test data. To prevent the nodes of different categories connected, for labeled nodes *i* and *j*, $\boldsymbol{A}$ is further processed as

$$\boldsymbol{A}_{ij} = 0 \quad \text{if} \quad y_i \neq y_j \tag{11}$$

where y_i and y_j are the labels of nodes *i* and *j,* respectively.

When either *i* or *j* is an unlabeled node, we require only that when $\boldsymbol{A}_{ij}$ is greater than a threshold η, there is an edge between nodes *i* and *j*. This helps to prevent nodes of different categories from being connected to each other. Therefore,$\boldsymbol{A}$ goes on processing as

$$\boldsymbol{A}_{ij} = 0 \quad \text{if} \quad \boldsymbol{A}_{ij} < \eta \tag{12}$$

$\eta \in (0, 1)$ is a threshold. If η is small, there are edge connections between nodes of different categories, which will have a negative impact on segmentation. Therefore, η should be a value close to 1.

Label propagation can be expressed as

$$\boldsymbol{F}_{t+1} = \alpha\tilde{\mathbf{A}}\boldsymbol{F}_t + (1-\alpha)\boldsymbol{Y} \tag{13}$$

where $\tilde{\mathbf{A}} \in \mathbb{R}^{L\times L}$ represents the graph adjacency matrix after normalization. *L* is the number of superpixels. $\boldsymbol{Y} \in \mathbb{R}^{L\times N}$ refers to the initial label. $\boldsymbol{Y}_{i,j} = 1$ if the *i*th superpixel is labeled with *j*; otherwise, $\boldsymbol{Y}_{i,j} = 0$. *N* is the number of categories. $\boldsymbol{F}_t \in \mathbb{R}^{L\times N}$denotes the predicted results after *t* iterations. $\alpha \in (0, 1)$ controls the amount of propagated information. Small α slows down propagation, while large α leads to faster propagation. But large α may cause an oversmoothing issue since there exists an edge connection among nodes with different categories in the graph. Because most of the nodes with different categories are ensured not to be connected by formulas (11) and (12), α is set to a relatively large value of 0.99, which is also a recommended value by most works [44], [45], [46]. The final results can be calculated as follows:

$$\boldsymbol{F}^* = \left(\boldsymbol{I} - \alpha\tilde{\mathbf{A}}\right)^{-1}\boldsymbol{Y} \tag{14}$$

where $\boldsymbol{F}^*$ is the prediction result and $\boldsymbol{I}$ is an identity matrix.

Because there are some nodes whose weights to all other nodes are less than threshold η, the above-constructed graph is usually a disconnected graph and labels cannot be propagated to some unlabeled nodes. Therefore, we predict the labels of such nodes with the nearest distance of them to each category prototype, as shown in Fig. 4(d). The prototype for each category is computed as

$$\boldsymbol{c}_k = \frac{1}{|S(k)|}\sum_{j\in S(k)} \boldsymbol{h}_j \tag{15}$$

where $|S(k)|$ represents the number of the labeled superpixels belonging to category *k*. Because some labeled superpixels in formula (15) are from the query set, we can reduce the gap between the support set and the query set to some extent. Then, we obtain the node's predicted probability by softmax based on the Euclidean distances of its feature and the prototypes [47]

$$p(y = k|\boldsymbol{x}) = \frac{\exp(-\left\|\boldsymbol{h}_i - \boldsymbol{c}_k\right\|_2^2)}{\sum_i \exp(-\left\|\boldsymbol{h}_i - \boldsymbol{c}_i\right\|_2^2)}. \tag{16}$$

After assigning labels to all the superpixels, we map the superpixels back to the size of the original image and then obtain the segmentation result of the image, as shown in Fig. 4(e).

E. Training Processing

The whole training process contains two stages: the pretraining stage and the training stage. In the pretraining stage, parameters are initialized by training the model on D_S. The loss function in the pretraining phase is the segmentation loss. The segmentation loss consists of two parts, one of which is the cross-entropy loss

$$L_{CE} = -\frac{1}{N_p}\sum_{x,y}\sum_{i\in C} y^i{}_{(x,y)} \log(\hat{y}^i_{(x,y)}) \tag{17}$$

where $y^i_{(x,y)}$ represents the real label, $\hat{y}^i_{(x,y)}$ represents the predicted label, N_p represents the number of pixels, and *C* represents the category set.

Cross-entropy loss usually has poor performance for categories with small sample sizes, so dice loss [48] is also used in the training process

$$L_{\text{DICE}} = \frac{1}{|C|}\sum_{i\in C}\left(1 - \frac{2\sum_{x,y} y^i_{(x,y)}\hat{y}^i_{(x,y)}}{\sum_{x,y}\left(y^i_{(x,y)}\right)^2 + \sum_{x,y}\left(\hat{y}^i_{(x,y)}\right)^2}\right) \tag{18}$$

where $|C|$ represents the number of categories.

The segmentation loss in the pretraining phase is

$$L_C = L_{\text{CE}} + L_{\text{DICE}}. \tag{19}$$

In the training stage, the model is trained with D_S and a few labeled samples from D_T simultaneously. All training data are

TABLE I
CHARACTERISTICS OF DIFFERENT DATASETS

Dataset	Size	Number of categories	Category	Number of samples	Pixel resolution
WHDLD	256×256×3	6	Bare Soil; Building; Pavement; Road; Vegetation; Water	4940	0.6 m
GID5	256×256×3	5	Building; Farmland; Forest; Water; Meadow	4368	0.8−2 m
Tibetan Plateau	512×512×3	8	Road; Farmland; Snow; Construction Land; Building; Vegetation; Bare Soil; Water	3976	0.75 m

transformed to increase the diversity of the training set. The loss function of this stage is expressed as follows:

$$L = L_C^{(S)} + L_C^{(T)} + \lambda L_D \tag{20}$$

where $L_C^{(S)}$ and $L_C^{(T)}$ denote the segmentation loss on D_S and D_T, respectively. L_D is the domain loss given by formula (8). λ is a weight of the domain loss function and $\lambda > 0$. The domain loss may overpower $L_C^{(S)}$ and $L_C^{(T)}$ for a large λ. Therefore, it is recommended to be less than 1.

In the test phase, the target dataset is fed into the model to obtain segmentation results for the unlabeled samples.

IV. EXPERIMENTAL RESULTS

In this section, we first introduce the datasets used in the experiment in Section IV-A. Then, we describe the implementation details and the compared methods in Sections IV-B and IV-C, respectively. Finally, we present and analyze the experimental results in Sections IV-D–IV-F, where Section IV-D shows the performance of the proposed method on two public datasets, Section IV-E presents a series of ablation studies to demonstrate the impact of each part in our proposed method, and Section IV-F shows the results on our collected Tibetan Plateau dataset.

A. Dataset

Experiments are conducted using two public datasets, WHDLD [49] and GID5 [50], along with one private dataset Tibetan Plateau. WHDLD (Wuhan dense labeling dataset) is cropped from a large RS image of Wuhan urban area obtained by GF-1 and ZY-3 satellites; the images in WHDLD are manually labeled with six categories. GID5 contains 150 high-quality GF-2 images acquired from more than 60 different cities in China with five categories. The Tibetan Plateau dataset covers the city of Lhasa and its surrounding areas obtained by Jilin-1 satellite and is manually annotated into eight categories. The three datasets are summarized in Table I.

The above three datasets are acquired over different geographical regions. There are distinct distribution discrepancies among them. They also have different sets of categories. Due to the large amount of data in the GID5 dataset, we selected six images from GID5 and cropped them to a total of 4368 images with a size of 256×256.[1] The other two datasets do not undergo additional processing. In the experiments, we select one dataset as D_S, and the other dataset as D_T. K images, which cover all categories of D_T, are randomly sampled from D_T as the support set. In the pretraining phase, D_S is used to train the model to obtain initial parameters. Then, D_S and the support set from D_T are augmented by task augmentation to train the model. Finally, the model is evaluated on D_T. Because different support sets lead to different results, our experimental results are the average results on ten different support sets.

[1]The cropped GID5 can be downloadable at https://drive.google.com/file/d/1xw5xi66u2mYQeMN6OZ11NWcrUIJ28ApQ/view?usp=sharing

B. Implementation Details

The experiments are implemented on PyTorch, accelerated by NVIDIA GeForce RTX 3090 GPU. The CPU is AMD Ryzen 5900X. The initial learning step is 5.0×10^{-4}. During the training process, the learning rate is reduced by half every 2000 episodes. VGG16 and ResNet50 pretrained in ImageNet [13] are used as the backbone to extract features. A 1×1 convolution is used to fuse features from backbones and outputs 64-D features. The feature disentanglement network consists of three fully connected layers and a ReLU activation layer. It receives the 64-D features from backbones and outputs two 32-D disentangled features. Best results are obtained when η is 0.9, and λ is 0.5 through cross-domain few-shot segmentation experiments on two public datasets. Therefore,η is fixed to 0.9, and λ is fixed to 0.5 in the experiments.

To quantitatively evaluate the performance, we use the mean intersection over union (mIoU), mean F1 score, and overall accuracy (OA) as evaluation indexes [51]. The three metrics are calculated as follows:

$$\text{mIoU} = \frac{1}{N}\sum_{i=1}^{N}\frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i + \text{FN}_i} \tag{21}$$

$$\text{meanF1} = \frac{1}{N}\sum_{i=1}^{N}\frac{2\text{TP}_i}{2\text{TP}_i + \text{FP}_i + \text{FN}_i} \tag{22}$$

$$\text{OA} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{23}$$

where TP, TN, FN, and FP are the true positives, true negatives, false negatives, and false positives, respectively, and N denotes the number of categories.

C. Compared Methods

To evaluate the effectiveness of the method TAFD, we compare it with some comparable methods: PANet [6], CAPL [13], and two improved methods based on PANet. Except for a few methods such as PANet and CAPL, most of the existing few-shot

TABLE II
COMPARISON OF DIFFERENT METHODS

Method	Number of Prototypes	Reducing the Gap between Query Set and Support Set	Traditional Data Augmentation (Adding Noise and Random Clipping)	Task Augmentation	Feature Disentangling	Prediction
PANet [6]	Single	×	✓	×	×	Cosine similarity
ASGNet* [21]	Multiple	×	✓	×	×	Cosine similarity
SSP* [6], [14]	Single	✓	✓	×	×	Cosine similarity
CAPL [13]	Single	×	✓	×	×	Cosine similarity
TAFD (Ours)	Single	✓	×	✓	✓	Label Propagation + L2 Metric

TABLE III
RESULTS OF DIFFERENT METHODS WHEN D_S IS WHDLD, D_T IS GID5, AND SHOT IS FIVE

Method	Backbone	Category IOU					mIoU	mean F1	OA
		Building (4.67%)	Farmland (43.50%)	Forest (4.55%)	Water (7.04%)	Meadow (40.22%)			
PANet	VGG16	14.70±2.21	24.78±3.91	14.04±2.20	25.33±5.13	19.67±0.94	19.71±2.31	32.06±3.51	33.29±4.03
TAFD	VGG16	17.35±4.66	18.76±2.55	**21.79**±4.70	51.98±0.83	25.85±6.24	27.15±0.91	40.86±1.65	41.25±2.67
PANet	ResNet50	16.25±4.16	30.04±5.54	9.67±3.61	42.58±4.17	22.12±5.57	24.13±2.13	35.22±2.64	38.37±4.32
ASGNet*	ResNet50	18.42±3.48	26.14±6.69	12.09±2.96	41.21±3.72	27.28±3.66	25.02±2.02	38.42±2.48	39.11±4.17
SSP*	ResNet50	17.19±4.62	**31.21**±5.02	16.84±2.55	36.55±4.31	19.83±4.86	24.32±2.84	34.29±2.77	35.05±4.76
CAPL	ResNet50	13.58±4.51	17.54±6.33	3.72±2.76	52.54±6.15	27.97±3.28	23.07±2.95	27.13±2.91	33.47±4.82
TAFD	ResNet50	**20.08**±3.62	30.88±5.83	21.41±1.17	**54.55**±4.04	**29.81**±4.63	**31.35**±1.72	**44.60**±2.05	**44.69**±3.57

The bold values denote the best performance.

TABLE IV
RESULTS OF DIFFERENT METHODS WHEN D_S IS WHDLD, D_T IS GID5, AND SHOT IS TEN

Method	Backbone	Category IOU					mIoU	mean F1	OA
		Building (4.67%)	Farmland (43.50%)	Forest (4.55%)	Water (7.04%)	Meadow (40.22%)			
PANet	VGG16	14.79±3.35	34.79±2.07	14.24±1.69	31.55±5.05	13.18±1.59	21.70±1.97	34.16±2.87	38.00±1.48
TAFD	VGG16	18.01±3.74	20.83±2.01	**22.06**±2.78	55.93±1.54	28.75±4.10	29.11±1.01	43.18±1.72	46.41±2.81
PANet	ResNet50	13.09±3.96	32.64±6.15	17.15±2.86	48.86±2.10	28.53±1.69	28.05±2.05	39.52±2.45	44.12±3.17
ASGNet*	ResNet50	20.31±4.02	34.66±5.94	15.04±3.57	44.18±1.94	29.63±1.93	28.76±2.27	40.37±2.60	43.26±3.42
SSP*	ResNet50	18.47±4.39	36.53±4.39	11.98±2.97	44.66±2.25	25.37±1.75	27.40±2.78	37.89±2.73	40.69±3.61
CAPL	ResNet50	15.32±3.79	30.31±4.66	5.68±2.12	56.29±3.61	28.53±2.55	27.23±2.53	32.16±2.69	34.75±3.36
TAFD	ResNet50	**26.90**±3.01	**37.48**±6.14	21.49±2.26	**61.79**±1.13	**32.61**±2.45	**36.06**±1.77	**46.31**±1.99	**51.96**±2.45

The bold values denote the best performance.

segmentation methods only tackle binary classification (background and foreground) but not multicategory classification tasks directly. PANet is a method based on a single prototype, and a single prototype is generally hard to represent the diversity of categories [21]. Besides, PANet ignores the gap between the support set and the query set when calculating the prototype [14]. To enrich the compared methods, we propose two improved PANet methods based on ASGNet [21] and SSP [14], respectively. The first improved method ASGNet∗ is based on ASGNet, which represents each category by multiple prototypes using *K*-means, and the prediction of labels adopts the same metric as PANet. The second improved method SSP∗ enhances features, which is derived from the idea of SSP. To reduce the gap between the support set and the query set in binary category segmentation, SSP enhances the prototypes of the foreground and background by the high-confidence areas in the query set, respectively. We utilize the foreground enhancement method of SSP to enhance the categories' prototypes in PANet since there is no background in our datasets. Other parts in SSP∗ are the same as in PANet. The main differences between the proposed method and the compared methods are listed in Table II.

D. Results of the Public Datasets

Tables III and IV show the results of the proposed method and the compared methods in terms of accuracy when D_S is WHDLD and D_T is GID5. Fig. 6 shows some segmentation results of different methods. There are five categories in D_T, where

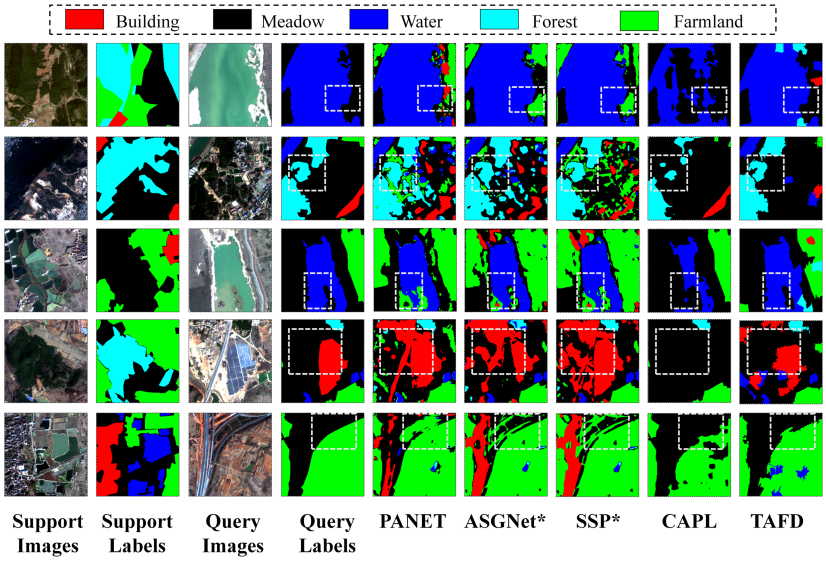


Fig. 6. Qualitative results of different methods when D_S is WHDLD, D_T is GID5, and shot is five.

TABLE V
RESULTS OF DIFFERENT METHODS WHEN D_S IS GID5, D_T IS WHDLD, AND SHOT IS FIVE

Method	Backbone	Category IOU						mIoU	mean F1	OA
		Building (4.04%)	Bare Soil (11.09%)	Pavement (11.52%)	Road (4.31%)	Vegetation (44.72%)	Water (24.31%)			
PANet	VGG16	6.64±2.21	14.94±4.67	14.97±3.95	6.10±2.43	25.97±4.09	29.35±5.99	16.34±2.39	26.61±2.81	31.98±4.01
TAFD	VGG16	**11.31**±3.58	**26.87**±2.39	17.96±4.37	8.75±0.91	**48.81**±4.65	47.74±5.02	26.90±1.99	39.50±2.56	49.86±3.31
PANet	ResNet50	5.43±3.33	20.96±3.16	9.07±2.90	5.34±2.30	37.53±3.05	44.98±3.34	20.55±1.63	32.63±2.48	42.82±3.38
ASGNet*	ResNet50	6.31±4.02	19.66±2.85	7.64±2.19	7.18±2.75	38.63±3.67	47.00±2.71	21.07±2.16	35.14±3.14	43.06±3.76
SSP*	ResNet50	9.15±3.52	25.88±2.64	6.04±2.27	9.85±3.19	40.80±2.65	44.93±4.12	22.78±1.51	33.66±2.25	40.22±2.98
CAPL	ResNet50	2.56±4.15	16.67±2.66	15.24±2.18	5.20±3.21	39.42±2.98	40.15±4.03	19.87±1.59	27.45±1.72	36.03±2.57
TAFD	ResNet50	9.32±2.34	20.03±1.66	**21.26**±1.03	**11.60**±1.34	47.14±2.20	**66.73**±1.90	**29.35**±1.01	**41.70**±1.60	**50.77**±2.17

The bold values denote the best performance.

building and water are the common categories and farmland, forest, and meadow are the private categories of D_T. The number below the category in tables represents the proportion. "shot" in the experiments denotes the total number of labeled images in D_T, which is different from the definition of "shot" in the existing literature. In the existing literature [5], "shot" generally refers to the number of labeled samples per category. In [11] and [14], PANet uses two backbones (VGG16 and ResNet50) in the experiments, and the other compared methods only use ResNet50. Therefore, we use both VGG16 and ResNet50 for the proposed method.

It can be seen from Tables III and IV that the proposed method is superior to the compared methods in terms of accuracy (mIoU, mean F1, and OA) when "shot" is five or ten. Concretely, the proposed method surpasses the suboptimal results by 6.32%, 6.18%, and 5.58% of mIoU, mean F1, and OA when shot is five, and by 7.30%, 5.94%, and 7.84% of mIoU, mean F1, and OA when shot is ten, respectively. In terms of each category's IoU, the proposed method achieves the best result in building, forest, water, and meadow and is similar to the best result (SSP$*$) in farmland. The results indicate that the proposed method can better alleviate the impact of interdomain discrepancy and effectively improve the generalization performance in cross-domain segmentation tasks.

From Fig. 6, we can see that the proposed method provides more satisfied segmentation results with only five labeled support images than the compared methods. For the private categories farmland, forest, and meadow in D_T, the proposed method can also provide relatively satisfactory results, which demonstrates the competitiveness of the proposed method in the cross-domain few-shot segmentation task.

Tables V and VI show the results of TAFD and the compared methods in terms of accuracy when D_S is GID5 and D_T is WHDLD; Fig. 7 shows some segmentation results of different methods. There are six categories in D_T, where building and water are the common categories and bare soil, pavement, road, and vegetation are the private categories of D_T.

It can be seen from Tables V and VI that the proposed method surpasses the compared methods in terms of accuracy (mIoU, mean F1, and OA) when shot is five or ten. Specifically, the proposed method surpasses the suboptimal method by 6.57%, 6.56%, and 7.71% of mIoU, mean F1, and OA when shot is

TABLE VI
RESULTS OF DIFFERENT METHODS WHEN D_S IS GID5, D_T IS WHDLD, AND SHOT IS TEN

Method	Backbone	Category IOU						mIoU	mean F1	OA
		Building (4.04%)	Bare Soil (11.09%)	Pavement (11.52%)	Road (4.31%)	Vegetation (44.72%)	Water (24.31%)			
PANet	VGG16	6.27±3.10	24.01±1.60	13.18±7.00	6.67±2.19	26.84±3.25	37.36±3.15	19.06±1.43	30.12±2.11	33.29±2.79
TAFD	VGG16	12.02±3.96	**27.14**±2.05	19.09±3.72	10.57±0.84	49.44±4.69	58.86±3.76	29.52±1.58	44.69±2.19	57.17±3.03
PANet	ResNet50	5.09±1.15	23.68±3.52	8.04±2.06	11.15±1.77	42.83±3.42	48.53±2.41	23.22±1.54	36.11±1.96	46.16±2.33
ASGNet*	ResNet50	8.99±1.67	26.87±3.88	7.15±2.61	10.33±1.36	42.10±2.85	51.44±3.12	24.48±1.79	37.24±1.83	48.71±2.41
SSP*	ResNet50	9.69±0.98	26.39±3.14	8.69±2.89	8.53±2.01	43.48±3.54	46.05±3.63	23.81±1.84	34.58±2.01	44.68±2.69
CAPL	ResNet50	2.38±1.75	18.50±4.30	17.48±3.17	6.20±2.29	41.76±3.66	45.57±3.98	21.98±1.89	32.06±2.12	41.76±2.88
TAFD	ResNet50	**12.56**±0.56	25.86±2.15	**20.81**±2.08	**12.72**±1.51	**56.63**±1.68	**74.69**±1.60	**33.88**±1.11	**46.31**±1.08	**61.80**±1.54

The bold values denote the best performance.

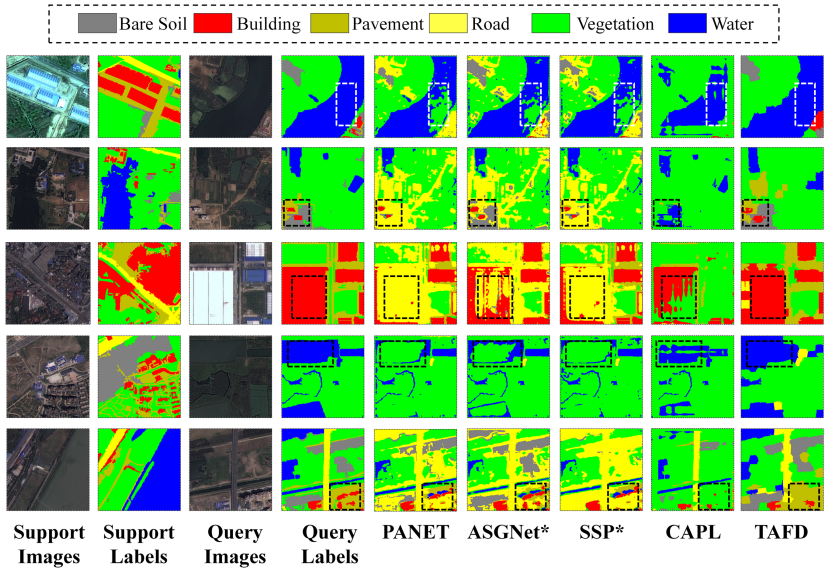


Fig. 7. Qualitative results of different methods when D_S is GID5, D_T is WHDLD, and shot is five.

five, and by 9.40%, 9.07%, and 13.09% of mIoU, mean F1, and OA when shot is ten, respectively. In terms of each category's IoU, all of the best results are achieved by TAFD. It can be concluded from Fig. 7 that TAFD enhances the generalizability of the model; therefore, TAFD provides more satisfied segmentation results in most categories than the compared methods. For the private categories of D_T (bare soil, pavement, road, and vegetation), the results are also relatively satisfactory.

From the above experiments, significant accuracy improvements are observed with the proposed method on both backbones. The main reason is that we propose the idea of TAFD, while other methods are not exploring ways to enhance the generalization capability of the model. In addition, the graph network provides the intrinsic structure information of data through the similarity matrix; more useful information further improves the segmentation accuracy. We can also see that for the backbones, ResNet50 is generally better than VGG16.

E. Ablation Experiments

To explore the effect of task augmentation, feature disentanglement, and graph network on segmentation, we conduct ablation experiments. Here, the backbones for all the methods are ResNet50. Tables VII and VIII show the ablation experiment results when D_S is WHDLD, D_T is GID5, and D_S is GID5, D_T is WHDLD, respectively, in which "M" refers to only using the metric method without label propagation, feature disentanglement, and task augmentation, which is equivalent to PANet at the superpixel level, "P" means label propagation, "T" denotes task augmentation, "D" indicates feature disentanglement, and "E" represents the traditional data augmentation method (adding noise and random clipping). We conclude the meaning of various combinations in Table IX.

From the above results, we have the following.

1) "M+P" is higher than "M" in terms of accuracy (mIoU, mean F1, and OA). "M" does not consider relationship information between regions on an image or regions across images. Besides, "M" computes the category prototypes only using the support set, ignoring the gap between the support set and the query set. "M+P" uses the local or long-range relationships between regions and uses label propagation to predict labels for some regions in the query set. In metric prediction, both the support set and the predicted part of the query set are used to calculate the

TABLE VII
RESULTS OF DIFFERENT METHODS WHEN D_S IS WHDLD, D_T IS GID5, AND SHOT IS FIVE

Method	Category IOU					mIoU	mean F1	OA
	Building (4.67%)	Farmland (43.50%)	Forest (4.55%)	Water (7.04%)	Meadow (40.22%)			
M	16.13±2.76	29.70±4.21	13.31±0.95	43.68±4.14	24.16±3.09	25.39±1.28	36.03±1.60	37.61±2.67
M+P	15.84±3.08	34.45±5.33	16.72±1.03	43.44±3.66	24.32±3.28	26.95±1.29	38.44±1.67	39.57±2.98
M+P+E	16.22±3.74	33.67±4.57	17.90±1.55	48.72±3.98	22.15±3.37	27.73±1.63	38.78±1.94	40.52±3.27
M+P+T	**20.85**±3.65	**35.21**±4.38	17.01±1.76	46.23±5.03	25.79±4.19	29.01±1.75	41.76±2.09	42.34±3.51
M+P+D	14.36±3.20	26.78±5.12	12.29±1.25	40.34±4.27	21.53±4.52	23.06±1.93	32.59±2.45	33.86±3.92
TAFD	20.08±3.62	30.88±5.83	**21.41**±1.17	**54.55**±4.04	**29.81**±4.63	**31.35**±1.72	**44.60**±2.05	**44.69**±3.57

The bold values denote the best performance.

TABLE VIII
RESULTS OF DIFFERENT METHODS WHEN D_S IS GID5, D_T IS WHDLD, AND SHOT IS FIVE

Method	Category IOU						mIoU	mean F1	OA
	Building (4.04%)	Bare Soil (11.09%)	Pavement (11.52%)	Road (4.31%)	Vegetation (44.72%)	Water (24.31%)			
M	7.01±1.94	23.10±1.59	12.00±0.98	7.41±1.19	37.05±1.73	49.79±1.21	22.72±0.79	32.63±1.31	36.69±1.91
M+P	6.07±2.12	**24.46**±1.65	12.78±0.95	8.16±1.25	39.04±2.06	58.76±1.79	24.87±0.72	35.41±1.27	42.17±1.86
M+P+E	7.41±2.04	20.54±1.76	13.65±1.27	8.77±1.74	41.30±2.55	60.46±1.82	25.36±0.95	36.95±1.48	43.66±2.02
M+P+T	7.69±2.37	19.21±1.92	19.90±0.96	7.99±1.52	43.55±2.84	66.72±1.93	27.51±1.33	38.76±1.70	47.58±2.77
M+P+D	6.22±2.26	16.55±1.79	12.84±1.12	6.89±1,48	34.98±2.63	63.21±2.03	23.45±1.24	32.95±1.69	38.05±2.65
TAFD	**9.32**±2.34	20.03±1.66	**21.26**±1.03	**11.60**±1.34	**47.14**±2.20	**66.73**±1.90	**29.35**±1.01	**41.70**±1.60	**50.77**±2.17

The bold values denote the best performance.

TABLE IX
MEANING OF DIFFERENT METHODS IN THE ABLATION EXPERIMENT

Method	Meaning of the method
M	Metric Prediction
M+P	Metric Prediction + Label Propagation
M+P+E	Metric Prediction + Label Propagation + Traditional Data Augmentation
M+P+T	Metric Prediction + Label Propagation + Task Augmentation
M+P+D	Metric Prediction + Label Propagation + Feature Disentanglement
M+P+T+D	TAFD

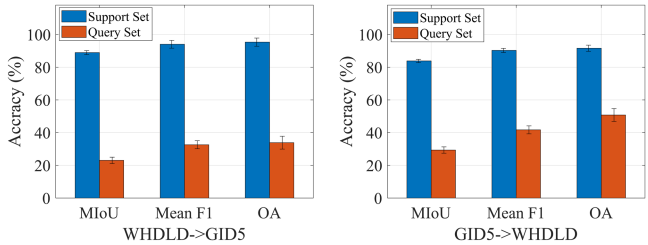


Fig. 8. Accuracy (mIoU, mean F1, and OA) of "M+P+D" on the support set and the query set.

prototypes, narrowing the gap between the support set and the query set, thus further improving the segmentation accuracy.

2) Compared with "M+P," "M+P+T" obtains higher accuracy, indicating that through augmentation of tasks, the generalization performance of the model can be improved. However, the accuracy of "M+P+D" is even lower than that of "M+P," and we do some more analysis to explain it. Fig. 8 gives the accuracy of "M+P+D" on the support set and the query set. We find that "M+P+D" achieves high accuracy on the support set but greatly degrades on the query set. This model overfitting phenomenon may be ascribed to the absence of task augmentation and too less training data from D_T.
3) The results of "M+P+E" and "M+P+T" indicate that the proposed task augmentation method is superior to the traditional data augmentation method (adding noise and random clipping); this is because the traditional data augmentation is unable to effectively expand the distribution space of data and cannot achieve good results in cross-domain segmentation tasks, while in task augmentation, training data will change in each iteration and become harder to classify during the training process. This implies that the distribution space of the training data is expanded, so that the generalization of the model can be improved. In addition, the labeled samples of D_T are also changing in each iteration, which further improves the accuracy of segmentation.
4) The proposed TAFD achieves the highest accuracy in the experiments. Task augmentation, feature disentanglement, and label propagation contribute to the significant performance improvement together. By task augmentation, the K-labeled training samples from D_T will change in each iteration, which is equivalent to increasing the

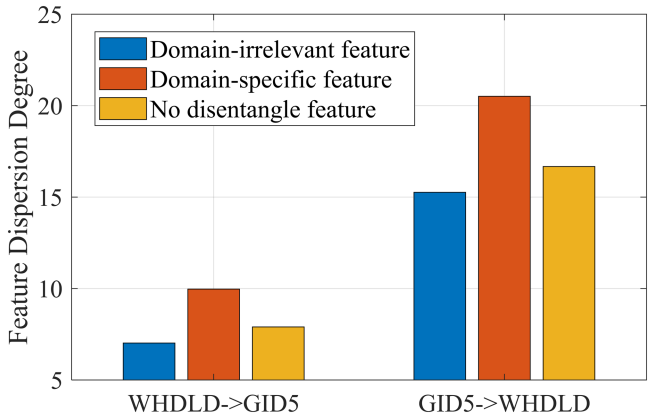


Fig. 9. Feature dispersion of three cases: no disentangled features, domain-specific features, and domain-irrelevant features.

number of training samples and can avoid the occurrence of overfitting to some extent; by feature disentangling, domain-irrelevant features are extracted, which reduces the impact of interdomain discrepancies. Moreover, label propagation leverages local and long-range relational information in prediction, which further improves the accuracy. The three measures together result in a significant increase in performance.

To explore the feature distribution before and after feature disentanglement, we calculate the feature dispersion [52] in three cases: features before disentanglement, domain-specific features, and domain-irrelevant features. Feature dispersion is defined as follows:

$$\alpha = \frac{1}{N}\sum_{i=1}^{N}\frac{S_i}{\hat{d}_i^2} \tag{24}$$

where S_i denotes the intracategory variance of category i, $\hat{d}_i$ denotes the nearest neighbor category distance, and N denotes the number of categories.

Generally, the smaller the feature dispersion, the easier the samples are to be classified. It can be seen from Fig. 9 that after feature disentanglement, domain-irrelevant features have the lowest feature dispersion, indicating that feature disentanglement is beneficial to obtain cross-domain invariance features. By feature disentanglement, the influence of domain-specific features on cross-domain data is reduced, thus improving the segmentation accuracy and generalization of the model.

F. Experiments on the Tibetan Plateau Dataset

The data in our research project have few labels; we carry on the research of few-shot segmentation to apply it in the study area's land cover classification. Thus, we also test the proposed method on the collected Tibetan Plateau dataset. In this experiment, the backbones for all the methods are ResNet50. We take the public dataset WHDLD as D_S, and the collected Tibetan Plateau dataset is D_T. There are eight categories in the Tibetan Plateau dataset: Road, Farmland, Snow, Construction Land, Building, Vegetation, Bare Soil, and Water, among which five categories (Road, Building, Vegetation, Bare Soil, and Water) are the same as in WHDLD. The experimental results are shown in Tables X and XI, and some segmentation results are given in Fig. 10.

Similar to the above results on the public dataset, TAFD produces better results on the Tibetan Plateau dataset than the comparison methods PANet, ASGNet∗, SSP∗, and CAPL in terms of accuracy mIoU, mean F1, and OA. Specifically, the proposed method surpasses the suboptimal method by 5.08%, 5.61%, and 20.72% of mIoU, mean F1, and OA when shot is five, and by 10.20%, 12.10%, 21.93% of mIoU, mean F1, and OA when shot is ten, respectively. For each category's IoU, TAFD achieves the highest IoU on most categories and closes to the best results on the remaining categories. In addition, the categories with larger sample proportion will generate more training data in task augmentation; therefore, TAFD is likely to achieve significantly higher accuracy than the compared methods on such categories (such as bare soil), while the performance improvement for the categories with a small sample proportion is relatively limited.

Based on the results depicted in Fig. 10, our proposed method demonstrates superior segmentation outcomes for bare soil, water, construction land, vegetation, and snow in comparison to alternative approaches. Furthermore, it yields relatively satisfactory segmentation results for other categories. This experiment effectively illustrates that our proposed method outperforms existing techniques, achieving the most accurate segmentation results for the Tibetan Plateau dataset.

V. Discussion

We propose a cross-domain few-shot segmentation method for RSIs. In this section, we discuss the advantages and limitations of the proposed method and provide prospects for future work. Compared to previous studies, the advantages of TAFD mainly manifest in the following three aspects.

1) The existing few-shot segmentation methods overlook the domain discrepancies between datasets. However, in RSIs, domain discrepancies often arise due to variations in sensors, shooting angles, capture times, etc., and hinder the model's performance. TAFD mitigates the negative impact of domain discrepancies from both the data and feature levels. At the data level, task augmentation is employed to generate more diverse and challenging training samples to enhance model generalization. At the feature level, features are disentangled to extract domain-invariant features, thereby enhancing the transferability of the model.
2) Prevalent few-shot segmentation and cross-domain few-shot segmentation methods mainly rely on CNNs for feature extraction; they often do not capture relationship information among regions in different images or within the same image. Such information is implicit in the data and helps to improve segmentation performance. TAFD models the relationship among image regions by graph and provides more useful information for segmentation by mining such implicit information.
3) Limited by the setting of current few-shot segmentation methods, they primarily focus on binary segmentation

TABLE X
RESULTS OF DIFFERENT METHODS WHEN D_S IS WHDLD, D_T IS TIBETAN PLATEAU, AND SHOT IS FIVE

Method	Category IOU								mIoU	mean F1	OA
	Road (4.33%)	Farmland (3.62%)	Snow (7.43%)	Construction Land(11.98%)	Building (5.47%)	Vegetation (2.54%)	Bare Soil (62.57%)	Water (2.05%)			
PANet	5.86±3.55	6.36±1.99	35.34±5.39	20.55±5.02	12.23±5.53	3.32±1.96	34.05±5.70	6.97±2.06	15.58±1.58	24.79±1.96	32.61±2.85
ASGNet*	**8.09**±4.21	**7.64**±2.73	37.86±4.38	19.63±5.66	6.80±4.02	6.77±2.17	33.19±4.14	**9.78**±3.11	16.22±1.63	25.93±2.09	31.95±3.03
SSP*	7.02±4.17	7.25±2.69	24.38±5.41	19.66±4.92	8.52±5.72	4.01±1.89	24.02±5.72	8.14±3.54	12.86±1.68	22.35±2.11	23.73±3.14
CAPL	6.10±4.57	6.27±3.01	36.52±4.69	20.85±6.25	12.98±4.12	6,95±2.75	29.71±5.21	5.41±3.77	14.73±1.49	22.79±1.85	26.58±2.64
TAFD	6.25±1.30	6.96±1.92	**46.74**±4.39	**21.14**±3.22	**17.73**±2.46	**7.67**±0.30	**55.39**±4.88	8.55±2.28	**21.30**±1.13	**31.54**±1.83	**53.33**±2.81

The bold values denote the best performance.

TABLE XI
RESULTS OF DIFFERENT METHODS WHEN D_S IS WHDLD, D_T IS TIBETAN PLATEAU, AND SHOT IS TEN

Method	Category IOU								mIoU	mean F1	OA
	Road (4.33%)	Farmland (3.62%)	Snow (7.43%)	Construction Land (11.98%)	Building (5.47%)	Vegetation (2.54%)	Bare Soil (62.57%)	Water (2.05%)			
PANet	6.14±2.29	7.17±2.85	38.84±4.62	23.87±3.09	12.43±3.95	6.71±1.12	41.14±2.07	11.49±4.66	18.47±1.34	27.32±1.29	39.83±2.01
ASGNet*	9.33±1.33	9.08±1.11	42.33±4.83	27.51±2.44	8.02±4.71	6.95±1.87	38.31±1.96	14.42±3.89	19.49±1.64	29.21±2.54	36.42±3.09
SSP*	7.65±1.41	**9.98**±1.42	28.52±5.72	**28.71**±1.61	10.51±3.66	4.27±1.92	30.41±2.78	10.08±5.01	16.26±1.97	26.40±2.71	30.89±3.89
CAPL	6.79±2.32	7.42±2.96	39.11±5.18	22.53±1.26	16.81±3.11	7.10±1.29	42.64±3.02	11.52±4.12	19.24±1.86	28.77±2.51	40.16±3.03
TAFD	**17.90**±0.97	7.95±0.70	**53.29**±4.65	26.95±0.59	**27.38**±4.40	**7.16**±0.37	**67.06**±2.22	**29.90**±4.25	**29.69**±1.57	**41.31**±2.15	**62.09**±2.93

The bold values denote the best performance.

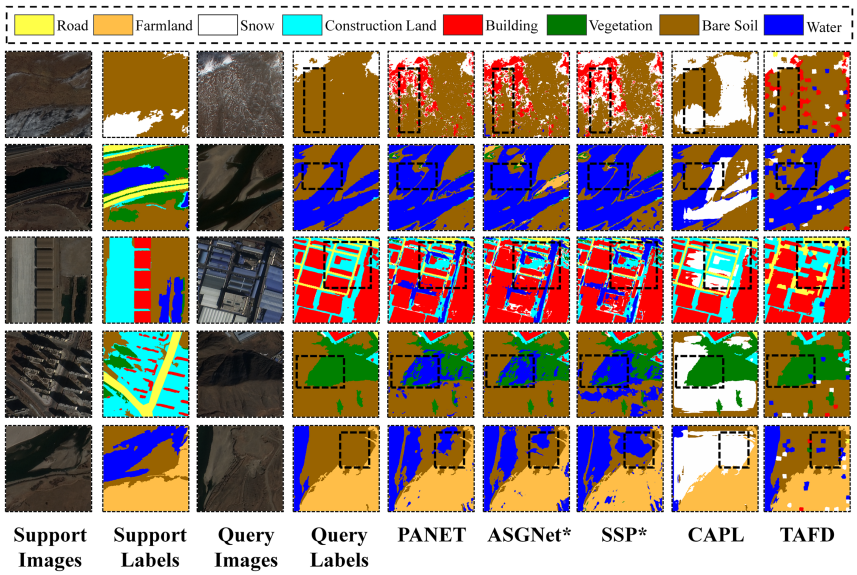


Fig. 10. Qualitative results of different methods when D_S is WHDLD, D_T is Tibetan Plateau, and shot is five.

tasks (i.e., foreground and background). However, RS applications often involve multicategory segmentation; implementing by binary segmentation methods is relatively cumbersome. In contrast, TAFD predicts all categories' labels directly by the label propagation algorithm on graphs, providing convenience for the practical RS applications.

The proposed method improves the transferability and generalization of the model and achieves better segmentation accuracy compared with the comparison methods. The average results of three cross-domain few-shot segmentation experiments show that the proposed method outperforms the suboptimal method by 5.99%, 6.12%, and 11.34% in terms of mIoU, mean F1, and OA when the labeled samples are five, and by 8.97%, 9.04%,

and 14.29% in mIoU, mean F1, and OA for ten labeled samples, respectively.

However, the segmentation accuracy is low for categories with fewer samples. Furthermore, only domain–independent features are used for prediction, but those domain-related features that may contain information conducive to the segmentation of private categories are not used. In future studies, we will attempt to solve the problem of sample imbalance and incorporate domain-related features in the segmentation of private categories to further improve the segmentation accuracy.

VI. Conclusion

In this article, we propose a novel method for the task of cross-domain few-shot segmentation in RS land cover. To improve the segmentation accuracy of the model on the target dataset, TAFD strategies are proposed. At the data level, task augmentation is proposed to generate more diverse and challenging training data in each iteration, expanding the distribution of training data, thus further enhancing the generalization ability of the model; at the feature level, feature disentanglement is proposed to extract domain-irrelevant features for segmentation, which reduces the negative impact of interdomain discrepancy on target domain segmentation brought by domain-specific features. Moreover, a graph is utilized to model and represent the intrinsic structural features of data, which provides more information for segmentation, and label propagation provides another convenient approach for predicting nodes' labels on the graph. Experimental results demonstrate that the proposed method significantly improves the few-shot segmentation accuracy (mIoU, mean F1, and OA) in cross-domain land cover tasks compared with the existing methods. In the future, we will focus on the problem brought by sample imbalance.

References

[1] J. Hou, Z. Guo, Y. Feng, Y. Wu, and W. Diao, "SPANet: Spatial adaptive convolution based content-aware network for aerial image semantic segmentation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 2192–2204, 2023, doi: 10.1109/JSTARS.2023.3244207.

[2] R. Xu, C. Wang, J. Zhang, S. Xu, W. Meng, and X. Zhang, "RSSFormer: Foreground saliency enhancement for remote sensing land cover segmentation," *IEEE Trans. Image Process*, vol. 32, pp. 1052–1064, 2023, doi: 10.1109/TIP.2023.3238648.

[3] C. Shang, S. Jiang, F. Ling, X. Li, Y. Zhou, and Y. Du, "Spectral-spatial generative adversarial network for super-resolution land cover mapping with multispectral remotely sensed imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 522–537, 2023, doi: 10.1109/JSTARS.2022.3228741.

[4] B. Wang, Z. Wang, X. Sun, Q. He, H. Wang, and K. Fu, "TDNet: A novel transductive learning framework with conditional metric embedding for few-shot remote sensing image scene classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 4591–4606, 2023, doi: 10.1109/JSTARS.2023.3263149.

[5] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, "One-shot learning for semantic segmentation," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 1–13.

[6] K. Wang, J. Liew, Y. Zou, D. Zhou, and J. Feng, "PANet: Few-shot image semantic segmentation with prototype alignment," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9197–9206.

[7] K. Han, Y. Wang, J. Guo, Y. Tang, and E. Wu, "Vision GNN: An image is worth graph of nodes," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, no. 35, pp. 8291–8303.

[8] H. Wang and H. Deng, "Cross-domain few-shot classification via adversarial task augmentation," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 1–7.

[9] Y. Fu, Y. Fu, J. Chen, and Y.-G. Jiang, "Generalized meta-FDMixup: Cross-domain few-shot learning guided by labeled target data," *IEEE Trans. Image Process*, vol. 31, pp. 7078–7090, 2022, doi: 10.1109/TIP.2022.3219237.

[10] M. Boudiaf, H. Kervadec, Z. Masud, P. Piantanida, I. Ben Ayed, and J. Dolz, "Few-shot segmentation without meta-learning: A good transductive inference is all you need," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 13979–13988.

[11] Y. Liu, N. Liu, X. Yao, and J. Han, "Intermediate prototype mining transformer for few-shot semantic segmentation," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 38020–38031.

[12] B. Wang, Z. Wang, X. Sun, H. Wang, and K. Fu, "DMML-Net: Deep meta metric learning for few-shot geographic object segmentation in remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5611118, doi: 10.1109/TGRS.2021.3116672.

[13] Z. Tian, X. Lai, L. Jiang, M. Shu, and J. Jia, "Generalized few-shot semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11563–11572.

[14] Q. Fan, W. Pei, W. Tai, and K. Tang, "Self-support few-shot semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 701–719.

[15] Y. Liu, X. Zhang, S. Zhang, and X. He, "Part-aware prototype network for few-shot semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 142–158.

[16] W. Zhuo, L. Yang, L. Qi, Y. Shi, and Y. Gao, "Mining latent classes for few-shot segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 8721–8730.

[17] X. Zhang, Y. Wei, Y. Yang, and T. S. Huang, "SG-One: Similarity guidance network for one-shot semantic segmentation," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3855–3865, Sep. 2020, doi: 10.1109/TCYB.2020.2992433.

[18] K. Nguyen and S. Todorovic, "Feature weighting and boosting for few-shot segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 622–631.

[19] B. Yang, C. Liu, B. Li, J. Jiao, and Q. Ye, "Prototype mixture models for few-shot semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 23–28.

[20] Z. Tian, H. Zhao, M. Shu, Z. Yang, R. Li, and J. Jia, "Prior guided feature enrichment network for few-shot segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 2, pp. 1050–1065, Feb. 2022, doi: 10.1109/TPAMI.2020.3013717.

[21] G. Li, V. Jampani, L. Sevilla-Lara, D. Sun, J. Kim, and J. Kim, "Adaptive prototype learning and allocation for few-shot segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8334–8343.

[22] X. Yao, Q. Cao, X. Feng, G. Cheng, and J. Han, "Scale-aware detailed matching for few-shot aerial image semantic segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5611711, doi: 10.1109/TGRS.2021.3119852.

[23] Q. Fu, Y. Xie, Y. Fu, J. Chen, and Y. Jiang, "Wave-SAN: Wavelet based style augmentation network for cross-domain few-shot learning," 2022, *arXiv:2203.07656*.

[24] H. Tseng, H. Lee, J. Huang, and M. Yang, "Cross-domain few-shot classification via learned feature-wise transformation," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–18.

[25] S. Lei, X. Zhang, J. He, F. Chen, B. Du, and C. LU, "Cross-domain few-shot semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 73–90.

[26] W. Wang, L. Duan, Y. Wang, Q. En, J. Fan, and Z. Zhang, "Remember the difference: Cross-domain few-shot semantic segmentation via meta-memory transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 7065–7074.

[27] Y. Lu, X. Wu, Z. Wu, and S. Wang, "Cross-domain few-shot segmentation with transductive fine-tuning," 2022, *arXiv:2211.14745*.

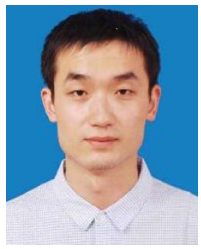
[28] S. Wan, C. Gong, P. Zhong, S. Pan, G. Li, and J. Yang, "Hyperspectral image classification with context-aware dynamic graph convolutional network," *IEEE Trans. Geosci. Remote Sens.*, no. 59, vol. 1, pp. 597–612, Jan. 2020, doi: 10.1109/TGRS.2020.2994205.

[29] J. Chen, L. Jiao, X. Liu, L. Li, F. Liu, and S. Yang, "Automatic graph learning convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5520716, doi: 10.1109/TGRS.2021.3135084.

[30] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–14.

[31] C. Zhang, G. Lin, F. Liu, J. Guo, Q. Wu, and R. Yao, "Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9587–9595, doi: 10.1109/ICCV.2019.00968.

[32] X. Li, Y. Yang, Q. Zhao, T. Shen, Z. Lin, and H. Liu, "Spatial pyramid based graph reasoning for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8950–8959.
[33] S. Yang, J. Hou, Y. Jia, S. Mei, and Q. Du, "Superpixel-guided discriminative low-rank representation of hyperspectral images for classification," *IEEE Trans. Image Process.*, vol. 30, pp. 8823–8835, 2021, doi: 10.1109/TIP.2021.3120675.
[34] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 5966–5978, Jul. 2021, doi: 10.1109/TGRS.2020.3015157.
[35] Y. Chen, W. Chen, Y. Chen, B. Tsai, Y. Wang, and M. Sun, "No more discrimination: Cross city adaptation of road scene segmenters," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2017, pp. 2011–2020.
[36] J. Chen, F. He, Y. Zhang, G. Sun, and M. Deng, "SPMF-Net: Weakly supervised building segmentation by combining superpixel pooling and multi-scale feature fusion," *Remote Sens.*, vol. 12, no. 6, 2020, Art. no. 1049, doi: 10.3390/rs12061049.
[37] Z. Wu et al., "DCAN: Dual channel-wise alignment networks for unsupervised scene adaptation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 518–534.
[38] J. Chen, J. Zhu, Y. Guo, G. Sun, Y. Zhang, and M. Deng, "Unsupervised domain adaptation for semantic segmentation of high-resolution remote sensing imagery driven by category-certainty attention," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5616915, doi: 10.1109/TGRS.2021.3140108.
[39] L. Zhong, Z. Fang, F. Liu, B. Yuan, G. Zhang, and J. Lu, "Bridging the theoretical bound and deep algorithms for open set domain adaptation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 8, pp. 3859–3873, Aug. 2023, doi: 10.1109/TNNLS.2021.3119965.
[40] A. Tavera et al., "Pixel-by-pixel cross-domain alignment for few-shot semantic segmentation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2022, pp. 1626–1635.
[41] J. Wang et al., "Generalizing to unseen domains: A survey on domain generalization," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 8, pp. 8052–8072, Aug. 2023, doi: 10.1109/TKDE.2022.3178128.
[42] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, "Image data augmentation for deep learning: A survey," 2022, *arXiv:2204.08610*.
[43] S. Li, F. Liu, L. Jiao, P. Chen, X. Liu, and L. Li, "MFNet: A novel GNN-based multi-level feature network with superpixel priors," *IEEE Trans. Image Process.*, vol. 31, pp. 7306–7321, 2022, doi: 10.1109/TIP.2022.3220057.
[44] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, and S. Hwang, "Learning to propagate labels: Transductive propagation network for few-shot learning," in *Proc. 25th Int. Conf. Pattern Recognit.*, 2019, pp. 1–11.
[45] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2004, pp. 321–328.
[46] A. Iscen1, G. Tolias, Y. Avrithis, and O. Chum, "Label propagation for deep semi-supervised learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5070–5079.
[47] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4077–4087.
[48] A. Davari et al., "On mathews correlation coefficient and improved distance map loss for automatic glacier calving front segmentation in SAR imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5213212, doi: 10.1109/TGRS.2021.3115883.
[49] Z. Shao, W. Zhou, X. Deng, M. Zhang, and Q. Cheng, "Multilabel remote sensing image retrieval based on fully convolutional network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 318–328, 2020, doi: 10.1109/JSTARS.2019.2961634.
[50] X. Tong, G. Xia, Q. Lu, H. Shen, and L. Zhang, "Learning transferable deep models for land-use classification with high-resolution remote sensing images," 2018, *arXiv:1807.05713*.
[51] Q. He, X. Sun, W. Diao, Z. Yan, F. Yao, and K. Fu, "Multimodal remote sensing image segmentation with intuition-inspired hypergraph modeling," *IEEE Trans. Image Process.*, vol. 32, pp. 1474–1487, 2023, doi: 10.1109/TIP.2023.3245324.
[52] J. Chen and X. Wang, "Open set few-shot remote sensing scene classification based on a multiorder graph convolutional network and domain adaptation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4709517, doi: 10.1109/TGRS.2022.3222449.



Jiehu Chen received the M.S. degree in communication and information systems from Xidian University, Xi'an, China, in 2016. He is currently working toward the Ph.D. degree in computer application technology with Shaanxi Normal University, Xi'an.

His research interests include computer vision and remote sensing image processing, especially on few-shot learning.

Xili Wang received the B.S. degree in computer science from Tianjin University, Tianjin, China, in 1991, and the M.S. and Ph.D. degrees in electronic information engineering from Xidian University, Xi'an, China, in 1994 and 2004, respectively.

She is currently a Professor with Shaanxi Normal University, Xi'an. She has authored more than 60 peer-reviewed articles in international journals from multiple domains, such as *Remote Sensing* and *Computer Vision*. Her research interests include image processing, deep learning, and their applications in remote sensing.

Dr. Wang is a Senior Member of the China Computer Society.

Ling Hong received the B.S. and Ph.D. degrees in electronic engineering from Xidian University, Xi'an, China, in 2008 and 2015, respectively.

She is currently an Associate Researcher with the School of Computer Science, Shaanxi Normal University, Xi'an. Her research interests include radar signal processing and intelligent information processing.

Ming Liu was born in Xi'an, China, in 1987. She received the B.S. degree in electronic engineering and the Ph.D. degree in pattern recognition and intelligence system from Xidian University, Xi'an, in 2009 and 2015, respectively.

She is currently an Associate Professor with the School of Computer Science, Shaanxi Normal University, Xi'an. Her research interests include synthetic aperture radar (SAR) target recognition and SAR image processing.