remote sensing

MDPI

Article

DILRS: Domain-Incremental Learning for Semantic Segmentation in Multi-Source Remote Sensing Data

Xue Rui [1], Ziqiang Li [2], Yang Cao [3], Ziyang Li [4] and Weiguo Song [1,*]

[1] State Key Laboratory of Fire Science, University of Science and Technology of China, Hefei 230026, China; ruixue27@mail.ustc.edu.cn
[2] CAS Key Laboratory of Technology in Geo-Spatial Information Processing and Application Systems, University of Science and Technology of China, Hefei 230026, China; iceli@mail.ustc.edu.cn
[3] Department of Automation, University of Science and Technology of China, Hefei 230026, China; forrest@ustc.edu.cn
[4] Key Laboratory of Quantitative Remote Sensing Information Technology, Aerospace Information Research Institute, Chinese Academy of Sciences (CAS), Beijing 100094, China; zyli@aoe.ac.cn
* Correspondence: wgsong@ustc.edu.cn

Abstract: With the exponential growth in the speed and volume of remote sensing data, deep learning models are expected to adapt and continually learn over time. Unfortunately, the domain shift between multi-source remote sensing data from various sensors and regions poses a significant challenge. Segmentation models face difficulty in adapting to incremental domains due to catastrophic forgetting, which can be addressed via incremental learning methods. However, current incremental learning methods mainly focus on class-incremental learning, wherein classes belong to the same remote sensing domain, and neglect investigations into incremental domains in remote sensing. To solve this problem, we propose a domain-incremental learning method for semantic segmentation in multi-source remote sensing data. Specifically, our model aims to incrementally learn a new domain while preserving its performance on previous domains without accessing previous domain data. To achieve this, our model has a unique parameter learning structure that reparametrizes domain-agnostic and domain-specific parameters. We use different optimization strategies to adapt to domain shift in incremental domain learning. Additionally, we adopt multi-level knowledge distillation loss to mitigate the impact of label space shift among domains. The experiments demonstrate that our method achieves excellent performance in domain-incremental settings, outperforming existing methods with only a few parameters.

Keywords: incremental learning; multi-source remote sensing; semantic segmentation; catastrophic forgetting

Citation: Rui, X.; Li, Z.; Cao, Y.; Li, Z.; Song, W. DILRS: Domain-Incremental Learning for Semantic Segmentation in Multi-Source Remote Sensing Data. *Remote Sens.* **2023**, *15*, 2541. https://doi.org/10.3390/rs15102541

Academic Editors: Silvia Liberata Ullo, Parameshachari Bidare Divakarachari and Pia Addabbo

Received: 20 April 2023
Revised: 9 May 2023
Accepted: 10 May 2023
Published: 12 May 2023

1. Introduction

The deployment of deep learning models on edge devices is emerging as a significant trend for future interpretation of Earth intelligence [1]. By locally processing real-time data, this approach eliminates the need for data transfer to cloud devices, saving significant processing time, data transmission, and resource consumption. This technique has matured in the field of autonomous driving [2,3]. Subsequently, the faster and more abundant acquisition of remote sensing data has created new standards and requirements for deep learning models, making it essential for these models to adapt and learn continuously over time. However, most existing deep learning models for semantic segmentation tasks [4,5] are trained offline and statically deployed [6,7]. These models require large amounts of data for long-term training and can only be applied to a specific domain, with no provision for adapting or expanding over time. When new domain data becomes available, the models cannot maintain their performance requirements for the original domain, leading

to catastrophic forgetting [8,9] problems. As illustrated in Figure 1, we show an example of catastrophic forgetting, when semantic segmentation models are applied in continual remote sensing domains. Therefore, it is critical to develop deep learning models [10,11] that can adapt to changing data and continuously learn to keep up with the evolving requirements of Earth intelligence interpretation.

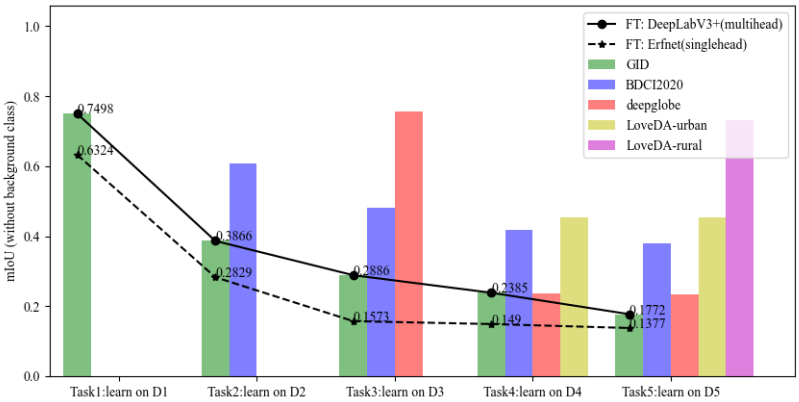

Figure 1. An example of catastrophic forgetting in a continual domain sequence. We deploy two segmentation models statically, where pre-trained models are sequentially fine-tuned (FT) [12] on five different domains (D1–D5) in the field of continual remote sensing. This domain sequence closely simulates the data collected by edge devices, such as satellite and aerial sensors capturing the images of urban and rural regions. The performance of the models is evaluated from two perspectives. Firstly, a bar chart demonstrates the performance of the DeepLabV3+ model when fine-tuned on D1–D5 (i.e., GID [13], BDCI2020 [14], deepglobe [15], LoveDA-urban [16], and LoveDA-rural [16]). The chart reveals a degradation in performance for the previous domains. Secondly, two line diagrams illustrate the performance of the GID domain on different tasks for the DeepLabV3+ [4] and Erfnet [5] models. The results show that models trained on new domains achieve good performance, while the performance of the models on previous domains gradually decreases. For a more detailed description, please see Section 4.4.

The existing training schemes to solve the challenge of deep learning models in continual domains can be summarized as follows [17]: (a) separate training in a single domain, storing each model, and then flexibly switching between different domains; (b) storing each domain data, then joint training with multiple domains when deploying on a specific domain; (c) sequential training with incremental domains, adopting the domain adaption method to improve the performance in the target domain. However, deploying deep learning models on edge devices requires the consideration of the operation rate, storage pressure, and data privacy problems. Obviously, the aforementioned methods do not meet these requirements. Instead, an extensible lightweight model with an incremental learning ability that can maintain good performance among all cumulative domains is more suitable for applications. Therefore, incremental learning (also known as continual learning) [8,9] is proposed.

The study of incremental learning in remote sensing is still in its early stages and is primarily focused on task-incremental learning or class-incremental learning, as evidenced by the works of [18–22]. However, domain-incremental learning is more relevant in practical applications, since deep learning models are expected to learn continual remote sensing domains when deployed on the edge devices. Despite the importance of domain-incremental learning, it has received relatively little attention. In this regard, our objective is to address this gap and explore methods for domain-incremental learning that can improve performance in continual domains. Additionally, we choose semantic segmentation as our downstream task.

In the domain-incremental setting, catastrophic forgetting can be attributed to two main factors: domain shift and label space shift. Figure 2 illustrates the properties of con-

tinual remote sensing domains, including: (1) inconsistent class distributions of different regions; and (2) spatial resolutions and spectral divergence of the specific object category in different sensors. Due to the diversity in sensors and regions, domain shift can occur as the image capture conditions change, such as variations in object scales, complex background, spatial resolution, spectral divergence, and weather conditions. Furthermore, previously unseen classes in new geographical regions and inconsistent class distributions can both contribute to label space shift. As a result, addressing domain shift and label space shift is crucial to achieving optimal segmentation model performance in domain-incremental learning.

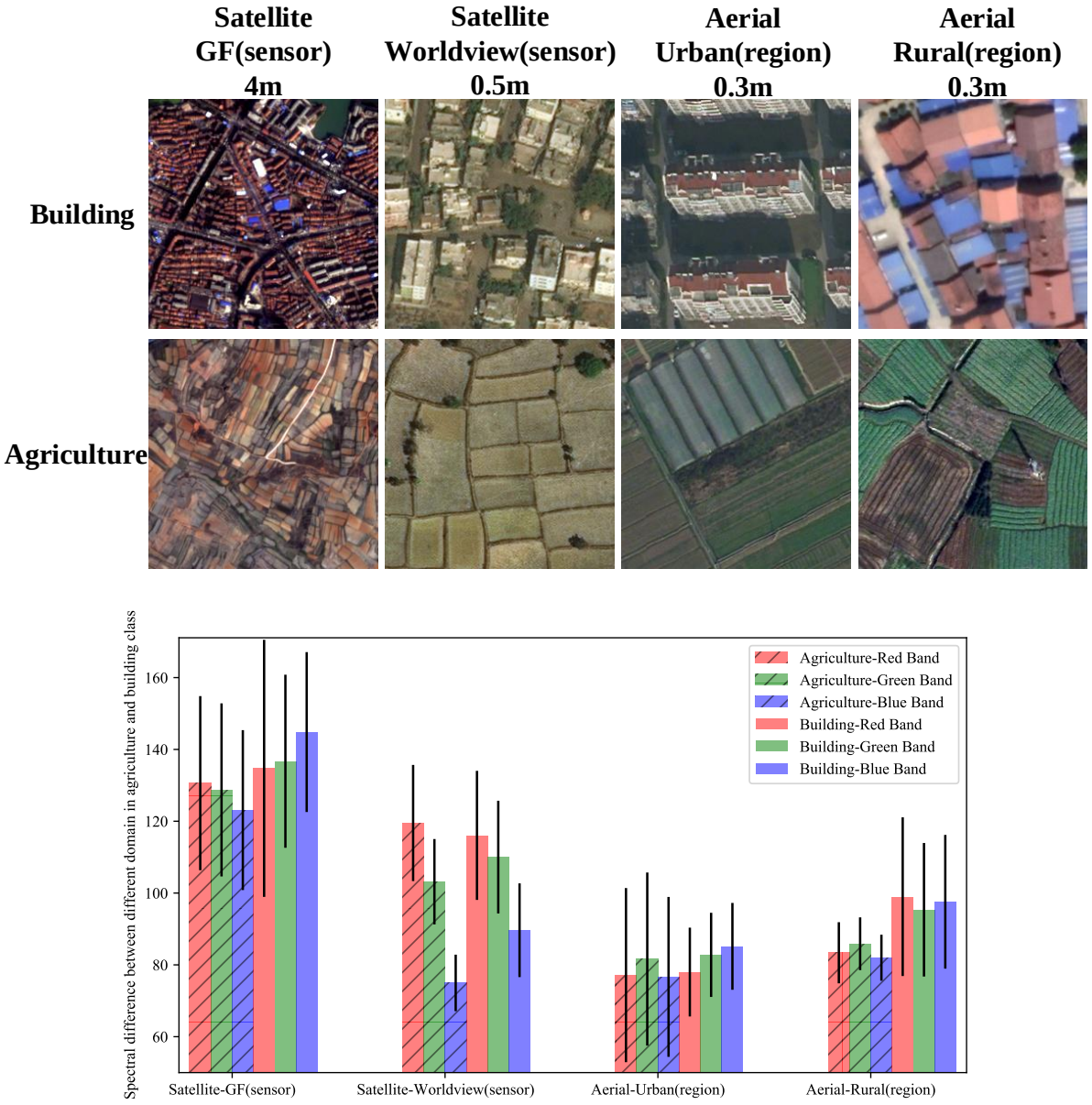

Figure 2. A brief illustration of the multi-source remote sensing data, including samples from multi-sensor (satellite and aerial sensors) and multi-region (urban and rural regions) data. Taking the category of building and agriculture for example, the upper half of the graph shows the visual difference in the sensor of different spatial resolutions, reflected in object scales and styles. The second half shows the spectral divergence of the building and agriculture in a series of images from different domains, by the mean and standard deviation in red, green, and blue wavelengths. We simplify our research and regard the remote sensing domains as images from multiple sensors and regions in our research.

Therefore, we propose the domain-incremental learning for the remote sensing framework (DILRS) as a solution in continual domains. Drawing inspiration from the effectiveness of the universal parametrization in multi-domain learning [23,24], we find that model parameters can be divided into those that learn domain-specific features and those that represent shared features. In domain-incremental learning settings, retaining domain-agnostic parameters and switching domain-specific parameters can be an effective approach to maintaining performance in current and previous domains with fewer parameters. To handle the label space shift problem during model updates in domains with non-overlapping new classes, we propose a multi-level knowledge distillation loss, which refers to knowledge distillation strategies [25–27]. Our main contributions can be summarized as follows:

(1) We define the problem of domain-incremental learning for remote sensing and propose a dynamic framework specific for this problem without using previous training data and labels. Experimental results demonstrate the excellent performance of our method with fewer parameters.
(2) To alleviate the domain shift among incremental domains, we adapt domain residual adapter modules in the structure, using different optimization strategies towards domain-specific and domain-agnostic parameters.
(3) Consider different label space shift, class-specific knowledge distillation loss is applied to distil the common class knowledge between domains, and we also use the distillation loss at intermediate feature space to avoid background class interference.

2. Related Work

While remote sensing visual perception models such as image scene classification and segmentation models have been thoroughly studied, incremental learning in remote sensing is still in its early stages. This section aims to provide an overview of incremental learning, followed by a focus on domain-incremental learning and incremental learning for semantic segmentation. In particular, we will briefly review the relevant research in remote sensing.

2.1. Incremental Learning

Incremental learning [8,9], aiming to tackle catastrophic forgetting during model learning and extending, can be divided into three scenarios: task-incremental learning, class-incremental learning, and domain-incremental learning. Three categories of incremental learning technologies have been proposed, including replay-based, regularization-based, and parameter isolation-based strategies. Replay-based methods [28] involve storing a portion of old data or training additional generators to produce pseudodata for replay, followed by joint training with new data. However, this approach may raise data privacy concerns and create storage pressures. To address the forgetting of previous knowledge, regularization-based strategies [12,29] typically employ knowledge distillation or regularization terms in loss functions. Compared with replay-based strategies, regularization-based strategies do not require the storage of previous data. However, the model is optimized based on the previous task, which could lead to the final model not converging towards the globally optimal solution and unsatisfactory performance. Parameter isolation-based strategies [30,31], on the other hand, typically isolate or freeze important model parameters from previous tasks and allow models to introduce new parameters to prevent forgetting in the new task. Given that remote sensing data storage is impractical, and we aim to maximize effectiveness, parameter isolation-based methods are the most suitable approach for our task [19]. Our proposed model is based on parameter isolation-based methods.

2.2. Domain-Incremental Learning

Domain-incremental learning refers to model learning from continual domains of changing distribution, where nonstationarity is reflected in background, blur, noise, and other factors [8,9]. However, there is a gap between this definition and real-world applica-

tions. A relevant example of domain-incremental learning in the real world is an agent that needs to learn to survive in different environments. Classic scenes include autonomous driving [23,32], person ReID [33], and crowd counting [34], among others [35]. For instance, Garg et al. [23] proposed a dynamic semantic segmentation model, which is effective in three driving scenes from visually disparate geographical regions. Mirza et al. [32] presented a robust object detection method for autonomous driving that learns incrementally across varying weather conditions. In fact, the research setting of domain-incremental learning is also applicable to remote sensing. Multi-source remote sensing data collected by revisiting multiple satellites implies that remote sensing intelligent interpretation models will require higher domain-incremental learning capabilities. To the best of our knowledge, there is no related research on domain-incremental learning for remote sensing.

Additionally, domain adaptation [36] and multi-domain learning [24,37] are closely related to our research. Multi-domain learning, with access to all domains' data, aims to retain good performance in all domains. Domain adaptation utilizes labeled data in the source domain to maximize performance in the target domain. The difference with our work is reflected in the model's goal, the availability of different domains, and the diversity in the label space.

2.3. Incremental Learning for Semantic Segmentation

Recently, the limitation of an offline setting used in semantic segmentation models is cause for concern. Incremental learning strategies for semantic segmentation have been proposed [25–27]. These studies are based on the assumption that models update for new, unseen categories in the class-incremental setting, where the background class or semantic shift is the primary challenge. Cermelli et al. [27] highlighted that the semantic distribution shift exists in the non-overlapping new classes of each learning step and proposed a distillation-based framework specific to solve this issue. Klingner et al. [26] introduced a knowledge distillation loss without relying on previous data in class-incremental learning for semantic segmentation.

However, compared with natural images, continual semantic segmentation in remote sensing is a relatively new field with few papers [20–22,38–41] studying the problem. Tasar et al. [22] were the first to study the incremental learning scenario of remote sensing segmentation, while Shan et al. [38] proposed two effective modules embedded in the proposed class-incremental segmentation framework without access to previous data. However, these experimental settings considered only a class-incremental semantic shift in the same domain. In reality, the semantic shift and domain shift may coexist in remote sensing applications, which motivates us to study semantic segmentation in the domain-incremental learning setting.

3. Method

This section provides an overview of our work, DILRS, which focuses on domain-incremental learning for remote sensing. Firstly, we present the problem formulation of domain-incremental learning that we use in our work. Next, we introduce the overall framework, DILRS, and provided a detailed description of its key component, the domain residual adapter module. Finally, we discuss the proposed loss function and the optimization strategy that we developed.

3.1. Problem Formulation

The domain incremental learning setting assumes the presentation of N tasks, each corresponding to n training domains $\mathcal{D}_k = \{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^{n}$, where $\mathbf{x}_k$ and $\mathbf{y}_k$ represent a data sample and its corresponding label, respectively. In contrast to current incremental semantic segmentation research [40], where only a semantic shift exists at each step, our experiments consider the coexistence of domain shift and label shift between D_k and D_{k-1}. Additionally, y_k may contain overlapping classes and new classes compared to y_{k-1}, and vice versa. We used the original dataset labels as domain labels and a multi-head decoder

structure to prevent semantic shifts in the background class. The classifier in each decoder predicts the category of pixels in different domains independently.

The overall training process of the domain-incremental semantic segmentation model is illustrated in Figure 3. The segmentation model is trained incrementally on the domain sequence. At step k of training, we train model $\mathcal{M}k(\mathbf{x}k, \mathbf{k})$ on domain $\mathcal{D}k = (\mathbf{x}k, \mathbf{y}k)$. We assume that the old data samples $\sum i = 1^{k-1} D_i$ become unavailable while D_k is provided. Our model aims to adapt to each new domain without degrading its performance on previous ones. During the inference phase, we have access to the ID of each domain, similarly to task-incremental learning. We evaluated the performance of the model at the end of the training sequence for the current and previous domains $\sum_{i=1}^{k} D_i$. To simplify the introduction, we refer to the domain, dataset, and task at step k as D_k.

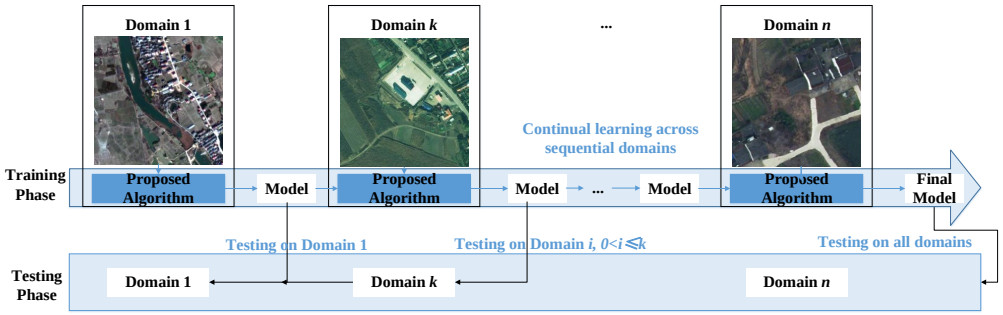

Figure 3. The DILRS framework for domain-incremental learning in remote sensing involves a process where, at each step, only the current domain is available. Our proposed model incrementally trains on the current domain while simultaneously testing on all previous domains.

3.2. Proposed Framework

The DILRS architecture consists of two components: a shared encoder $\mathcal{E}$ and K parat-actic domain-specific decoders $\mathcal{C}_k$, as depicted in detail in Figure 4. The shared encoder is based on the lightweight efficient residual factorized network (Erfnet) [5], which incorporates domain residual adapter (DRA) modules. These DRA modules learn both domain-specific features and domain-agnostic features. For the domain $\mathcal{D}_k = (\mathbf{x}_k, \mathbf{y}_k)$, our framework learns a mapping,

$$\overline{\mathbf{y}}_k = M_k(x_k, k; \mathcal{W}_k, \mathcal{W}_s) = \mathcal{C}_k(\mathcal{E}(x_k, k; \alpha_k, \mathcal{W}_s)), \tag{1}$$

where $\overline{\mathbf{y}}_k$ represents the predictions of the model, $\mathcal{W}_k = \{\alpha_k, \mathcal{C}_k\}$ and $\mathcal{W}_s$ are domain-specific parameters and domain-agnostic parameters in the model, respectively; α_k are domain-specific parameters in encoder $\mathcal{E}$.

3.3. Domain Residual Adapter Module

The domain residual adapter (DRA) module is a critical component in the DILRS architecture, as it is responsible for reparametrizing the network into domain-specific and domain-agnostic parameters. As shown in Figure 5, the features in the jth module are denoted as u_k^j, which consist of the features $\hat{u}_k^{j-1}$ introduced by the DRA module with a skip connection of the features u_k^{j-1} from the previous module:

$$u_k^j = u_k^{j-1} + \hat{u}_k^{j-1}, j \geq 2. \tag{2}$$

Specifically, $\hat{u}_k^{j-1}$ is formed by a concatenation of the domain-agnostic part $g^j(\cdot)$ and domain-specific part $h_k^j(\cdot)$:

$$\hat{u}_k^{j-1} = f_k^j\left(h_k^j\left(u_k^{j-1}\right); g^j\left(u_k^{j-1}\right)\right), j \geq 2, \tag{3}$$

where g^j is a domain-agnostic structure across all domains, and f_k^j and h_k^j constitute a parallel domain-specific residual adapter structure for each domain. Among them, g^j is composed of [3 × 1] and [1 × 3] convolutional layers, followed by a ReLU activation function. As for the domain-specific structure, h_k^j is a [1 × 1] convolutional layer denoted as a domain-specific layer of the domain k in parallel. Additionally, f_k^j represents the batch normalization layers, which are also domain-specific structures in parallel. The setting of the DRA fellows the residual adapter module in [23,24,42].

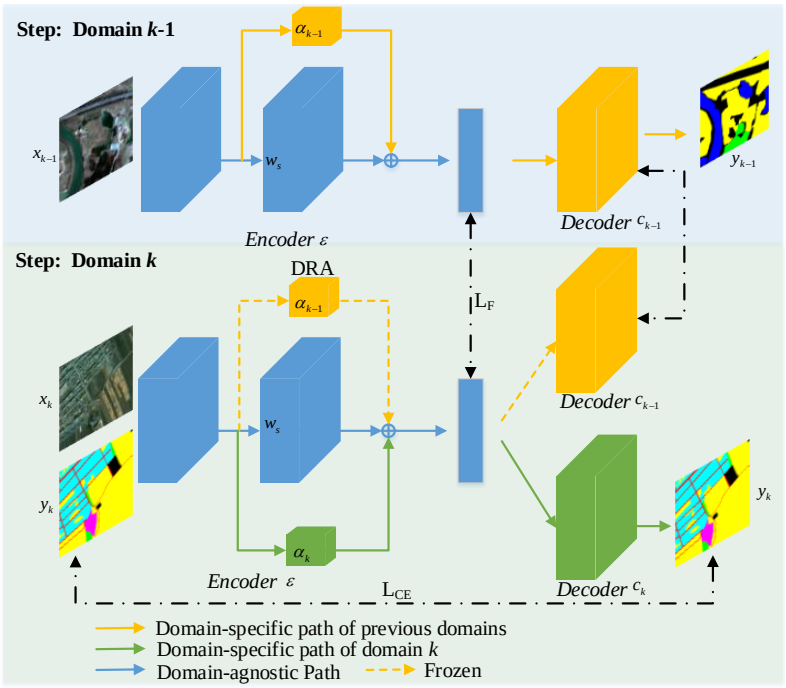

Figure 4. Our proposed approach is composed of a shared encoder and domain-specific decoders. The encoder is made up of several domain residual adapter (DRA) modules, as illustrated in detail in Figure 5. At each step, denoted by k, the domain-specific paths for the previous domains are frozen, as indicated by the yellow dotted line in the figure. The model then trains on the domain-agnostic path (in blue) and the current domain-specific path (in green).

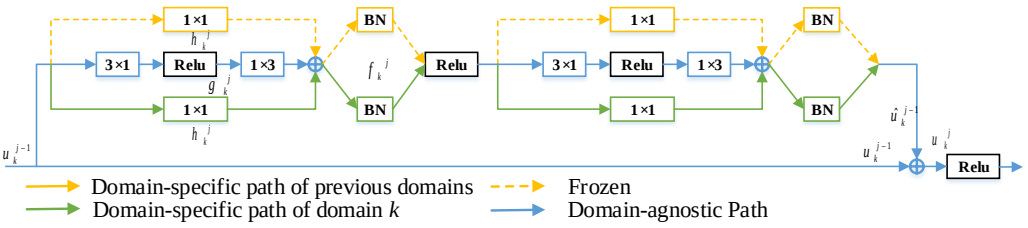

Figure 5. The detailed structure of the domain residual adapter module (DRA). Parameters of the domain-specific and domain-agnostic part in DRA are shown in different colors.

3.4. Loss

As mentioned above, the domain shift is the primary challenge when the model adapts to continual domains. Additionally, the label space shift between these domains is another factor that needs to be considered. Recent studies [26,27] of incremental learning for semantic segmentation have focused on the semantic shift of the background class, where the old classes of the previous step are divided into the background class at each step, and all classes belong to the same domain in their setting. Although there is no domain shift in this research, the methods that adapt knowledge distillation strategies to solve the semantic shift are worth referencing, as they are a common strategy to transfer knowledge

from the old model into the new one. However, in the DILRS setting, a naive application of previous knowledge distillation loss functions would not suffice.

Considering the fact that domain shift coexists with label space shift in DILRS, we revisit the classical knowledge distillation and optimization strategies by introducing a multi-level class-specific knowledge distillation function and different optimization for domain-specific and domain-agnostic parameters.

(1) Class-specific knowledge distillation function. To optimize the domain-agnostic parameters' cross-domain sequence, we adopt the knowledge distillation loss. At step k, we initialize the domain-agnostic parameters from W_s of M_{k-1} and then distill the predictions of the new and old models' output. Considering the fact that we only have data of domain $\mathcal{D}_k = (\mathbf{x}_k, \mathbf{y}_k)$, we input x_k into both the current model M_k and the previous model M_{k-1} for the previous task $i, 0 < i < k$:

$$\begin{aligned} q_i^{new} &= M_k(x_k, k-1; W_s, W_{k-1}), \\ q_i^{old} &= M_{k-1}(x_k, k-1; W_s, W_{k-1}), \\ L_D &= \sum_{i=1}^{k-1} J\left(q_i^{new}, q_i^{old}, \mu_k\right), \end{aligned} \tag{4}$$

where q_i^{new} and q_i^{old} represent the output probabilities of the current model M_k and the previous model M_{k-1}, respectively. We define the class-specific knowledge distillation loss as J, which is computed over all previous tasks. The detailed diagram of J can be seen in Figure 6. We design the class-specific knowledge distillation strategy on both spatial and channel dimensions to further mitigate the influence of the label space shift. Since we use the current domain $\mathcal{D}k$ to replace all previous domains $\sum i = 1^{k-1} D_i$, we empirically found that the data shift in the background class and non-overlapping classes leads to worse distillation results. Hence, we only distill knowledge in the overlapping classes between the current domain D_k and the previous domain $D_i, 0 < i < k$ separately. Specifically, we define the class-specific knowledge distillation loss as follows:

$$J(q_i^{new}, q_i^{old}, \mu_k) = -\frac{1}{N_k} \sum_{x_k \in \mathcal{D}_k} \sum_{c=1}^{C_\mu} \mu_k q_i^{old} \log(q_i^{new}). \tag{5}$$

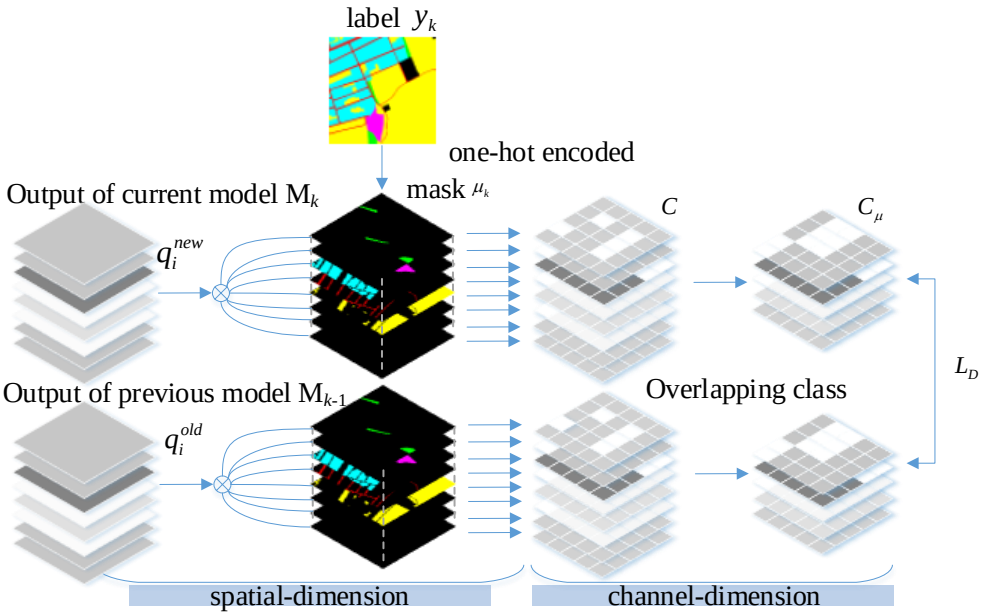

Figure 6. The detailed diagram of class-specific knowledge distillation loss J which can be divided into the spatial and channel dimension parts.

Here, $\mu_k \in \{0,1\}^{C\times H\times W}$ denotes the pixel-wise elements of a binary mask, getting from one-hot encoded label y_k. C_μ represents the overlapping classes between the current domain D_k and the previous domain $D_i, 0 < i < k$, respectively. N_k is the set domain D_k of all pixels contributing to the loss.

(2) Knowledge distillation function in feature space. Our model can be decomposed by one shared encoder $\mathcal{E}$ and multi-head decoder $\mathcal{C}_k$. As mentioned above, the parameters of the decoder $\mathcal{C}_k$ are separate domain-specific parameters, while the parameters of the encoder mix domain-specific parameters α_k and domain-agnostic parameters $\mathcal{W}_s$. Inspired by [25,38], we try to preserve knowledge by keeping the encoder $\mathcal{E}$ of model M_k and M_{k-1} having a similar representation capability at feature space. We compute this by

$$\begin{aligned} p_i^{new} &= \mathcal{E}(x_k, k-1; W_s, \alpha_{k-1}), \\ p_i^{old} &= \mathcal{E}(x_k, k-1; W_s, \alpha_{k-1}), \\ L_F &= -\frac{1}{N_k}\sum_{i=1}^{k-1}\sum_{x_k\in\mathcal{D}_k} \|p_i^{new} - p_i^{old}\|^2 \end{aligned} \tag{6}$$

where p_i^{new} and p_i^{old} represent the intermediate features of the current model M_k and the previous model M_{k-1} before the decoding stage, and $\|\cdot\|$ denotes $L2-norm$.

(3) Overall loss. Additionally, during the incremental step k, we used cross-entropy loss to train model M_k, defined by

$$L_{CE} = -\frac{1}{N_k}\sum_{x_k\in\mathcal{D}_k} \psi_k(y_k, \overline{y}_k), \tag{7}$$

where $\overline{y}_k$ denotes the prediction output of M_k. ψ_k is the SoftMax cross-entropy loss, and we set the class weights of each category to better solve the category imbalance problem.

The total loss is defined as a weighted sum of these three kinds of loss

$$L_{total} = \lambda_1 \cdot L_{CE} + \lambda_2 \cdot L_D + \lambda_3 \cdot L_F, \tag{8}$$

where λ_1, λ_2 and λ_3 are the weights of cross-entropy loss L_{CE}, class-specific knowledge distillation function L_D, and knowledge distillation function in feature space L_F. The details of the parameter setting λ_1, λ_2 and λ_3 can be seen in Section 4.5.

(4) Optimization strategy. At step k of our work, we take a different optimization strategy on domain-specific parameters $\mathcal{W}_k = \{\alpha_k, C_k\}$ and domain-agnostic parameters $\mathcal{W}_s$. We initialize the $\mathcal{W}_k$ based on $\mathcal{W}_{k-1}$, while the output classification layer is randomly initialized considering the label space shift between y_k and y_{k-1}. Additionally, all previous domain-specific parameters $\sum_{i=1}^{k-1} W_i$ are frozen at step k. Similarly, domain-agnostic parameters $\mathcal{W}_s$ initialize the model M_{k-1}, which are shared with all domains. Knowledge distillation on feature space and class-specific output prediction makes the current model preserve previous domain knowledge by domain-agnostic weights $\mathcal{W}_s$ as much as possible. In contrast, the cross-entropy loss trained on each domain improves the domain-specific performance. In addition, the domain-specific paths corresponding to the multi-domain intertwine with domain-agnostic parameters. For each evaluation of each domain, only the corresponding domain-specific path within domain-agnostic parameters is activated in the forward pass. Overall, the process of training can be described as fellows Algorithm 1.

Algorithm 1: Process of learning a new domain of kth step in DILRS

Require:
D_k: new domain (dataset) of current step k
M_{k-1}: model of previous step $k-1$
Initialization:$M_k \leftarrow$ add new domain-specific structure to M_{k-1}
$initW_k$: W_k in $M_k \leftarrow W_{k-1}$ in M_{k-1}
Freeze: domain-specific weights of all previous domains: $\sum_{i=1}^{k-1} W_i$
for epochs **do**
  Forward pass $M_k(x_k, k)$ via W_k
  Compute cross-entropy loss L_CE for D_k
  Forward pass $M_k(x_k, k-1)$ via W_{k-1}
  Forward pass $M_{k-1}(x_k, k-1)$ via W_{k-1}
  Compute knowledge distillation loss L_D, L_F
  Compute loss L_{total}
  Update: M_k and M_s at learning rate lr
end for

4. Experiments

In this section, we first introduce the overview of the datasets, and then we show the implementation details and evaluation metrics. Finally, we briefly introduce the compared methods and show the experimental results from different perspectives in detail, respectively.

4.1. Datasets

In the remote sensing field, there is currently no clear definition of what constitutes a 'domain', despite extensive research on domain adaptation [36]. For the purposes of our research, we have chosen several representative datasets [13–16] to form an experimental domain sequence. Table 1 presents the statistics of our chosen domain sequence, with all datasets resized to 256 × 256. The domains are incrementally ordered based on an increasing sensor resolution (D1–D5), and include a range of satellite (GF-2, GF-1/6, WorldView-2) and airborne sensors, covering various complex scenarios in different regions and countries. Given the regional diversity, we further divided the rural and urban areas of LoveDA [16] into two independent domains. Additionally, our domain sequence has non-overlapping categories, which are detailed in Table 1. The class distribution is shown in Figure 7, which highlights the differences and challenges posed by the domain incremental learning setting. Our experimental setting involves simultaneous domain shift and label space shift. We will make our dataset available on our website at http://complex.ustc.edu.cn/, accessed on 11 May 2023.

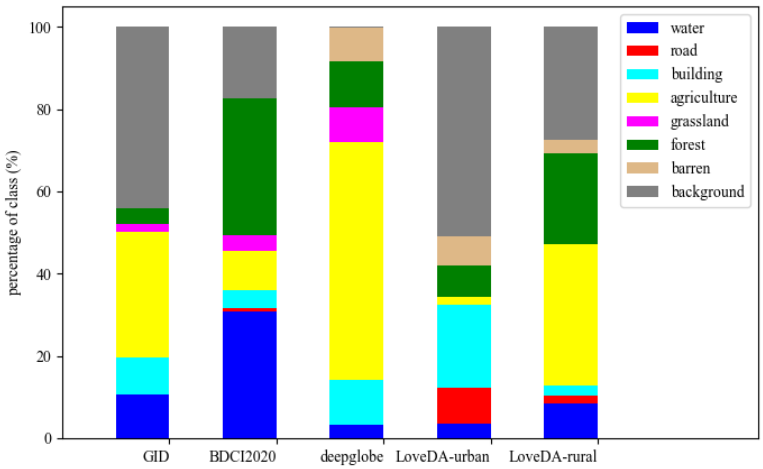

Figure 7. The class distribution of our datasets.

Table 1. Comparison and statistics among datasets.

Dataset	Sensor	Resolution (m)	Image Width	Images	Classes
GID [13]	GF-2	4	256	109,201	6 (Water/Building/Agricultural/Forest/Grassland/Background)
BDCI2020 [14]	GF-1/6	2	256	145,982	7 (Water/Building/Agricultural/Forest/Grassland/Road/Background)
deepglobe [15]	WorldView-2	0.5	256	65,044	7 (Water/Building/Agricultural/Forest/Grassland/Barren/Background)
LoveDA-urban [16]	Airborne	0.3	256	29,328	7 (Water/Building/Agricultural/Forest/Road/Barren/Background)
LoveDA-rural [16]	Airborne	0.3	256	37,728	7 (Water/Building/Agricultural/Forest/Road/Barren/Background)

4.2. Implementation Details

We utilize the Erfnet [5] as our model's backbone, with the integration of our domain's residual adapter module. Specifically, the encoder embeds with our domain residual adapter module, while each head of the multi-head decoder seamlessly follows the Erfnet structure. We use the Adam optimizer and set the batch size to 36. In light of the data imbalance in LoveDA compared to the other datasets, we adopted data augmentation strategies such as random horizontal flip and rotation for LoveDA-rural and LoveDA-urban.

Considering the stability-plasticity trade-off problem in incremental learning, we use a different learning rate for domain-specific $\mathcal{W}_k$ and domain-agnostic parameters $\mathcal{W}_s$. In particular, $\mathcal{W}_k$ is related to the plasticity of a new domain, while $\mathcal{W}_s$ is close to the stability of previous domains. There is an imbalance between the representation learning on a new domain and representation maintenance on a previous domain when using the same learning rate for $\mathcal{W}_k$ and $\mathcal{W}_s$. Experiments indicate that $\frac{LR_{W_k}}{LR_{W_f}}$ valued as 100 obtains a good stability–plasticity trade-off. The learning rate of the $\mathcal{W}_k$ and $\mathcal{W}_s$ are set to 5×10^{-4} and 5×10^{-6}, respectively. Our experiments are implemented using Pytorch, and we use an NVIDIA Tesla V100 GPU.

In accordance with the evaluation metrics used in studies such as [19,23,34,43], we utilize Δ_m and BWT to evaluate the performance of our incremental learning model, DILRS. Specifically, Δ_m measures the average performance degradation compared to the single-task baseline b:

$$\Delta_m = \frac{1}{T}\sum_{t=1}^{T}\frac{mIoU_{T,t} - mIoU_{b,t}}{mIoU_{b,t}}. \tag{9}$$

Here, the mean intersection over union $mIoU_{T,t}$ denotes the evaluation accuracy of the incremental-learning segmentation model on task t, which can also be considered a domain. On the other hand, $mIoU_{b,t}$ represents the evaluation accuracy of the single-task baseline for task t, and T denotes the total number of tasks. If $\Delta_m < 0$, it indicates that the performance of the incremental-learning task is worse than the single-task baseline for each domain, while $\Delta_m > 0$ indicates a better performance.

The backward transfer BWT metric measures the forgetting of the model on old tasks after training on a new task, which is particularly relevant for evaluating incremental learning methods. Theoretically, if BWT < 0, this means that the model learning a new task will improve the performance of the model on the previous tasks. In contrast, if BWT > 0, it denotes that the performance of previous tasks decreases when learning a new task, which is also known as catastrophic forgetting.

$$\text{BWT} = \frac{1}{T-1}\sum_{t=1}^{T-1}(mIoU_{t,t} - mIoU_{T,t}), \tag{10}$$

where $mIoU_{T,t}$ represents the accuracy on task t after learning task T.

4.3. Compared Methods

When training multiple domains, several choices of training paradigms are optional, including disjoint, joint-training, and fine-tuning. We compare our proposed methods

with these baseline methods. Additionally, the three latest incremental learning methods are compared, including EwC [29], LwF [12], and ILTSS [25]. Note that all the compared methods are based on the Erfnet model.

(1) Benchmark Methods for Comparison. Disjoint (single-task) trains a separate model independently for each domain, which aligns with the i.i.d. (independent identically distributed) assumption of model training. Joint-training (multi-task) trains a unified model by combining all domain data. Both disjoint and joint-training belong to the offline setting, and we record results when training until convergence on each new task. We use a multi-head decoder to train all accumulated datasets. Fine-tuning (FT) trains each domain incrementally by fine-tuning the pre-trained model on the new domain until convergence. Fine-tuning is a standard baseline in incremental learning, while nothing has been implemented to avoid forgetting. We used a single-head decoder in our experimental setting. Feature extraction (FE) freezes the weights of previous domains, which theoretically preserves the model's performance over the previous domains as much as possible. We fix all encoder weights and only train the new domain's decoder weights.

(2) Latest Incremental Learning Methods. There are only a few methods explicitly designed for domain-incremental learning. Given the similarity between our experimental setting and incremental learning for semantic segmentation, we compare our method with the three latest incremental learning methods, namely EwC [29], LwF [12], and ILTSS [25]. Since the data of the previous domain are not available in our setting, we do not consider replay-based incremental learning as a compared method. EWC [29] ranks the weights of the old task and then optimizes them differently depending on their importance. It does this by introducing the Fisher information matrix based on the analysis of sequential Bayesian estimation. LwF [12] was the first method to use knowledge distillation to prevent catastrophic forgetting when learning new tasks or classes. On the other hand, ILTSS [25] extends the knowledge distillation strategy in segmentation by adopting a feature-space loss.

4.4. Experimental Results

(1) Multi-source domain-incremental learning scenario. As mentioned previously, our incremental learning approach follows the domain sequence (D1–D5): GID [13]; BDCI2020 [14]; deepglobe [15]; LoveDA-urban [16]; and LoveDA-rural [16]. We begin by training a model on GID in step 1 and then incrementally learning the same model on BDCI2020 in step 2, followed by the remaining domains in steps 3–5. Table 2 presents the performance evolution of different methods in the domain-incremental learning setting, with all domain test results recorded when the model achieves its best results on the new domain. Our results show that our method outperforms multi-task learning. While multi-task training can access all training data and is typically considered an upper bound, the differences in data distribution among the multiple domains in our experiment may affect each task's ability to achieve optimal results. As anticipated, fine-tuning (FT) performs poorly on the previous domains, indicating that the model entirely forgets the knowledge of old domains. Feature extraction (FE) cannot achieve satisfactory results in the new domain, demonstrating that freezing the encoder weights leads to the lowest plasticity in the new domain. Moreover, the performance of FE in previous domains was ideal, which we consider as reference results. Compared with FT, LwF [12] and ILTSS [25] only have a slight effect on relieving catastrophic forgetting, while LwF (single-head) yields even worse results than FT. Additionally, the performance of the multi-head decoder setting is slightly better than that of the single-head decoder. Similarly, inevitable catastrophic forgetting exists in the EwC [29] method.

Table 2. Results of the 5-step domain-incremental learning. We record performance (IoU) on current and all previous domains at each step. Parentheses indicate the drop in performance compared with the domain's first trained step. Δ_m and BWT are calculated from the results at step 5. ↑ indicates that larger is better, while ↓ is the opposite. Convention: **best**.

DIL Step	**Step1**	**Step2**		**Step3**			**Step4**			
Methods	**GID**	**GID**	**BDCI**	**GID**	**BDCI**	**Deepglobe**	**GID**	**BDCI**	**Deepglobe**	**Urban**
Single-task	0.6190	0.6190	0.6332	0.6190	0.6332	0.6226	0.6190	0.6332	0.6226	0.4312
Multi-task	0.6190	0.5526 (−0.0664)	0.5666	0.4949 (−0.1241)	0.5147 (−0.0520)	0.5655	0.4194 (−0.1996)	0.4084 (−0.1582)	0.3722 (−0.1933)	0.2674
FT (single-head)	0.6190	0.2512 (−0.3678)	0.6365	0.1315 (−0.4875)	0.2359 (−0.4006)	0.6147	0.1555 (−0.4635)	0.0901 (−0.5464)	0.1002 (−0.5145)	0.4126
FE (multi-head)	0.6190	0.6190 (ref)	0.3034	0.6190 (ref)	0.3034 (ref)	0.1301	0.6190 (ref)	0.3034 (ref)	0.1301 (ref)	0.2674
EwC (single-head) [29]	0.6190	0.2518 (−0.3672)	**0.6408**	0.1736 (−0.4454)	0.3034 (−0.3374)	0.5172	0.1776 (−0.4414)	0.1533 (−0.4875)	0.1301 (−0.3871)	0.3553
LwF (single-head) [12]	0.6443	0.4954 (−0.1489)	0.5944	0.2863 (−0.3580)	0.3127 (−0.2817)	0.5827	0.2184 (−0.4258)	0.1438 (−0.4506)	0.1425 (−0.4402)	0.3914
LwF (multi-head) [12]	0.6532	0.2538 (−0.3994)	0.6074	0.1376 (−0.5156)	0.1994 (−0.408)	**0.6362**	0.1898 (−0.4634)	0.0905 (−0.5169)	0.0582 (−0.5779)	0.4345
ILTSS (single-head) [25]	0.6532	0.2629 (−0.3903)	0.5954	0.1531 (−0.5001)	0.2067 (−0.3887)	0.5902	0.1663 (−0.4869)	0.1228 (−0.4726)	0.1273 (−0.4629)	0.4113
ILTSS (multi-head) [25]	0.6443	0.4347 (−0.2096)	0.6217	0.2954 (−0.3489)	0.3717 (−0.2499)	0.6289	0.2213 (−0.4230)	0.2625 (−0.3592)	0.2331 (−0.3959)	**0.4307**
Ours	0.6532	**0.6510 (−0.0022)**	0.6064	**0.6245 (−0.0287)**	**0.5622 (−0.0442)**	0.6046	**0.5530 (−0.1002)**	**0.5694 (−0.0370)**	**0.5398 (−0.0648)**	0.4306

DIL step			Step5				
Methods	GID	BDCI	deepglobe	urban	rural	Δ_m (%) ↑	**BWT** (%) ↓
Single-task	0.6190	0.6332	0.6226	0.4312	0.5467	-	-
Multi-task	0.4052 (−0.2138)	0.3896 (−0.1770)	0.4183 (−0.1472)	0.2628 (−0.0046)	0.4527	−32.41	13.57
FT (single-head)	0.1560 (−0.4630)	0.2069 (−0.4296)	0.2404 (−0.3743)	0.3845 (−0.0281)	0.5701	−42.01	32.37
FE (multi-head)	0.6190 (ref)	0.3034 (ref)	0.1301 (ref)	0.2674 (ref)	0.3102	-	-
EwC (single-head) [29]	0.2169 (−0.4021)	0.3125 (−0.3283)	0.2676 (−0.2497)	0.3123 (−0.0430)	0.5101	−41.38	25.57
LwF (single-head) [12]	0.1861 (−0.4582)	0.1181 (−0.4763)	0.1295 (−0.4532)	0.3282 (−0.0632)	0.5539	−50.61	36.27
LwF (multi-head) [12]	0.1887 (−0.4645)	0.1980 (−0.4094)	0.2463 (−0.3899)	0.3678 (−0.0667)	**0.6543**	−38.74	33.26
ILTSS (single-head) [25]	0.1809 (−0.4723)	0.1881 (−0.4073)	0.2495 (−0.3407)	0.3768 (−0.0345)	0.6128	−40.30	31.37
ILTSS (multi-head) [25]	0.2095 (−0.4348)	0.2207 (−0.4010)	0.2059 (−0.4230)	**0.4272 (−0.0035)**	0.6777	−35.04	31.55
Ours	**0.5601 (−0.0931)**	**0.5507 (−0.0558)**	**0.5248 (−0.0798)**	0.4180 (−0.0127)	0.6233	**−5.46**	**6.03**

Compared with the above methods, our method achieves significantly better results in both old and new domains. As highlighted in bold in Table 2, our approach shows the minimum degradation of mIoU at each step, with good performance concurrently in the current new domain. Additionally, our method achieves Δ_m of -5.46% and BWT of 6.03%, demonstrating plasticity and stability.

(2) Single-source domain-incremental learning scenario. In addition to discussing domain-incremental learning in multi-source scenarios, we also considered the single-source scenario where the rural and urban domains come from the same aerial dataset LoveDA [16]. These domains share the same semantic categories and sensor resolution but exhibit a domain shift. Furthermore, the t-SNE visualization results of these five domains in the feature space, as shown in Figure 8, suggest that the domains are independently identically distributed, with rural and urban having a more similar distribution. Based on this observation, we conduct experiments in which the model incrementally learns from rural to urban and from urban to rural. The results, as shown in Table 3, are compared with the performance of rural and urban in the single-task setting. It is worth noting that the two parentheses in step 2 indicate different meanings. Specifically, the left one shows the drop/gain in performance concerning step 1, while the right one compares the performance with the single-task baseline for the corresponding dataset. Our method performs well in the single-source domain-incremental learning scenarios, with little catastrophic forgetting. Furthermore, the performance in step 2 surpasses that of the corresponding dataset in step 1, and we observe a gain of 14.78% concerning the single-task baseline for the rural and urban. Our experiments suggest that our model achieves forward transfer from the previous domain by capturing and adapting domain-agnostic and domain-specific features between the rural and urban domains.

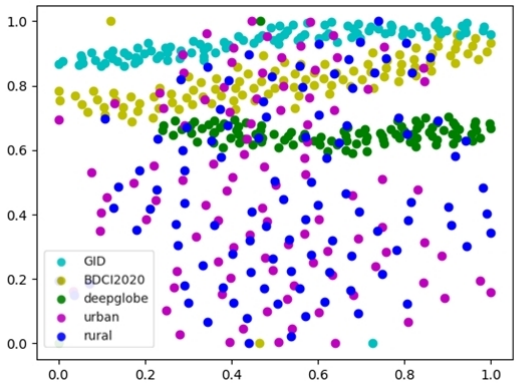

Figure 8. The t-stochastic neighbor embedding (t-SNE) visualization results of the features of domains 1–5 (GID, BDCI2020, deepglobe, LoveDA-rural, LoveDA-urban).

Table 3. Results obtained on a single-source domain-incremental learning scenario: rural → urban and urban → rural. The left parenthesis in step 2 indicates a drop/gain in performance concerning step 1, while the right one compares with a single-task baseline for the corresponding dataset.

DIL Step	Single-Task	Single-Task	Step1	Step2: Rural → Urban		Step1	Step2: Urban → Rural	
IoU per Category	Rural	Urban	Rural	Rural	Urban	Urban	Urban	Rural
mIoU	0.4312	0.5467	0.4236	0.6091 (0.1855)	0.5790 (0.1478)	0.5280	0.5388 (0.0108)	0.6945 (0.1477)
Water	0.5569	0.6278	0.5054	0.7677 (0.2623)	0.6976 (0.1407)	0.6175	0.5866 (−0.0309)	0.7517 (0.1239)
Road	0.4947	0.4771	0.3039	0.5240 (0.2201)	0.5375 (0.0428)	0.5457	0.5421 (−0.0036)	0.5763 (0.0992)
Building	0.5056	0.4667	0.3278	0.5511 (0.2233)	0.5948 (0.0892)	0.5637	0.5535 (−0.0102)	0.6241 (0.1574)
Agriculture	0.3722	0.7042	0.4491	0.4959 (0.0468)	0.7006 (0.3284)	0.3332	0.3051 (−0.0281)	0.8266 (0.1224)
Forest	0.4026	0.7480	0.1710	0.4471 (0.2761)	0.6756 (0.2730)	0.3662	0.4360 (0.0698)	0.8332 (0.0852)
Barren	0.3434	0.3815	0.2444	0.4507 (0.2063)	0.4702 (0.1268)	0.3864	0.3752 (−0.0112)	0.6386 (0.2571)

(3) Class-wise qualitative analysis. In this section, we delve into the class-wise accuracy of previous domains during domain-incremental learning. Figure 9 presents the

comparative results of our method with a single-task and ILTSS (multi-head) [25], where we recorded the test results of domains 1–4 (GID, BDCI2020, deepglobe, LoveDA-rural) at step 5. Additionally, all test results are noted when the model is optimal in the current domain (LoveDA-rural). As shown in Figure 9, the ILTSS method suffers heavy catastrophic forgetting and fails to maintain performance in the last domain (LoveDA-urban). The model almost forgets all the knowledge of the previous domain (GID and BDCI2020), particularly in classes such as building, agriculture, and grassland in domain GID and water, road, and grassland in domain BDCI2020. On the other hand, our method successfully mitigates forgetting in all previous domains, and there is a considerable gap between our performance and ILTSS in the three domains (GID, BDCI2020, and deepglobe), of 41.80%, 31.11%, and 31.14%, respectively. It is worth noting that our method's performance differs slightly from the single-task method, performing even slightly better in some categories. Moreover, to evaluate the current domain LoveDA-rural, we observe that our method and ILTSS (multi-head) outperformed single-task performance by 7.55–13.32%. Considering the similarity between the rural and urban areas of LoveDA, we attribute the knowledge of the previous domain to getting forward transfer to the current domain. While ILTSS performs better than our approach in the current domain.

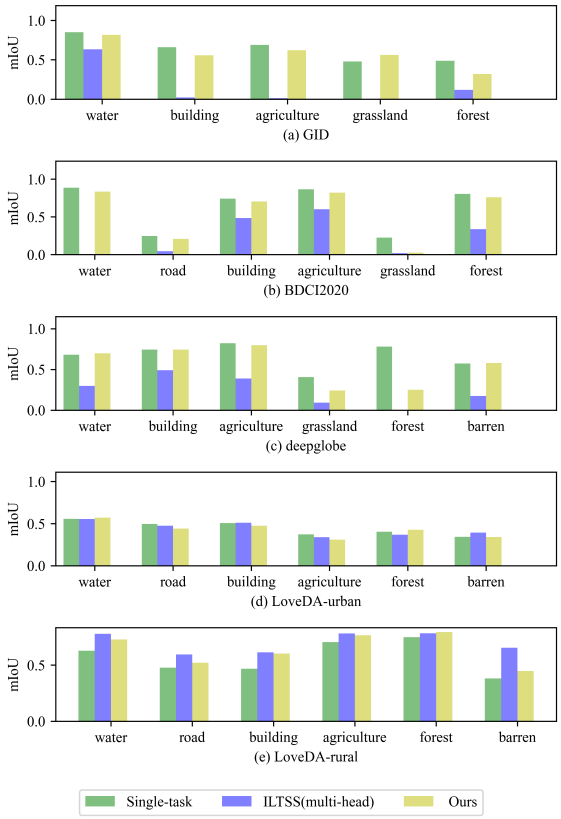

Figure 9. The test class-wise accuracy of domains 1–5 (GID, BDCI2020, deepglobe, LoveDA-rural, LoveDA-urban) after model optimally learns on domain5 (LoveDA-urban) at step 5.

LoveDA-rural highlights the plasticity–stability trade-off problem in incremental learning. This trade-off refers to the need to compromise between learning a new domain while also preserving the knowledge acquired from previously learned domains.

(4) Visualization analysis. Figure 10 presents the visualization results of representative samples obtained by our method and the comparative methods on the experimental domains. These datasets are uniformly cropped as 256 × 256, and some regions are cut into small pieces, making the segmentation difficult due to the lack of contextual semantic understanding. As shown in the first three domains (six rows) of Figure 10b,c, much noise is introduced, and significant misclassification is present. Moreover, the model loses the

ability to classify some categories in the previous domain, such as the 'road' category in domain BDCI2020 of the ILTSS method, while misclassifying categories that did not exist before, such as the 'water' category in domain BDCI2020 of the LwF method, indicating catastrophic forgetting in both methods. In contrast, the results of the last domain (urban) and the current domain (rural) show better performance in Figure 10b,c.

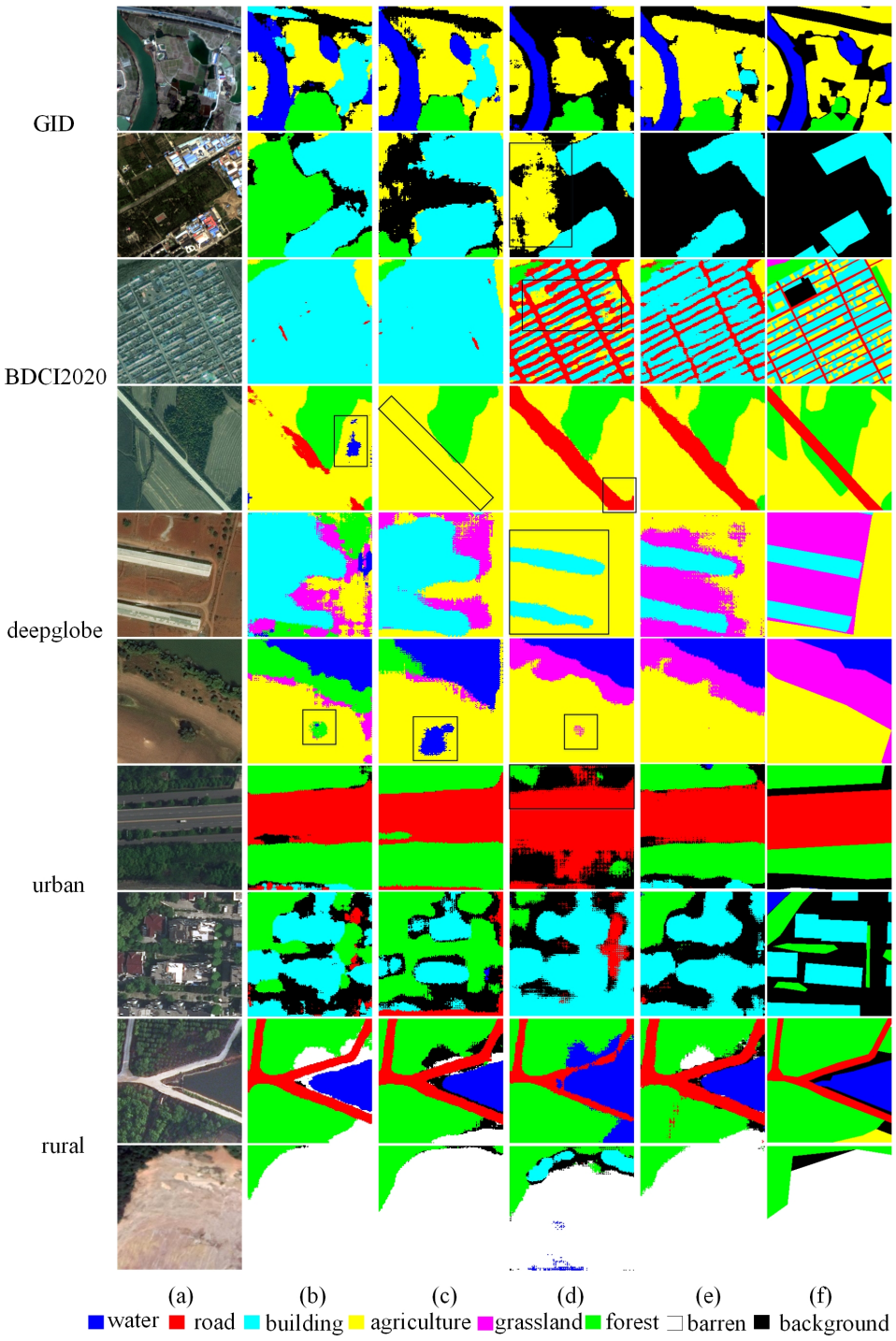

Figure 10. Visualization of semantic segmentation results in five domain as step 5: each domain displays in two rows according to the training order (GID–BDCI2020–deepglobe–urban–rural). Each line: (**a**) Input image; (**b**) LwF (multi-head); (**c**) ILTSS (multi-head); (**d**) multi-task; (**e**) Ours; (**f**) Ground truth. The black bounding boxes highlight the details in the images. The color corresponding to each category is shown at the bottom.

Furthermore, we discuss the performance of multi-task training in Figure 10d. Although multi-task training can access all domain data, the model optimizes the training data from all domain sequences simultaneously, leading to suboptimal performance on all domains. The performance of multi-task training in the rural and urban domains is worse than the other three domains, likely due to the smaller amount of data and the feature space gap, as shown in Figure 8. Although multi-task training achieves a more precise prediction in some categories, such as the 'agriculture' category embedded in the 'road' and 'building' of BDCI2020 in Figure 10d, it still confuses misclassification in the 'grassland' category of deepglobe, reflecting its performance instability in different samples.

In comparison, our method significantly improves the performances in these five domains, as shown in Figure 10e, with minor catastrophic forgetting even in the previous domain. Additionally, our method can correctly classify the 'grassland', 'forest', and 'agriculture' categories, as they are similar in appearance, such as the bounding box region in the sixth row, thanks to our understanding of contextual semantic knowledge. The model's classification ability is susceptible to multi-source data, especially when it already has knowledge of the previous domain, posing a challenge for utilizing different domain knowledge, while our result benefits from the domain-specific structure.

4.5. Ablation Study

(1) Loss analysis. The proposed method comprises two key components: the DILRS architecture and the multiple loss function. In this section, we present ablation experiments to evaluate the effectiveness of the proposed loss function and architecture, as depicted in Table 4. In all experiments, we used the proposed DILRS architecture, except for the 'ours' entry in Table 2, which is based on Erfnet [5]. We report the optimal results for each experimental setting, with varying weights for different loss functions. Ablation experiment 1 is conducted to examine the performance of only using the cross-entropy loss L_{CE} for training, which is similar to FT (multi-head, in Table 2) for incremental training. In contrast, our DILRS model yields better results, which we attribute to the utilization of domain-specific and domain-agnostic structures. By comparing ablation experiment 1 with experiments 2, 3, and 4, we evaluate the impact of different types of distillation loss. The results show that our proposed class-specific loss L_D outperforms the others, as both Δ_m and BWT are improved. Additionally, we combine the class-specific loss L_D with distillation loss at the feature space, as shown in ablation experiment 5. As mentioned above, using these two losses jointly maximizes the distillation of previous domain knowledge while minimizing the effects of the label space shift. The excellent performance demonstrated in ablation experiment 5 is also utilized in our method.

Table 4. Results of the ablation study for different loss functions. Δ_m and BWT are calculated based on our proposed model at step 5. L_{CE}, L_D, and L_F are the loss function in our proposed method, while L_{dist} represents the classical distillation loss proposed by [44].

	L_{CE}	L_{dist}	L_D	L_F	Δ_m (%) ↑	BWT (%) ↓
1	✓				−33.12	26.46
2	✓	✓			−15.25	15.21
3	✓		✓		−9.24	11.44
4	✓			✓	−10.01	14.04
5	✓		✓	✓	−5.46	6.03

Moreover, we investigate the influence of the weights of loss λ_1, λ_2, and λ_3 in (8), which is also a critical factor to balance plasticity and stability. In our experiments, we set $\lambda_2 = \lambda_3$ to simplify research, and $\frac{\lambda_2}{\lambda_1}$ represents the ratio of distillation to cross-entropy loss. The results of varying the ratio $\frac{\lambda_2}{\lambda_1}$ are shown in Table 5. As introduced in Section 3.2, Δ_m indicates the performance of each domain in domain-incremental models, while BWT measures the ability to retain old knowledge. Ideally, as the $\frac{\lambda_2}{\lambda_1}$ increases, the model should

focus more on retaining old knowledge. Thus, the value of Δ_m should gradually decrease, and the value of BWT should decrease accordingly. The results show that the performance conforms to this law only in a specific range, and $\frac{\lambda_2}{\lambda_1} = 1$ achieves better results compared to the other four parameter settings.

Table 5. Results of an ablation study for the weight ratio $\frac{\lambda_2}{\lambda_1}$.

$\frac{\lambda_2}{\lambda_1}$	Δ_m (%) ↑	BWT (%) ↓
0.1	−7.45	8.48
1	−5.46	6.03
10	−27.75	24.19
100	−32.41	13.57

(2) Parameters and FLOP analysis. As discussed in [19] and the related work, parameter isolation-based methods are the most suitable option for incremental learning in the remote sensing field when compared with replay-based and regularization-based methods. Our method also belongs to the parameter isolation-based method. However, as the model expands to different domains, the increased number of parameters will inevitably burden the application. Therefore, it is necessary to analyze the evolution of parameters and FLOPs in the domain-incremental learning setting, as these reflect the model's space and time computational complexity. In Figure 11, we present the growth in the number of parameters and floating point operations (FLOPs) of the single-task baseline and our method with incremental domains. It can be observed that our method exhibits a 21.09% growth in parameters, while the FLOPs remain constant. In contrast, although the single-task model has fewer parameters and FLOPs at domain 1, the growth rate as incremental domains are added is tremendous.

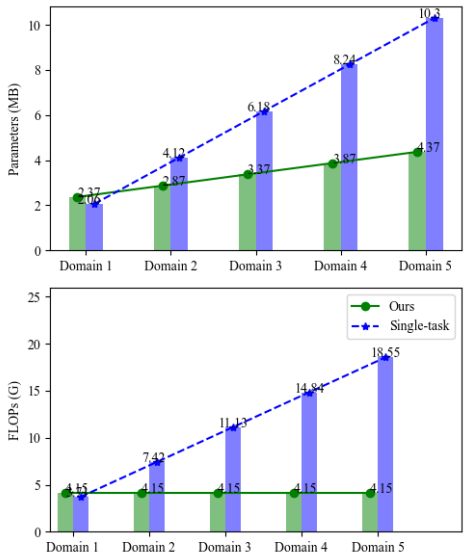

Figure 11. Parameters and FLOP growth with the incremental domain.

5. Conclusions

In this paper, we investigate the domain-incremental learning challenge in the context of remote sensing, where the model needs to incrementally learn new out-of-domain distribution data. Catastrophic forgetting caused by the coexistence of a domain shift and label space shift has limited the performance of previous works in this area. To tackle this issue, we propose a model that utilizes domain adapter modules to reparametrize domain-agnostic and domain-specific parameters as well as introduce a novel multi-level knowledge distillation loss. Our experimental results demonstrate that our approach out-

performs existing methods for both multi-source and single-source remote sensing domains. Additionally, class-wise qualitative analysis and visualization support the superiority of our method.

As the deployment of deep learning models on edge devices gains importance in earth intelligence interpretation, developing domain-incremental learning methods that are suitable for remote sensing multi-source data becomes essential. Currently, there are few relevant studies, and our dataset and experimental settings can serve as a benchmark in the future. However, our research still has some limitations, as data collected by the same satellite in different seasons and regions can be considered different domains, which better align with the actual deployment of remote sensing edge devices. Due to limited data, this setting was not followed in this study. We will continue to improve our method in future studies.

Author Contributions: Methodology, X.R. and Z.L. (Ziqiang Li); Experiments, X.R.; Data analysis, X.R.; Writing—original draft preparation, X.R.; Writing—review and editing, Z.L., Y.C. and Z.L. (Ziyang Li); Supervision, W.S., Y.C. and Z.L. (Ziyang Li). All authors have read and agreed to the published version of the manuscript.

Funding: The work was supported by the National Key R&D Program of China (2021YFC3000300, 2021YFC3000305).

Data Availability Statement: Not applicable.

Conflicts of Interest: We state that we do not have known competing financial interests or personal relationships which may influence the work in this paper.

Abbreviations

The following abbreviations are used in this manuscript:

DRA	Domain residual adapter
DILRS	Domain-incremental learning for remote sensing
FT	Fine-tuning
FE	Feature extraction

References

1. Sun, X.; Liang, W.; Diao, W.; Cao, Z.Y.; Feng, Y.C.; Wang, B.; Fu, K. Progress and challenges of remote sensing edge intelligence technology. *J. Image Graph.* **2020**, *25*, 1719–1738. [CrossRef]
2. Gan, Y.; Pan, M.; Zhang, R.; Ling, Z.; Zhao, L.; Liu, J.; Zhang, S. Cloud-Device Collaborative Adaptation to Continual Changing Environments in the Real-world. *arXiv* **2022**, arXiv:2212.00972.
3. Wang, Q.; Fink, O.; Van Gool, L.; Dai, D. Continual test-time domain adaptation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 7201–7211.
4. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818. [CrossRef]
5. Romera, E.; Alvarez, J.M.; Bergasa, L.M.; Arroyo, R. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Trans. Intell. Transport. Syst.* **2017**, *19*, 263–272. [CrossRef]
6. Li, Z.; Xia, P.; Rui, X.; Li, B. Exploring The Effect of High-frequency Components in GANs Training. *ACM Trans. Multimed. Comput. Commun. Appl.* **2023**, *19*, 1–22. [CrossRef]
7. Li, Z.; Xia, P.; Tao, R.; Niu, H.; Li, B. A New Perspective on Stabilizing GANs Training: Direct Adversarial Training. *IEEE Trans. Emerg. Top. Comput. Intell.* **2023**, *7*, 178–189. [CrossRef]
8. Mai, Z.; Li, R.; Jeong, J.; Quispe, D.; Kim, H.; Sanner, S. Online continual learning in image classification: An empirical survey. *Neurocomputing* **2022**, *469*, 28–51. [CrossRef]
9. Van de Ven, G.M.; Tolias, A.S. Three scenarios for continual learning. *arXiv* **2019**, arXiv:1904.07734. [CrossRef]
10. Li, Z.; Wang, C.; Zheng, H.; Zhang, J.; Li, B. FakeCLR: Exploring Contrastive Learning for Solving Latent Discontinuity in Data-Efficient GANs. In Proceedings of the Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, 23–27 October 2022; Proceedings, Part XV, pp. 598–615.
11. Li, Z.; Tao, R.; Wang, J.; Li, F.; Niu, H.; Yue, M.; Li, B. Interpreting the latent space of gans via measuring decoupling. *IEEE Trans. Artif. Intell.* **2021**, *2*, 58–70. [CrossRef]
12. Li, Z.; Hoiem, D. Learning without forgetting. *IEEE Trans. Pattern Anal. Machine Intell.* **2017**, *40*, 2935–2947. [CrossRef]

13. Tong, X.Y.; Xia, G.S.; Lu, Q.; Shen, H.; Li, S.; You, S.; Zhang, L. Land-cover classification with high-resolution remote sensing images using transferable deep models. *Remote Sens. Environ.* **2020**, *237*, 111322. [CrossRef]
14. BDCI2020, C. Remote Sensing Image Segmentation Datatset. Available online: https://www.datafountain.cn/competitions/475 (accessed on 17 April 2022).
15. Demir, I.; Koperski, K.; Lindenbaum, D.; Pang, G.; Huang, J.; Basu, S.; Hughes, F.; Tuia, D.; Raskar, R. Deepglobe 2018: A challenge to parse the earth through satellite images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 172–181. [CrossRef]
16. Wang, J.; Zheng, Z.; Ma, A.; Lu, X.; Zhong, Y. LoveDA: A Remote Sensing Land-Cover Dataset for Domain Adaptive Semantic Segmentation. *arXiv* **2021**, arXiv:2110.08733. [CrossRef]
17. Li, W.H.; Liu, X.; Bilen, H. Universal Representations: A Unified Look at Multiple Task and Domain Learning. *arXiv* **2022**, arXiv:2204.02744.
18. Liu, W.; Nie, X.; Zhang, B.; Sun, X. Incremental Learning With Open-Set Recognition for Remote Sensing Image Scene Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5622916. [CrossRef]
19. Lu, X.; Sun, X.; Diao, W.; Feng, Y.; Wang, P.; Fu, K. LIL: Lightweight Incremental Learning Approach Through Feature Transfer for Remote Sensing Image Scene Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5611320. [CrossRef]
20. Feng, Y.; Sun, X.; Diao, W.; Li, J.; Gao, X.; Fu, K. Continual learning with structured inheritance for semantic segmentation in aerial imagery. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5607017. [CrossRef]
21. Rong, X.; Sun, X.; Diao, W.; Wang, P.; Yuan, Z.; Wang, H. Historical Information-Guided Class-Incremental Semantic Segmentation in Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5622618. [CrossRef]
22. Tasar, O.; Tarabalka, Y.; Alliez, P. Incremental learning for semantic segmentation of large-scale remote sensing data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 3524–3537. [CrossRef]
23. Garg, P.; Saluja, R.; Balasubramanian, V.N.; Arora, C.; Subramanian, A.; Jawahar, C. Multi-Domain Incremental Learning for Semantic Segmentation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2022; pp. 761–771. [CrossRef]
24. Rebuffi, S.A.; Bilen, H.; Vedaldi, A. Efficient parametrization of multi-domain deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8119–8127. [CrossRef]
25. Michieli, U.; Zanuttigh, P. Incremental learning techniques for semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27–28 October 2019. [CrossRef]
26. Klingner, M.; Bär, A.; Donn, P.; Fingscheidt, T. Class-incremental learning for semantic segmentation re-using neither old data nor old labels. In Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020; pp. 1–8. [CrossRef]
27. Cermelli, F.; Mancini, M.; Bulo, S.R.; Ricci, E.; Caputo, B. Modeling the background for incremental learning in semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9233–9242. [CrossRef]
28. Rebuffi, S.A.; Kolesnikov, A.; Sperl, G.; Lampert, C.H. icarl: Incremental classifier and representation learning. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2001–2010. [CrossRef]
29. Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A.A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 3521–3526. [CrossRef]
30. Rajasegaran, J.; Hayat, M.; Khan, S.; Khan, F.S.; Shao, L. Random path selection for incremental learning. *arXiv* **2019**, arXiv:1906.01120. [CrossRef]
31. Rusu, A.A.; Rabinowitz, N.C.; Desjardins, G.; Soyer, H.; Kirkpatrick, J.; Kavukcuoglu, K.; Pascanu, R.; Hadsell, R. Progressive neural networks. *arXiv* **2016**, arXiv:1606.04671. [CrossRef]
32. Mirza, M.J.; Masana, M.; Possegger, H.; Bischof, H. An Efficient Domain-Incremental Learning Approach to Drive in All Weather Conditions. *arXiv* **2022**, arXiv:2204.08817. [CrossRef]
33. Lu, Y.; Wang, M.; Deng, W. Augmented Geometric Distillation for Data-Free Incremental Person ReID. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 7329–7338. [CrossRef]
34. Gao, J.; Li, J.; Shan, H.; Qu, Y.; Wang, J.Z.; Zhang, J. Forget Less, Count Better: A Domain-Incremental Self-Distillation Learning Benchmark for Lifelong Crowd Counting. *arXiv* **2022**, arXiv:2205.03307. [CrossRef]
35. Wang, M.; Yu, D.; He, W.; Yue, P.; Liang, Z. Domain-incremental learning for fire detection in space-air-ground integrated observation network. *Int. J. Appl. Earth Obs. Geoinf.* **2023**, *118*, 103279. [CrossRef]
36. Elshamli, A.; Taylor, G.W.; Areibi, S. Multisource domain adaptation for remote sensing using deep neural networks. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 3328–3340. [CrossRef]
37. Wang, X.; Cai, Z.; Gao, D.; Vasconcelos, N. Towards universal object detection by domain attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7289–7298. [CrossRef]
38. Shan, L.; Wang, W.; Lv, K.; Luo, B. Class-incremental Learning for Semantic Segmentation in Aerial Imagery via Distillation in All Aspects. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5615712. [CrossRef]

39. Arnaudo, E.; Cermelli, F.; Tavera, A.; Rossi, C.; Caputo, B. A contrastive distillation approach for incremental semantic segmentation in aerial images. In Proceedings of the International Conference on Image Analysis and Processing, Lecce, Italy, 23–27 May 2022; pp. 742–754. [CrossRef]
40. Michieli, U.; Zanuttigh, P. Continual semantic segmentation via repulsion-attraction of sparse and disentangled latent representations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 1114–1124. [CrossRef]
41. Li, J.; Sun, X.; Diao, W.; Wang, P.; Feng, Y.; Lu, X.; Xu, G. Class-incremental learning network for small objects enhancing of semantic segmentation in aerial imagery. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5612920. [CrossRef]
42. Rebuffi, S.A.; Bilen, H.; Vedaldi, A. Learning multiple visual domains with residual adapters. *Adv. Neural Inf. Process. Syst.* **2017**, 506–516. [CrossRef]
43. Kanakis, M.; Bruggemann, D.; Saha, S.; Georgoulis, S.; Obukhov, A.; Gool, L.V. Reparameterizing convolutions for incremental multi-task learning without task interference. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 689–707. [CrossRef]
44. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531. [CrossRef]