Progressive Self-Supervised Pretraining for Hyperspectral Image Classification

Peiyan Guan, *Graduate Student Member, IEEE*, and Edmund Y. Lam, *Fellow, IEEE*

***Abstract*— Self-supervised learning has demonstrated considerable success in hyperspectral image (HSI) classification when limited labeled data are available. However, inherent dissimilarities among HSIs require self-supervised pretraining from scratch for each HSI dataset. Pretraining on a large amount of unlabeled data can be time-consuming. In addition, the poor quality of some HSIs can limit the performance of self-supervised learning (SSL) algorithms. To address these issues, we propose to enhance self-supervised pretraining on HSIs with transfer learning. We introduce a progressive self-supervised pretraining (PSP) framework that acquires strong initialization for the final pretraining on the target HSI dataset by sequentially performing self-supervised pretraining on datasets that are increasingly similar to the target HSI, specifically, first on a large general vision dataset and then on a related HSI dataset. This sequential strategy enables the model to progressively learn from domain-general vision knowledge to target-specific hyperspectral knowledge. To mitigate the catastrophic forgetting in sequential training, we develop a regularization method, called self-supervised elastic weight consolidation (SS-EWC), to impose adaptive constraints on the changes to model parameters. Thorough classification experiments on various HSI datasets demonstrate that our framework significantly and consistently improves the self-supervised pretraining on HSIs in terms of both convergence speed and representation quality. Furthermore, our framework exhibits high generalizability and can be applied to various SSL algorithms. Transfer learning continues to prove its usefulness in self-supervised settings.**

***Index Terms*— Hyperspectral image (HSI) classification, limited labels, self-supervised learning (SSL), transfer learning.**

I. Introduction

A HYPERSPECTRAL image (HSI) captures the target scene with hundreds of narrow spectral bands and is widely applied in many applications, such as environmental monitoring [1], [2] and precision agriculture [3]. HSI classification, which refers to classifying the category of each pixel, is one important analysis technique for these applications. Early methods rely on extracting handcrafted features, which show poor performance [4], [5], [6]. Recently, deep learning has seen a lot of developments in image processing [7], [8], [9], [10], [11]. Deep neural networks are able to extract discriminative features, which benefits many pattern recognition tasks, including HSI classification [12], [13], [14]. However, despite higher accuracy, deep neural networks require a lot of labeled data for supervision. When the number of labeled HSI data is limited, deep networks can encounter overfitting problems and suffer from severe performance degradation. However, the collection and cleaning of so much labeled data are not only time-consuming but also laborious. With regard to the HSI data, this is even harder, as expertise is also required. To address this issue, many approaches of different learning paradigms, such as unsupervised learning [15], semisupervised learning [16], and few-show learning [17], have been proposed. However, these methods usually work much less efficiently and effectively than supervised learning in semantic feature extraction.

Recently, self-supervised learning (SSL) has become one of the hottest topics in representation learning [18], [19]. It proposes to acquire free labels from the unlabeled data itself and performs the training in a supervised manner. It aims to acquire strong pretrained models that can be transferred to downstream tasks, such as classification and detection. One representative SSL algorithm is contrastive learning where a model is trained to learn semantically meaningful representations by pushing the matched data closer and those unmatched far apart [20], [21]. Self-supervised pretraining is particularly effective when labels are scarce. Thus, some researchers propose to adopt SSL to solve the lack of labels for HSI classification and show great improvement [22], [23], [24]. However, despite the remarkable performance, several major issues remain when applying SSL for HSI classification. First, since HSIs are different from each other, for each HSI dataset, self-supervised pretraining needs to be performed from scratch, which requires expensive computation and long training time on large amounts of unlabeled data. In addition, SSL algorithms are sensitive to the quality of HSI, which, however, may suffer from low spatial resolution and noise. Thus, the performance of SSL methods is not consistent across different HSIs.

One common approach to overcoming the long training time and data sensitivity is transfer learning [25]. Reusing a pretrained model accelerates and stabilizes the training process on the target data. This inspires our original idea: instead of starting the self-supervised pretraining on HSIs from scratch, initialize it with an existing self-supervised pretrained model. In other words, enhance self-supervised pretraining with transfer learning. Transfer learning can be useful when the new data are similar to the old data. Thus, one possible solution is to reuse a model that pretrained on a related HSI dataset. The high similarity between the old and new data makes the reused model easily adapted to the target domain. However, the size of a related HSI dataset is usually quite

Manuscript received 22 November 2023; revised 11 March 2024; accepted 5 April 2024. Date of publication 6 May 2024; date of current version 16 May 2024. This work was supported in part by the Research Grants Council of Hong Kong under Grant GRF 17201822 and in part by the University of Hong Kong under Grant 104006536. *(Corresponding author: Peiyan Guan.)*

The authors are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, SAR, China (e-mail: pyguan@eee.hku.hk; elam@eee.hku.hk).

Digital Object Identifier 10.1109/TGRS.2024.3397740

small, which limits the effectiveness of transfer learning. The great development of large-scale self-supervised pretraining of vision models today leads us to ask whether such models can help the pretraining of HSIs [26], [27], [28]. These models are very strong, because they are trained on large and general vision datasets, such as ImageNet [29] and CIFAR-10 [30], which contain a huge amount of images of numerous classes. However, the general images are quite different from HSIs in spectral characteristics (three channels versus >100 channels). In addition, the acquisition scenarios of general vision datasets and HSI datasets are usually different. The huge data discrepancy brings great challenges to transferring pretrained vision models to the pretraining of HSIs.

To achieve fast and stable self-supervised pretraining on HSIs, we propose a simple and effective framework, called progressive self-supervised pretraining (PSP). Our framework performs self-supervised pretraining on datasets that are increasingly similar to the target HSI dataset. Specifically, it first pretrains a model on a large and general vision dataset (base dataset). Then, it continues to pretrain the model on an HSI dataset (source dataset), which is in the same field with the target HSI dataset but larger than it. These two steps acquire a pretrained model that can be transferred to the final pretraining on the target HSI dataset. Our PSP employs a large general vision dataset and a related HSI dataset sequentially for the preparatory pretraining according to their similarities to the target HSI. This sequential strategy enables the model to learn progressively from the general vision domain to the target hyperspectral domain. Moreover, each step provides a strong initialization for the next step, which makes the whole pretraining process robust.

Two points are worth noting about our sequential pretraining framework. First, different SSL algorithms are needed for the general vision dataset and the HSI dataset due to the image difference. We select two contrastive learning algorithms, SimCLR [20] and XDCL [22], for the two kinds of datasets, respectively. These two algorithms have similar frameworks and learning tasks, i.e., instance discrimination. In addition, to improve the transfer performance, we make the inputs to these two models have similar spectral characteristics. Second, our sequential pretraining strategy introduces a potential challenge known as catastrophic forgetting, where the model may forget knowledge obtained from previous steps [31]. Specifically, during the pretraining on the source dataset, the general knowledge from the base dataset will be lost, which is undesirable, since these knowledge holds potential value for the final pretraining on the target dataset. Instead, we aim to acquire the new hyperspectral knowledge and preserve the old general knowledge simultaneously in this step. To mitigate the forgetting, we develop a regularization method, called self-supervised elastic weight consolidation (SS-EWC), to impose constraints on the changes in model parameters during training.

Besides, while transfer learning for HSI has been carefully studied in supervised settings [32], it has not been formally investigated in self-supervised contexts. This problem is focused in our work, and some key questions remain to be answered. Transfer learning can be used to overcome overfitting when labeled HSI data are limited. However, when abundant unlabeled HSI data are available for the self-supervised pretraining, which means that overfitting is no longer an problem, is transfer learning still useful? How much can the convergence of the model be accelerated? Can the transferred model bring extra performance improvement, or will the effectiveness be nullified? We analyze the effectiveness of our method with extensive experiments on various HSI datasets, demonstrating that PSP improves SSL for HSI classification significantly in terms of both training speed and representation quality. Consistent improvement of PSP is observed across different SSL algorithms. Furthermore, the robustness of our framework against critical factors is investigated. Our main contributions can be summarized as follows.

1) To the best of our knowledge, we are the first to propose to enhance self-supervised pretraining on HSIs with transfer learning.
2) We build a PSP framework to provide powerful initialization for self-supervised pretraining on the target HSI dataset. It facilitates gradual learning from the general vision domain to the target hyperspectral domain by sequentially performing self-supervised pretraining on datasets that are increasingly similar to the target HSI.
3) We develop a regularization method, SS-EWC, to alleviate catastrophic forgetting during the sequential pretraining. It preserves the general vision knowledge and extends the model to the HSI domain simultaneously by constraining the deviation of model parameters.

The remaining sections are organized as follows. Section II give provides an overview of the related work. Section III introduces the details our PSP framework. Section IV presents the experimental results obtained from diverse HSI datasets, along with key analysis of our method. Finally, we draw the conclusions in Section V.

II. Related Work

1) *Self-supervised learning* refers to a type of methods that capture intrinsic features of the data without relying on human-annotated labels but rather self-generated labels [19], [33], [34], [35]. Unlike traditional unsupervised learning, SSL obtains labels from the data itself and designs pretext tasks for the representation learning. In this work, we mainly focus on one representative algorithm: contrastive learning [20], [36], [37]. It designs an instance discrimination task to find which samples are matched. A contrastive loss is adopted to push the positive samples closer and pull the negative samples apart. The positive sampling strategy plays a critical role. For instance, CMC proposes to use different channels (chrominance and luminance) of an image as the positive samples [38], while MoCo leverages a momentum encoder to generate positive samples [18]. Similarly, the samples augmented from the same image are considered matched in SimCLR, while those from different images are negative [20]. Contrastive learning has also been extended in multimodal representation learning [39], [40]. One example is CLIP, which contrasts images

with corresponding text descriptions [26]. Due to the great success of contrastive learning in various areas, it has been introduced to the representation learning of HSIs [22], [41]. These methods usually follow the structure of common contrastive learning algorithms but have different positive sampling strategies. For example, DMVL constructs the positive samples by splitting different spectral bands of HSI cubes [23], while SSCL relies on data augmentation [42]. Another example is XDCL, which separates the spectral and spatial information of each HSI cube and performs contrast between the two domains [22]. In addition to view construction, some researchers propose to combine contrastive learning with other pretext tasks, e.g., masked reconstruction [43]. However, the need to perform self-supervised pretraining from scratch on each HSI dataset and the inconsistent performance across different HSIs restrict the application of SSL in HSI classification. Therefore, our focus in this work is to efficiently acquire a robust self-supervised pretrained model for HSIs.

2) *Transfer learning* is a technique studying how to leverage knowledge learned from a *source* dataset to improve the performance of the model on a *target* dataset. A common scheme is to initialize the model weights with those trained on the source data and then fine-tune them with labeled target data, thereby avoiding overfitting. It is an effective method for lack of labeled data. Inspired by this success, transfer learning has been widely applied in HSI classification. One key factor influencing transfer performance is the similarity between the source and target datasets. Early methods focus on selecting an HSI collected by the same sensor with the target HSI as the source dataset to avoid large domain shift in transferring [44], [45], [46]. However, some researchers later demonstrate that transfer learning can also be applied between source and target datasets acquired by different hyperspectral sensors [47], [48], which greatly expands the selection of source datasets. To overcome the domain shift issue, a domain adaptation (DA) technique is explored. Some work focuses on minimizing the maximum mean discrepancy distance between domains [49], while some others employ the adversarial learning to adapt the model [50], [51]. Recently, some researchers try to apply contrastive learning to alleviate the domain shift [52]. Another important factor for a success transfer is the size of the source dataset. Since one single HSI is usually not large enough, some work proposes to combine multiple HSIs to create a larger source dataset, leading to improved classification performance [53], [54]. Nevertheless, the scale of such combined datasets is still considerable smaller compared with general vision datasets, such as ImageNet [29]. To address this limitation, He et al. [55] seek the use of large-scale vision datasets by incorporating a mapping layer to accommodate the differences between HSIs and RGB images. Previous work in transfer learning for HSI classification has primarily focused on supervised settings, which leaves large room for exploration in the context of self-supervised pretraining.

3) *Catastrophic forgetting* refers to the phenomenon where networks tend to forget the knowledge learned from previous tasks when trained with a new task [56]. This issue can be viewed as a stability–plasticity dilemma, where stability denotes the ability to retain old knowledge, while plasticity refers to the capability of adapting to new data. Typically, the more of one aspect usually results in the less of the other and vice versa. An ideal solution is to strike a balance between the two aspects, that is, to achieve the old knowledge preserving and the new knowledge adaption simultaneously. This problem is commonly observed in sequential learning paradigms, and many methods have been proposed to address it, which can be broadly divided into three types. The first one is the rehearsal approach, which saves or generates old data for replay during training on the new task [57], [58], [59]. The disadvantages include the memory cost of saving old data and the extra training time. Besides, the selection of saved data can be a problem. Another type is the architecture approach, which allocates different parameters to different tasks [60], [61]. For example, Rusu et al. [62] propose to expand the network with new branches for new tasks, while freezing the old parts and only training the new branches. However, this approach changes the model structure and is not suitable for model transfer. The last type is the regularization approach, which imposes constraints on parameter changes by adding a penalty loss or modifying the updating of the parameters [63], [64], [65]. This approach is efficient and effective for model transfer. One representative method is elastic weight consolidation (EWC) [66], which uses the Fisher matrix to estimate the importance of each weight and apply adaptive penalty correspondingly. However, EWC needs a supervised classification task to calculate the Fisher matrix, which is not applicable to self-supervised pretraining. In this work, our SS-EWC proposes an alternative for computing the Fisher matrix using the contrastive loss, which is specially developed to solve catastrophic forgetting in self-supervised contexts.

III. Methodology

Performing self-supervised pretraining on an HSI dataset from scratch requires long training time and is sensitive to data quality. We propose to use transfer learning to achieve fast and robust self-supervised pretraining on HSIs. Our PSP is a sequential pretraining framework where the model trained in each step is used to initialize the training process in the next step. The overflow of PSP is depicted in Fig. 1. Before the final pretraining on the target HSI dataset, it first conducts self-supervised pretraining on a large general vision dataset and then on a related HSI dataset. Since the datasets used throughout the whole training are increasingly similar to the target HSI, the model is able to gradually adapt from the general vision domain to the target HSI domain. In addition, a parameter regularization method, SS-EWC, is proposed

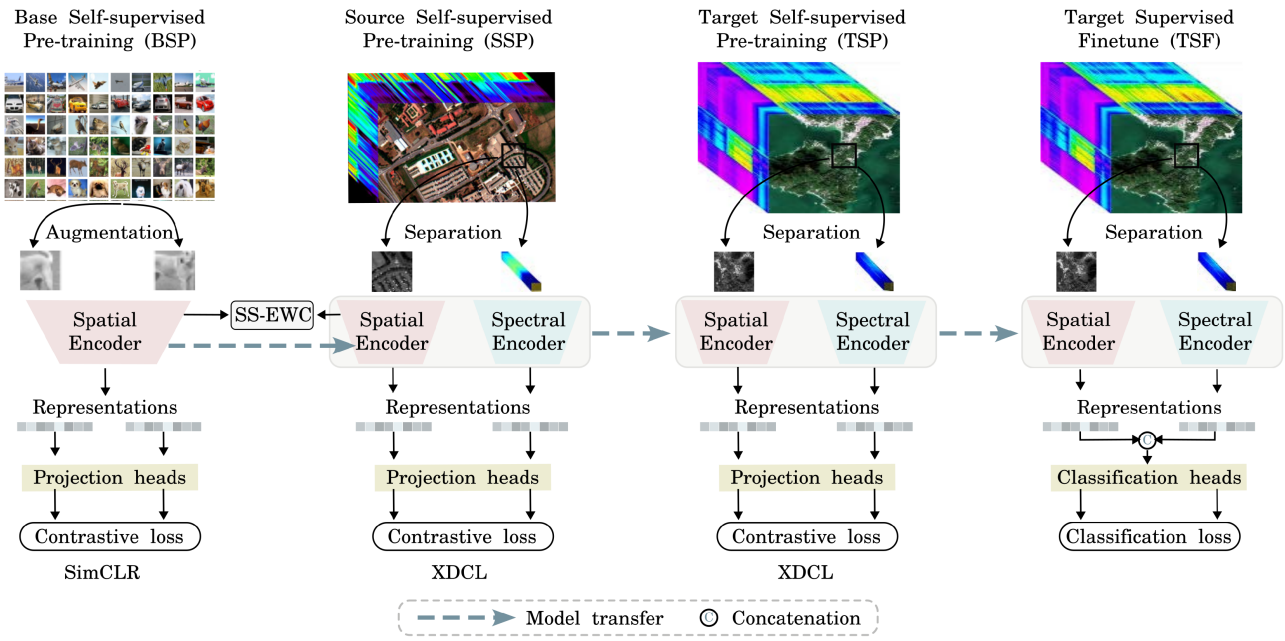

Fig. 1. Diagram of our PSP, which consists of several major steps. First, it performs the pretraining with SimCLR on the base dataset and acquires a trained spatial encoder. This step is called base self-supervised pretraining. In the next step, SSP, we conduct another contrastive learning algorithm, XDCL, on an HSI dataset that is larger than the target HSI dataset, where the spatial encoder is transferred from the base pretrained model, while the spectral encoder is trained from scratch. We also adopt a parameter regularization method called SS-EWC to refrain the spatial encoder from deviating too far away from the base pretrained model by constraining the parameter space. Then, in TSP, XDCL is conducted on the target HSI dataset with spatial and spectral encoders both initialized with the source pretrained model. In the end, the final pretrained model is transferred to the target HSI classification task. We call this step target supervised finetune.

to alleviate catastrophic forgetting in the sequential training process. In this section, we begin by discussing the selection and design of SSL algorithms employed within our framework. Then, we provide a detailed introduction to each step of our PSP.

A. Self-Supervised Learning Algorithm

We first explore the selection of SSL algorithms used in our method, which is of crucial importance. Given the discrepancy between general vision datasets and HSI datasets (RGB versus hyperspectral), it is essential to employ different SSL algorithms for pretraining on these two types of datasets. In the meantime, our sequential pretraining process requires smooth model transfer between different steps. Thus, suitable algorithms should support encoders with the same structure and inputs with similar spectral characteristics. According to these considerations, we adopt SimCLR for the pretraining on the general vision dataset and XDCL for the pretraining on HSI datasets, as depicted in Fig. 1. By employing the two algorithms, we ensure compatibility between the encoders throughout the sequential pretraining process.

SimCLR and XDCL are contrastive learning algorithms designed for the representation learning of general images and HSIs, respectively. The two algorithms share a similar framework, where they initially construct different samples for each instance and then use the base encoder to learn representations for them. Finally, a contrastive loss is applied on the representations to push the matched samples closer and those unmatched far away. The primary difference between SimCLR and XDCL lies in the construction of paired samples. SimCLR applies image augmentation, including random resized crop, flip, color jittering, and Gaussian blur, on general images. On the other hand, XDCL separates the spectral and spatial information of HSI cubes. Specifically, the center spectrum is taken as the spectral sample, and a random band of the spatial neighborhood is selected as the spatial sample.

The smooth model transfer between SimCLR and XDCL is facilitated by the design of their respective encoders. As illustrated in Fig. 1, SimCLR builds a single encoder, which takes augmented general images as the input. In contrast, XDCL builds a spatial encoder and a spectral encoder for the spatial and spectral samples, respectively. Notably, similar to SimCLR's encoder, the spatial encoder of XDCL also takes visual images, specifically, a 2-D band of the HSI cube, as input. Thus, we build the two encoders with the same structure and conduct model transfer between them. In addition, to make the inputs to them have more similar spectral characteristics, we apply an additional processing step to the inputs of SimCLR's encoder. Specifically, following the common image augmentation operation, we randomly select one of the three channels (RGB) from the augmented images as the new input. Consequently, the inputs to SimCLR's encoder and XDCL's spatial encoder are both single-channel spatial images. To maintain consistency in the reference of the two algorithms, we refer to the encoder of SimCLR also as spatial encoder in the following. Moreover, we omit the signal masking step in XDCL. The structures of the spectral and spatial encoders are presented in Table I.

B. Progressive Self-Supervised Pretraining

1) Base Self-Supervised Pretraining: As depicted in Fig. 1, our PSP starts from the self-supervised pretraining on the base

TABLE I
STRUCTURE OF THE SPECTRAL AND SPATIAL ENCODERS. "CONV" REFERS TO CONVOLUTION LAYER. "BN" DENOTES BATCH NORMALIZATION LAYER. "MP" REPRESENTS MAX-POOLING LAYER

Spectral encoder		Spatial encoder	
Layer	Filters	Layer	Filters
Conv+ReLU	3, 64	Conv+ReLU	3×3, 64
Conv+BN+ReLU	[3, 128]×2	Conv+BN+ReLU	[3×3, 128]×2
Conv+BN+ReLU	[3, 128]×2	Conv+BN+ReLU	[3×3, 128]×2
MP	3, 128	MP	2×2, 128
Conv+BN+ReLU	[3, 256]×2	Conv+BN+ReLU	[3×3, 256]×2
Conv+BN+ReLU	[3, 256]×2	Conv+BN+ReLU	[3×3, 256]×2
MP	3, 256	MP	2×2, 256
Conv+BN+ReLU	[3, 512]×2	Conv+BN+ReLU	[3×3, 512]×2
Conv+BN+ReLU	[3, 512]×2	Conv+BN+ReLU	[3×3, 512]×2
MP	3, 512	MP	2×2, 512
Flatten	N/A	Flatten	N/A
Fully connected	N/A	Fully connected	N/A

dataset. The base dataset is a general vision dataset consisting of common images of various classes. Our PSP framework has flexibility in selecting the base dataset, and an ideal one is expected to be large-scale and domain-general. Several publicly available visual datasets, such as ImageNet [29] and CIFAR-10 [30], can serve as suitable options for the base dataset. We employ the contrastive learning algorithm, SimCLR, to train a spatial encoder from scratch. This initial pretraining step yields a base pretrained model, which is used as the foundation for subsequent steps.

2) Source Self-Supervised Pretraining: Subsequently, we perform self-supervised pretraining on the source dataset using the base pretrained model as initialization. For this step, we employ the contrastive learning algorithm, XDCL, for the model training. The source dataset should be large enough and more similar to the target HSI dataset than the base dataset. To fulfill these requirements, we select an HSI that is not only in the same field with the target HSI but also larger than it as the source dataset. This selection ensures that the pretrained model gradually drifts from a general domain to a domain closer to the target dataset. XDCL has two encoders: a spatial encoder and a spectral encoder. During source self-supervised pretraining (SSP), the spectral encoder is trained from scratch, while the spatial encoder is transferred from the base pretrained model. This transfer ensures the utilization of previously acquired knowledge. Consequently, at this step, we obtain a source pretrained model.

Self-Supervised EWC: The objective of SSP is to preserve the acquired general vision knowledge and expand the model to the hyperspectral domain of the source dataset concurrently. However, this step inherently encounters the catastrophic forgetting problem. The transferred model might be adapted to the new distribution of the source dataset too much and lose valuable general vision knowledge useful for the target HSI data. To mitigate catastrophic forgetting, a classical approach is to impose constraints on the deviation of network parameters. With a general regularization, the learning objective of SSP becomes

$$\min_{\theta} \mathcal{L}_S + \lambda \cdot \Omega(\theta) \tag{1}$$

where $\mathcal{L}_S$ refers to the contrastive loss of XDCL, θ is the parameters of the spatial encoder, λ denotes the trade-off weight, and Ω is a general form of regularization. To prevent the model from drifting too much far away from the base pretrained model during SSP, we design a regularization method, called SS-EWC. It incorporates an adaptive L_2 penalty on each parameter of the spatial encoder according to their importance to the previous pretraining step. For a parameter important for base self-supervised pretraining (BSP), we give it a higher penalty weight to discourage it from large changes in SSP. As found in [66], the importance of each parameter to a task can be estimated using the diagonal elements of Fisher matrix F. The regularization term of SS-EWC can be formulated as follows:

$$\Omega(\theta) = \sum_k \frac{1}{2} F_k \left(\theta_k - \theta_k^*\right)^2 \tag{2}$$

where θ_k^* is the optimized value of the kth parameter after BSP and F_k is the kth diagonal element of F. One key property of F is that it can be estimated from the square of the first-order derivative of the loss near a minimum. Our SS-EWC uses the self-supervised contrastive loss $\mathcal{L}_B$ of SimCLR in BSP to calculate F as follows:

$$F_k = \frac{1}{D} \sum_{d \in D} \frac{\partial \mathcal{L}_B(d, \theta_k)^2}{\partial \theta_k^2} \bigg|_{\theta_k = \theta_k^*} \tag{3}$$

where D is the base dataset and d is a mini-batch of D. With this regularization, the spatial encoder is able to learn new hyperspectral knowledge of the source dataset with minimal forgetting of the general vision knowledge learned from BSP.

3) Target Self-Supervised Pretraining: Finally, the self-supervised pretraining is conducted on the target HSI dataset. Similar to the pretraining on the source dataset, target self-supervised pretraining (TSP) also utilizes XDCL as the SSL algorithm. Both the spatial and spectral encoders are initialized with the source pretrained model, which serves as a crucial starting point for adaptation to the target domain. In this step, the main goal of the transferred model is to rapidly adapt to the target domain. Thus, we do not apply any parameter regularization method during TSP.

4) Target Supervised Fine-Tune: Following the completion of the whole self-supervised pretraining process, we acquire the final pretrained model, which is then transferred to the downstream task, i.e., HSI classification, on the target HSI dataset. We attach a new classification head to the model for the adaptation to the classification task. The two representations learned from the spectral and spatial encoders are concatenated to form the full representation for each HSI cube. During target supervised fine-tune (TSF), the model is trained with labeled target HSI data.

IV. EXPERIMENTS

A. Datasets

We extensively evaluate the performance of the proposed PSP framework on numerous datasets, which are introduced below in three categories: base dataset, source dataset, and target dataset.

1) Base Dataset: We select **CIFAR-10** [30] as the base dataset. It is a ten-class, object centric, color imagery dataset. It consists of 60 000 32 × 32 RGB images of various objects, including airplane, bird, ship, and so on, with 6000 images per class. We use the training split, which contains 50 000 images for base self-supervised pretraining.

2) Source Dataset: **Houston** is an HSI classification dataset and is selected as the source dataset. The Houston image is captured over the campus of University of Houston and its neighborhood area. After removing noisy bands, it consists of 144 bands covering the spectral range between 380 and 1050 nm. The image consists of 1905 × 349 pixels and is divided into 15 classes, including various land-cover objects.

3) Target Dataset: Three hyperspectral datasets are selected as the target dataset, including Pavia University (PU), Salinas, and Indian Pines (IPs). The details of these datasets are presented in the following.

1) *PU* is gathered by the reflective optics system image spectrometer (ROSIS) over the PU, Italy. It consists of 610 × 610 pixels. After removing noisy bands, each pixel contains 103 bands covering the spectral range of 430–860 nm. There are nine classes of various ground objects in this dataset. Only the top-left 610 × 340 pixels are used in the experiment, since the other areas have no information.
2) *Salinas* is collected by the airborne visible/infrared imaging spectrometer (AVRIS) sensor over Salinas Valley in California. After removing 20 noisy bands, it consists of 204 bands covering the spectral range of 400–2500 nm. The spatial size of each band is 512 × 217. This image differentiates 16 classes in total.
3) **IP** is captured by AVRIS sensor over the IPs test site in Northwestern Indiana. After removing the noisy bands, it consists of 145 × 145 pixels with 200 spectral bands in the 400–2500-nm region. It also contains 16 ground-truth classes.

CIFAR-10 is very large and domain-general, while Houston is in the same field with the three target HSI datasets (remote sensing) and larger than them. The three target datasets are collected by different sensors and over different land covers, which enables us to test the universality of our PSP framework. Following the settings of XDCL, for the HSI datasets, we select cubes with a size of $9 \times 9 \times C$ as the inputs, where C represents the channel of the HSI and the class of each cube is decided by the center spectrum. Besides, all data of each HSI are used for the self-supervised pretraining. For the CIFAR-10 dataset, to achieve the scale consistency between HSI cubes and general images, we resize the general images to 12 × 12 after applying the random crop augmentation.

B. Implementation Details

Unless otherwise specified, the settings of SimCLR and XDCL algorithms follow the original work. The pixel intensity of CIFAR-10 images and HSIs is all normalized to [0, 1]. The dimension of the representation is 512. The projection head is a two-layer MLP, which outputs projections with a size of 128. Projection heads are not included in the transferring process. Since the dimension of the fully connected layer of the spectral encoder varies with HSI band number C, this layer is also not transferred. The training epoch is 1000 for the BSP and 10 for the SSP. We train the model with a batch size of 512. The learning rate is set to 3×10^{-4} with a decay schedule of cosine annealing for the two contrastive learning algorithms. The model is optimized with an Adam optimizer [67] with $\beta_1 = 0.9$. The temperature parameter of the contrastive loss is set to 0.07, and the trade-off weight λ is set to 10 000. SimCLR and XDCL are both implemented with the PyTorch framework. The experiments are performed on a computation platform with a Linux OS, an AMD Ryzen Threadripper PRO 3955WX CPU (16 Cores), 128-GB physics memory, and four NVIDIA RTX 4090 graphics cards.

C. Evaluations

We evaluate the final pretrained models acquired by our PSP with HSI classification tasks under limited label condition. The supervised classification evaluation is conducted on the three target datasets. During the evaluations, unless other specified, the middle band of each HSI cube, rather than a random band, is used as the spatial sample for the input of XDCL. We quantitatively evaluate the classification results with overall accuracy (OA), average accuracy (AA), and the kappa coefficient κ.

D. Analysis of the Pretraining Quality

In this section, we evaluate the effectiveness of our PSP framework with two downstream tasks: linear HSI classification and semisupervised HSI classification. We mainly compare PSP with the conventional self-supervised pretraining strategy that performs XDCL on the target HSI dataset from scratch. We refer to this pretraining strategy as **Scratch**.

1) Linear HSI Classification: We first test the separability of learned representations with a linear HSI classification task. A good pretrained model should produce features that can be accurately differentiated with limited labels. Specifically, after the pretraining procedure, the spectral and spatial encoders are frozen as the feature extractor. A linear classifier is then trained on top of the encoders with few labeled data. In our experiments, five labeled HSI data per class are selected as the training data, and the rest are used for the testing. The classifier is trained for 20 epochs with a learning rate of 3×10^{-2}.

This experiment is used to verify the effectiveness of PSP in accelerating self-supervised pretraining and improving representation quality. First, for the evaluation of training speed, we perform linear classification on the models of different target pretraining epochs. The accuracy curves of the two pretraining strategies are presented in Fig. 2. As can be seen, with the additional pretraining on the base and source dataset for initialization, our PSP significantly increases the convergence speed of the self-supervised pretraining on the target HSI dataset. For example, PSP achieves the highest classification accuracy in less than five epochs, while the Scratch pretraining strategy takes around 40 epochs (8× longer) on the PU dataset. Similar levels of training acceleration can also be found in the experimental results of the Salinas and IP datasets.

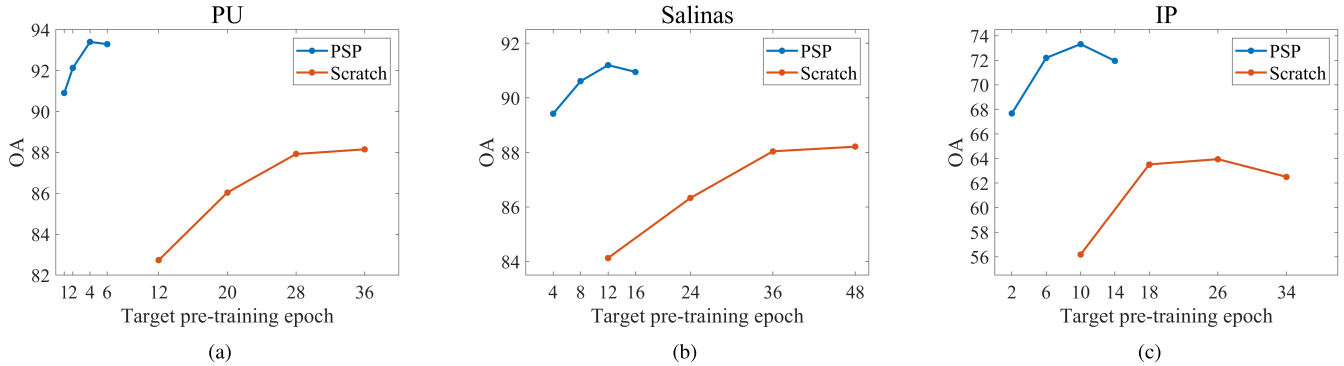

Fig. 2. Linear HSI classification. Overall classification accuracy of the PSP and Scratch pretraining strategies with respect to the target pretraining epoch over (a) PU dataset, (b) Salinas dataset, and (c) IP dataset.

In addition to the convergence speed, it is observed that our PSP improves the classification accuracy greatly over all three datasets, which cover a wide range of object categories, when only few labels are available. Specifically, the OA is increased by around 5% on PU, around 3% on Salinas, and over 9% on IP. It shows that our PSP demonstrates better separability and quality of learned representations compared with the Scratch pretraining strategy. We also note that performing self-supervised pretraining on the IP dataset has the lowest accuracy among all three datasets, which may be due to the poor image quality of IP. Surprisingly, our PSP brings the most significant accuracy improvement on it compared with the other two datasets, which demonstrates that PSP is beneficial to achieve consistent performance of SSL algorithms across different HSIs. What is even more impressive is that our PSP takes much less time to achieve higher accuracy than performing the self-supervised pretraining from scratch. Taking the PU dataset as an example, after only one epoch, the accuracy of PSP is already higher than the highest accuracy of the Scratch pretraining strategy, which takes it about 40 epochs. These results verify that transfer learning is still necessary and very effective in self-supervised settings.

2) Semisupervised HSI Classification: We also conduct a semisupervised classification task to further evaluate the pretrained models. Specifically, for each pretraining strategy, we select the pretrained model, which has the best performance in linear classification as the feature extractor and also append a classification head to it. These top-performing pretrained models are then fine-tuned with the same labeled data as in the linear classification experiment. The learning rate for the pretrained model is set to 3×10^{-5}, and that for the classification head is 3×10^{-3}. The tuning process is completed after 20 epochs.

This experiment is used to verify whether the effectiveness of additional pretraining on the base and source datasets will be nullified when all parameters of the per-trained models are fine-tuned with labeled HSI data. The fine-tuning performance for the best pretrained model of each pretraining strategy is shown in Table II. As can be seen, similar to the linear classification results, our PSP has much better performance than the Scratch pretraining strategy on all three datasets, which shows that PSP is still useful in semisupervised HSI classification, and the benefit of additional pretraining will not be nullified when all parameters are fine-tuned. It can be inferred from the experimental results that the quality of representations learned by our PSP is superior enough to that of the Scratch pretraining strategy that fine-tuning all model parameters cannot compensate for this difference, which further verifies the effectiveness of transfer learning in enhancing self-supervised HSI pretraining.

TABLE II
SEMISUPERVISED HSI CLASSIFICATION. OVERALL CLASSIFICATION ACCURACY OF FINE-TUNED MODELS OF PSP AND SCRATCH PRETRAINING (PT) STRATEGIES OVER THE THREE DATASETS. THE BEST VALUE IS IN BOLD

Pre-training strategy	PU	Salinas	IP
Scratch	88.65%	88.54%	64.62%
PSP	**93.43%**	**91.59%**	**73.84%**

E. Analysis of the Sequential Pretraining

One key design of our PSP framework is the sequential pretraining strategy where self-supervised pretraining is conducted on the base, source, and target datasets step by step. These three datasets are progressively more similar to the target HSI data, which enables the model learn knowledge that converges from the general domain to the target domain. Under this strategy, each step could provide a powerful pretrained model as initialization for the next step, which accelerates and stabilizes the whole training process. Thus, each pretraining step matters in our PSP framework. To evaluate the effect of each step, we compare the implementations of the following strategies.

1) *Target (T):* We discard the base and source datasets and perform self-supervised pretraining on the target HSI dataset from scratch, which is also the conventional Scratch pretraining strategy.
2) *Base Then Source ($B + S$):* We perform the pretraining on the base dataset first and the source dataset second. Then, we directly transfer the pretrained model to the target HSI classification task without pretraining on the target dataset.

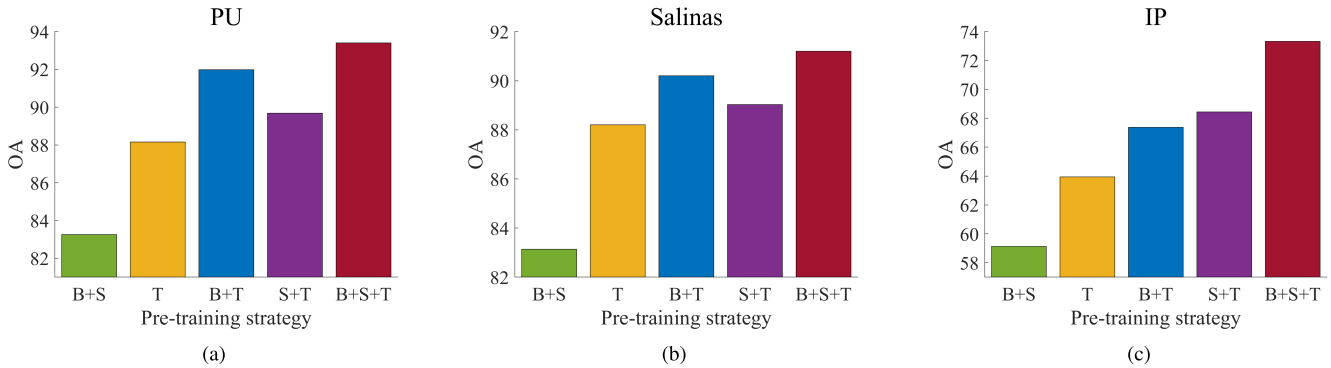

Fig. 3. Sequential pretraining analysis. Overall classification accuracy of best-performing models of different strategies from linear HSI classification over (a) PU dataset, (b) Salinas dataset, and (c) IP dataset.

3) *Base Then Target (B + T):* After the pretraining on the base dataset, we skip the source dataset and go straight to the pretraining on the target dataset.
4) *Source Then Target (S + T):* Self-supervised pretraining is first conducted on the source dataset and then on the target dataset, while the base dataset is discarded.
5) *Base Then Source Then Target (B + S + T):* This is the full framework of our PSP where self-supervised pretraining is conducted on the base, source, and target datasets sequentially.

Parameter regularization method, SS-EWC, is applied in the SSP step of B + S and B + S + T but not in T, B + T, and S + T. We use the top-performing model of each strategy from linear HSI classification for comparison. Fig. 3 shows the highest linear classification accuracy for each strategy over the three target datasets. It can be observed that our pretraining strategy, B + S + T, demonstrates consistently strong performance across different datasets. Besides, B + T and S + T both show improvement compared with the Scratch pretraining strategy (T), which indicates that either the base pretraining or the source pretraining is helpful. However, which one of these two steps is more useful depends on the target dataset according to our experimental results. As can be seen, S + T performs better on the IP dataset, while B + T has higher accuracy on the PU and Salinas datasets. We also note that directly transferring the model pretrained on the base and source datasets to classify the target HSI data obtains obviously worse results than pretraining on the target HSI dataset from scratch. This demonstrates that the domain gap between the target HSI dataset and the other two datasets, including the base dataset and the source dataset, is hard to fully bridge, which verifies the necessity of the TSP.

F. Analysis of the Stability of SSP

One issue for our sequential pretraining strategy is the catastrophic forgetting problem, which is particularly serious for the second pretraining step, SSP, where the model tends to learn new knowledge of the source dataset and forget the general knowledge of the base dataset, i.e., more plasticity but less stability. To enhance the stability of SSP, we propose SS-EWC to adaptively constrain the changes in model parameters. It identifies which parameters are important to BSP and prevent them from large changes, while the remaining parameters are mainly updated in SSP. In this way, the model retains the general knowledge and adapts to the source data simultaneously. To evaluate the effect of our SS-EWC, we compare it with some alternatives.

TABLE III
STABILITY ANALYSIS OF SSP. OVERALL CLASSIFICATION ACCURACY OF DIFFERENT REGULARIZATION METHODS OVER THE THREE DATASETS. THE BEST VALUE IS IN BOLD

Regularization	PU	Salinas	IP
None	91.89%	90.16%	71.34%
Freeze	92.70%	90.78%	71.03%
Uniform L_2	92.91%	90.94%	71.82%
SS-EWC	**93.40%**	**91.20%**	**73.32%**

1) *None:* No regularization is applied in SSP.
2) *Freeze:* It has been found that the shallow layers usually learn general features, while the deep layers tend to learn task-specific abstract features. Thus, to make the model retain the general knowledge from BSP and fit the training in SSP, one possible solution is to freeze the shallow layers and only update the deep layers. Specifically, in our experiment, the first three convolution layers are frozen, and the last four are trained.
3) *Uniform L_2:* We do not distinguish the importance of each parameter to BSP and apply L_2 penalty on them uniformly.

We also use the best-performing model from linear HSI classification for the evaluation. Table III shows the highest classification accuracy of each method over the three target datasets. As can be seen, applying regularization method is useful in most cases, except for freeze over IP. This may be due to the great domain gap between IP and the base dataset. The general knowledge is not as useful for the pretraining on IP as it is for the pretraining on PU and Salinas. This can also be verified by the observation from Fig. 3 that B + T achieves better results than S + T on PU and Salinas but worse result on IP. Thus, for different target datasets, SSP may require different levels of stability. In this case, the pretraining on IP needs a less stable SSP than that on PU and Salinas. However, freezing the shallow layers is too crude to consider such dataset difference. In comparison, constraining the parameter

TABLE IV
EFFECT ANALYSIS OF THE TRADE-OFF PARAMETER λ. OVERALL CLASSIFICATION ACCURACY OF DIFFERENT VALUES OF λ OVER THE THREE DATASETS. THE BEST VALUE IS IN BOLD

λ	PU	Salinas	IP
1000	91.57	89.67	71.58
5000	91.88	90.24	72.36
10000	**93.40**	**91.20**	**73.32**
50000	92.24	90.53	71.49
100000	91.75	90.11	70.62

changes is more careful and effective, which can also be seen from Table III. In particular, our SS-EWC obtains the best performance over all three datasets compared with Uniform L_2, which demonstrates the superiority of adaptive constraint over uniform constraint. The results also verify that SS-EWC succeeds in identifying the importance of different parameters to BSP, and it could achieve appropriate plasticity–stability balance that is applicable to the training of various target datasets.

In addition to the regularization strategy, the trade-off parameter λ in (1) also has a great impact on the stability of SSP. We study the effect of different values of λ with the linear classification task. The results over three target datasets are presented in Table IV. It can be observed that our PSP gives the best performance on all three datasets when $\lambda = 10\,000$. Too large λ makes the model more difficult to change parameters and, therefore, harder to adapt to the source data, while too little λ could make the model forget too much general knowledge. In our experiment, setting λ to 10 000 achieves a great stability–plasticity balance.

G. Comparison With the State-of-the-Art Methods

In this section, our PSP is compared with several state-of-the-art (SOTA) methods proposed to overcome the lack of HSI labels. These methods belong to different categories and are summarized as follows.

1) *Deep Few-Shot Learning Methods:* RN-FSC [17] employs meta-learning algorithm to train the model to learn how to learn on abundant source data and then transfers the model to the target domain.
2) *Deep Semisupervised Learning Methods:* AROC-DPNet [68] leverages a clustering algorithm to generate pseudo-labels, which are then used along with the true labels to optimize the network.
3) *Deep Unsupervised Learning Methods:* The 3DCAE [69] builds a 3-D convolutional autoencoder to learn unsupervised representations by trying to reconstruct the input HSI cubes.
4) *Domain Adaptation Methods:* ADA-Net [50] builds a generator and a discriminator and trains them in an adversarial manner to adapt the model from the source domain to the target domain.
5) *Deep Self-Supervised Learning Methods:* DMVL [23] applies contrastive learning algorithm and constructs different views of HSI data by splitting the spectral bands into different groups. XDCL [22] is also based on contrastive learning and proposes to separate different domain information in each HSI cube for making different views. It is used as the SSL algorithm in our PSP framework, while the signal masking operation in the original paper is discarded.

We follow the optimal parameters and settings in the original paper to reimplement these methods and use the same labeled target samples for the training for fair comparison. Our PSP uses the best-performing model from linear HSI classification for comparison. The classification results are presented in Table V. As can be seen, in general, SSL methods achieve better performance than the conventional unsupervised methods, demonstrating the strength of pretext tasks in discriminating useful semantics. As shown in Table V, our method outperforms comparison methods significantly over all three datasets in terms of all three metrics, which validates the effectiveness and robustness of our PSP on learning strong representations. In addition, compared with XDCL, our PSP not only improves the classification accuracy but also alleviates the issue of inconsistent performance across different HSIs.

H. Robustness to the Number of Target Pretraining Data

One key factor that influences the performance of SSL algorithms is the size of the pretraining dataset. In this section, we evaluate the robustness of our PSP to the number of target pretraining data. Specifically, we use 5%, 10%, 25%, and 100% of each target dataset for the target pretraining of PSP and Scratch.

Same as before, for different target dataset sizes, we use the top-performing models from linear classification for comparison. The classification results over the three datasets are presented in Fig. 4. As can be seen, self-supervised pretraining always performs better when more data are used. Nevertheless, compared with Scratch, PSP shows a much more slight decrease in accuracy when the amount of target pretraining data is reduced, which demonstrates the robustness of our method to the number of pretraining data. We also note that across the three datasets, the SSL algorithm is most sensitive to changes in the amount of training data on IP, where the accuracy declines significantly when the percentage of training data decreases from 100% to 25%. This is because IP itself contains the least amount of HSI data among the three datasets. However, even when the amount of training data is reduced to such an extreme, our PSP still shows great pretraining performance, which further validates its robustness.

I. Robustness to the Base and Source Datasets

Another crucial factor for the performance of our PSP is the base and source datasets. CIFAR-10 and Houston are selected as the base and source datasets in our experiment. However, the PSP framework allows for many options. To evaluate the robustness of PSP to the base and source datasets, we select different datasets and perform the sequential pretraining on them. Specifically, we select **RESISC** [70] as the new base dataset and **PU** as the new source dataset, while **Salinas** and **IP** are still used as the target datasets. RESISC is a 45-class remote sensing imagery dataset. It contains 31 500 RGB

TABLE V
CLASSIFICATION RESULTS OF DIFFERENT METHODS OVER THE THREE DATASETS. XDCL WITH * DENOTES THE ORIGINAL FULL XDCL METHOD WITH SIGNAL MASKING OPERATION, WHILE XDCL WITHOUT * REPRESENTS THE ALGORITHM USED IN OUR PSP. THE BEST VALUE IS IN BOLD

Method	PU			Salinas			IP		
	OA(%)	AA(%)	κ(%)	OA(%)	AA(%)	κ(%)	OA(%)	AA(%)	κ(%)
RN-FSC [17]	75.60	85.73	69.73	83.75	89.26	83.26	59.31	65.27	56.88
AROC-DPNet [68]	84.26	85.89	81.04	86.85	88.94	85.06	61.88	66.57	59.67
3DCAE [69]	72.43	83.62	65.97	82.54	86.75	81.02	57.35	66.98	52.19
ADA-Net [50]	83.65	83.03	81.54	85.26	86.96	80.17	60.54	68.19	58.94
DMVL [23]	81.94	88.25	77.32	89.87	92.96	89.06	69.51	79.64	66.93
XDCL [22]	88.15	90.11	84.03	88.24	93.64	86.80	63.95	76.65	59.03
XDCL* [22]	91.34	91.44	88.18	89.45	93.51	88.66	66.07	78.49	63.71
PSP	**93.40**	**93.43**	**91.36**	**91.20**	**95.19**	**90.04**	**73.32**	**81.52**	**69.82**

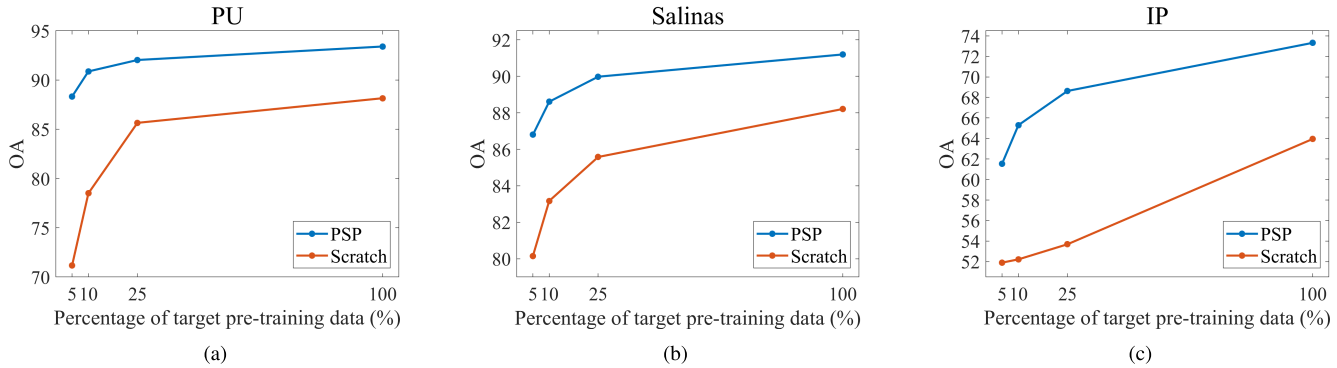

Fig. 4. Analysis of robustness to the size of target datasets. Overall accuracy of the best-performing models from linear HSI classification with different numbers of target pretraining data over (a) PU dataset, (b) Salinas dataset, and (c) IP dataset.

images of various remotely sensed objects with 700 images per class. Thus, compared with CIFAR-10, RESISC is more similar to the target HSI datasets. In addition, PU is in the same field with Salinas and IP. It is larger than the two target datasets but smaller than Houston.

We evaluate the pretraining quality when using the new base and source datasets through linear HSI classification experiments. The learning curves of different base and source datasets are shown in Fig. 5. As can be seen, compared with pretraining from scratch, whether using the new base dataset, RESISC, or the new source dataset, PU, our method delivers significant boosts to the pretraining in terms of both the model convergence and the representation quality. However, they both perform a bit worse than with the original base and source datasets, CIFAR-10 and Houston. The performance drop with the new source dataset may be because PU is smaller than Houston, while the drop with RESISC as the base dataset demonstrates that domain-general vision knowledge is useful for continual pretraining on HSI datasets. These results verifies the effectiveness of our dataset selection strategy that the base dataset should be large and domain-general, while the source dataset is expected to be large and similar to the target data.

J. Generalizability to Different Self-Supervised Learning Algorithms

Our PSP is proposed to enhance the self-supervised pretraining for HSI classification. Thus, it is expected be applicable to different SSL algorithms. To evaluate the generalizability of PSP, we use another contrastive learning algorithm, DMVL, instead of XDCL, for the self-supervised pretraining on HSI datasets. Unlike XDCL, DMVL constructs positive samples for each HSI cube by dividing the spectral bands into two parts and then applying PCA on each part. The first three components of each part are selected as the positive samples. These samples have the same number of channels as RGB images but different spectral characteristics. DMVL has only one encoder, and we build it with the same structure as the spatial encoder in Table I. The other parameters and settings follow the suggestions of the original paper. For the base pretraining, we still use SimCLR.

We use the linear classification task to evaluate the pretraining quality of DMVL. The accuracy curves of PSP and Scratch with respect to target pretraining epoch over PU are shown in Fig. 6. It is observed that PSP demonstrates great improvement compared with the Scratch strategy in terms of both training speed and representation quality. Specifically, PSP converges more than 5× faster than performing DMVL from scratch (eight epochs versus 45 epochs). The classification accuracy is improved by over 4%. Besides, the accuracy of PSP after two training epochs is already higher than the highest accuracy of Scratch after 45 epochs. The improvement that PSP brings to DMVL is consistent with that it brings to XDCL, as seen in Fig. 2(a). These results validate the strong generalizability of our PSP to different SSL algorithms.

K. Analysis of the Computational Complexity

In this section, we evaluate the computational complexity of our PSP. The pretraining, fine-tuning, and feature extraction

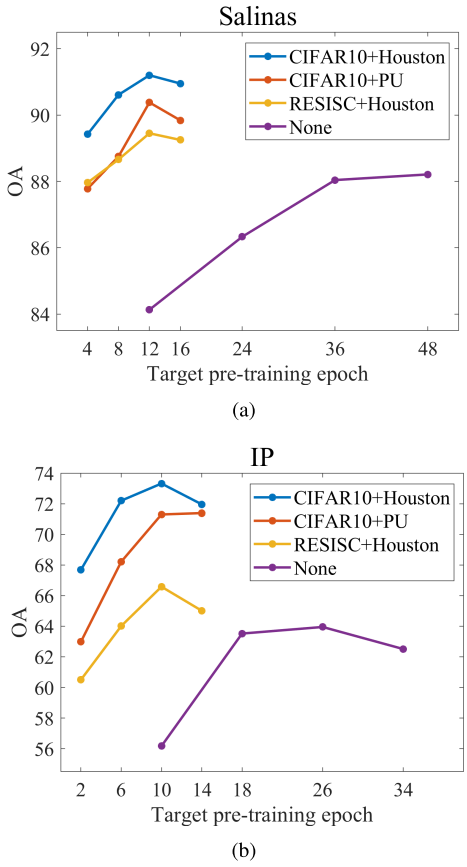

Fig. 5. Analysis of robustness of our PSP to the base and source datasets. The accuracy curves of different base and source datasets with respect to the target pretraining epoch over (a) Salinas dataset and (b) IP dataset.

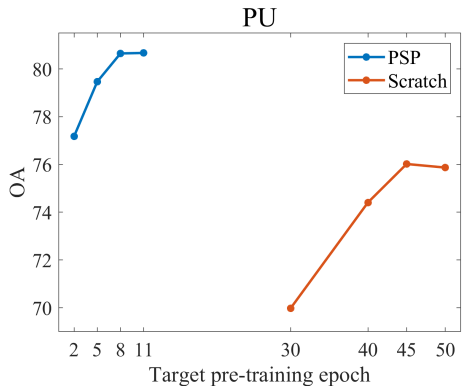

Fig. 6. Analysis of generalizability of our PSP to different SSL algorithms. With DMVL as the new SSL algorithm, accuracy curves of PSP and Scratch with respect to the target pretraining epoch over the PU dataset.

time of different SSL methods and unsupervised learning methods over PU are listed in Table VI. All experiments are performed on the same computation platform as stated in Section IV-B. It can be observed that the pretraining time of our full PSP framework is much more than the others, which is due to the fact that the pretraining on the base dataset and the source dataset takes a lot of time. However, such increase in time is worthwhile, since the pretrained model, once acquired, can be transferred to any HSI dataset, which will significantly reduce the training time on the target HSI data. As can be seen, with the pretrained model as initialization, the target

TABLE VI

PT, FINE-TUNING (FT), AND FEATURE-EXTRACTION (FE) TIME OF DIFFERENT METHODS ON THE PU DATASET. XDCL* DENOTES THE ORIGINAL FULL XDCL METHOD WITH SIGNAL MASKING OPERATION. PSP DENOTES THE WHOLE PT PROCESS FROM BSP TO TSP, WHILE PSP(T) REPRESENTS TSP ALONE. THE BEST VALUE IS IN BOLD

Time	3DCAE [69]	DMVL [23]	XDCL* [22]	PSP	PSP(T)
PT (min)	7.68	78.95	12.14	186.22	**0.59**
FT (s)	6.84	97.31	3.90	3.90	3.90
FE (s)	2.01	27.64	**1.33**	**1.33**	**1.33**

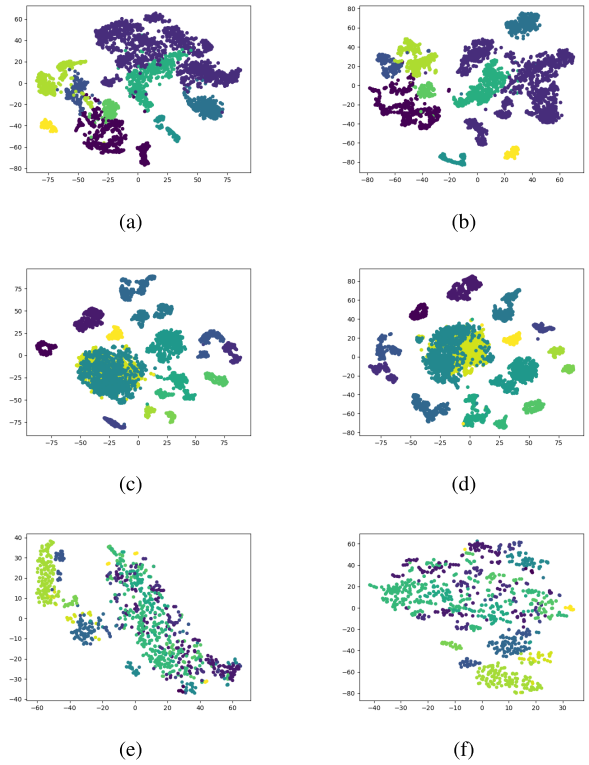

Fig. 7. Feature visualization results on three datasets. (a) Scratch pretraining on PU. (b) PSP on PU. (c) Scratch pretraining on Salinas. (d) PSP on Salinas. (e) Scratch pretraining on IP. (f) PSP on IP.

pretraining on PU only takes less than 1 min, which is considerably lower than the other methods. During the fine-tuning stage, the encoder is frozen as feature extractor, and a classifier is trained. It is observed that our PSP takes the least fine-tuning time, which is the same as XDCL, since they share the same encoder structure. Feature extraction time is also an important metric for evaluating the practicality of methods. Benefiting from a lightweight network and small size of network inputs, our PSP consumes the least time to extract features for all samples of PU. These results demonstrate that our method is both effective and efficient for learning HSI representations without labels.

L. Feature Visualization

In order to further evaluate the pretraining quality of our method, we use the t-distributed stochastic neighbor embedding (t-SNE) method [71] to visualize the learned representations of PSP and Scratch pretraining. The top-performing

model from linear classification is used as the feature extractor. We show the visualization results over the three target datasets in Fig. 7, where different colors represent different categories. As can be seen, compared with pretraining from scratch, PSP improves the separability of extracted features significantly, which demonstrates the power of our method. Higher quality representations of our PSP lead to better classification performance, just as illustrated in Fig. 2.

V. Conclusion

In this work, we provide the first empirical analysis of using transfer learning to enhance the self-supervised pretraining for HSI classification. The proposed PSP framework obtains strong initialization for the final pretraining by sequentially pretraining on datasets that are increasingly similar to the target HSI data. The common catastrophic forgetting issue is alleviated with the proposed SS-EWC. In the experiments, our PSP is observed to yield faster and more robust self-supervised pretraining process on various target HSI datasets. Moreover, the quality of learned representations is significantly improved. These results demonstrate that transfer learning is still useful in the context of self-supervised pretraining. Our framework is easy to implement, and its applicability to different SSL algorithms further enhances its practicality. This work reduces the need of SSL for unlabeled HSI data and significantly increases the efficiency and stability of self-supervised pretraining, greatly broadening the use of SSL algorithms in real-world HSI applications.

References

[1] M. B. Stuart, A. J. S. McGonigle, and J. R. Willmott, "Hyperspectral imaging in environmental monitoring: A review of recent developments and technological advances in compact field deployable systems," *Sensors*, vol. 19, no. 14, p. 3071, Jul. 2019.

[2] M. J. Khan, H. S. Khan, A. Yousaf, K. Khurshid, and A. Abbas, "Modern trends in hyperspectral image analysis: A review," *IEEE Access*, vol. 6, pp. 14118–14129, 2018.

[3] B. Lu, P. Dao, J. Liu, Y. He, and J. Shang, "Recent advances of hyperspectral imaging technology and applications in agriculture," *Remote Sens.*, vol. 12, no. 16, p. 2659, Aug. 2020.

[4] A. Plaza, P. Martinez, J. Plaza, and R. Perez, "Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 466–479, Mar. 2005.

[5] L. Ma, M. M. Crawford, and J. Tian, "Local manifold learning-based k-nearest-neighbor for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4099–4109, Nov. 2010.

[6] B.-C. Kuo and D. A. Landgrebe, "Nonparametric weighted feature extraction for classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 5, pp. 1096–1105, May 2004.

[7] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[8] P. Guan and E. Y. Lam, "Multistage dual-attention guided fusion network for hyperspectral pansharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5515214.

[9] T. Zeng and E. Y. Lam, "Robust reconstruction with deep learning to handle model mismatch in lensless imaging," *IEEE Trans. Comput. Imag.*, vol. 7, pp. 1080–1092, 2021.

[10] P. Guan and E. Y. Lam, "Three-branch multilevel attentive fusion network for hyperspectral pansharpening," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2022, pp. 1087–1090.

[11] N. Meng, H. K.-H. So, X. Sun, and E. Y. Lam, "High-dimensional dense residual convolutional neural network for light field reconstruction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 3, pp. 873–886, Mar. 2021.

[12] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, pp. 1–12, Jan. 2015.

[13] X. Li, M. Ding, and A. Pižurica, "Deep feature fusion via two-stream convolutional neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 4, pp. 2615–2629, Apr. 2020.

[14] B. Zhang, L. Zhao, and X. Zhang, "Three-dimensional convolutional neural network model for tree species classification using airborne hyperspectral images," *Remote Sens. Environ.*, vol. 247, Sep. 2020, Art. no. 111938.

[15] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.

[16] B. Liu, X. Yu, P. Zhang, X. Tan, A. Yu, and Z. Xue, "A semi-supervised convolutional neural network for hyperspectral image classification," *Remote Sens. Lett.*, vol. 8, no. 9, pp. 839–848, Sep. 2017.

[17] K. Gao, B. Liu, X. Yu, J. Qin, P. Zhang, and X. Tan, "Deep relation network for hyperspectral image few-shot classification," *Remote Sens.*, vol. 12, no. 6, p. 923, Mar. 2020.

[18] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," 2019, *arXiv:1911.05722*.

[19] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 16000–16009.

[20] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn.*, vol. 119, 2020, pp. 1597–1607.

[21] J.-B. Grill et al., "Bootstrap your own latent—A new approach to self-supervised learning," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 21271–21284.

[22] P. Guan and E. Y. Lam, "Cross-domain contrastive learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5528913.

[23] B. Liu, A. Yu, X. Yu, R. Wang, K. Gao, and W. Guo, "Deep multiview learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7758–7772, Sep. 2021.

[24] P. Guan and E. Y. Lam, "Spatial–spectral contrastive learning for hyperspectral image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2022, pp. 1372–1375.

[25] F. Zhuang et al., "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jul. 2020.

[26] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, vol. 139, 2021, pp. 8748–8763.

[27] J. Shao et al., "INTERN: A new learning paradigm towards general vision," 2021, *arXiv:2111.08687*.

[28] M. Oquab et al., "DINOv2: Learning robust visual features without supervision," 2023, *arXiv:2304.07193*.

[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[30] A. Krizhevsky G. Hinton, "Learning multiple layers of features from tiny images," M.S. thesis, Univ. Toronto, Toronto, ON, Canada, 2009.

[31] Z. Mai, R. Li, J. Jeong, D. Quispe, H. Kim, and S. Sanner, "Online continual learning in image classification: An empirical survey," *Neurocomputing*, vol. 469, pp. 28–51, Jan. 2022.

[32] S. Jia, S. Jiang, Z. Lin, N. Li, M. Xu, and S. Yu, "A survey: Deep learning for hyperspectral image classification with few labeled samples," *Neurocomputing*, vol. 448, pp. 179–204, Aug. 2021.

[33] X. Liu et al., "Self-supervised learning: Generative or contrastive," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 857–876, Aug. 2021.

[34] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1422–1430.

[35] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," 2018, *arXiv:1803.07728*.

[36] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with exemplar convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1734–1747, Sep. 2016.

[37] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[38] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 776–794.

[39] P. Morgado, N. Vasconcelos, and I. Misra, "Audio-visual instance discrimination with cross-modal agreement," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12470–12481.

[40] P. Guan et al., "PIDRo: Parallel isomeric attention with dynamic routing for text-video retrieval," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 11164–11173.

[41] Q. Liu, J. Peng, G. Zhang, W. Sun, and Q. Du, "Deep contrastive learning network for small-sample hyperspectral image classification," *J. Remote Sens.*, vol. 3, p. 0025, Jan. 2023.

[42] S. Hou, H. Shi, X. Cao, X. Zhang, and L. Jiao, "Hyperspectral imagery classification based on contrastive learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 5521213.

[43] L. Huang, Y. Chen, and X. He, "Spectral–spatial masked transformer with supervised and contrastive learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5508718.

[44] J. Yang, Y. Zhao, J. C. Chan, and C. Yi, "Hyperspectral image classification using two-channel deep convolutional neural network," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 5079–5082.

[45] C. Deng, Y. Xue, X. Liu, C. Li, and D. Tao, "Active transfer learning network: A unified deep joint spectral–spatial feature learning model for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1741–1754, Mar. 2019.

[46] J. Lin, C. He, Z. J. Wang, and S. Li, "Structure preserving transfer learning for unsupervised hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 10, pp. 1656–1660, Oct. 2017.

[47] Y. Liu, L. Gao, C. Xiao, Y. Qu, K. Zheng, and A. Marinoni, "Hyperspectral image classification based on a shuffled group convolutional neural network with transfer learning," *Remote Sens.*, vol. 12, no. 11, p. 1780, Jun. 2020.

[48] J. Lin, L. Zhao, S. Li, R. Ward, and Z. J. Wang, "Active-learning-incorporated deep transfer learning for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 11, pp. 4048–4062, Nov. 2018.

[49] Y. Zhu et al., "Deep subdomain adaptation network for image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 4, pp. 1713–1722, Apr. 2020.

[50] X. Ma, X. Mou, J. Wang, X. Liu, J. Geng, and H. Wang, "Cross-dataset hyperspectral image classification based on adversarial domain adaptation," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 4179–4190, May 2021.

[51] Y. Huang et al., "Two-branch attention adversarial domain adaptation network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5540813.

[52] Q. Liu et al., "Refined prototypical contrastive learning for few-shot hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5506214.

[53] B. Liu, X. Yu, A. Yu, P. Zhang, G. Wan, and R. Wang, "Deep few-shot learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2290–2304, Apr. 2018.

[54] H. Lee, S. Eum, and H. Kwon, "Exploring cross-domain pretrained model for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5526812.

[55] X. He, Y. Chen, and P. Ghamisi, "Heterogeneous transfer learning for hyperspectral image classification based on convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3246–3263, May 2019.

[56] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of Learning and Motivation*, vol. 24. Amsterdam, The Netherlands: Elsevier, 1989, pp. 109–165.

[57] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, "Gradient based sample selection for online continual learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 11816–11825.

[58] A. Chaudhry et al., "On tiny episodic memories in continual learning," 2019, *arXiv:1902.10486*.

[59] T. Lesort, H. Caselles-Dupré, M. Garcia-Ortiz, A. Stoian, and D. Filliat, "Generative models from the perspective of continual learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[60] A. Mallya and S. Lazebnik, "PackNet: Adding multiple tasks to a single network by iterative pruning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7765–7773.

[61] J. Serra, D. Suris, M. Miron, and A. Karatzoglou, "Overcoming catastrophic forgetting with hard attention to the task," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4548–4557.

[62] A. A. Rusu et al., "Progressive neural networks," 2016, *arXiv:1606.04671*.

[63] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, Dec. 2017.

[64] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 139–154.

[65] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with A-GEM," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[66] K. James et al., "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.

[67] D. P. Kingma and J. Ba, "ADAM: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, May 2015.

[68] B. Fang, Y. Li, H. Zhang, and J. C.-W. Chan, "Collaborative learning of lightweight convolutional neural network and deep clustering for hyperspectral image semi-supervised classification with limited training samples," *ISPRS J. Photogramm. Remote Sens.*, vol. 161, pp. 164–178, Mar. 2020.

[69] S. Mei, J. Ji, Y. Geng, Z. Zhang, X. Li, and Q. Du, "Unsupervised spatial–spectral feature learning by 3D convolutional autoencoder for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6808–6820, Sep. 2019.

[70] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct. 2017.

[71] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

Peiyan Guan (Graduate Student Member, IEEE) received the B.S. degree in communication engineering from Nanjing University, Nanjing, China, in 2018. He is currently pursuing the Ph.D. degree with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong.

His research interests include hyperspectral image analysis, self-supervised learning, representation learning, and multimodal learning.

Edmund Y. Lam (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 1995, 1996, and 2000, respectively.

From 2010 to 2011, he was a Visiting Associate Professor with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA. He is currently a Professor of electrical and electronic engineering with The University of Hong Kong, Hong Kong, where he also serves as the Computer Engineering Program Director. His research interests include computational imaging.

Dr. Lam is a Fellow of the Optical Society of America (OSA), the Society of Photo-Optical Instrumentation Engineers (SPIE), the Society for Imaging Science and Technology (IS&T), and the Hong Kong Institution of Engineers (HKIE). He was a recipient of the IBM Faculty Award.