electronics

MDPI

Article

Cross-Scene Hyperspectral Image Classification Based on Graph Alignment and Distribution Alignment

Haisong Chen [1], Shanshan Ding [2] and Aili Wang [2,*]

[1] School of Undergraduate Education, Shenzhen Polytechnic University, Shenzhen 518115, China; hschen@szpt.edu.cn
[2] Heilongjiang Province Key Laboratory of Laser Spectroscopy Technology and Application, Harbin University of Science and Technology, Harbin 150080, China; 2120600047@stu.hrbust.edu.cn
* Correspondence: aili925@hrbust.edu.cn

Abstract: A domain alignment-based hyperspectral image (HSI) classification method was designed to address the heterogeneity in resolution and band between the source domain and target domain datasets of cross-scene hyperspectral images, as well as the resulting reduction in common features. Firstly, after preliminary feature extraction, perform two domain alignment operations: image alignment and distribution alignment. Image alignment aims to align hyperspectral images of different bands or time points, ensuring that they are within the same spatial reference framework. Distribution alignment adjusts the distribution of features of samples of different categories in the feature space to reduce the distribution differences of the same type of features between two domains. Secondly, adjust the consistency of the two alignment methods to ensure that the features obtained through different alignment methods exhibit consistency in the feature space, thereby improving the comparability and reliability of the features. In addition, this method considers multiple losses in the model from different perspectives and makes comprehensive adjustments through a unified optimization process to more comprehensively capture and utilize the correlation information between data. Experimental results on Houston 2013 and Houston 2018 datasets can improve the hyperspectral prediction performance between datasets with different resolutions and bands, effectively solving the problems of high cost and limited training samples in HSI labeling and significantly improving cross-scene HSI classification performance.

Keywords: hyperspectral image; image classification; domain alignment; cross-scene

Citation: Chen, H.; Ding, S.; Wang, A. Cross-Scene Hyperspectral Image Classification Based on Graph Alignment and Distribution Alignment. *Electronics* **2024**, *13*, 1731. https://doi.org/10.3390/electronics13091731

Academic Editor: Stefanos Kollias

Received: 3 February 2024
Revised: 13 March 2024
Accepted: 21 March 2024
Published: 1 May 2024

1. Introduction

Hyperspectral remote sensing technology integrates traditional spectral detection and photographic imaging techniques, which can simultaneously obtain multiple types of information (radiation, spectral, spatial information, etc.) and integrate them into a graph-integrated data cube [1]. By capturing spatial features at multiple spectral levels, hyperspectral remote sensing technology can provide richer spectral information, further enhancing the recognition and classification capabilities of features, and has important applications in military [2], agricultural and forestry monitoring [3], vegetation research [4], urban remote sensing [5], food quality control [6], chip detection [7], and tumor diagnosis [8].

Due to differences in the composition of ground objects and the angle of sunlight exposure, there will be significant differences in the spectral curves formed by different ground objects. The principle of hyperspectral remote sensing image classification is to determine the substance it represents based on this difference and then set a corresponding land feature category label for each pixel [9].

In recent years, neural network classification algorithms have been widely applied in the field of hyperspectral remote sensing images, such as 2D convolution [10], 3D convolution [11], and graph convolution [12]. Although significant breakthroughs have

been made, improving the generalization performance of the model requires a large number of labeled samples. However, the imaging complexity in hyperspectral remote sensing images (such as "same spectral foreign objects" and "same object but different spectra") makes this task challenging. Manual labeling of samples is difficult and usually very expensive and time-consuming [13], resulting in limited labeled samples. In practice, it is common to encounter situations where the classification accuracy cannot reach the predetermined goal due to insufficient labeled samples.

Therefore, when the number of instances in the training set is small, improving the generalization performance of hyperspectral image classification models has become an urgent problem to be solved [14]. If we can use images with a large number of labeled samples (source images) to classify new images with similar distributions but limited labeled samples (target images), it will save a lot of resources. This transfer idea is called cross-scene classification. However, due to the fact that different datasets are collected at different locations and time periods, as well as the influence of non-human factors such as climate, there may be significant differences in spectral characteristics between the two datasets, namely the spectral shift phenomenon [15]. Therefore, it is not possible to directly use two datasets for mutual prediction. How to eliminate spectral differences between images in order to reuse existing labeled samples in cross-scene hyperspectral image classification is a challenging problem, yet it is worth researching.

In cross-scene hyperspectral image classification tasks, domain adaptation (DA) is widely used to reduce spectral shifts between cross-scene images and improve classification accuracy [16]. At present, the domain adaptation techniques used for cross-scene hyperspectral image prediction can be summarized into two categories: shallow domain adaptation and deep domain adaptation. The shallow domain adaptation method mainly utilizes instance-based and feature representation-based methods to adjust the distribution of the source and target domains [17]. The deep domain adaptation method utilizes deep neural networks to achieve cross-domain learning.

Deep domain adaptation uses deep neural networks as feature learning tools. Due to the stronger transfer ability of deep features learned by deep neural networks, they can more effectively improve the classification progress and performance of cross-scene hyperspectral models compared to shallow domain adaptation. In deep domain adaptation methods, spectral shifts between the source and target domains are often reduced from the perspectives of distribution differences and adversarial approaches.

The deep domain adaptation method based on distribution differences will add adaptive layers to the deep neural network for specific tasks to match the edge distribution or conditional distribution between domains. Zhu proposed the method of the multi-representation adaptation network (MRAN) to perform cross domain classification tasks through multi-representation alignment [18]. In addition, they proposed a method called the deep subdomain adaptation network (DSAN), which utilized local maximum mean differences to align the distribution of corresponding subdomains in different domains in order to achieve learning of transfer networks [19]. Based on maximum mean discrepancy (MMD), Li proposed a two-stage deep data mining method. In the first stage, MMD is used to minimize the inter-domain distribution distance to learn deep embedding space. In the second stage, a spatial–spectral twin network is used to minimize the distance between instances of the same category of two domains based on pairwise loss and to maximize the distance between instances of different categories of two domains, reducing data drift while learning more discriminative deep embedding spaces as much as possible [20]. Zhu proposed a three-stage network—an attention-based multi-scale residual adaptive network for cross-scene classification, which adds an attention module before the multi-scale robust feature extraction, and conditional distribution alignment adaptive modules [21]. Considering the superiority of the attention mechanism, Liang proposed a few-shot learning method with three modules and multi-source fusion based on the attention mechanism. Compared with the previous methods, although both have three modules and apply attention mechanisms, the original intention of this method is to be used for multi-source small-scale HSI

classification, which can transfer the learned classification ability from multiple-source data to target data [22]. However, due to the more complex features that need to be aligned for multi-source domain adaptation, single-source deep neural network domain adaptation is still the main focus of the current research.

In further research, graph neural network (GNN) is gradually being applied to HIS classification. Wang et al. proposed a deep domain adaptive method for multi-temporal hyperspectral remote sensing images based on GNN, constructing graphs for both source and target data, and then using graphics in each hidden layer to obtain features [23]. Due to the fact that GNN operates on graph structures and utilizes the relationships between data samples, it can aggregate features and propagate information through graph nodes. Therefore, the extracted features have better smoothness in each spectral neighborhood, which is beneficial for classification. However, graph convolutional neural networks are unable to effectively handle inductive semi-supervised learning problems in the early stages. To address this issue, Hamilton et al. proposed the GraphSAGE algorithm, which updated the node representation by randomly sampling neighboring nodes and aggregating the sampled neighboring nodes. On the other hand, in the graph convolution operation of algorithms such as GraphSAGE, the contribution of all neighboring nodes to the central node is considered the same or predetermined, and this method may not accurately capture the connection relationship between nodes at the spectral level [24]. Therefore, this paper will introduce attention to distinguish the importance of nodes in different layers in GNN and enhance the weight of important nodes at the spectral level.

The above content introduces the current progress in the field of hyperspectral image classification and related techniques. Compared with cross-domain classification methods such as MRAN and DSAN previously proposed by Zhu et al. [18], the deep domain adaptation method proposed in this paper innovatively adopts multiple kernel function strategies to achieve more accurate and comprehensive feature matching in response to the complex feature alignment requirements of hyperspectral images, breaking through the possible limitations of traditional single kernel function in dealing with such problems. On the other hand, in the application of graph neural networks, this paper not only draws on the advantages of using GNN for deep domain adaptation, such as Wang et al. [23], but also improves the shortcomings of early GNN in dealing with inductive semi-supervised learning problems. In addition, this paper draws on and further develops an attention-mechanism-based approach to make full use of the attention mechanism to address the complex feature alignment challenges in hyperspectral images. Specifically, this paper integrates an attention mechanism into GNN, which can be weighted at the spectral level according to node importance, so as to effectively distinguish the influence of nodes in different layers, improve feature extraction and smoothness, and help improve classification performance. This series of improvements and highly specific application technology demonstrate for the proposed method a unique novelty in the field of hyperspectral image classification.

Our proposed method aims to solve the problem of different resolution and band between source domain data and target domain data in cross-scene hyperspectral image classification. The contribution of our proposed method is as follows:

1. In response to the problem of different attributes between source domain images and target domain images in cross-scene hyperspectral image prediction and the difficulty of prediction, this paper proposed a domain adaptation method that focuses more on spatial features, which can effectively cope with knowledge transfer under limited conditions of similar spectral information and further expand the applicability of cross-scene HSI prediction.
2. In distribution alignment, this paper employed three kernel functions to extract linear and high-dimensional nonlinear features from hyperspectral images. By collaborating with three different kernel functions, we can extract features while avoiding the negative impact of outlier noise points, thereby improving the stability of the model. In graph alignment, to measure the similarity of the aligned graph structure, Sinkhorn

loss is used in the GraphSAGE step. By integrating the three Sinkhorn loss, the sampling aggregation ability of GraphSAGE is continuously improved during the back-propagation process.

3. Attention mechanism is a method of processing multidimensional data, which can help models focus on important parts of input data. The attention mechanisms were used to adaptively assign different importance weights to nodes in different spectral layers of GraphSAGE for accurately capturing the intrinsic connectivity between nodes.

2. Methods

The two HSI datasets used in this paper are almost completely different in terms of effective spectral bands and spatial resolution. There is a more severe spectral shift phenomenon between such datasets, with greater differences in feature distribution and less similar information. Therefore, this paper focuses on mining the spatial feature correlation between two HSI datasets, with spectral information as an auxiliary, and designing a domain-aligned HSI classification method based on graph alignment and distribution alignment (GADA). The overall model architecture is shown in Figure 1.

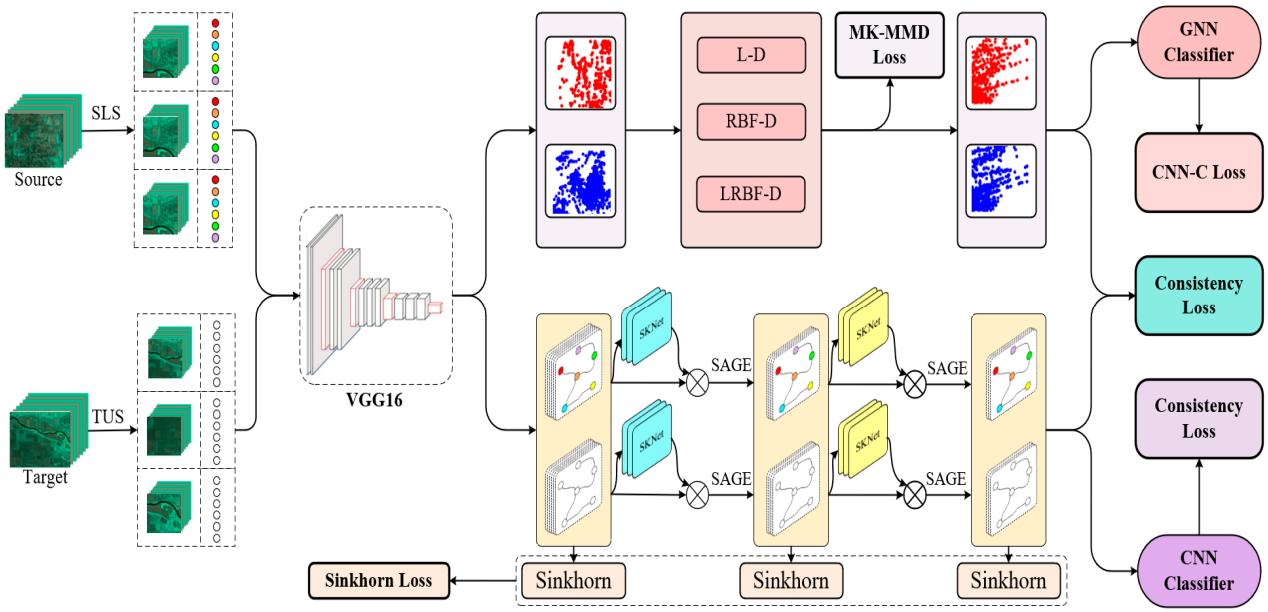


Figure 1. GADA model architecture for cross-scene HSI classification.

Firstly, input source labeled samples (SLS) and target unlabeled samples (TUS) into feature extractor VGG16 to extract meaningful spatial and spectral features. Secondly, in domain alignment, two methods are used to transfer features: graph alignment and distribution alignment. In distribution alignment, multi-kernel MMD is used for feature alignment, including Linear Discrepancy (L-D), Radial Basis Function Discrepancy (RBF-D), and Laplacian Radial Basis Function Discrepancy (LRBF-D). In graph alignment, the attention mechanism SKNet is used for feature importance selection, and SAGE and Sinkhorn are used for optimal graph transmission. Then, GNN classifiers and CNN classifiers are trained using distribution alignment and graph alignment data, respectively, to further improve domain alignment ability by optimizing the classifier's loss. Finally, consistency constraints are applied to the two domain alignment methods to ensure that the features obtained through different alignment methods have a consistent representation in the feature space, improving the comparability and reliability of the features.

2.1. GraphSAGE Combined with Graph Attention Mechanism

In order to effectively align the graph structure information, this paper uses two layers of GraphSAGE based on the mean aggregation function in the model, which is a highly flexible deep learning method. GraphSAGE considers both node feature information and structural information to generate a mapping for graph embedding. Unlike previous methods, GraphSAGE preserves the generation of embedding mapping strategies rather than just the mapped results, thus having stronger scalability. More specifically, GraphSAGE aims to learn the representation method of nodes, that is, how to capture the relationships between nodes by selecting samples and aggregating features from their surrounding neighbors. During the testing or inference phase, use the trained model and aggregation function to perform spatial mapping on new samples. The detailed process can be represented as the three steps shown in Figure 2.

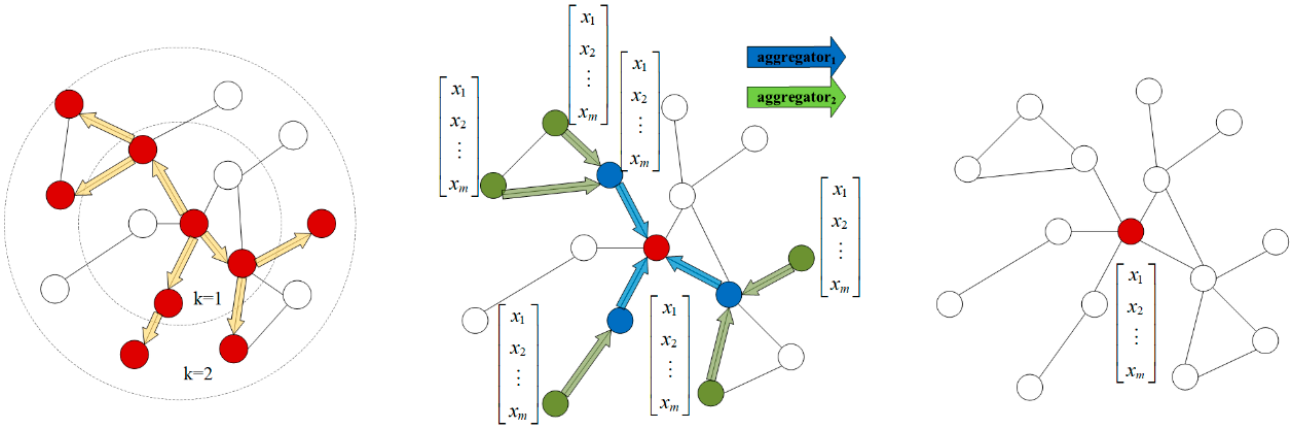


Figure 2. Visual illustration of GraphSAGE samples and aggregation methods. (**a**) Sampling neighbor nodes; (**b**) aggregating feature information from neighbors; (**c**) predicting graphic context and labels.

1. Sampling neighbor nodes

For each node $v \in V$, GraphSAGE selects a certain number of nodes from its neighboring nodes through random walks or fixed-length neighbor sampling, forming a sampling set $N(v)$, where V is the set of all nodes. The purpose of sampling is to control computational complexity and ensure effective representation learning in large-scale graph data.

2. Aggregating Neighbor Features

For each node $v \in V$, GraphSAGE combines its own features $\mathbf{x}_v$ with the features of its neighboring nodes $\mathbf{x}_u, \forall u \in N(v)$ through aggregation operations. Its purpose is to integrate the information of neighboring nodes into the target node, better capturing the relationships between nodes. Aggregation operation can be expressed as:

$$h_v = \mathrm{AGG}(\mathbf{z}_u, \forall u \in N(v)) \tag{1}$$

$\mathbf{z}_u$ is the representation vector of the node, and $\mathrm{AGG}(\cdot)$ is the aggregation function.

3. Updating node representation

GraphSAGE maps the aggregated feature $\mathbf{h}_v$ to a low-dimensional representation space through a learning function $f(\cdot)$ to obtain the node representation $\mathbf{z}_v = f(\mathbf{h}_v)$.

By iterating the above steps, GraphSAGE can learn the feature representation of nodes in low-dimensional space, where the node representation vector $\mathbf{z}_v$ can capture the topological structure and similarity information between nodes.

A topological node in a graph structure will use the information of its surrounding nodes to generate features to represent itself. However, nodes in the topology do not have equal importance, and their representation importance in different spectral layers also

varies. In graph optimization problems, attention mechanisms can be used to represent the importance of nodes, thereby better capturing information about the network structure. The attention module structure of SKNet [25] used in this paper is shown in Table 1.

Table 1. SKNet structure diagram.

Size of Input	Attention Module
112×112	7×7, 64, stride 2
56×56	3×3 max pool, stride 2
56×56	$\begin{bmatrix} 1 \times 1, 128 \\ \mathrm{SK}[M = 3, G = 32, r = 16], 128 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
28×28	$\begin{bmatrix} 1 \times 1, 256 \\ \mathrm{SK}[M = 3, G = 32, r = 16], 256 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
14×14	$\begin{bmatrix} 1 \times 1, 512 \\ \mathrm{SK}[M = 3, G = 32, r = 16], 512 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$
7×7	$\begin{bmatrix} 1 \times 1, 1024 \\ \mathrm{SK}[M = 3, G = 32, r = 16], 1024 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
1×1	7×7 global average pool, 1000-d fc, softmax

SKNet is multiple sets of repetitive bottleneck blocks, known as "SK (Selective Kernel) units". Each SK unit consists of a set of 1×1 convolutions, SK convolutions, and 1×1 convolutions. In the SK unit, there are three main hyperparameters: the number of paths M determines the number of different kernel selections to aggregate, the number of groups G controls the cardinality of each path, and the reduction ratio r controls the number of parameters in the fusion operator. The SK structure of convolution is shown in Figure 3.

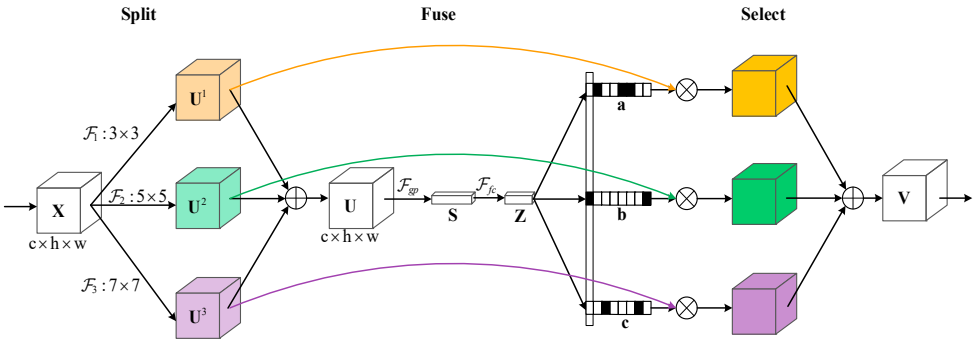


Figure 3. The structure of SK convolution.

SK convolution is achieved through three operators: split, fuse, and select as follows.

1. Split

For the input feature map $\mathbf{X} \in \mathbb{R}^{h \times w \times c}$, two transformations $\mathcal{F}_i : \mathbf{X} \to \mathbf{U}^i \in \mathbb{R}^{H \times W \times C}$, $i \in \{1, 2, 3\}$ will be performed in parallel, with convolution kernel sizes of 3, 5, and 7, respectively. $\mathcal{F}_i$ is composed of grouping/deep convolution, batch regularization, and ReLU activation functions in order.

2. Fuse

The goal of SK convolution is to enable neurons to automatically adjust the size of their receiving domain based on the activated content. Therefore, "gates" are used to control the flow of information in different branches, and this information with different scales is integrated into the next layer of neurons. To achieve this goal, the results obtained from various branches in the network can be merged by summing them element-by-element:

$$\mathbf{U} = \sum_{i=1}^{3} \mathbf{U}^i \tag{2}$$

Then, the global average pooling is used to aggregate the overall information and fuse the relevant features of the channels, denoted as $\mathbf{s} \in \mathbb{R}^C$.

$$\mathbf{s}_c = \mathcal{F}_{gp}(\mathbf{U}_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbf{U}_c(i, j) \tag{3}$$

Thus, compact features are generated to achieve precise and adaptive size selection.

$$\mathbf{z} = \mathcal{F}_{fc}(s) = \delta(\mathcal{B}(\mathbf{W}s)) \tag{4}$$

δ represents ReLU function, and $\mathcal{B}$ represents batch regularization and $\mathbf{W} \in \mathbb{R}^{d \times C}$. To learn the impact of d on the model efficiency, the restoration ratio r is used to control its value.

$$d = \max(C/r, L) \tag{5}$$

L represents the minimum value of d.

3. Select

Under the influence of compact feature descriptor z, cross channel soft attention is used to adaptively select information at different spatial scales. Specifically, it involves normalizing each spectrum.

$$a_c = \frac{e^{\mathrm{A}_c z}}{e^{\mathrm{A}_c z} + e^{\mathrm{B}_c z} + e^{\mathbf{C}_c z}}, b_c = \frac{e^{\mathrm{B}_c z}}{e^{\mathrm{A}_c z} + e^{\mathrm{B}_c z} + e^{\mathbf{C}_c z}}, b_c = \frac{e^{\mathbf{C}_c z}}{e^{\mathrm{A}_c z} + e^{\mathrm{B}_c z} + e^{\mathbf{C}_c z}} \tag{6}$$

a, **b**, and **c** are soft attention vectors for $\mathbf{U}^1$, $\mathbf{U}^2$, and $\mathbf{U}^3$, respectively. $\mathrm{A}_c \in \mathbb{R}^{1 \times d}$ is the c-th element of **A**, and a_c is the c-th element of **a**. The final attention map $\mathrm{A}, \mathrm{B}, \mathrm{C} \in \mathbb{R}^{C \times d}$ can be obtained by the attention weights on each kernel:

$$\mathbf{V}_c = a_c \cdot \mathbf{U}_c^1 + b_c \cdot \mathbf{U}_c^2 + c_c \cdot \mathbf{U}_c^3, \; a_c + b_c + c_c = 1 \tag{7}$$

where $\mathbf{V} = [\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_c], \mathbf{V}_c \in \mathbb{R}^{H \times W}$.

SKNet itself is a channel attention network. Due to the differences in the representation of graph nodes in different spectral layers, the purpose of using a channel attention mechanism is to highlight the node representation of important spectral layers more prominently.

2.2. GraphSAGE Optimized by Sinkhorn Algorithm

The Sinkhorn algorithm is an iterative algorithm used to solve the optimal transport problem which aims to find the optimal mapping between two probability distributions, such that the total cost from one distribution to another is minimized under a given cost function. The optimal transmission of GraphSAGE using the Sinkhorn algorithm involves calculating the cost matrix and applying the Sinkhorn algorithm to optimize the probability transition matrix, as shown in Figure 4.

The cost matrix **C** is used to represent the similarity or distance between nodes in the source domain and the target domain of HSI, with a dimension of $n \times m$, where n is the number of nodes in the source image and m is the number of nodes in the target image. The common cost measurement method is to use the distance between feature vectors between nodes, assuming the node feature matrix in the source domain is $\mathbf{X} \in \mathbb{R}^{n \times d}$, where d is the dimension of the feature vector, and the node feature matrix in the target image is $\mathbf{Y} \in \mathbb{R}^{m \times d}$. The Euclidean distance can be used to measure the distance between node feature vectors.

$$C_{ij} = \left| \mathbf{x}_i - \mathbf{y}_j \right|_2 \tag{8}$$

$\mathbf{x}_i$ is the feature vector of the i-th node in the source image, and $\mathbf{y}_j$ is the feature vector of the j-th node in the target image.

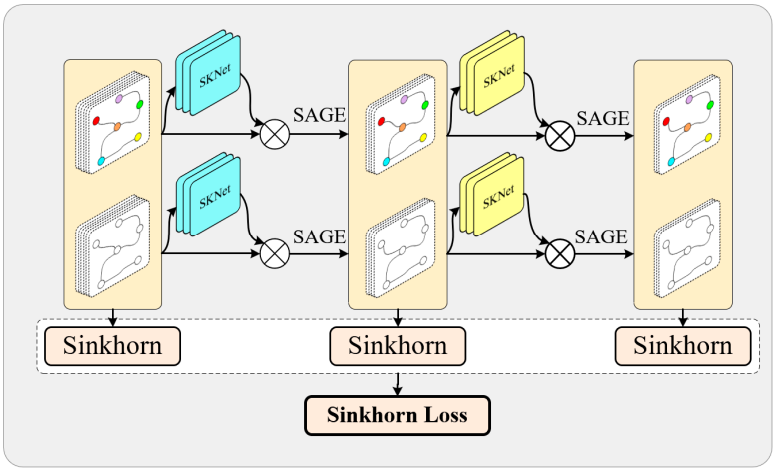


Figure 4. GraphSAGE optimized by Sinkhorn.

The probability transition matrix **P** represents the probability mapping between each node in the source image and each node in the target image. The optimization process using the Sinkhorn algorithm is as follows: first, define the edge distribution vectors **h** and **g**. For GraphSAGE, the degree of the node is usually used as the edge distribution vector. Assuming the node degree in the source image is $\mathbf{d}_x$ and the node degree in the target map is $\mathbf{d}_y$, the edge distribution vector can be represented as:

$$\mathbf{h} = \frac{\mathbf{d}_x}{|\mathbf{d}_x|_1}, \ \mathbf{g} = \frac{\mathbf{d}_y}{|\mathbf{d}_y|_1} \tag{9}$$

where $|\cdot|_1$ represents L_1 norm.

Next, initialize the probability transition matrix **P** as a non-negative square matrix. Then, the elements of **P** can be iteratively updated until the convergence condition is met. In each iteration, use the following formula to update the elements of **P**:

$$P_{ij} = \frac{h_i}{\sum\limits_j P_{ij}} \cdot \frac{g_j}{\sum\limits_i P_{ij}} \tag{10}$$

It is necessary to normalize **P** to become a probability distribution:

$$\bar{\mathbf{P}} = \frac{\mathbf{P}}{\sum\limits_{ij} P_{ij}} \tag{11}$$

Finally, the probability transition matrix $\bar{\mathbf{P}}$ can be used to map the node features in the source domain to the target domain. The node feature matrix in the target map can be calculated as:

$$\mathbf{Y} = \bar{\mathbf{P}} \cdot \mathbf{X} \tag{12}$$

Through iterative updates and normalization operations, the Sinkhorn algorithm can gradually converge to the optimal probability transition matrix, achieving ordered transmission of node features.

2.3. Probability Distribution Alignment

The goal of edge distribution alignment is to reduce the distance between the edge probability distributions of the source and target domains in hyperspectral data, thereby achieving domain adaptation, assuming there is a feature mapping that results in similar edge distributions of the mapped data in the same space. The task of aligning edge distributions is to find this feature mapping **Q**. Firstly, assuming that this mapping is known, then calculate and optimize the distance between two distributions to obtain this mapping while reducing the distance between edge distributions. For the calculation of the

inter-domain distance, among existing methods, multi-kernel maximization of the mean difference is an effective non-parametric distance measure, which is a widely used strategy. In practice, the unbiased estimation of a single kernel MMD compares the square distance between empirical kernels.

$$Dis(\mathbf{X}, \mathbf{Y}) = \left\| \frac{1}{n} \sum_{i=1}^{n} \mathbf{Q}(x_i) - \frac{1}{m} \sum_{j=1}^{m} \mathbf{Q}(y_j) \right\|_H^2 \tag{13}$$

n is the number of source domain features, and m is the number of target domain features. Multiple-kernel MMD (MK-MMD) expands the representation ability of MMD by using multiple kernel functions, thereby better adapting to the differences in data distribution at different scales and shapes. The kernel matrix calculated by multiple kernel functions can reflect the feature similarity of data under different kernel function representations. Through comparative analysis, this section will use the linear kernel function, Gaussian radial basis function kernel function, and Laplace kernel function, as shown in Figure 5.

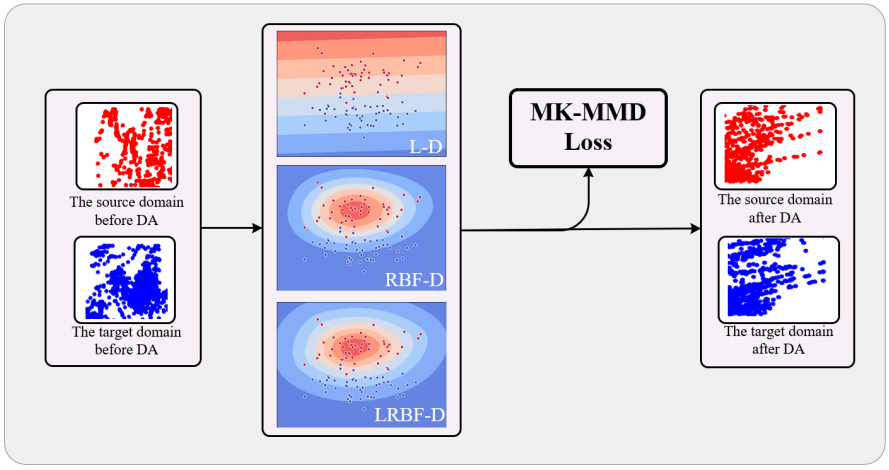


Figure 5. The structure of multi-kernel MMD.

The selection reasons are summarized as follows:

1. The linear kernel function is one of the simplest kernel functions, which can transform linearly indivisible data in low-dimensional space into linearly separable data in high-dimensional space. Compared with other complex kernel functions, linear kernel functions have relatively simple and efficient calculations, making them faster to train on large-scale datasets.

$$K_{lin}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \cdot \mathbf{x}_j \tag{14}$$

$\mathbf{x}_i$ and $\mathbf{x}_j$ are two sample points in the original feature space, $\cdot$ representing the inner product of the vector.

2. The Gaussian radial basis function has strong nonlinear ability and can capture complex nonlinear relationships between data, which can complement linear kernel function. It has smooth properties and can classify and fit data gently, which helps the model's generalization ability. It also has high fitting accuracy, can handle various data distribution problems, and can extract effective features from them. The disadvantage is that it is more sensitive to outliers.

$$K_{lap}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{2\sigma^2} \right) \tag{15}$$

σ is a hyperparameter that controls the width of the kernel function, which can adjust the shape of the function and can be flexibly adjusted to adapt to data distributions of different scales and densities.

3. Laplace kernel functions have nonlinear characteristics, which can handle nonlinear problems and better capture the nonlinear relationships between data. In addition, Laplace kernel functions have stronger robustness against outliers compared to Gaussian kernel functions. The Laplace kernel calculates the distance between samples using the L1 norm, which is the sum of the absolute values of the differences between two vector elements. The L1 norm is very sensitive to outliers because outliers cause an increase in distance. Even if one sample is very different from the others, the distance between them will still be calculated, thus affecting the value of the kernel function. In contrast, Gaussian kernel functions compute the distance between samples using the L2 norm, which is the square root of the sum of squares of the difference between two vector elements. The L2 norm is more robust than the outlier because the effect of the outlier is amplified in the calculation of the sum of squares, but suppressed in the calculation of the square root. Therefore, even if there are outliers, their effect on the distance between samples is relatively small, and the Gaussian kernel function is less sensitive to outliers.

Due to the exponential term of the Laplace kernel function, the influence of outliers on the function value is relatively small, which helps to improve the stability of the model. This can complement the Gaussian kernel function.

$$K_{gau}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{|\mathbf{x}_i - \mathbf{x}_j|_1}{\sigma}\right) \tag{16}$$

2.4. Loss Analysis

The HSI classification method based on domain alignment undergoes two feature alignments after extracting features, further enhancing the expression ability of features and better capturing the similarity and distribution information between images. On the basis of feature alignment, this method introduces consistency constraints to ensure the consistency of the two alignment methods, thereby improving the accuracy and stability of classification. Finally, in order to achieve the classification task, different classifiers were trained for each of the two alignment methods.

In order to comprehensively evaluate the effectiveness of each stage of operation, this section designs four corresponding loss functions for each stage to comprehensively consider different optimization objectives. Firstly, the Sinkhorn loss is used to measure the similarity between two alignment methods and optimize the quality of alignment by minimizing the difference between the two. Next is the multi-core MMD loss, which measures the distribution difference from the source domain to the target domain and enhances alignment performance by maximizing the difference between distributions. In addition, consistency constraint loss is used to constrain the consistency between two alignment methods to ensure their consistency in feature representation. Finally, the classifier loss is used to train the classifier to minimize classification errors and improve classification accuracy. By comprehensively considering these four loss functions, the domain-aligned HSI classification method can fully utilize the information of feature extraction, alignment, and classification, thereby achieving better classification performance.

1. The loss of Sinkhorn

Sinkhorn loss is used to solve optimal transmission problems, which refers to how to transform one probability distribution into another in order to minimize the total cost during the transformation process. In the image alignment stage, in order to achieve optimal image alignment, this section adopts a two-step optimal transmission scheme. When considering input feature maps, three sets of Sinkhorn costs will be involved. The Sinkhorn loss is composed of these three sets of costs.

For two given probability distributions, the source domain distribution $\mathbf{X}$ and the target domain distribution $\mathbf{Y}$, Sinkhorn loss can be used to measure the distance between them. The calculation formula for Sinkhorn loss is as follows:

$$\mathcal{L}_{Sin}(\mathbf{X}, \mathbf{Y}) = \min\gamma \in \Gamma(\mathbf{X}, \mathbf{Y})\langle \mathbf{C}, \gamma \rangle - \epsilon \cdot H(\gamma) \tag{17}$$

$\Gamma(\mathbf{X}, \mathbf{Y})$ is the set of all joint probability distributions between probability distributions $\mathbf{X}$, and $\mathbf{Y}$ is a cost matrix used to measure the cost of transferring one element from $\mathbf{X}$ to another element of $\mathbf{Y}$. $\langle \mathbf{C}, \gamma \rangle$ represents the dot product of the cost matrix $\mathbf{C}$ and joint probability distribution γ. $H(\gamma)$ represents the entropy of the joint probability distribution γ.

By minimizing the Sinkhorn loss, the optimal joint probability distribution can be obtained, thereby obtaining the optimal transmission scheme. When calculating the Sinkhorn loss, the parameter ϵ is a regularization term used to balance the weights between the cost term and the entropy term. Smaller values of ϵ will focus more on the cost term, while larger values of ϵ will focus more on the entropy term.

2. The loss of multi-kernel MMD

Multiple kernel functions are independent of each other when calculating the maximum mean difference loss, which means that for a given sample set, different kernel functions can be used independently to measure the distribution differences between them. That is to say, each kernel function can independently calculate the MMD loss between sample sets without being affected by other kernel functions. This independence allows for a more flexible selection of different kernel functions to adapt to different data features and distribution patterns, thus more accurately describing the differences between sample sets. By using multiple kernel functions, it is possible to comprehensively consider the distribution differences of different scales and angles and obtain a more comprehensive and accurate MMD loss measurement. The loss calculation formulas for the linear kernel function, Gaussian radial basis function kernel function, and Laplace kernel function used in this article are shown as follows, respectively.

$$\mathcal{L}_{lin}(\mathbf{X}, \mathbf{Y}) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} K_{lin}(\mathbf{x}_i, \mathbf{x}_j) - \frac{2}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m} K_{lin}(\mathbf{x}_i, \mathbf{y}_j) + \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} K_{lin}(\mathbf{y}_i, \mathbf{y}_i) \tag{18}$$

$$\mathcal{L}_{gau}(\mathbf{X}, \mathbf{Y}) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} K_{gau}(\mathbf{x}_i, \mathbf{x}_j) - \frac{2}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m} K_{gau}(\mathbf{x}_i, \mathbf{y}_j) + \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} K_{gau}(\mathbf{y}_i, \mathbf{y}_j) \tag{19}$$

$$\mathcal{L}_{lap}(\mathbf{X}, \mathbf{Y}) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} K_{lap}(\mathbf{x}_i, \mathbf{x}_j) - \frac{2}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m} K_{lap}(\mathbf{x}_i, \mathbf{y}_j) + \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} K_{lap}(\mathbf{y}_i, \mathbf{y}_j) \tag{20}$$

The contribution of different kernel functions to feature extraction varies, and their relative contribution can be measured by weighting the losses to form the overall loss function.

Specifically, each kernel function has its unique feature extraction ability, which can capture feature information of different scales and angles. To quantify the relative importance of different kernel functions, each kernel function can be assigned a weight, such as ς_1, ς_2, and ς_3, which represents the contribution of the kernel function to the overall loss. Therefore, the loss function composed of multiple kernel functions can be regarded as the weighted sum of the losses of each kernel function, where the weight of each kernel function determines its influence in the overall loss.

$$\mathcal{L}_{MK-MMD}(\mathbf{X}, \mathbf{Y}) = \varsigma_1\mathcal{L}_{lin}(\mathbf{X}, \mathbf{Y}) + \varsigma_2\mathcal{L}_{gau}(\mathbf{X}, \mathbf{Y}) + \varsigma_3\mathcal{L}_{lap}(\mathbf{X}, \mathbf{Y}) \tag{21}$$

3. The loss of consistency constraint

In the consistency constraint, this section uses the consistency of cross entropy loss imbalance domain alignment to measure the consistency between two domains, which is used to minimize the distribution difference between two domains, making their feature representations more consistent. G and D represent the features after graph alignment and distribution alignment, respectively. The consistency constraint loss can be expressed as:

$$\mathcal{L}_{con}(\mathbf{G}, \mathbf{D}) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{c=1}^{C} p(\mathbf{G}_t^c, \mathbf{G}_t)\log(p(\mathbf{D}_t^c, \mathbf{D}_t)) \tag{22}$$

T represents the number of samples, C represents the number of terrain categories, and $p(\mathbf{G}_t^c, \mathbf{G}_t)$ and $p(\mathbf{D}_t^c, \mathbf{D}_t)$ represent the probability distribution of samples with real labels c in $\mathbf{G}$ and $\mathbf{D}$, respectively.

4. The loss of classifier

The domain adaptation method based on graph alignment will establish a graph-aligned classifier GCN, while the domain adaptation method based on distribution alignment will construct a distribution-aligned classifier CNN. Through the predictions of these two classifiers on the training set, two loss of GCN-C and CNN-C can be obtained, which are called classifier losses and will be used together in the backpropagation process of the neural network. Both GCN and CNN classifiers use cross entropy loss for their classification losses, expressed as follows:

$$\mathcal{L}_{GNN-C}(\mathbf{G}) = -\frac{1}{T_G}\sum_{t=1}^{T_G}\sum_{c}^{C}(y_t^c \times \log(\hat{y}_t)) \tag{23}$$

$$\mathcal{L}_{CNN-C}(\mathbf{D}) = -\frac{1}{T_D}\sum_{t=1}^{T_D}\sum_{c}^{C}(y_t^c \times \log(\hat{y}_t)) \tag{24}$$

T_G and T_D represent the number of samples after graph alignment and distribution alignment, and y_t and $\hat{y}_t$ represent the real label and predicted probability of the samples, respectively.

The cross entropy loss function measures the difference between the predicted probability distribution of the model and the real labels. For the overall loss of the classifier, the regularization coefficients λ_1 and λ_2 are also added to represent the final classifier loss.

$$\mathcal{L}_C(\mathbf{G}, \mathbf{D}) = \lambda_1\mathcal{L}_{GNN-C}(\mathbf{G}) + \lambda_2\mathcal{L}_{CNN-C}(\mathbf{D}) \tag{25}$$

When the model proceeds backpropagation, the parameters can be updated by optimizing its overall loss, which is composed of the aforementioned losses.

$$\mathcal{L} = \alpha_1\mathcal{L}_{Sin}(\mathbf{X}, \mathbf{Y}) + \alpha_2\mathcal{L}_{MK-MMD}(\mathbf{X}, \mathbf{Y}) + \alpha_3\mathcal{L}_{con}(\mathbf{G}, \mathbf{D}) + \alpha_4\mathcal{L}_C(\mathbf{G}, \mathbf{D}) \tag{26}$$

2.5. The Description of Experimental Dataset

In order to verify the effectiveness and progressiveness of the methods proposed in this section, domain adaptation and classification experiments were conducted on two public datasets, Houston 2013 and Houston 2018, respectively. Predicting cross-scene hyperspectral images with different spatial resolutions and spectral bands not only requires source data and target data to have different spatial and spectral resolutions but also requires category intersection. Houston 2013 and Houston 2018 are currently two datasets that meet the requirements. Houston 2013 has 48 effective spectral bands, with a total of 20 types of land cover, and the spatial resolution of the image is 1m. Houston 2018 has 144 effective spectral bands, with a total of 15 land cover categories. The spatial resolution of the image is 2.5 m. There are seven corresponding land cover classifications for the two datasets. The Houston 2013 dataset was taken from 17:37 to 17:39 on 23 June 2012, while the Houston 2018 dataset was taken from 16:31 to 18:18 on 16 February 2017. Through the false color map, it can be found that the surface of the region has changed greatly in a period of five years. The two datasets were shot in completely different seasons, with Houston 2013 filmed in winter and Houston 2018 filmed in summer. In addition, the light angle around 17:37 in winter is small and the intensity is weak, while the light angle around 17:37 in summer is large and the intensity is large. Their detailed contents are shown in Table 2.

Table 2. Public land cover categories of Houston dataset.

No.	Category	Houston 2013	Houston 2018
C1	Healthy grass	345	1353
C2	Stressed grass	365	4888
C3	Trees	365	2766
C4	Water	285	22
C5	Residential	319	5347
C6	Commercial	408	32,459
C7	Road	443	6365

Figure 6a,b shows the false color and ground-truth maps of the Houston 2013 dataset, while Figure 6c,d present the false color and ground-truth maps of the Houston 2018 dataset.

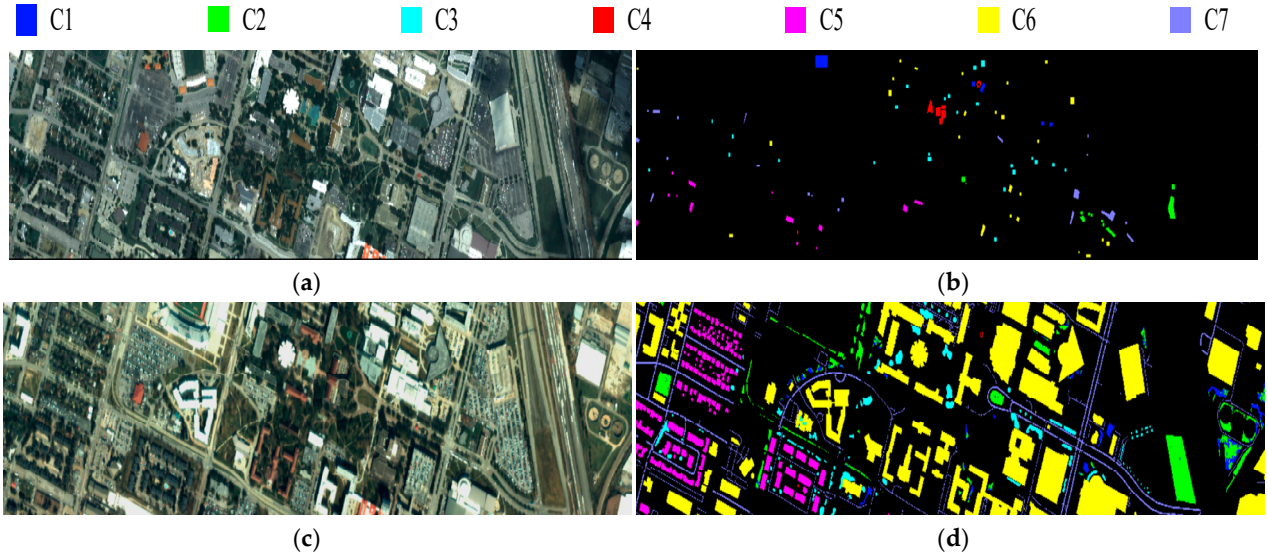


Figure 6. False-color map and ground-truth map of the Houston dataset. (**a**) False-color map of Houston 2013; (**b**) ground-truth map of Houston 2013; (**c**) false-color map of Houston 2018; (**d**) ground-truth map of Houston 2018.

3. Results

3.1. The Settings' Experimental Parameters

The experiment uses the Windows 10 operating system with an Intel (R) Core (TM) i5-6300HQ processor CPU @ 2.30GHz, with 12GB of RAM and NVIDIA GeForce GTX 960M GPU. The experimental programming language is based on the Python language and the popular Pytorch framework. This paper adopts dynamic adjustment of learning rate, with an initial learning rate set to 0.03, and the learning rate is updated according to the predetermined rules every 10 epochs. This setting aims to gradually reduce the learning rate so that the network can learn quickly in the initial stage and pay more attention to detail adjustment and stable convergence in the later stage. As an optimization algorithm, stochastic gradient descent (SGD) was chosen, which is a widely used optimization technique in the field of deep learning. In the experiment, 5% of samples are selected from the source domain of the datasets. In order to demonstrate in detail the key parameter settings of the model in this section, various parameters are listed in Table 3.

Table 3. Model parameters setting.

Layer Name	Output Shape	Filter Size	Padding	Dilation	Groups
Conv2d_0	(W-2) × (W-2) × 32	3 × 3	1 × 1	1 × 1	1
Conv2d_3	(W-4) × (W-4) × 32	3 × 3	1 × 1	1 × 1	1
Conv2d_6	(W-6) × (W-6) × 64	3 × 3	1 × 1	1 × 1	1
Conv2d_9	(W-8) × (W-8) × 64	3 × 3	1 × 1	1 × 1	1
Linear_0	4096	-	-	-	-
Linear_0_1	256	-	-	-	-
Linear_fc	7	-	-	-	-
SAGEConv_0	64	-	-	-	-
SKConv_3×3	(W-8) × (W-8) × 64	3 × 3	1 × 1	1 × 1	1
SKConv_5×5	(W-8) × (W-8) × 64	5 × 5	2 × 2	1 × 1	1
SKConv_7×7	(W-8) × (W-8) × 64	7 × 7	3 × 3	1 × 1	1
Linear_fc	7	-	-	-	-

In Table 3, W represents the size of the input data with a value set to 12, which is crucial for the design of the input layer and subsequent layers of the model. During the experiment, all test sets were used for model evaluation to ensure comprehensiveness and accuracy. Ultimately, the experimental results are based on the average of 10 independent experiments, which can increase the reliability of the results and reduce the error impact caused by the randomness of a single experiment. Through rigorous experimental design and meticulous parameter adjustments, this paper aims to explore and validate the performance of the proposed model on specified tasks.

3.2. Comparative Experimental Results and Analysis

3.2.1. DA Analysis

In this study, specific feature visualization images were used to demonstrate the data distribution of the Houston dataset after feature dimensionality reduction. The horizontal and vertical coordinates in Figure 7a,b represent the first and second feature components after dimensionality reduction, respectively. Through this visualization method, it can be visually observed in Figure 7a that there is a significant difference in the distribution of the Houston 2013 dataset and the Houston 2018 dataset in the feature space. This difference not only reveals the variation characteristics of data at different time points but also reflects the heterogeneity of data distribution in different spectral band numbers and spatial resolutions. This difference poses additional challenges for machine learning algorithms, especially in hyperspectral image prediction tasks.

By comparing the data distribution before and after domain adaptation shown in Figure 7a,b, Figure 7c can be obtained. It can be clearly observed that compared with the situation without domain adaptation, the difference in data distribution after domain adaptation is significantly reduced. This indicates the effectiveness of domain adaptation, which plays a crucial role in narrowing the distribution differences between different scenarios. The reduction of this difference indicates that after domain adaptation processing, the similarity between different datasets has been enhanced, providing more favorable conditions for more accurate and robust prediction of hyperspectral images. Therefore, the introduction of domain adaptation provides effective assistance for hyperspectral image prediction in improving model generalization ability and coping with scene changes and attribute differences.

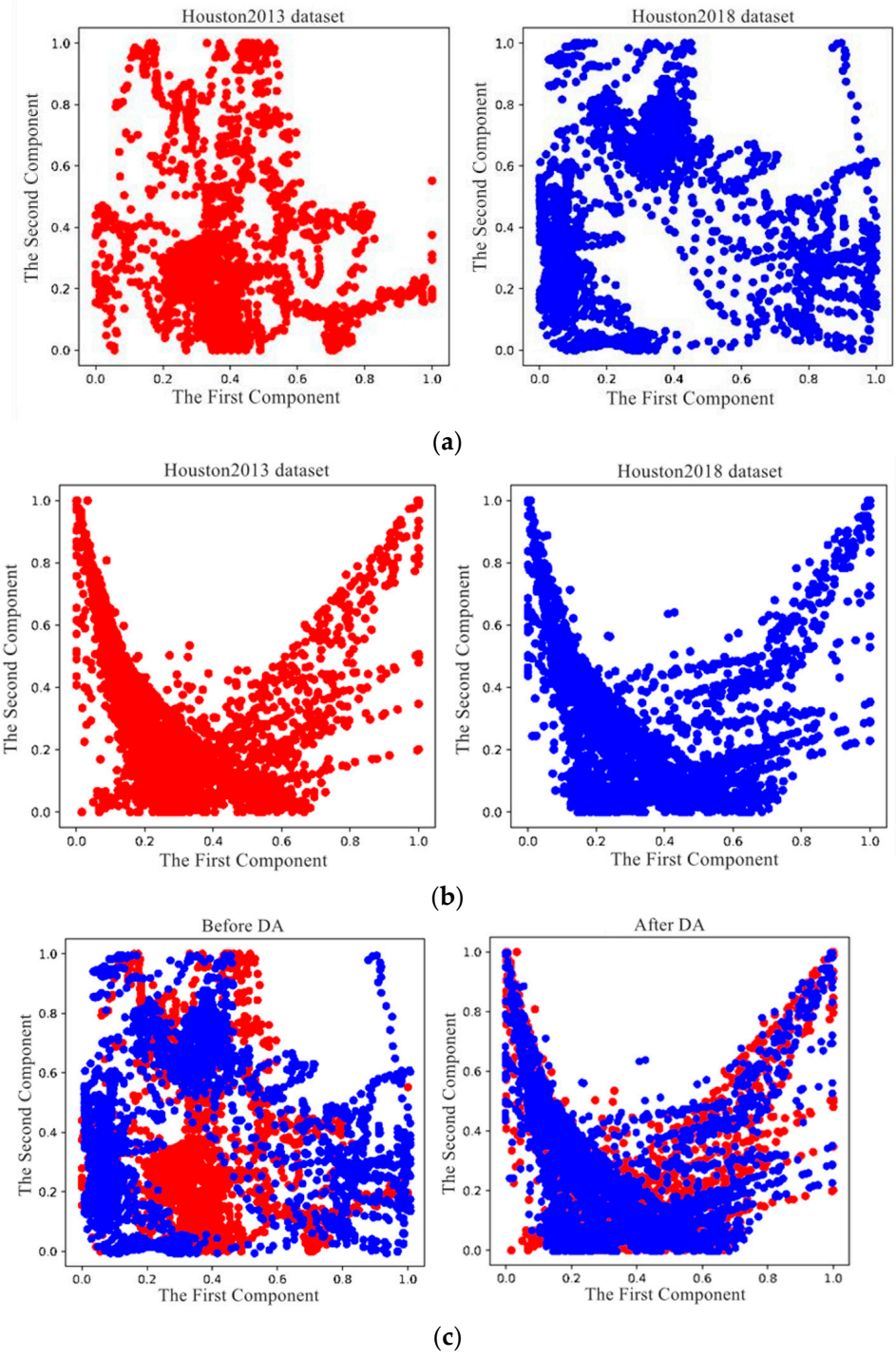


Figure 7. The DA analysis for feature distribution. (**a**) Feature distribution before DA; (**b**) feature distribution after DA; (**c**) comparison of feature distribution before and after DA (combination of Houston 2013 and Houston 2018 data sets).

3.2.2. Comparison of Different Classification Methods

In order to fully verify the progressiveness and effectiveness of the designed cross-scene and cross-attribute hyperspectral image classification model, five groups of comparative experiments were conducted, and the results are shown in Table 4. These experiments aim to compare the performance of the model with traditional methods and advanced methods in the past five years, including SVM, HTCNN [26], CDA [27], CLDA [28], and TSTNet [29]. This series of comparisons aims to comprehensively evaluate the performance

of each model in dealing with cross-domain hyperspectral image classification tasks, in order to confirm the excellence of the proposed model in this paper.

Table 4. Comparison of different models' classification accuracy.

Class \ Method	SVM	HTCNN	CDA	CLDA	TSTNet	GADA
C1	99.78	4.85	65.04	64.97	85.03	64.22
C2	65.30	71.57	86.32	88.04	68.73	91.06
C3	25.70	35.75	66.92	71.73	52.82	65.04
C4	100.00	53.64	77.27	93.16	100.00	100.00
C5	73.26	54.40	98.02	96.13	61.71	62.00
C6	69.12	90.80	60.16	61.71	84.44	87.22
C7	43.63	44.05	73.46	77.71	53.07	62.06
OA (%)	64.67	74.72	68.44	70.12	75.34	80.30
AA (%)	68.11	50.72	75.31	79.06	72.26	75.94
K × 100	45.59	55.24	55.65	57.75	59.69	67.45

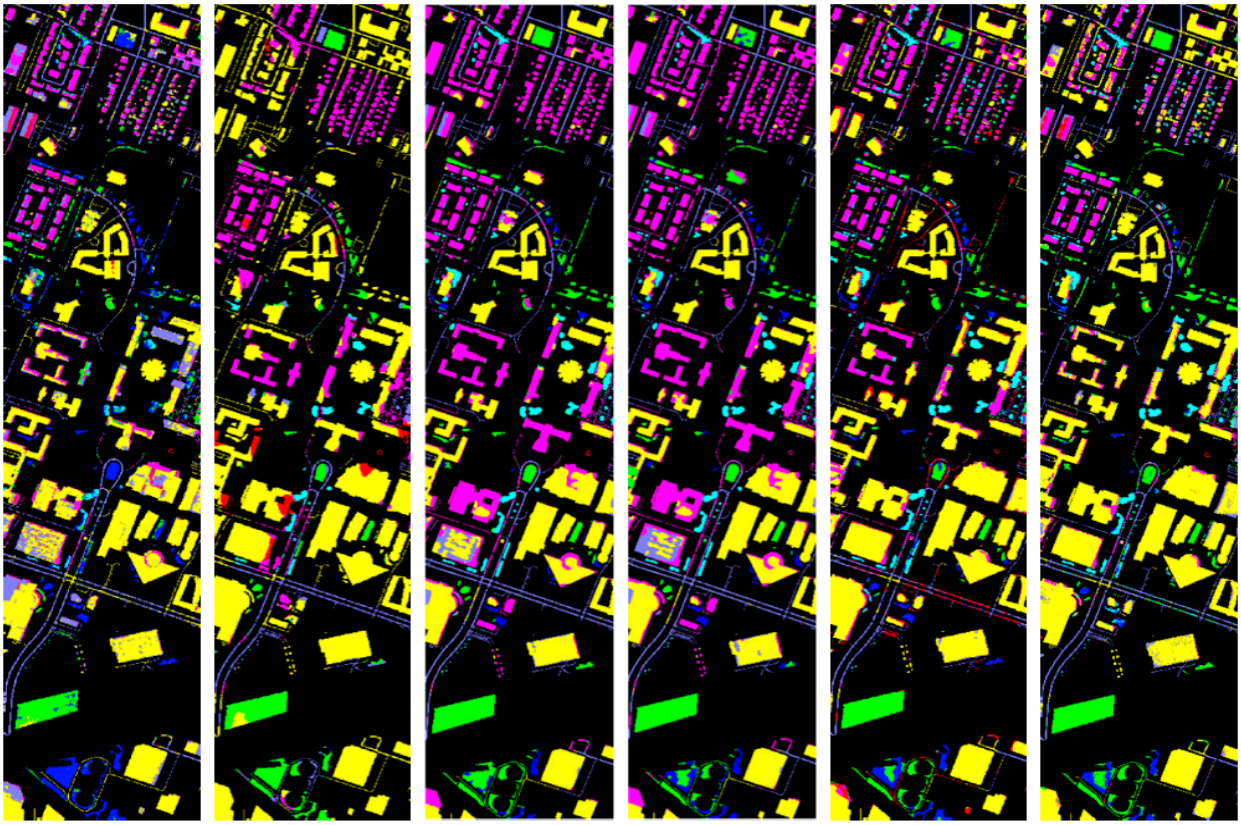
From Table 4, it can be analyzed that GADA has excellent performance compared to other models, demonstrating its advantages in terms of overall performance indicators and specific category analysis. Firstly, in terms of overall accuracy OA, GADA is significantly ahead of other models at a level of 80.30%, providing highly reliable overall classification performance for this task, which reflects the superior generalization ability of the GADA model in complex and variable datasets. The average classification accuracy is significantly higher than the traditional method SVM (68.11%). Although the GADA model has slightly decreased in average accuracy compared to CLDA, the decrease is only 3.12%, but it has improved by 10.18% in OA.

In addition, it is necessary to consider the calculation method of AA, which averages the accuracy of each category. In some cases, the model may perform well in most categories and slightly decline in a few categories. If this decline occurs in a relatively small category, it may not have a significant negative impact on the overall task. Although AA has decreased, focusing on specific category performance and improving Kappa coefficients can help to gain a more comprehensive understanding of the performance of the proposed model.

In terms of the Kappa coefficient, GADA achieved the best result with a score of 67.45%. The Kappa coefficient corrects for the uncertainty of random prediction, thus better reflecting the true performance of the model. The performance of GADA on this indicator further proves its efficient processing and consistent prediction of tasks. In terms of category analysis, taking the C4 category as an example, GADA demonstrated excellent performance with an accuracy of 100%. In contrast, the performance of other models in this category is significantly lower than that of the proposed model, indicating that the proposed model in this paper has a unique advantage in distinguishing this category. By comparing the experimental results, it can be found that the classification effect of the model in the C1 category is slightly inferior to other categories. The C2 category, meanwhile, performed very well. The possible reason for this is that the C1 category is healthy grass and the C2 category is stress grass, which have very similar characteristics. In addition, the distribution of the C2 category in the data set is concentrated and the area is large. The C1 category is scattered at the edge of C2 and has a small area. Therefore, when the topology of the graph is generated, the features of the central node are greatly affected by C2, which eventually leads to a large number of C1 samples being misjudged as C2. Similarly, the classification effect of the C4 category is also very good, which may be due to the characteristics of water itself. The distribution of water itself is rather clustered and not dispersed, so the model can fully learn adjacent features in the topology of the graph.

In order to demonstrate the effectiveness of GADA, a detailed visualization was presented by comparing all the above methods as shown in Figure 8. From Figure 8, it can be observed that the number of erroneous pixels labeled by GADA is relatively small, and its classification results are more accurate, which is consistent with the results detailed in

Table 4. For the C4 category, the classification results of GADA are completely consistent with the visualization results of ground truth (GT), highlighting its outstanding performance in this category. This series of visual comparisons further confirms the significant advantages of GADA in the field of image processing. Overall, the model designed in this paper not only significantly surpasses traditional methods in overall accuracy but also demonstrates outstanding advantages in various categories and overall performance compared to existing excellent methods, providing a more feasible and effective solution for solving the problem of cross-scene hyperspectral image classification.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 8. Visual comparison of classification results by different methods. (**a**) SVM; (**b**) HTCNN; (**c**) CDA; (**d**) CLDA; (**e**) TSTNet; (**f**) GADA.

4. Discussions

4.1. Ablation Experiment

This paper has made improvements on the basis of TSTNet (base) and verified the rationality and effectiveness of the improvement strategy. To evaluate the impact of these improvements, four sets of ablation experiments were conducted, gradually introducing MK-MMD, Sinkhorn, and attention mechanisms into TSTNet. For the convenience of description, the above three modules are represented by A, B, and C. The corresponding experimental results are detailed in Table 5 to display the comparative results of each ablation experiment.

In the first set of experiments, TSTNet was used as the baseline model. In the classification accuracy of various categories (C1 to C7), the performance of the model fluctuates between 52.82% and 100%, with relatively scattered performance. OA reached 75.34%, AA reached 72.26%, and the Kappa coefficient was 0.5969, which serves as a benchmark for subsequent improvements. After the introduction of MK-MMD in the second group of experiments, the performance of each category was improved, especially in the C2 and C3 categories where the improvement was more significant. OA increased from 75.34% to 76.90%, AA increased from 72.26% to 77.82%, and the Kappa coefficient increased from

0.5969 to 0.6352. This indicates that MK-MMD has a positive impact on the performance of the model, especially in improving the classification of specific categories. The performance of the model in the C2, C3, and C6 categories improved after the introduction of Sinkhorn in the third group of experiments. OA increased from 76.90% to 78.69%, AA decreased from 77.82% to 76.44%, but the Kappa coefficient further increased to 0.6537. The introduction of Sinkhorn significantly improved the performance of the model in certain specific categories. The fourth group of experiments introduced all improvement content, with OA increasing from 78.69% to 80.30% and AA slightly decreasing from 76.44% to 75.94%, but the Kappa coefficient continued to increase to 0.6745. The attention mechanism has further improved the overall performance.

Table 5. Comparison of ablation experimental results.

Class \ Method	Base	Base + A	Base + A + B	Base + A + B + C
C1	85.03	87.29	75.46	64.22
C2	68.73	72.00	78.25	91.06
C3	52.82	70.75	70.93	65.04
C4	100.00	100.00	100.00	100.00
C5	61.71	69.96	66.50	62.00
C6	84.44	81.55	85.52	87.22
C7	53.07	63.17	58.44	62.06
OA (%)	75.34	76.90	78.69	80.30
AA (%)	72.26	77.82	76.44	75.94
K × 100	59.69	63.52	65.37	67.45

Through comprehensive improvement strategies, the model in this section has achieved significant improvements in OA, AA, and Kappa coefficients, verifying the rationality and effectiveness of the improvement strategy.

4.2. Validity Verification

In view of the limited amount of data in the cross-scene hyperspectral dataset with different attributes and overlapping categories, and the relatively few relevant research methods; in order to make full use of the existing data sets and more effectively prove the advancement of this method, this chapter further takes Houston 2018 as the source domain to conduct prediction verification on Houston 2013. It is also compared with TSTNet, a baseline method that has been better so far. The experimental results are shown below.

By comparing the experimental results of TSTNET and GADA on Houston18-13, the following comprehensive analysis can be obtained. Overall, GADA shows a significant advantage in overall accuracy (OA) and achieves higher performance than TSTNET. This shows that GADA is more effective in dealing with cross-scene hyperspectral image classification. GADA generally outperforms TSTNET in classification performance across categories. In particular, on certain categories (e.g., C2, C3, C5, C6), GADA's classification accuracy is significantly higher, indicating that it is better able to capture the characteristics of different categories. The improvement in average accuracy (AA) further confirms the robustness of GADA's performance across the entire dataset. GADA performed better than TSTNET, achieving higher average accuracy not only in individual categories but also overall. The increase of the Kappa coefficient indicates that GADA achieves better classification consistency when considering classification random factors. This shows that GADA is more reliable in the classification of various categories, which enhances the consistency and reliability of classification results. In summary, the experimental results clearly show that GADA has more significant advantages than TSTNET in the task of cross-scene hyperspectral image classification.

4.3. Hyperparameter Discussion

In order to explore the impact of the proportion of training samples on the prediction effect of the model, we conducted a set of comparative experiments, and the proportion of training samples was set as 1%, 3%, 5%, 7%, and 10% respectively. The experimental results are shown in Figure 9. Figure 10 gives the comparison of model effects under different training sample proportions.

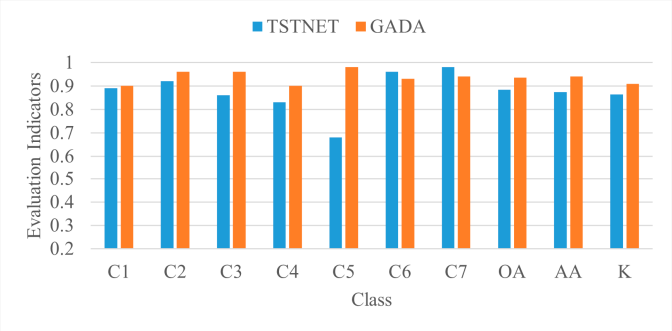


Figure 9. Houston 2018–2013 baseline experimental results.

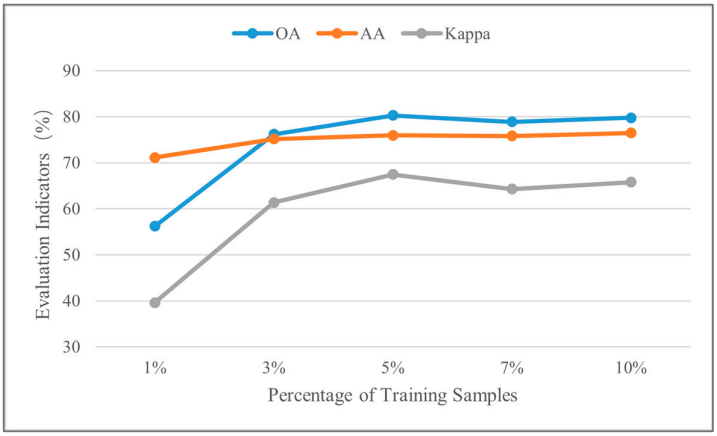


Figure 10. Comparison of model effects under different training sample proportions.

The experimental results show that with the increase in the proportion of training samples, the overall performance of the model is improved in a certain range. Specifically, when the proportion of training samples increases from 1% to 5%, the overall accuracy (OA) significantly increases from 56.2% to 80.3%, which indicates that more training samples are conducive to the model learning more abundant features and rules, thus effectively improving the overall prediction accuracy. However, when the sample proportion increased to 10%, the upward trend of OA value slowed down and leveled off, which means that under the current model structure and parameter settings, after a certain threshold (about 5%), continuing to increase the training sample has a limited effect on the improvement of global prediction ability. The average accuracy (AA) of each category also presents a similar trend, that is, with the increase of sample proportion, it first increases and then stabilizes, especially in the stage of low sample proportion. However, after the increase of training samples exceeds 5%, the benefit of improving the model's equilibrium prediction ability in each category gradually decreases. The Kappa coefficient is used as an index to measure the gap between the actual prediction performance and random prediction performance of classification models, and its variation trend is consistent with OA and AA. That is, the Kappa coefficient increased significantly from 39.59% to 67.45% during the period when the sample proportion increased from 1% to 5%, and then the increase speed gradually slowed down. In summary, from the perspective of the OA, AA, and Kappa coefficients of the model, a 5% training sample proportion is the best choice.

5. Conclusions

This article proposes an HSI classification method based on graph alignment and distribution alignment to address the issues of high HSI labeling cost, limited training sample size, and prediction between different attributes. This method first uses VGG16 to extract spatial and spectral features from the original hyperspectral image and then uses two methods of image alignment and distribution alignment for domain alignment. In addition, this method also adopts an attention mechanism and three kernel functions to extract linear and high-dimensional nonlinear features of hyperspectral images, thereby improving the stability of the model. In terms of the Kappa coefficient, GADA achieved the best result with a score of 67.45%, while OA was significantly ahead of the other models with a value of 80.30%. The experimental results show that this method can effectively improve the HSI classification performance of different attributes across scenes. Future research can focus on improving the image alignment and distribution alignment algorithms to enhance the accuracy and stability of alignment for hyperspectral images of different bands or time points and improving the feature extraction model by introducing more advanced network structures and attention mechanisms to extract more informative and discriminative features before domain alignment.

There are also some potential applications for our method. For example, in urban planning, hyperspectral image classification can provide detailed information on urban land use and cover. By classifying hyperspectral images of urban areas, different types of buildings, green spaces, roads, and water bodies can be identified and monitored, providing data support for urban planning decisions, such as assessing urban expansion, land use change, and urban greening planning.

Author Contributions: Conceptualization, A.W., S.D. and H.C.; methodology, A.W. and S.D.; software, S.D.; validation S.D.; writing—review and editing A.W., S.D. and H.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Key Research and Development Plan Project of Heilongjiang (JD2023SJ19), the Natural Science Foundation of Heilongjiang Province (LH2023F034), the High-end Foreign Experts Introduction Program (G2022012010L), and the Key Research and Development Program Guidance Project of Heilongjiang (GZ20220123).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Houston 2018: https://hyperspectral.ee.uh.edu/?page_id=1075, accessed on 16 February 2018. Houston 2013: https://hyperspectral.ee.uh.edu/?page_id=459, accessed on16 February 2013.

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

References

1. Yi, Q.; Zhang, Y.H.; Wang, Q.; Zong, Y.T. Research on Hyperspectral Image Classification Algorithm Based on Improved Residual Network. *Ordnance Ind. Autom.* **2023**, *42*, 15–20.
2. Song, R.X.; Feng, Y.N.; Cheng, W.; Wang, X.H. Research Progress on Hyperspectral Image Change Detection. *Spectrosc. Spectr. Anal.* **2023**, *43*, 2354–2362.
3. Wang, K.Q.; Peng, X.W.; Zhang, Y.Z.; Luo, Z.; Jang, D.P. A Hyperspectral Classification Method for Agroforestry Vegetation Based on Improved U-Net. *For. Eng.* **2022**, *38*, 58–66.
4. Tømmervik, H.; Julitta, T.; Nilsen, L.; Lennart, N.; Park, T.; Burkart, A.; Ostapowicz, K.; Karlsen, S.R.; Parmentier, F.; Pirk, N.; et al. The Northernmost Hyperspectral FLoX Sensor Dataset for Monitoring of High-Arctic Tundra Vegetation Phenology and Sun-Induced Fluorescence (SIF). *Data Brief* **2023**, *50*, 109581. [CrossRef] [PubMed]
5. Begliomini, F.N.; Barbosa, C.C.F.; Martins, V.S.; Novo, E.M.; Paulino, R.S.; Maciel, D.A.; Lima, T.M.; O'Shea, R.E.; Pahlevan, N.; Lamparelli, M.C. Machine Learning for Cyanobacteria Mapping on Tropical Urban Reservoirs using PRISMA Hyperspectral Data. *ISPRS J. Photogramm. Remote Sens.* **2023**, *204*, 378–396. [CrossRef]
6. Li, Y.L.; Chen, Y.X. Research Progress of Imaging Technology in Food Safety and Quality Control. *Modern Food* **2020**, *17*, 114–115.
7. Wang, T.T.; Cai, H.X.; Li, S. Research Progress of Novel Metasurface Spectral Imaging Chips. *Laser Optoelectron. Prog.* **2023**, *60*, 203–212.

8. Song, N.; Guo, H.Z.; Shen, C.Y. Research on Detection Technology of Brain Glioma Based on Hyperspectral Imaging. *Spectrosc. Spectr. Anal.* **2020**, *12*, 3784–3788.
9. Zhang, J.W.; Chen, Y.J. Overview of hyperspectral image classification methods. *J. Nanjing Univ. Inf. Sci. Technol.* **2020**, *1*, 89–100.
10. Liu, Q.C.; Xiao, L.; Yang, J.X.; Chan, J.C. Content-Guided Convolutional Neural Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 6124–6137. [CrossRef]
11. Islam, R.; Zakaria, Z. Hybrid 3DNet: Hyperspectral Image Classification with Spectral-spatial Dimension Reduction using 3D CNN. *Int. J. Comput. Appl.* **2022**, *184*, 6–11.
12. Jia, S.; Jiang, S.G.; Zhang, S.Y.; Xu, M.; Jia, X.P. Graph-in-Graph Convolutional Network for Hyperspectral Image Classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 1157–1171. [CrossRef]
13. Zhao, C.H.; Li, T.; Feng, S. Hyperspectral Image Classification Based on Dense Convolution and Domain Adaptation. *Acta Photonica Sin.* **2021**, *50*, 3653–3666.
14. Shen, J.J. Research on Spatial-Spectral Feature Extraction and Classification of Cross-scene Hyperspectral Imagery. Master's Thesis, Shenzhen University, Shenzhen, China, 2021; pp. 1–69.
15. Ye, M.C.; Qian, Y.T.; Zhou, J.; Tang, Y.Y. Dictionary Learning-Based Feature-Level Domain Adaptation for cross scene Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 1544–1562. [CrossRef]
16. Li, J.J.; Meng, L.C.; Zhang, K.; Lu, K.; Shen, H.Y. Review of Studies on Domain Adaptation. *Comput. Eng.* **2021**, *47*, 1–13.
17. Wang, M.; Deng, W.H. Deep Visual Domain Adaptation: A Survey. *Neurocomputing* **2018**, *312*, 135–153. [CrossRef]
18. Zhu, Y.C.; Zhuang, F.Z.; Wang, J.D.; Chen, J.W.; Shi, Z.P.; Wu, W.J.; Qing, H. Multi-representation Adaptation Network for Cross-domain Image Classification. *Neural Netw.* **2019**, *119*, 214–221. [CrossRef]
19. Zhu, Y.C.; Zhuang, F.Z.; Wang, J.D.; Ke, G.L.; Chen, J.W.; Bian, J.; Xiong, H.; He, Q. Deep Subdomain Adaptation Network for Image Classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 1713–1722. [CrossRef]
20. Li, Z.K.; Tang, X.Y.; Li, W.; Wang, C.Y.; Liu, C.W.; He, J. A Two-stage Deep Domain Adaptation Method for Hyperspectral Image Classification. *Remote Sens.* **2020**, *12*, 1054. [CrossRef]
21. Zhu, S.H.; Du, B.; Zhang, L.P.; Li, X. Attention-Based Multiscale Residual Adaptation Network for cross scene Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–15. [CrossRef]
22. Liang, X.J.; Zhang, Y.; Zhang, J.P. Attention Multisource Fusion-Based Deep Few Shot Learning for Hyperspectral Image Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 8773–8788. [CrossRef]
23. Wang, W.J.; Ma, L.; Chen, M.; Li, X. Joint Correlation Alignment-Based Graph Neural Network for Domain Adaptation of Multitemporal Hyperspectral Remote Sensing Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 3170–3184. [CrossRef]
24. Wan, S.; Yang, J.; Gong, C. Advances of Hyperspectral Image Classification Based on Graph Neural Networks. *Acta Electron. Sin.* **2023**, *51*, 1687–1709.
25. Li, X.; Wang, W.H.; Hu, X.L.; Yang, J. Selective Kernel Networks. *Proc. Int. Conf. Comput. Vis.* **2021**, 7263–7272. [CrossRef]
26. He, X.; Chen, Y.S.; Ghamisi, P. Heterogeneous Transfer Learning for Hyperspectral Image Classification Based on Convolutional Neural Network. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 3246–3263. [CrossRef]
27. Liu, Z.X.; Ma, L.; Du, Q. Class-Wise Distribution Adaptation for Unsupervised Classification of Hyperspectral Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 508–521. [CrossRef]
28. Fang, Z.Q.; Yang, Y.X.; Li, Z.K.; Li, W.; Chen, Y.S.; Ma, L.; Du, Q. Confident Learning-Based Domain Adaptation for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–16. [CrossRef]
29. Zhang, Y.X.; Li, W.; Zhang, M.M.; Qu, Y.; Tao, R.; Qi, H.R. Topological Structure and Semantic Information Transfer Network for cross scene Hyperspectral Image Classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 2817–2830. [CrossRef]