Cross-Domain Invariant Feature Absorption and Domain-Specific Feature Retention for Domain Incremental Chest X-Ray Classification

Mengchu Wang, Yuhang He*, Lin Peng, Xiang Song, Songlin Dong, Yihong Gong, *Fellow, IEEE*

***Abstract*— Chest X-ray (CXR) images have been widely adopted in clinical care and pathological diagnosis in recent years. Some advanced methods on CXR classification task achieve impressive performance by training the model statically. However, in the real clinical environment, the model needs to learn continually and this can be viewed as a domain incremental learning (DIL) problem. Due to large domain gaps, DIL is faced with catastrophic forgetting. Therefore, in this paper, we propose a Cross-domain invariant feature absorption and Domain-specific feature retention (CaD) framework. To be specific, we adopt a Cross-domain Invariant Feature Absorption (CIFA) module to learn the domain invariant knowledge and a Domain-Specific Feature Retention (DSFR) module to learn the domain-specific knowledge. The CIFA module contains the C(lass)-adapter and an absorbing strategy is used to fuse the common features among different domains. The DSFR module contains the D(omain)-adapter for each domain and it connects to the network in parallel independently to prevent forgetting. A multi-label contrastive loss (MLCL) is used in the training process and improves the class distinctiveness within each domain. We leverage publicly available large-scale datasets to simulate domain incremental learning scenarios, extensive experimental results substantiate the effectiveness of our proposed methods and it has reached state-of-the-art performance.**

***Index Terms*— Domain incremental learning, Chest X-ray classification, dual-adapter, multi-label contrastive learning**

I. Introduction

In recent years, with the rapid development of Deep Neural Networks (DNNs), applying DNNs in medical image analysis has attracted massive attention and achieved promising progress. With a large number of chest X-ray (CXR) generations, the DNN-based computer-aided disease (CAD) diagnosis is used widely in clinical care [1]. For example, the Coronavirus 2019 (COVID-19), from its first case reported in 2019, has affected people all over the world and caused millions of deaths [2]–[4]. By employing the DNNs for lung lesions discovery of COVID-19, the DNN-based medical image analysis significantly improves the efficiency and accuracy of disease diagnosis, especially with limited medical resources. Despite these outstanding achievements [5]–[7], the current mainstream of DNNs is prone to the close-world assumption, that the classes, distribution, modalities, *etc* of the testing data are required to be identical to the ones of training data. In real-world applications, however, the DNN models have to face dynamic environments, newly encountered diseases, and continuously changed symptoms. For instance, the lung lesions caused by COVID-19 are constantly changing at different times and in different regions. In 2020, peripheral lower lung–predominant ground glass and consolidation were the main manifestations of the ancestral strain in Asia [8]. In 2023, lobar pneumonia, bronchopneumonia, and bronchiolitisit were commonly found in patients infected with the Omicron variant in South America [9]. The distribution shifts (*i.e.*, the domain gaps) caused by the changes of ethnic groups, ages, *etc*, [10]–[12] in real-world scenarios make it difficult to obtain a model by close-set training. Therefore, there is an urgent technical need to enable the DNN models to continuously and incrementally learn new knowledge of new domains while maintaining the performance on old ones, *i.e.*, the *domain incremental learning* (DIL).

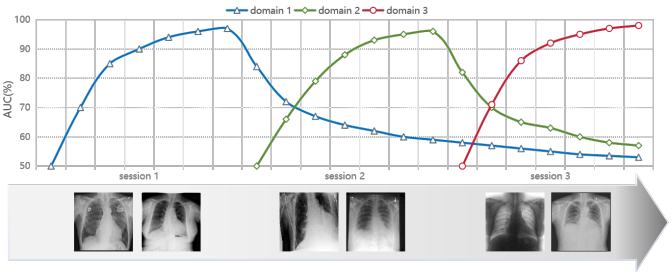


Fig. 1. The catastrophic forgetting on the domain incremental multi-label chest X-ray classification. Although the model is trained to converge on the past domain, when the new data comes, the past accuracy drops sharply.

In this paper, we focus on the DIL of chest X-ray classification task [5]. Due to the distribution shifts between different domains, the DIL is faced with one challenge, *catastrophic forgetting* problem [13], [14], as shown in Fig. 1, *i.e.*, learning new leading to forgetting old, where the parameters related

This work was funded by the Natural Science Foundation of China under Grant No.U21B2048 and No.62302382 and Shenzhen Key Technical Projects under Grant CJGJZD2022051714160501, China Postdoctoral Science Foundation No.2024M752584. *(Corresponding author: Yuhang He)*

Mengchu Wang and Xiang Song are with the School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China. (e-mail: {xuanfei1000, songxiang}@stu.xjtu.edu.cn).

Yuhang He, Lin Peng, Songlin Dong and Yihong Gong are with the College of Artificial Intelligence, Xi'an Jiaotong University, Xi'an 710049, China. (e-mail: heyuhang@xjtu.edu.cn, {penglin, dsl972731417}@stu.xjtu.edu.cn, ygong@mail.xjtu.edu.cn)

to the old knowledge are overwritten by the newly learned knowledge, resulting in dramatic performance decreases on old tasks. To meet this challenge, existing methods solve the forgetting from three aspects, including rehearsal-based methods, regularization-based methods, and architecture-based methods. Firstly, for rehearsal-based methods, recent solutions use a buffer to store a subset of old representative samples [15]–[19] or features [20], [21] and replay them during the new learning session. However, they face several limitations. First, due to data privacy, the previous samples are not accessible, which means the data cannot be transferred during incremental sessions [22], [23]. This makes these approaches infeasible. Second, resource constraints are essential, especially in clinical settings. Infrastructures among different areas and institutions may constrain the storage of model parameters and data samples, and make large-scale data transmission unreliable [24], [25]. Secondly, for regularization-based methods, recent works [17], [26], [27] apply regularization skills in the loss. Most of them are designed for single-label classification, however, for complex label co-occurrence relationships in chest X-ray images, they no longer work. Although a few of them can be adapted to medical tasks, they cannot show advanced performance when there is a huge difference between the distribution of old and new data. Lastly, the architecture-based methods, which are based on a pre-trained backbone and add an extra network module to learn the knowledge of each session [28], [29]. These methods can avoid the data privacy problem and make an impressive performance on natural images. Unfortunately, the pre-acquired knowledge of natural images is unhelpful for medical image classification. When the image style transfers, these methods produce less satisfactory predictions.

To tackle these limitations, we propose a novel Cross-domain invariant feature absorption and Domain-specific feature retention (CaD) framework for the DIL of chest x-ray classification task. The proposed CaD contains two major components, including a Cross-domain Invariant Feature Absorption (CIFA) module to learn the domain invariant knowledge across different domains and a Domain-Specific Feature Retention (DSFR) module to learn domain-specific knowledge within each domain. Additionally, a multi-label contrastive loss (MLCL) is designed to improve the model performance in handling multiple diseases in a single image. Specifically, 1) in CIFA, for each domain, we design an absorption adapter (C-adapter) to extract domain-invariant classification information and then fuse the newly obtained C-adapter to the previous ones by moving the average. Concretely, we first incentivize the model to map the parameters of the C-adapter into a shared knowledge of different domains. Then, an absorbing strategy of the C-adapter keeps the model structure consistent and avoids parameter redundancy. These enable the model to explore more domain-invariant information and improve the model robustness to continuously emerging domains. 2) In DSFR, we learn an independent retention adapter (D-adapter) for each domain, respectively. During the incremental period, the D-adapter extracts image features for the currently obtained data and is frozen after training to preserve the current domain knowledge unforgotten in the following incremental learning process. Compared to the rehearsal-based methods that select a few local samples of each domain, the D-adapter learns the global information of each domain and captures fine-grained domain-specific features by an extra 2D convolution layer. On these basis, we design a multi-label contrastive loss to increase the intra-class compactness and inter-class separability, addressing the complex relationships between pathology labels (such as Edema and Pneumonia are highly related).

We conduct extensive and comprehensive experiment results on five benchmark datasets, including ChestXray14 [30], Chexpert [31], MIMIC-CXR [32], VinBig [33] and OpenI [34]. The experimental results show that our method steadily and significantly outperforms the current state-of-the-art methods (5.92% average improvement on the *ChestXray14* $\rightarrow$ *VinBig* $\rightarrow$ *OpenI* datasets), which demonstrates the superiority and efficiency of the proposed CaD method.

In summary, our main contributions include:

- We propose a novel Cross-domain invariant feature absorption and Domain-specific feature retention (CaD) framework for the domain incremental multi-label chest X-ray classification problem.
- We design the Cross-domain Invariant Feature Absorption module (CIFA) with a C-adapter to extract domain-invariant classification information across different domains.
- We design the Domain-Specific Feature Retention module (DSFR) with domain-specific D-adapters, where each D-adapter is initialized to learn the global information of each domain and is frozen after training.
- Extensive and comprehensive experiment results on five benchmark datasets show that our CaD method steadily and significantly outperforms the state-of-the-art methods by a large margin (at least 2.98% improvements on the five benchmark datasets).

II. Related Work

A. Chest X-ray Multi-label Classification

Chest X-ray classification is an important task in computer-aided disease diagnosis. In recent years, more and more Deep Neural Networks (DNNs) have been used and proven beneficial on this task, such as ChexNet [5] based on DenseNet121 [35], and other supervised methods [6], [36] based on ViT [37]. DualCheXNet [38] fused image features at two levels, the feature-level and decision-level, to form the complementary information in the model. Recent work [36] has explored the impact of different backbones on CXR image classification, including different sizes of CNNs, ViTs, and DeiTs [39]. [40] focused on radiology reports and utilized them to help the image encoder capture CXR image features. They show an impressive performance on different CXR image datasets. On the other hand, more and more chest X-ray datasets have been published with larger scale and more complex data distribution, such as ChestXray14 [30], Chexpert [31], MIMIC-CXR [32].

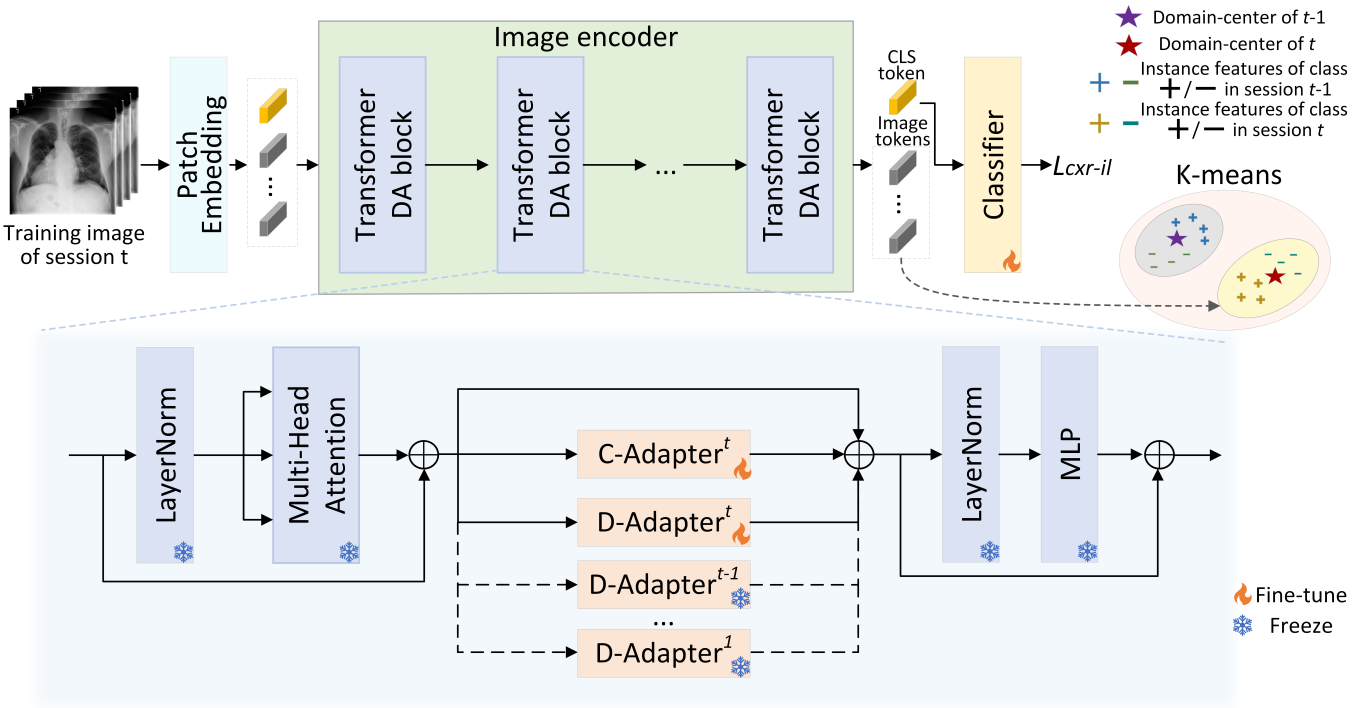


Fig. 2. The overview of our proposed CaD framework with dual-adapter layers for domain incremental learning on chest X-ray classification. For incremental training session *t*, we adopt the following steps: 1) updating the image encoder through a parameter-efficient fine-tuning strategy with D-adapter and C-adapter. 2) getting the image feature after training all the specific domain images. 3) using K-means to get each domain center and store them. 4) using an absorbing strategy to merge the parameters of the C-adapter. For the inference, first, the test image tokens will be fed into each group of dual-adapter layers and get the corresponding test image feature. Second, computing the distance of test image features and the domain centers, the nearest one will be selected and the domain-id will be obtained. Last, the test image will be evaluated on the selected model with dual-adapter layers.

B. Domain Incremental learning

Rehearsal-based methods store a set of exemplars of the old data when the model is training the new data. The exemplar set can be the selected images [41], the features of model layers [20], [21], or the samples from the generative model [42]–[44]. Early rehearsal method ER [45] uses a memory buffer to store samples from old session images and adds them to new data. DER [19] is an improvement of ER, which stores the output of the old model and uses a loss to make it close to the output of the new model. For DIL of medical images, there are some works [12], [25], [46]. [25] proposes a method that stores activations of the model and replays them when new data is trained. Another similar method is Vector quantization (VQ) [46], which stores and replays hidden representations of the old model.

Regularization-based methods make regularization skills in the loss to constrain learning new knowledge. Some of them use regularization on parameters [47]–[50], and others are about knowledge distillation [16], [26], [27], [51]. Elastic Weight Consolidation (EWC) [17] uses a fisher matrix to preserve the old parameters from the last session. Based on EWC, EWC_on [26] calculates the parameter importances from the classification loss. LWF [27] is a representative method using knowledge distillation, which regards the previous model as soft teachers to mitigate forgetting.

Architecture-based methods use independent parameters of the model for each session. Some of them divide the model into different parts and assign them to each session, such as [51]–[53]. The other methods aim to expand the network, which keeps the old network unchanged and adds new parameters for the new data. L2P [28] is a method based on a pre-trained model of ViT, which exploits a dependent prompt pool to learn continually. Different from L2P, S-Prompts [29] learns independent prompts for each session respectively and achieves desirable performance on DIL of natural images.

III. Methodology

A. Problem Formulation

In this paper, we focus on Domain Incremental Learning (DIL) on CXR classification. Specifically, assuming there are T+1 different domains total, which means the incremental sessions number is T (the first domain does not belong to incremental session). Take $D^t = \{x_i^t, y_i^t\}_{i=1}^{N^t}$ as the training set, where x_i^t denotes the i-th image from domain t, y_i^t is the label related to x_i^t, and N^t is the image number of domain t. Denote C^t as the label set of the domain t, and $C^1 = C^2 = ... = C^t$. At each session, D^t is the only accessible training set.

B. Overall Framework

As shown in Fig. 2, we propose a novel framework called CaD. Firstly, CaD uses a pre-trained vision transformer [37] on ImageNet [54] as the original backbone. To mitigate the

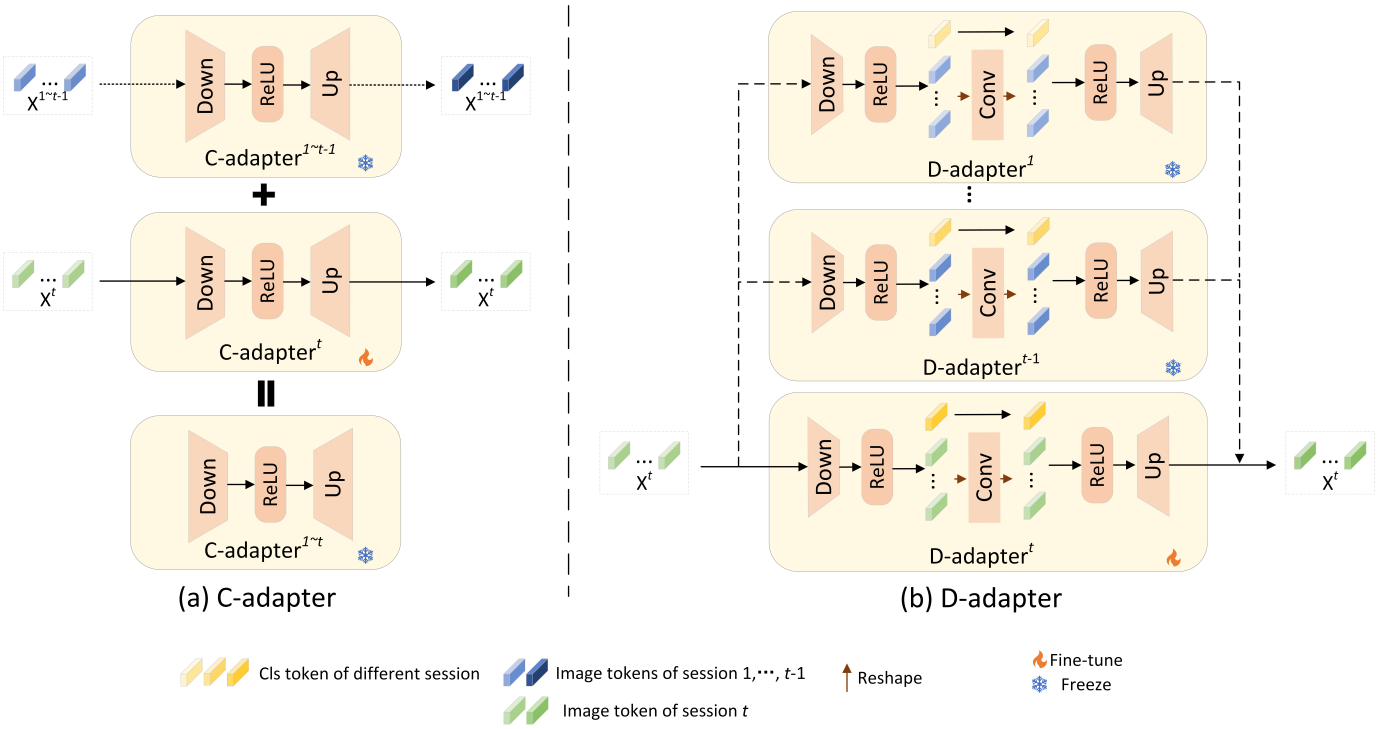


Fig. 3. The detailed structure of CIFA and DSFR. C-adapter consists of the trainable down → up bottleneck module (left). The absorbing strategy is illustrated on the left. Specifically, during the t-th session, the C-adapter is trained with images from the current domain. After this session concludes, the parameters of the C-adapter trained in sessions $\mathbf{1} \sim \boldsymbol{t-1}$ are fused with the current C-adapter. This updated C-adapter, now representing the t-th session, is subsequently used for training in the following session. The D-adapter (right) differs from the C-adapter by incorporating an additional 2D convolutional layer, which is applied on image tokens after reshaping operations. Each D-adapter is added in parallel. During each session, a new D-adapter is trained on current images and saved without modification. The next session repeats this process to get the new D-adapter.

extreme domain shift between natural images and chest X-ray images, we make the first domain data to help attention blocks in ViT to learn specialized medical knowledge about lung diseases. During the incremental learning process, we design D(omain)-adapter and C(lass)-adapter (the dual-adapter layer) to learn each domain private message and total domain shared category message respectively, where the rest of the backbone is frozen during training. In the testing stage, select the domain-specific D-adapter according to feature similarities and integrate the selected D-adapter and the unified C-adapter into the network.

Specifically, given an image $x \in \mathbb{R}^{H\times W\times 3}$, where H, W denote the height and width of the image respectively, and 3 is the channels. After the patch embedding layer, x is flattened into patch tokens $x_{img} \in \mathbb{R}^{N\times(P^2\times 3)}$, where (P, P) is the resolution of each image patch, $N = HW/P^2$ is the number of image tokens. A [CLS] token $x_{cls} \in \mathbb{R}^{1\times(P^2\times 3)}$ is concatenated with image tokens x_{img} and we get a feature map $M \in \mathbb{R}^{(N+1)\times(P^2\times 3)}$. Then, the feature map $M \in \mathbb{R}^{(N+1)\times d}$ (d is equal to $(P^2 \times 3)$) is fed into DA-block in order. Each DA block contains a multi-head self-attention (MHSA) layer, a dual-adapter layer, and an MLP layer.

The output $M^l_{SA,out} \in \mathbb{R}^{(N+1)\times d}$ of MHSA layer is sent to C-adapter and D-adapter, and gets $M^l_{C,out}, M^l_{D,out} \in \mathbb{R}^{(N+1)\times d}$ respectively. The output of the dual-adapter layer can be expressed as:

$$M^l_{DA,out} = M^l_{C,out} + M^l_{D,out} + M^l_{SA,out} \quad (1)$$

Then, $M^l_{DA,out} \in \mathbb{R}^{(N+1)\times d}$ passes the MLP layer and we get $M^l_{out} \in \mathbb{R}^{(N+1)\times d}$.

After the last DA block, the [CLS] token and image tokens are split from M^l_{out}, the former passes into the classifier for the image recognition, and the latter are the image features stored by K-Means.

C. Cross-domain Invariant Feature Absorption (CIFA) module with C-adapter

To excavate and preserve the shared knowledge across different domains, we propose a C(lass)-adapter and integrate it into DA blocks in the ViT model.

In ViT, the MLP structure is crucial for feature extraction. The C-adapter, comprising two fully connected layers, enhances the model's nonlinearity, improving its ability to capture complex features. Although the neural network includes numerous dense layers with full-rank matrices to boost performance, [55] demonstrates that a large model can be projected onto a smaller subspace without substantially impairing learning capability. This suggests that simple, lower-parameter layers can approximate the performance of larger models. Consequently, a low-dimensional feature representation can maintain high accuracy while filtering out unnecessary noise. Instead of increasing the intermediate dimension of the C-adapter, we reduce it, which helps mitigate overfitting without a significant accuracy loss.

Another consideration for the design of the C-adapter is the parameter efficiency. Reducing the feature dimensions

significantly reduces the number of parameters that need to be trained. Compared with directly conducting large-scale parameter training on a high dimension, this operation can reduce the computational complexity.

As shown in Fig. 3, each C-adapter consists of a linear down-projection layer $W_{C,down} \in \mathbb{R}^{d\times d'}$, an activation layer, and a linear up-projection layer $W_{C,up} \in \mathbb{R}^{d'\times d}$, where d' is middle dimension. After an image $x \in \mathbb{R}^{H\times W\times 3}$ passes through the MHSA layer, we could obtain a feature map $M^l_{C,in} \in \mathbb{R}^{(N+1)\times d}$ (it is the same as $M^l_{SA,out}$), where d represents the dimension and l is the l-th block of image encoder. Because the MHSA layer does not change the size of its input feature map, the input of the C-adapter is still the shape of $M^l_{C,in} \in \mathbb{R}^{(N+1)\times d}$. During the C-adapter, the down-projection layer reduces the dimension into a more compact representation with dimension d' and gets $M^l_{C,mid} \in \mathbb{R}^{(N+1)\times d'}$. Then, the up-projection layer restores $M^l_{C,mid}$ to the original dimension d (it becomes the $\tilde{M}^l_C \in \mathbb{R}^{(N+1)\times d}$) and connects to the original feature map $M^l_{C,in}$ through the residual connection. If we use *ReLU* as an activation function, the output feature map is $M^l_{C,out} \in \mathbb{R}^{(N+1)\times d}$, then the C-adapter can be expressed as:

$$M^l_{C,mid} = \textit{ReLU}(\textit{LN}(M^l_{C,in}) \cdot \boldsymbol{W}_{C,down}) \tag{2}$$

$$\tilde{M}^l_C = \textit{LN}(M^l_{C,mid}) \cdot \boldsymbol{W}_{C,up} \tag{3}$$

$$M^l_{C,out} = \textit{ReLU}(\tilde{M}^l_C + \textit{shortcut}(M^l_{C,in})) \tag{4}$$

where the *LN* is the layer normalization operation [56] and the *shortcut* represents the residual connection.

After each incremental session training, we absorb the last C-adapter parameters to the current C-adapter. Specifically, in the first session, we save parameters of the C-adapter trained on the base domain data, which is continually trained across all the sessions. In the session t, we update weights $W^{l,t}_{down}$,$W^{l,t}_{up}$, and bias $b^{l,t}_{down}$,$b^{l,t}_{up}$, where $W^{l,t}_{down} \in \mathbb{R}^{d\times d'}$, $W^{l,t}_{up} \in \mathbb{R}^{d'\times d}$ and $b^{l,t}_{down} \in \mathbb{R}^{d'}$, $b^{l,t}_{up} \in \mathbb{R}^{d}$ are the down-projection layer and up-projection layer parameters of C-adapter in the l-th DA block respectively, L is the total block number of image encoder. Then we add current weights $W^{l,t}_{down}, W^{l,t}_{up}$ and bias $b^{l,t}_{down}, b^{l,t}_{up}$ to saved weights $W^{l,1\sim t-1}_{down}, W^{l,1\sim t-1}_{up}$ and bias $b^{1\sim t-1}_{down}, b^{1\sim t-1}_{up}$, and calculate average to get new weights $W^{l,1\sim t}_{down}, W^{l,1\sim t}_{up}$ and bias $b^{l,1\sim t}_{down}, b^{l,1\sim t}_{up}$. The new parameters will be saved and used in the next session. For t-th session, the above absorb strategy can be described by:

$$W^{l,1\sim t}_{down} = \frac{1}{t}((t-1)\cdot W^{l,1\sim t-1}_{down} + W^{l,t}_{down}) \tag{5}$$

$$W^{l,1\sim t}_{up} = \frac{1}{t}((t-1)\cdot W^{l,1\sim t-1}_{up} + W^{l,t}_{up}) \tag{6}$$

$$b^{l,1\sim t}_{down} = \frac{1}{t}((t-1)\cdot b^{l,1\sim t-1}_{down} + b^{l,t}_{down}) \tag{7}$$

$$b^{l,1\sim t}_{up} = \frac{1}{t}((t-1)\cdot b^{l,1\sim t-1}_{up} + b^{l,t}_{up}) \tag{8}$$

D. Domain-Specific Feature Retention (DSFR) module with D-adapter

We design a series of D-adapters to learn and store domain-specific information. Different from the C-adapter, the D-adapter contains a 2D convolutional layer to preserve spatial details. Intuitively, relying solely on fully connected layers for feature extraction in a new domain may result in suboptimal performance due to a lack of locality, while convolutional layers can capture local priors in images. In medical imaging, local features are especially critical because diseased areas often appear only in specific regions. Fully connected layers map global features across the entire image, which may hinder the model's focus on key areas. For an incremental new domain, making the D-adapter structure identical to the C-adapter is insufficient, thus, adding a convolutional layer is necessary. Additionally, this convolutional layer is lightweight within the D-adapter, introducing minimal parameters and enabling efficient training.

For each image, we extract the image tokens $x'_{img} \in \mathbb{R}^{N\times d'}$ from the feature map $M^l_{D,mid} \in \mathbb{R}^{(N+1)\times d'}$, and resize it into $\tilde{x}_{img} \in \mathbb{R}^{H/P\times W/P\times d'}$:

$$M^l_{D,mid} = \textit{ReLU}(\textit{LN}(M^l_{D,in}) \cdot \boldsymbol{W}_{D,down}) \tag{9}$$

$$x'_{img}, x_{cls} = M^l_{D,mid}[1:,:], M^l_{D,mid}[1,:] \tag{10}$$

$$\tilde{x}_{img} = \text{Reshape}(x'_{img}) \tag{11}$$

Then, the $\tilde{x}_{img}$ is fed into the 2D convolutional layer and captures the fine-grained features. After that, it is reshaped back to the size of x'_{img} and becomes new image tokens $x_{img} \in \mathbb{R}^{N\times d'}$. The image tokens after the convolutional operation will concatenate with [CLS] token and we get a feature map $\tilde{M}^l_{D,mid} \in \mathbb{R}^{(N+1)\times d'}$. The process of generating $\tilde{M}^l_{D,mid}$ is as follows:

$$x_{img} = \text{Reshape}(\text{Conv2D}(\tilde{x}_{img})) \tag{12}$$

$$\tilde{M}^l_{D,mid} = \textit{ReLU}(\textit{Concat}([x_{img}, x_{cls}])) \tag{13}$$

After that, the $\tilde{M}^l_{D,mid}$ is fed into the up-projection layer and obtained $\tilde{M}^l_D \in \mathbb{R}^{(N+1)\times d}$. The output $M^l_{D,out}$ maintains detailed information by residual connection, which can be written as:

$$\tilde{M}^l_D = \textit{LN}(\tilde{M}^l_{D,mid}) \cdot \boldsymbol{W}_{D,up} \tag{14}$$

$$M^l_{D,out} = \textit{ReLU}(\tilde{M}^l_D + \textit{shortcut}(M^l_{D,in})) \tag{15}$$

To avoid interference between different incremental data, as shown in Fig. 3, each series of D-adapter is added in parallel independently. When trained on a new session, a new series D-adapter is updated. Correspondingly, the classifier follows the same way as the D-adapter, which will be fine-tuned during the new incremental session. We use a normal fully connected (FC) layer as the classifier, and the parameters of these layers are saved after training.

The domain-id is unknown during the inference, so after finishing each training session, the model features (the feature map without [CLS] token before the classifier layer) with unique D-adapter are clustered by K-Means. Each learning session owns one clustering center after K-Means and the center will be saved. When finishing all incremental sessions, there is a set of adapter and classifier parameters (layer set), as well as a set of clustering centers (center set). The elements in these two sets are in one-to-one correspondence. The layer set can be defined as $P = \{P^1, P^2, ..., P^T\}$, where

$P^t = [P^t_{C-adapter}, P^t_{D-adapter}, P^t_{fc}]$, and the center set is $A = \{A^1, A^2, ..., A^T\}$. During the inference, each one of the layer set processes features of given test images and there will be a feature set $F = \{F^1, F^2..., F^T\}$. By calculating the distances between *t*-th features and all centers in the center set, we can get the shortest distance and its corresponding serial number, which is the domain-id of this domain feature. After the domain-id is certain, the associated adapters and classifier will be chosen for predicting these test images. For the test images of the *t*-th session, the domain-id can be determined by:

$$domain\_id^t = \arg\min_{1 \le i \le T} \sqrt{(A^t - F^i)^2} \tag{16}$$

E. Loss Function

The model is trained on L_{cxr-il}, which consists of two components. One is the classification loss $L_{soft-bce}$ (it is the binary cross-entropy loss with weights), and the other is the multi-label contrastive loss $L_{soft-con}$. The total loss can be written as:

$$L_{cxr-il} = L_{soft-bce} + \lambda L_{soft-con} \tag{17}$$

where λ is a hyper-parameter to control the trade-off between two losses.

The first term of L_{cxr-il} is $L_{soft-bce}$, which is used to calculate the distance between image prediction and multiple labels. For the predicted category probabilities of images, it can be symbolized as

$$\mathbf{p} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & \cdots & p_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ p_{C1} & p_{C2} & \cdots & p_{CN} \end{bmatrix} \in \mathbb{R}^{C \times N}$$

, then, the loss is formulated as:

$$\begin{aligned} L_{soft-bce} = -\frac{1}{C} \cdot \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} (y_{ij} \cdot log(\sigma(p_{ij})) \\ +(1 - y_{ij}) \cdot log(1 - \sigma(p_{ij}))) \end{aligned} \tag{18}$$

where *C* and *N* are the number of categories and the number of images respectively. p_{ij} represents the prediction probability that image *i* belongs to category *j* and y_{ij} is its corresponding label. σ is the sigmoid function.

Another term is the $L_{soft-con}$ inspired by contrastive learning [57]. In common contrastive learning tasks, each image has a single label, so for any two instances, there are only two scenarios: they either belong to the same class (positive samples) or they belong to different classes (negative samples). However, for two instances with multi-labels, they may share partial or all the labels or have completely different labels. Therefore, we optimize the InfoNCE loss [58] to adjust complex co-occurrence relationships between labels. Specifically, in a data batch of size *b*, our $L_{soft-con}$ will output all the other instances for a specific instance *i*, where $i \in N(i)$, $N(i) = \{k | k \in \{1, 2, \cdots, b\}, k \neq i\}$. The loss can be written as:

$$L_{soft-con} = -\sum_{j=1}^{N(i)} r_{ij} log \frac{exp(\langle f_i, f_j \rangle)}{\sum_{k=1}^{N(i)} exp(\langle f_i, f_k \rangle)} \tag{19}$$

where r_{ij} is the label similarity between instance *i* and instance *j*, $f_i = f(x_i)$ denotes the prediction of image encoder, $\langle , \rangle$ represents the cosine similarity.

IV. Experimental Results

A. Datasets

We evaluated the method on five chest X-ray datasets, which are ChestXray14 [30], Chexpert [31], MIMIC-CXR [32], VinBig [33] and OpenI [34]. ChestXray14 [30] has 14 common chest pathological descriptions and a category of no-finding with 112120 images. For the Chexpert dataset [31] and MIMIC-CXR dataset [32], 224316 images and 377110 images are labeled respectively. Due to following the same marking rules, they have the same 14 labels. The fourth dataset is the VinBig dataset [33], which comes from a chest X-ray image competition, and contains 15k images and 14 different labels. The OpenI dataset [34] is the last one, comprising 7465 images and covering 18 labels. We filter each dataset to include the shared labels and the frontal view. For the ChestXray14, Chexpert, and MIMIC-CXR datasets, we follow the official split of the train/test set, and for each one of the rest, 20% is randomly chosen as an independent testing set, while the rest 80% of the samples are used as training.

The domain gaps of datasets. We assess the significant domain differences between datasets by calculating the Maximum Mean Discrepancy (MMD) [59] for each domain, thereby demonstrating the existence of these domain gaps. MMD is a widely used metric for evaluating the differences between two distributions and is commonly applied in domain adaptation tasks [60]–[64]. It measures the mean deviation across domains and can be formulated as follows:

$$\mathrm{MMD}^2_k(X, Y) = \left\| \frac{1}{N} \sum_{i=1}^{N} \phi(x_i) - \frac{1}{M} \sum_{j=1}^{M} \phi(y_i) \right\|^2_{\mathcal{H}_k} \tag{20}$$

where $\mathcal{H}_k$ is the Reproducing Kernel Hilbert Space (RKHS) and refers to the Hilbert space with reproducing kernel *k*. *X* and *Y* are two different distributions and $X = \{x_i\}_i^N$, $Y = \{y_j\}_j^M$ are the corresponding samples respectively. The mapping function is ϕ.

TABLE I
MMD OF EVERY TWO DOMAINS AMONG 5 CHEST X-RAY IMAGE DATASETS.

MMD	ChestXray14	VinBig	Chexpert	MIMIC-CXR	OpenI
ChestXray14	——				
VinBig	0.034	——			
Chexpert	1.560	0.208	——		
MIMIC-CXR	0.049	0.065	0.183	——	
OpenI	0.014	0.015	0.201	0.016	——

We calculate the MMD between every two datasets and report the values on Table I. As a benchmark, we calculate MMD on the Office-Home dataset [65], commonly used in domain adaptation research [66]–[68]. This dataset comprises four distinct domains: Artistic images, Clip Art images, Product images, and Real-World images. Table II presents the MMD values between each domain pair, with higher MMD values indicating greater domain discrepancies.

TABLE II
MMD OF EVERY TWO DOMAINS AMONG 4 DOMAINS OF OFFICE-HOME DATASET.

MMD	Art	Clipart	Product	Real World
Art	——			
Clipart	0.011	——		
Product	0.011	0.0073	——	
Real World	0.010	0.0072	0.0071	——

From Table I, each MMD value in Table I exceeds 0.011 (the highest value recorded in Table II), indicating a significant difference in sample distributions. These observations highlight the significant domain gaps present in our study.

B. Evaluation Protocols

For the multiple-step incremental process, at step t, the model is trained on the specific dataset D^t without access to any other datasets. Among all the steps, each dataset shares the same category set C. To cover the most common chest diseases, we examine two orders with three learning steps and the hardest order with five steps. Specifically, the Order A is: *ChestXray14* $\rightarrow$ *Chexpert* $\rightarrow$ *MIMIC-CXR*, and it contains the shared 7 labels with Cardiomegaly, Pneumonia, Pneumothorax, Consolidation, Edema, Atelectasis, Effusion. The Order B is: *ChestXray14* $\rightarrow$ *VinBig* $\rightarrow$ *OpenI*, and it contains the shared 9 labels with Cardiomegaly, Pleural_Thickening, Pneumothorax, Atelectasis, Effusion, Infiltration, Mass, Nodule, Fibrosis. The Order C is: *ChestXray14* $\rightarrow$ *VinBig* $\rightarrow$ *Chexpert* $\rightarrow$ *MIMIC-CXR* $\rightarrow$ *OpenI*, and it contains the shared 4 labels with Atelectasis, Cardiomegaly, Effusion, Pneumothorax. Besides, we also exchange the position of datasets in each order to demonstrate the robustness of our method.

We adopt the common evaluation metric, AUROC (the Area Under the Receiver Operating Characteristic Curve) to evaluate the disease classification results. Following the DIL works on natural images, we report the AUC of each dataset during the current session and the decreased AUC on old domains that have been learned. After each incremental session, we also report the average AUC of all datasets.

C. Implementation Details

Our method adopts the ImageNet-1k [54] pre-trained ViT-B/16 [37] as the backbone (All compared methods use the same backbone). For each incremental session, the model is trained using the SGD optimizer of 35 epochs. The batch size is set to 32 and the training patch size is $3 \times 224 \times 224$. Default the learning rate of 0.01 and λ of 0.1 on multi-GPU (2). For data augmentation, we adopt horizontal flipping, multi-scale crop, and random affine transformation for each dataset.

For comparison methods, we use the same ViT-B/16 model as the backbone and follow the same hyper-parameter settings and consistent data augmentation among all methods. The baseline method is 'FT', which is to fine-tune the model without any anti-forgetting skill. 'upper bound' is to train the model by combining data from all past domains and current domains together. Besides, we also compared the latest leading architecture-based methods on natural images including L2P [28] and S-Prompts [29], the state-of-the-art method (VQ [46]) on medical images, regularization-based method like EWC_on [26] and rehearsal-based method like ER [45].

D. Comparison Results

1) Overall Performance: Table III, Table IV, Table V, and Table VI present the results for each domain across sessions in Orders A, B, and C. The "All" column provides the average AUC values across all trained domains. In Table III, Table IV, each method is represented by three rows. The first row is the first session, which includes only one domain with no degree of forgetting. The second row is the second session, there is a new incremental domain and we get two AUCs of the new domain and the old domain. The drop of the first domain is calculated. In the third row, with the addition of a third domain, we provide AUC values for all three domains. The forgetting degrees of the first and second domains are reported after $\downarrow$. Table V and Table VI follow the same format, with the only difference being that the session is five.

Results on Order A. From Table III, our method achieves the best mean AUC, which is 84.29% on Order A-1 and 83.80% on Order A-2 respectively. Our method outperforms the second best performing method by 3.97% on Order A-1, and 2.98% on Order A-2.

As the number of incremental sessions increases, the forgetting of previous domains becomes more pronounced, as observed with the FT baseline method. The AUC drop in FT is significant for old domains, reaching as high as 21.44% on Order A-1. The ER method, which utilizes a buffer, has a performance that varies considerably depending on the sequence of domains. Specifically, the AUC drop for ER is smaller on Order A-2 than on Order A-1, indicating that this method does not have strong order robustness. Moreover, preserving model features (VQ) causes less forgetting than preserving the samples of datasets (ER) on large-scale datasets, for example, VQ shows better than ER in preserving old knowledge on Order A. The drop AUC of ER is higher by 1.83% and 3.94% than that of VQ. The performances of these rehearsal-based methods are unstable when the domain order changes, which is similar to EWC_on.

Most existing leading methods (L2P and S-Prompts) using pre-trained models show poor performance on medical images unless they are transferred first. This two-stage strategy is redundant but our method is end-to-end and resource-saving. Without transferring, the AUCs of these methods on old domains increase gradually, as more and more medical information becomes available. This is a process to adapt to medical images instead of preventing forgetting. Therefore, the drop AUCs are negative values in some suborders. The advanced methods of natural images are not suitable for medical images. We focus on the final session because the long incremental sessions show the ability of anti-forgetting truly and accurately. Across all suborder scenarios of Order A, our approach consistently surpasses others, achieving an average AUC improvement of at least 2.98% and maintaining

TABLE III

DOMAIN-INCREMENTAL RESULTS ON ORDER A-1 AND ORDER A-2 ON EACH SESSION. THE AUCS(%) OF EACH DATASET AND THE MEAN AUC (ALL) ARE EVALUATED. THE BEST RESULTS ARE MARKED IN BOLD.

Methods	**Buffer size**	ChestXray14 AUC / Forgrt	Chexpert AUC / Forgrt	MIMIC-CXR AUC / Forgrt	All AUC / Forgrt	MIMIC-CXR AUC / Forgrt	Chexpert AUC / Forgrt	ChestXray14 AUC / Forgrt	All AUC / Forgrt
		Order A-1: ChestXray14 → Chexpert → MIMIC-CXR				Order A-2: MIMIC-CXR → Chexpert → ChestXray14			
upper bound	——	78.55	96.11	81.82	85.49	81.78	94.28	78.92	84.99
FT	0	78.26 ↓—— / 63.26 ↓15.00 / 56.82 ↓21.44	94.17 ↓—— / 86.53 ↓7.64	**81.43** ↓——	78.26 ↓—— / 78.72 ↓15.00 / 74.93 ↓14.54	79.83 ↓—— / 70.83 ↓9.00 / 74.67 ↓5.16	93.61 ↓—— / 80.00 ↓13.61	**78.81** ↓——	79.83 ↓—— / 82.22 ↓9.00 / 77.83 ↓9.38
ER [45]	1000	74.35 ↓—— / 68.12 ↓6.23 / 67.26 ↓7.09	93.61 ↓—— / 87.22 ↓6.39	74.91 ↓——	74.35 ↓—— / 80.87 ↓6.23 / 76.46 ↓6.74	79.87 ↓—— / 78.47 ↓1.40 / 76.55 ↓3.32	93.61 ↓—— / 85.70 ↓7.91	78.81 ↓——	79.87 ↓—— / 86.04 ↓1.40 / 80.35 ↓5.61
EWC_on [26]	0	78.20 ↓—— / 73.74 ↓4.46 / 73.53 ↓4.67	95.27 ↓—— / 86.53 ↓8.74	80.94 ↓——	78.20 ↓—— / 84.51 ↓4.46 / 80.32 ↓6.71	80.31 ↓—— / 73.06 ↓7.25 / 74.63 ↓5.68	93.61 ↓—— / 86.94 ↓6.67	78.32 ↓——	80.31 ↓—— / 83.33 ↓7.25 / 79.96 ↓6.17
L2P [28]	0	70.92 ↓—— / 70.92 ↓0.00 / 70.92 ↓0.00	61.54 ↓—— / 62.22 ↓-0.68	66.06 ↓——	70.92 ↓—— / 66.23 ↓0.00 / 66.40 ↓-0.34	73.73 ↓—— / 73.73 ↓0.00 / 73.73 ↓0.00	73.59 ↓—— / 73.59 ↓0.00	61.78 ↓——	73.73 ↓—— / 73.66 ↓0.00 / 69.70 ↓0.00
S-Prompts [29]	0	51.81 ↓—— / 51.81 ↓0.00 / 60.66 ↓-8.85	60.65 ↓—— / 70.54 ↓-9.89	70.35 ↓——	51.81 ↓—— / 56.23 ↓0.00 / 67.18 ↓-9.37	52.98 ↓—— / 52.98 ↓0.00 / 62.12 ↓-9.14	60.65 ↓—— / 83.35 ↓-22.70	71.25 ↓——	52.98 ↓—— / 56.82 ↓0.00 / 72.24 ↓-15.92
VQ [46]	1000	78.18 ↓—— / 72.04 ↓6.14 / 71.31 ↓6.87	94.21 ↓—— / 91.26 ↓2.95	76.09 ↓——	78.18 ↓—— / 83.12 ↓6.14 / 79.55 ↓4.91	79.79 ↓—— / 77.38 ↓2.41 / 76.74 ↓3.05	93.61 ↓—— / **93.33** ↓0.28	72.40 ↓——	79.79 ↓—— / 85.49 ↓2.41 / 80.82 ↓1.67
Ours	0	78.33 ↓—— / 78.33 ↓0.00 / **78.33** ↓0.00	95.27 ↓—— / **95.27** ↓0.00	79.27 ↓——	78.33 ↓—— / 86.80 ↓0.00 / **84.29** ↓0.00	81.07 ↓—— / 81.07 ↓0.00 / **81.07** ↓0.00	93.19 ↓—— / 93.19 ↓0.00	77.15 ↓——	81.07 ↓—— / 87.13 ↓0.00 / **83.80** ↓0.00

TABLE IV

DOMAIN-INCREMENTAL RESULTS ON ORDER B-1 AND ORDER B-2 ON EACH SESSION. THE AUCS(%) OF EACH DATASET AND THE MEAN AUC (ALL) ARE EVALUATED. THE BEST RESULTS ARE MARKED IN BOLD.

Methods	**Buffer size**	ChestXray14 AUC / Forgrt	VinBig AUC / Forgrt	OpenI AUC / Forgrt	All AUC / Forgrt	VinBig AUC / Forgrt	ChestXray14 AUC / Forgrt	OpenI AUC / Forgrt	All AUC / Forgrt
		Order B-1: ChestXray14 → VinBig → OpenI				Order B-2: VinBig → ChestXray14 → OpenI			
upper bound	——	78.82	81.36	81.77	80.65	78.32	80.45	78.24	79.00
FT	0	78.47 ↓—— / 69.44 ↓9.03 / 66.43 ↓12.04	81.36 ↓—— / 69.28 ↓12.08	**80.83** ↓——	78.47 ↓—— / 75.40 ↓9.03 / 72.18 ↓12.06	78.32 ↓—— / 64.56 ↓13.76 / 63.88 ↓14.44	78.61 ↓—— / 70.73 ↓7.88	**76.88** ↓——	78.32 ↓—— / 71.58 ↓13.76 / 70.49 ↓11.16
ER [45]	1000	73.79 ↓—— / 69.77 ↓4.02 / 69.17 ↓4.62	79.84 ↓—— / 71.39 ↓8.45	74.80 ↓——	73.79 ↓—— / 74.81 ↓4.02 / 71.79 ↓6.54	76.34 ↓—— / 60.59 ↓15.75 / 61.23 ↓15.11	77.67 ↓—— / **72.80** ↓4.87	77.79 ↓——	76.34 ↓—— / 69.13 ↓15.75 / 70.61 ↓9.99
EWC_on [26]	0	78.43 ↓—— / 67.28 ↓11.15 / 65.98 ↓12.45	79.54 ↓—— / 65.46 ↓14.08	74.99 ↓——	78.43 ↓—— / 73.41 ↓11.15 / 68.81 ↓13.27	78.44 ↓—— / 62.55 ↓15.89 / 61.22 ↓17.22	78.49 ↓—— / 66.66 ↓11.83	74.57 ↓——	78.44 ↓—— / 70.52 ↓15.89 / 67.48 ↓14.53
L2P [28]	0	70.94 ↓—— / 70.94 ↓0.00 / 70.94 ↓0.00	54.49 ↓—— / 54.49 ↓0.00	57.35 ↓——	70.94 ↓—— / 62.71 ↓0.00 / 60.89 ↓0.00	70.07 ↓—— / 69.46 ↓0.61 / 69.85 ↓0.22	56.21 ↓—— / 56.21 ↓0.00	61.11 ↓——	70.07 ↓—— / 62.84 ↓0.61 / 62.39 ↓0.11
S-Prompts [29]	0	49.02 ↓—— / 49.02 ↓0.00 / 53.54 ↓-4.52	50.09 ↓—— / 51.38 ↓-1.29	59.56 ↓——	49.02 ↓—— / 49.56 ↓0.00 / 54.83 ↓-2.91	49.66 ↓—— / 48.84 ↓0.82 / 50.01 ↓-0.35	49.02 ↓—— / 53.54 ↓-4.52	59.56 ↓——	49.66 ↓—— / 48.93 ↓0.82 / 54.37 ↓-2.44
VQ [46]	1000	78.35 ↓—— / 74.71 ↓3.64 / 76.60 ↓1.75	74.67 ↓—— / 62.41 ↓12.26	68.39 ↓——	78.35 ↓—— / 74.69 ↓3.64 / 69.13 ↓7.01	78.06 ↓—— / 66.15 ↓11.91 / 66.14 ↓11.92	68.43 ↓—— / 67.82 ↓0.61	66.76 ↓——	78.06 ↓—— / 67.29 ↓11.91 / 66.91 ↓6.26
Ours	0	78.52 ↓—— / 78.52 ↓0.00 / **78.52** ↓0.00	78.93 ↓—— / **78.93** ↓0.00	76.85 ↓——	78.52 ↓—— / 78.73 ↓0.00 / **78.10** ↓0.00	78.15 ↓—— / 78.15 ↓0.00 / **78.15** ↓0.00	72.69 ↓—— / 72.69 ↓0.00	72.84 ↓——	78.15 ↓—— / 75.42 ↓0.00 / **74.56** ↓0.00

a 0.00% AUC drop. These demonstrate our proposed method has strong order robustness and leads to no forgetting.

Results on Order B. Table IV summaries each session results of all the methods on Order B. Our method reaches the mean AUC of 78.10% on Order B-1 and 74.56% on Order B-2, and outperforms second method 5.92% on Order B-1, 3.95% on Order B-2.

The data distribution of Order B changes much more drastically than that of Order A, so there are some different conclusions. It is worth noting that rehearsal-based methods

TABLE V

DOMAIN-INCREMENTAL RESULTS ON ORDER C-1 ON EACH SESSION. THE AUCS(%) OF EACH DATASET AND THE MEAN AUC (ALL) ARE EVALUATED. THE BEST RESULTS ARE MARKED IN BOLD.

Methods	**Buffer size**	ChestXray14 AUC	ChestXray14 Forgrt	VinBig AUC	VinBig Forgrt	Chexpert AUC	Chexpert Forgrt	MIMIC-CXR AUC	MIMIC-CXR Forgrt	OpenI AUC	OpenI Forgrt	All AUC	All Forgrt
		Order C-1: ChestXray14 → VinBig → Chexpert → MIMIC-CXR → OpenI											
upper bound	——	82.11		91.74		98.21		98.50		91.43		92.39	
FT	0	82.07	↓——									82.07	↓——
		72.21	↓9.86	87.67	↓——							79.94	↓9.86
		75.63	↓6.44	76.63	↓11.04	98.21	↓——					83.49	↓8.74
		79.70	↓2.37	73.78	↓13.89	95.17	↓3.04	82.67	↓——			82.83	↓6.43
		72.48	↓9.59	66.11	↓21.56	89.04	↓9.17	73.90	↓8.77	**88.09**	↓——	77.92	↓12.27
ER [45]	1000	81.80	↓——									81.80	↓——
		74.51	↓7.29	91.46	↓——							82.99	↓7.29
		80.47	↓1.33	80.79	↓10.67	96.25	↓——					85.83	↓6.00
		79.56	↓2.24	72.57	↓18.89	94.99	↓1.26	84.11	↓——			82.81	↓7.46
		75.64	↓6.16	68.64	↓22.82	90.71	↓5.54	79.74	↓4.37	85.38	↓——	80.02	↓9.72
EWC_on [26]	0	81.35	↓——									81.35	↓——
		70.58	↓10.77	91.74	↓——							81.16	↓10.77
		73.11	↓8.24	63.05	↓28.69	96.25	↓——					77.47	↓18.47
		79.79	↓1.56	70.69	↓21.05	89.64	↓6.61	83.37	↓——			80.87	↓9.74
		72.23	↓9.12	72.21	↓19.53	88.39	↓7.86	71.61	↓11.76	82.82	↓——	77.45	↓12.07
L2P [28]	0	72.53	↓——									72.53	↓——
		72.53	↓0.00	53.93	↓——							63.23	↓0.00
		72.53	↓0.00	54.17	↓-0.24	72.26	↓——					66.32	↓-0.12
		72.53	↓0.00	54.26	↓-0.33	72.26	↓0.00	68.24	↓——			66.82	↓-0.11
		72.53	↓0.00	64.49	↓-10.56	72.26	↓0.00	68.24	↓0.00	63.59	↓——	68.22	↓-2.64
S-Prompts [29]	0	45.82	↓——									45.82	↓——
		45.82	↓0.00	41.92	↓——							43.87	↓0.00
		45.82	↓0.00	44.65	↓-2.73	20.77	↓——					37.08	↓-1.37
		45.82	↓0.00	45.82	↓-3.90	20.77	↓0.00	46.98	↓——			39.85	↓-1.30
		45.82	↓0.00	43.35	↓-1.43	20.77	↓0.00	46.98	↓0.00	62.05	↓——	43.79	↓-0.36
VQ [46]	1000	82.03	↓——									82.03	↓——
		77.53	↓4.50	81.09	↓——							79.31	↓4.50
		78.98	↓3.05	75.48	↓5.61	87.00	↓——					80.48	↓4.33
		77.92	↓4.11	59.64	↓21.45	87.00	↓0.00	78.75	↓——			75.82	↓8.52
		78.03	↓4.00	63.96	↓17.13	87.00	↓0.00	78.06	↓0.69	79.63	↓——	77.34	↓5.45
Ours	0	81.97	↓——									81.97	↓——
		81.97	↓0.00	86.68	↓——							84.33	↓0.00
		81.97	↓0.00	86.68	↓0.00	95.00	↓——					87.88	↓0.00
		81.97	↓0.00	86.68	↓0.00	95.00	↓0.00	81.09	↓——			86.19	↓0.00
		81.97	↓0.00	**86.68**	↓0.00	**95.00**	↓0.00	**81.09**	↓0.00	84.18	↓——	**85.78**	↓0.00

show opposite results to those on Order A. Specifically, the ability of VQ to learn new knowledge is weaker than ER on Order B. On the other hand, the regularization-based method (EWC_on) also produces less satisfactory predictions in this situation compared to those on Order A. For L2P and S-Prompts, ours outperforms both of them by a large margin, which shows the outstanding work on natural images is inapplicable to medical images. When order changes, our method is flexible and efficient and performs far ahead of others.

Results on Order C. For Order C, Table V indicates that the mean AUC on the final session is 85.78% and higher 5.76% than the second method. To evaluate order robustness, we modify the order of multiple datasets in Order C, with the experimental results presented in Table VI. Except for OpenI, the order of the remaining four datasets has been adjusted. We arrive at the same conclusion as with Order C-1: our method achieves the highest mean AUC of 84.77% across both the new domain and all previous domains in the final session. Importantly, the change in order does not affect the degree of forgetting, which remains at 0.00%. These show our proposed method suits longer incremental learning scenarios and avoids memory costs.

E. Performance on 3D data

1) Experimental setup: We adapt our DIL protocol to 3D medical images. In clinical settings, CT scans are the primary modality for diagnosing chest diseases. Accordingly, we collect three COVID-19 CT datasets (CNCB [69], Covid_CT_MD [70], and Covid_Radiopaedia) to evaluate our method. The CNCB dataset includes 58,766 CT scans across three categories: COVID-19 pneumonia, common pneumonia, and normal controls. Covid_CT_MD contains 4,957 slices showing infection and 18,392 normal slices. Covid_Radiopaedia, sourced from the radiology website Radiopaedia, includes 4,057 CT scans with the same categories as CNCB. We structure the dataset stream for the DIL task by following the chest X-ray data sequence and designate the order as *CNCB* →

TABLE VI
DOMAIN-INCREMENTAL RESULTS ON ORDER C-2 ON EACH SESSION. THE AUCS(%) OF EACH DATASET AND THE MEAN AUC (ALL) ARE EVALUATED. THE BEST RESULTS ARE MARKED IN BOLD.

Methods	Buffer size	VinBig		ChestXray14		MIMIC-CXR		Chexpert		OpenI		All	
		AUC	Forgrt	AUC	Forgrt	AUC	Forgrt	AUC	Forgrt	AUC	Forgrt	AUC	Forgrt
		Order C-2: VinBig → ChestXray14 → MIMIC-CXR → Chexpert → OpenI											
upper bound	——	91.89		91.74		84.05		98.50		91.43		91.52	
FT	0	90.51	↓——									90.51	↓——
		72.95	↓17.56	81.90	↓——							77.42	↓17.56
		75.65	↓14.86	80.85	↓1.05	83.96	↓——					80.15	↓7.96
		67.35	↓23.16	75.57	↓6.33	78.99	↓4.97	95.17	↓——			79.27	↓11.49
		71.38	↓19.13	73.26	↓8.64	76.58	↓7.38	90.89	↓4.28	**90.52**	↓——	80.52	↓9.85
ER [45]	1000	88.50	↓——									88.50	↓——
		74.89	↓13.61	81.39	↓——							78.14	↓13.61
		73.18	↓15.32	80.74	↓0.65	82.14	↓——					78.68	↓7.99
		68.31	↓20.19	75.76	↓5.63	80.55	↓1.59	96.24	↓——			80.21	↓9.14
		67.06	↓21.44	72.44	↓8.95	74.64	↓7.50	86.96	↓9.28	88.87	↓——	77.99	↓11.79
EWC_on [26]	0	91.38	↓——									91.38	↓——
		70.50	↓20.88	80.89	↓——							75.69	↓20.88
		72.33	↓19.05	79.04	↓1.85	82.49	↓——					77.95	↓10.45
		66.92	↓24.46	71.41	↓9.48	74.87	↓7.62	97.50	↓——			77.67	↓13.85
		62.66	↓28.72	70.82	↓10.07	76.73	↓5.76	88.92	↓8.58	84.86	↓——	76.79	↓13.28
L2P [28]	0	81.03	↓——									81.03	↓——
		80.28	↓0.75	52.05	↓——							66.16	↓0.75
		78.98	↓2.05	52.05	↓0.00	56.84	↓——					62.62	↓1.02
		79.23	↓1.80	52.11	↓-0.06	56.80	↓0.04	52.66	↓——			60.20	↓0.59
		81.25	↓-0.22	52.53	↓-0.48	57.41	↓-0.57	55.93	↓-3.27	54.80	↓——	60.38	↓-1.13
S-Prompts [29]	0	45.45	↓——									45.45	↓——
		41.55	↓3.90	45.82	↓——							43.68	↓3.90
		41.42	↓4.03	45.82	↓0.00	46.98	↓——					44.74	↓2.01
		46.52	↓-1.07	45.82	↓0.00	46.98	↓0.00	20.77	↓——			40.02	↓-0.35
		42.21	↓3.24	45.82	↓0.00	46.98	↓0.00	20.77	↓0.00	64.84	↓——	44.12	↓0.81
VQ [46]	1000	90.26	↓——									90.26	↓——
		86.95	↓3.31	72.81	↓——							79.88	↓3.31
		84.35	↓5.91	71.48	↓1.33	76.86	↓——					77.56	↓3.62
		78.88	↓11.38	70.45	↓2.36	74.04	↓2.82	78.85	↓——			75.55	↓5.52
		79.68	↓10.58	71.76	↓1.05	73.89	↓2.97	73.72	↓5.13	77.99	↓——	75.40	↓4.93
Ours	0	91.43	↓——									91.43	↓——
		91.43	↓0.00	76.02	↓——							83.73	↓0.00
		91.43	↓0.00	76.02	↓0.00	78.62	↓——					82.02	↓0.00
		91.43	↓0.00	76.02	↓0.00	78.62	↓0.00	93.12	↓——			84.80	↓0.00
		91.43	↓0.00	**76.02**	↓0.00	**78.62**	↓0.00	**93.12**	↓0.00	84.65	↓——	**84.77**	↓0.00

Covid_CT_MD → *Covid_Radiopaedia* (Order D). Since 3D CT images consist of stacks of 2D slices, we divide each CT scan into individual 2D slices and treat them as separate 2D images, a common and efficient approach in 3D medical imaging [71]–[74]. For consistency, we use the same backbone (ViT-B/16) and data augmentation settings as in the DIL task for chest X-ray images. In terms of metrics, we report the F1 score (rather than AUC for 2D data) and use the F1 score drop to quantify the degree of forgetting.

2) comparison results: Table VII presents the results for 3D CT data. Our method outperforms all other methods in terms of the mean F1 score, which is 98.66%, leading the second-best method by 1.96% in the last session. Additionally, it maintains a forgetting degree of 0.00% across all sessions, surpassing the second method by 0.86% in the last session. As the number of incremental sessions increases, our method demonstrates superior performance and reduced forgetting. While L2P achieves the same 0.00% drop in F1 score for old domains, it struggles to effectively learn new domains. In contrast, VQ shows slight forgetting in old domains while effectively acquiring new knowledge. However, it requires a large buffer to store and replay features, which is inefficient in terms of storage and computational resources. Our method, on the other hand, only necessitates a few additional parameters and remains lightweight. These results further validate the effectiveness of our method for 3D medical modalities, highlighting its superior performance on CT images.

F. Generalization

To mitigate the risk of overfitting, we evaluate our model on an unseen dataset, Padchest [75], which includes over 160,000 images and an extensive category set. Adhering to the DIL standard, we ensure that the label set remains consistent throughout the incremental learning process. We filter the Padchest images based on the category set of Order A and test Padchest across all compared methods without any additional training and data augmentation techniques.

TABLE VII

DOMAIN-INCREMENTAL RESULTS ON ORDER D ON EACH SESSION. THE F1 SCORES (%) OF EACH DATASET AND THE MEAN F1 SCORE (ALL) ARE EVALUATED. THE BEST RESULTS ARE MARKED IN BOLD.

Methods	**Buffer size**	CNCB F1	CNCB Forgrt	Covid_CT_MD F1	Covid_CT_MD Forgrt	Covid_Radiopaedia F1	Covid_Radiopaedia Forgrt	All F1	All Forgrt
		Order D: CNCB → Covid_CT_MD → Covid_Radiopaedia							
upper bound	——	100.00		99.10		100.00		99.70	
FT	0	99.83	↓——					99.83	↓——
		96.17	↓3.66	98.18	↓——			97.17	↓3.66
		78.44	↓21.39	85.89	↓12.29	**99.77**	↓——	88.03	↓16.84
ER [45]	1000	99.92	↓——					99.92	↓——
		95.95	↓3.97	98.55	↓——			97.25	↓3.97
		84.04	↓15.88	96.17	↓2.38	99.65	↓——	93.28	↓9.13
EWC_on [26]	0	99.78	↓——					99.78	↓——
		95.44	↓4.34	98.55	↓——			96.99	↓4.34
		77.86	↓21.92	74.14	↓24.41	99.26	↓——	83.75	↓23.16
L2P [28]	0	99.79	↓——					99.79	↓——
		99.79	↓0.00	68.14	↓——			83.96	↓0.00
		99.79	↓0.00	68.14	↓0.00	93.77	↓——	87.23	↓0.00
S-Prompts [29]	0	11.02	↓——					11.02	↓——
		10.34	↓0.68	25.17	↓——			17.75	↓0.68
		65.22	↓-54.20	69.14	↓-43.97	98.08	↓——	77.48	↓-49.08
VQ [46]	1000	99.87	↓——					99.87	↓——
		99.25	↓0.62	94.34	↓——			96.79	↓0.62
		99.42	↓0.45	93.08	↓1.26	97.60	↓——	96.70	↓0.86
Ours	0	99.96	↓——					99.96	↓——
		99.96	↓0.00	96.58	↓——			98.27	↓0.00
		99.96	↓0.00	**96.58**	↓0.00	99.43	↓——	**98.66**	↓0.00

TABLE VIII

GENERALIZATION RESULTS ON THE UNSEEN TEST SET OF ORDER A-1. THE AUCS(%) OF PADCHEST ARE REPORTED. THE BEST RESULT IS MARKED IN BOLD.

Methods	**Buffer size**	Padchest AUC
Order A-1: ChestXray14 → Chexpert → MIMIC-CXR		
FT	0	71.37
ER [45]	1000	73.05
EWC_on [26]	0	71.86
L2P [28]	0	65.75
S-Prompts [29]	0	61.37
VQ [46]	1000	73.89
Ours	0	**74.45**

From Table VIII, our method demonstrates the highest performance on the unseen Padchest dataset, achieving an AUC of 74.45%, which is 0.56% higher than the second-best method, VQ. The baseline method, FT, suffers from overfitting in the final domain, limiting its generalizability. Both ER and VQ leverage a buffer to retain prior information, preserving richer medical features and enabling them to perform reasonably well on unseen data. However, architecture-based methods such as L2P and S-prompts struggle, showing low test accuracy on unseen datasets due to insufficient medical knowledge acquisition.

Meanwhile, for our method, the C-adapter effectively stores invariant domain features, enhancing performance on the unseen Padchest domain. This shared knowledge aligns closely with the new dataset, underscoring our method's superior generalization capability. Furthermore, the results indicate that our method can classify medical images in a zero-shot setting.

G. Ablation Results

1) The contribution of dual-adapter: Table IX presents the ablation results for the C-adapter and D-adapter on Order A-1 within the incremental learning process, clearly demonstrating the effectiveness of these components. In Table IX, Row 1 represents the baseline without adapters, using a training strategy where the backbone remains frozen throughout incremental sessions. Without adapter involvement, the model struggles to generalize to new domains, achieving a mean AUC of 76.96%, which is 7.33% lower than our approach in Row 4. Both the C-adapter and D-adapter play critical roles in enhancing recognition accuracy, with the D-adapter showing a greater impact on the medical image classification task. Specifically, the C-adapter improves accuracy by 3.95%, while the D-adapter contributes an increase of 6.91%. The C-adapter structure is simpler compared to that of the D-adapter, and the fusion strategy pays more attention on the shared domain knowledge, resulting in incomplete learning of new domains. In contrast, the D-adapter, with an additional convolutional layer, offers stronger feature extraction capability. Given the significant difference of each domain, learning the features of each domain separately yields a more comprehensive understanding of domain knowledge. That is why the D-adapter contributes greater than the C-adapter. However, when the DA layer is applied, the mean AUC reaches 84.29%, which has a large gap of other configurations. In this case, the domain-specific features learned by the D-adapter are complemented

TABLE IX

D-ADAPTER AND C-ADAPTER WHOLE ABLATION EXPERIMENTS ON ORDER A-1. WE REPORT THE MEAN AUC(%) ON EACH DOMAIN TEST SET ON THE LAST SESSION.

C-adapter	D-adapter	ChestXray14 AUC	Chexpert AUC	MIMIC-CXR AUC	All AUC
		Order A-1: ChestXray14 → Chexpert → MIMIC-CXR			
		72.80	82.55	75.54	76.96
✓		77.87	88.35	76.53	80.91
	✓	78.18	95.27	78.16	83.87
✓	✓	**78.33**	**95.27**	**79.27**	**84.29**

TABLE X

D-ADAPTER AND C-ADAPTER WHOLE ABLATION EXPERIMENTS ON ORDER B-1. WE REPORT THE MEAN AUC(%) ON EACH DOMAIN TEST SET ON THE LAST SESSION.

C-adapter	D-adapter	ChestXray14 AUC	VinBig AUC	OpenI AUC	All AUC
		Order B-1: ChestXray14 → VinBig → OpenI			
		75.17	73.22	69.59	72.66
✓		78.37	73.34	70.98	74.23
	✓	78.41	78.76	73.81	76.99
✓	✓	**78.52**	**78.93**	**76.85**	**78.10**

TABLE XI

D-ADAPTER ABLATION EXPERIMENTS ON ORDER A-1. WE REPORT THE MEAN AUC(%) ON EACH DOMAIN TEST SET ON THE LAST SESSION.

embed form		embed layers				middle dimension *d'*					ChestXray14 AUC	Chexpert AUC	MIMIC-CXR AUC	All AUC
seq	parallel	1-4	5-8	9-12	1-12	64	128	512	768	1024	Order A-1: ChestXray14 → Chexpert → MIMIC-CXR			
	✓				✓	✓					**78.33**	**95.27**	**79.27**	**84.29**
✓					✓	✓					77.81	93.19	77.31	82.77
	✓				✓	✓					78.33	**95.27**	**79.27**	**84.29**
	✓	✓				✓					**78.34**	92.77	76.85	82.65
	✓		✓			✓					78.26	94.33	77.42	83.33
	✓			✓		✓					77.99	91.83	78.23	82.68
	✓				✓	✓					78.33	**95.27**	**79.27**	**84.29**
	✓				✓		✓				78.41	92.22	78.99	83.21
	✓				✓			✓			**78.56**	91.11	76.84	82.17
	✓				✓				✓		78.39	93.19	78.93	83.50
	✓				✓					✓	78.25	90.27	79.16	82.56

TABLE XII

2D CONVOLUTIONAL LAYER OF D-ADAPTER ABLATION EXPERIMENTS ON ORDER A-1. WE REPORT THE MEAN AUC(%) ON EACH DOMAIN TEST SET ON THE LAST SESSION.

2D conv layer				ChestXray14 AUC	Chexpert AUC	MIMIC-CXR AUC	All AUC
True			False	Order A-1: ChestXray14 → Chexpert → MIMIC-CXR			
1×1	3×3	5×5					
			✓	78.18	88.15	78.16	81.50
✓				78.33	**95.27**	**79.27**	**84.29**
	✓			**78.45**	94.05	79.17	83.89
		✓		77.57	87.65	76.68	**80**.63

by domain-invariant features. Consequently, these two modules work synergistically and are indispensable.

From Table X, we reach a conclusion consistent with that from Table IX. Row 2 presents the results without the D-adapter, showing a 1.57% increase in mean AUC over the baseline (Row 1). When only the D-adapter is added, the mean AUC improves by 4.33% compared to Row 1. Applying both the C-adapter and D-adapter achieves the highest accuracy, with the mean AUC reaching 78.10%. This result strongly validates the necessity of each component and the effectiveness of our modules. The structure is dataset-agnostic and exhibits high robustness.

2) Effectiveness of D-adapter: Table XI demonstrates the results on order A-1 of D-adapter, we compare different embed types and different embed positions of the backbone, as well as different middle dimensions d' of the down-projection linear.

TABLE XIII
C-ADAPTER ABLATION EXPERIMENTS ON ORDER A-1. WE REPORT THE MEAN AUC(%) ON EACH DOMAIN TEST SET ON THE LAST SESSION.

embed form		embed layers		middle dimension d'					Chestxray14	Chexpert	MIMIC-CXR	All
seq	parallel	front	back	64	128	512	768	1024	AUC	AUC	AUC	AUC
Order A-1: ChestXray14 → Chexpert → MIMIC-CXR												
	✓		✓	✓					**78.33**	**95.27**	**79.27**	**84.29**
✓			✓	✓					78.21	93.89	77.78	83.29
	✓		✓	✓					**78.33**	**95.27**	**79.27**	**84.29**
	✓	✓		✓					78.19	93.67	77.73	83.20
	✓		✓	✓					**78.33**	**95.27**	**79.27**	**84.29**
	✓		✓		✓				78.10	93.61	76.89	82.87
	✓		✓			✓			78.02	91.53	77.44	82.33
	✓		✓				✓		78.15	91.11	78.67	82.64
	✓		✓					✓	78.09	89.44	79.03	82.18

First, the parallel way to add a D-adapter shows better performance than the sequential way. The AUC of each dataset decreases when the sequential D-adapter is used. Specifically, the AUC drops 0.52% in ChestXray14, 2.08% in Chexpert, 1.96% in MIMIC-CXR, and the average of three domains drops 1.52%.

Second, we test the impact of the D-adapter structure on experimental results at different positions of the model. For ViT-B/16 with 12 blocks, Rows 3-6 of Table XI show the average AUC on all 12 blocks, the first 4 blocks, the middle 4 blocks, and the last 4 blocks with D-adapter respectively. Because shallow layers and deep layers capture different features, unfreezing each block for training learns knowledge more comprehensively. Although the highest AUC of ChestXray14 appears when the front layers unfix (78.34%), on Chexpert and MIMIC-CXR, it appears on all layers (95.27% and 79.27%).

Third, we adjust the D-adapter complexity by changing the middle dimension d'. As the dimension increases, the performance of the model does not show significant improvement, instead, it leads to an increase in complexity. For instance, as the dimension increases from 64 to 512, the AUC of Chexpert and MIMIC-CXR both decrease significantly, which could be seen by comparing Rows 7-11 in Table XI. With the dimension increasing larger, we draw a similar conclusion to that in the C-adapter.

Last, when the D-adapter is modified to match the C-adapter without a convolutional layer, the average AUC drops by 2.79%. The first row of Table XII shows results across datasets without the 2D convolutional layer, with AUC reductions observed in each domain: ChestXray14 decreases by 0.15%, Chexpert by 7.12%, and MIMIC-CXR by 1.11%. Furthermore, we evaluate different kernel sizes, finding that the highest mean AUC is achieved with a 1×1 convolution layer. Although the 3×3 convolution layer performs slightly lower (by 0.40% compared to the 1×1 layer), it still exceeds the performance of the no-convolution configuration by 2.39%. These findings affirm the effectiveness of the 2D convolutional layer in extracting fine details and local features.

3) Effectiveness of C-adapter: Row 1 and Row 2 in Table XIII present the classification results using the C-adapter in parallel and serially. The AUC of each domain drops when we connect C-adapter modules in series into the backbone.

For the positions of model blocks, we follow the same setting as the D-adapter to test its influence. The C-adapter embedded in later blocks of the model enhances the model's capability to extract domain-shared information. The AUC when the C-adapter at the back is 1.09% higher than that in the front, which proves the deeper blocks will learn more knowledge about categories.

The ablation experimental results about dimension d' can be seen in Rows 5-9 of Table XIII, and we draw a similar conclusion to that of D-adapter. The highest accuracy is reached when $d' = 64$ (the mean AUC among three domains is higher 1.42% than that when $d' = 128$, 1.96% than that when $d' = 512$, 1.65% than that when $d' = 768$ and 2.11% than that when $d' = 1024$). We set the larger dimension d', not all domains show improved AUC, but it causes greater waste of resources.

TABLE XIV
COMPUTATIONAL COMPLEXITY OF THE D-ADAPTER AND C-ADAPTER ON ORDER A-1. WE REPORT THE FLOPS AND PARAMETERS, ALONG WITH THEIR RESPECTIVE PROPORTIONS RELATIVE TO THE BACKBONE.

C-adapter	D-adapter	FLOPs		Params	
Order A-1: ChestXray14→Chexpert→MIMIC-CXR					
✓		58.14M	0.33%	295.87K	0.31%
	✓	697.68M	3.31%	3.99M	3.48%
✓	✓	755.82M	3.64%	4.29M	3.79%

4) The complexity of C-adapter and D-adapter: We compute the FLOPs of the C-adapter and D-adapter, as shown in Table XIV. The FLOPs for the C-adapter and D-adapter are 58.14M and 697.68M, respectively, which constitute only 0.33% and 3.31% of the backbone's FLOPs. This indicates that both the C-adapter and D-adapter modules are low in complexity and highly efficient during inference. These modules are lightweight, making them suitable for fast training on limited hardware, which is advantageous for deployment in hospitals or regions with underdeveloped medical infrastructure.

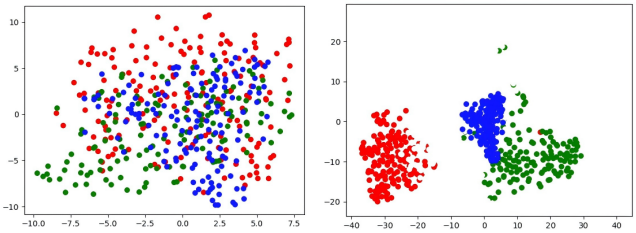
Additionally, we report the parameters of each component: the C-adapter introduces 295.87K parameters, and the D-adapter introduces 3.99M parameters. This demonstrates that our DA layer is lightweight and parameter-efficient, allowing for the integration of multiple parallel instances within the

TABLE XV

OTHER FINE-TUNING METHODS RESULTS ON ORDER A-1 ON EACH SESSION. THE AUCS(%) OF EACH DATASET AND THE MEAN AUC (ALL) ARE EVALUATED. THE BEST RESULTS ARE MARKED IN BOLD.

Methods	Buffer size	ChestXray14 AUC	ChestXray14 Forgrt	Chexpert AUC	Chexpert Forgrt	MIMIC-CXR AUC	MIMIC-CXR Forgrt	All AUC	All Forgrt
		Order A-1: ChestXray14 → Chexpert → MIMIC-CXR							
upper bound	—	78.55		96.11		81.82		85.49	
FT	0	78.26	↓——					78.26	↓——
		63.26	↓15.00	94.17	↓——			78.72	↓15.00
		56.82	↓21.44	86.53	↓7.64	**81.43**	↓——	74.93	↓14.54
LORA [76]	0	73.45	↓——					73.45	↓——
		73.45	↓0.00	85.85	↓——			79.65	↓0.00
		73.45	↓0.00	85.85	↓0.00	77.15	↓——	78.82	↓0.00
BitFit [77]	0	73.84	↓——					73.84	↓——
		73.84	↓0.00	88.19	↓——			81.02	↓0.00
		73.84	↓0.00	88.19	↓0.00	76.52	↓——	79.51	↓0.00
Prefix-Tuning [78]	0	74.91	↓——					74.91	↓——
		74.91	↓0.00	86.45	↓——			80.68	↓0.00
		74.91	↓0.00	86.45	↓0.00	77.95	↓——	79.77	↓0.00
P-Tuning [79]	0	78.20	↓——					78.20	↓——
		78.20	↓0.00	95.27	↓——			86.74	↓0.00
		78.20	↓0.00	95.27	↓0.00	74.99	↓——	82.82	↓0.00
Ours	0	78.33	↓——					78.33	↓——
		78.33	↓0.00	95.27	↓——			86.80	↓0.00
		78.33	↓0.00	**95.27**	↓0.00	79.27	↓——	**84.29**	↓0.00

backbone without significantly increasing storage demands. This efficiency is particularly beneficial for long-term incremental learning scenarios, holding significant application value in real-world medical settings.



(a) C-adapter (b) D-adapter

Fig. 4. Visualization of C-adapter and D-adapter

5) Visualization of C-adapter and D-adapter: Fig. 4 presents the T-SNE visualizations of the C-adapter and D-adapter on Order A-1. In panel (a) on the left, we visualize the C-adapter's output by excluding the D-adapter, while panel (b) on the right shows the D-adapter's output after removing the C-adapter. Red, green, and blue points represent the ChestXray14, CheXpert, and MIMIC-CXR datasets, respectively.

As shown in Fig. 4, the C-adapter captures more common features across domains, resulting in feature representations, which intermingle from different domains. In contrast, the D-adapter preserves domain-specific knowledge, leading to distinct clustering of each domain. This visualization intuitively demonstrates the characteristics of each module and aligns well with our design intentions.

6) The comparison of other fine-tuning methods: We compare other representative fine-tuning skills with our C-adapter and D-adapter, including LORA (Low-Rank Adaptation) [76], BitFit (Bias-Only Fine-Tuning) [77], Prefix-Tuning [78], and P-Tuning (Prompt-Tuning) [79]. We apply these fine-tuning methods to vision transformers and compare them under the same backbone (ViT-B/16) and hyper-parameter settings as used in our method on Order A-1.

From Table XV, our method has the best performance among all compared methods, which is higher 1.47% than the second method on the mean AUC. Although the forgetting rate is still 0.00% among these fine-tuning methods, the ability to learn new domains has a large gap with ours. For P-Tuning, although it performs better than the other three methods in the first two domains, it is significantly lower in the third domain. This illustrates that P-Tuning cannot suit long continual steps. While Prefix-Tuning shows comparable performance to ours on the last domain, it still lags significantly on the first domain. Consequently, these fine-tuning methods are not well-suited for incremental learning in the medical imaging scenario.

7) Influence of backbone: To assess the impact of model architecture, we conduct ablation studies using different CNN-based backbones, specifically ResNet with varying depths. In the case of ResNet, each block consists of three convolutional layers: a 1×1 convolutional layer, a $K \times K$ convolutional layer, and another 1×1 convolutional layer. Analogous to the embedding position in ViT, we insert our DA layer between the $K \times K$ and 1×1 convolutional layers in ResNet. The structural details of the C-adapter and D-adapter remain consistent with those in ViT-B/16. For comparative analysis, we also evaluate the FT method using ResNet as the backbone, with all data processing and optimizer settings kept identical. To illustrate the effect of different backbones, we present an ablation table (Table XVI) that includes results for ViT-B/16, as well as ResNet101 and ResNet152, which are based on CNN architectures.

TABLE XVI

DIFFERENT BACKBONES COMPARED RESULTS ON ORDER A-1 ON EACH SESSION. THE AUCS(%) OF EACH DATASET AND THE MEAN AUC (ALL) ARE EVALUATED. THE BEST RESULTS ARE MARKED IN BOLD.

Methods	**Backbone**	ChestXray14 AUC Forgrt	Chexpert AUC Forgrt	MIMIC-CXR AUC Forgrt	All AUC Forgrt
		Order A-1: ChestXray14 → Chexpert → MIMIC-CXR			
upper bound	ResNet101	77.46	94.32	81.78	84.52
FT	ResNet101	77.19 ↓——			77.19 ↓——
		64.81 ↓12.38	93.49 ↓——		79.15 ↓12.38
		54.49 ↓22.70	85.27 ↓8.22	**81.55** ↓——	73.77 ↓15.46
ours	ResNet101	77.39 ↓——			77.39 ↓——
		77.39 ↓0.00	89.11 ↓——		83.25 ↓0.00
		77.39 ↓0.00	**89.11** ↓0.00	77.35 ↓——	**81.28** ↓0.00
upper bound	ResNet152	78.46	95.28	81.67	85.14
FT	ResNet152	77.10 ↓——			77.10 ↓——
		63.88 ↓13.22	90.27 ↓——		77.07 ↓13.22
		54.26 ↓22.84	80.27↓10.00	**81.03** ↓——	71.85 ↓16.42
ours	ResNet152	78.08 ↓——			78.08 ↓——
		78.08 ↓0.00	93.75 ↓——		85.91 ↓0.00
		78.08 ↓0.00	**93.75** ↓0.00	77.50 ↓——	**83.11** ↓0.00
upper bound	ViT-B/16	78.55	96.11	81.82	85.49
FT	ViT-B/16	78.26 ↓——			78.26 ↓——
		63.26 ↓15.00	94.17 ↓——		78.72 ↓15.00
		56.82 ↓21.44	86.53 ↓7.64	**81.43** ↓——	74.93 ↓14.54
ours	ViT-B/16	78.33 ↓——			78.33 ↓——
		78.33 ↓0.00	95.27 ↓——		86.80 ↓0.00
		78.33 ↓0.00	**95.27** ↓0.00	79.27 ↓——	**84.29** ↓0.00

TABLE XVII

D-ADAPTER AND C-ADAPTER WHOLE ABLATION EXPERIMENTS WITH RESNET152 ON ORDER A-1. WE REPORT THE MEAN AUC(%) ON EACH DOMAIN TEST SET ON THE LAST SESSION.

C-adapter	D-adapter	ChestXray14 AUC	Chexpert AUC	MIMIC-CXR AUC	All AUC
		Backbone: ResNet152			
		Order A-1: ChestXray14 → Chexpert → MIMIC-CXR			
		74.42	82.77	71.85	76.34
✓		77.23	83.95	72.12	77.76
	✓	77.27	92.86	72.14	80.76
✓	✓	**78.08**	**93.75**	**77.50**	**83.11**

Table XVI presents the results across different backbone architectures. Compared with FT using the same backbone, our method demonstrates significant improvements, with a minimum gain of 7.51%. This indicates that catastrophic forgetting persists when switching to a ResNet backbone. In the case of ResNet, models with more parameters tend to achieve higher AUCs in the last session. Specifically, the AUC of ResNet152 is 1.83% higher than that of ResNet101 using our method. However, all ResNet-based results are consistently lower than those obtained with ViT-B/16, at least by 1.18%. This suggests that the DA layer collaborates more effectively with the MLP layer in feature extraction when paired with ViT, reinforcing the suitability of our method for ViT architectures. Furthermore, regardless of the backbone used, our method achieves a 0.00% forgetting rate, confirming that our CaD framework leads to no forgetting. This finding highlights the robustness of our approach across different architectures.

To further highlight the contributions of each component (the C-adapter and the D-adapter) within a CNN model, we conducted an ablation study on Order A-1 using ResNet152, which can be seen in Table XVII.

Table XVII presents the ablation results for each component using ResNet152. Both the C-adapter and D-adapter contribute to improving accuracy. Referring to Table IX, we observe that without any adapter, the mean AUC remains similar across backbones, achieving 76.34% with ResNet152 and 76.96% with ViT-B/16. Only using the C-adapter results in a mean AUC increase of 3.95% with ViT-B/16 and 1.42% with ResNet152. In contrast, only using the D-adapter yields more substantial improvements, achieving a mean AUC of 83.87% on ViT-B/16 and 80.76% on ResNet152. Notably, the D-adapter consistently contributes more than the C-adapter across different backbones. Combining both the D-adapter and C-adapter results in the best performance, boosting mean AUC by 7.33% with ViT-B/16 and 6.77% with ResNet152. This demonstrates that our DA layer significantly enhances accuracy across different backbones.

8) Influence of MHSA layer: **Experimental setup**. We add the ablation studies about the MHSA layer and show the results in Table XVIII. We remove the MHSA layer of the ViT backbone and report AUC on the last session on Order A-1, which can be seen in Row 1 of Table XVIII.

TABLE XVIII
MHSA LAYER ABLATION EXPERIMENTS ON ORDER A-1. WE REPORT THE MEAN AUC(%) ON EACH DOMAIN TEST SET ON THE LAST SESSION.

MHSA layer					ChestXray14	Chexpert	MIMIC-CXR	All
True				False	AUC	AUC	AUC	AUC
MHSA	+MHSA	+SHSA	+Conv		Order A-1: ChestXray14 → Chexpert → MIMIC-CXR			
				✓	50.00	50.00	50.00	50.00
✓					**78.33**	**95.27**	**79.27**	**84.29**
	✓				78.23	92.13	77.79	82.72
		✓			**78.15**	**92.36**	**79.13**	**83.21**
			✓		77.41	92.17	78.83	82.80

Since we adopt the ImageNet-1k pre-trained ViT-B/16 as the backbone, the parameters matrix of each layer remains fixed and unchanged. Using pre-trained models for incremental learning is common in natural image tasks [28], [29]. Therefore, changing the MHSA layer to other feature extraction layers, such as convolutional layers, proves unfeasible and impractical. The original MHSA layer structure must remain intact to ensure compatibility with the pre-trained parameters. Based on this, we design the ablation study for the MHSA layer as follows: in addition to the original MHSA layer, we introduce additional feature extraction layers, including the MHSA layer, the SHSA layer, and the convolutional layer, to evaluate their contributions.

Specifically, the SHSA layer is a single-head self-attention layer, where the number of heads is reduced from 12 in the MHSA layer to 1. For the conv layer, we split the feature map $M \in \mathbb{R}^{(N+1)\times d}$ into image tokens $x_{img} \in \mathbb{R}^{N\times d}$ and [CLS] token $x_{cls} \in \mathbb{R}^{1\times d}$. The image tokens are reshaped to $M_{conv} \in \mathbb{R}^{H\times W\times 3}$. The $M_{conv} \in \mathbb{R}^{H\times W\times 3}$ is fed into the conv layer and get the output $M'_{conv} \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times(3\times2\times2)}$. After the conv layer, the feature map $M'_{conv} \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times(3\times2\times2)}$ is reshaped to $\tilde{M}_{conv} \in \mathbb{R}^{N\times d}$. Then, the $\tilde{M}_{conv} \in \mathbb{R}^{N\times d}$ concatenates [CLS] token, and we obtain the feature map $M_{conv,out} \in \mathbb{R}^{N\times(d+1)}$. $M_{conv,out} \in \mathbb{R}^{N\times(d+1)}$ is fed into the DA block, which sequentially passes through the MHSA layer, the DA layer, and the MLP layer. The process is formulated as:

$$x_{img}, x_{cls} = M^l[1:,:], M^l[1,:] \tag{21}$$

$$M^l_{conv} = \mathrm{Reshape}(x_{img}) \tag{22}$$

$$M'^l_{conv} = \mathit{ReLU}(\mathit{LN}(\mathrm{Conv2D}(M^l_{conv}))) \tag{23}$$

$$\tilde{M}^l_{conv} = \mathrm{Reshape}(M'^l_{conv}) \tag{24}$$

$$M^l_{conv,out} = \mathit{Concat}([\tilde{M}^l_{conv}, x_{cls}]) + \mathit{shortcut}(M^l) \tag{25}$$

where l means the l-th block.

Ablation results. Rows 1-2 in Table XVIII present the performance outcomes with and without the MHSA layer. When the backbone lacks the MHSA layer, it reduces to a simple multilayer perceptron with skip connections. This structure fails to capture the complex features of medical images, leading to a catastrophic drop in classification accuracy, with the AUC across each domain fixed at 50.00%. These results highlight the indispensable role of the MHSA layer in feature extraction. However, due to the model's uniformly poor classification across datasets, it does not exhibit any capacity to mitigate forgetting in the DIL task.

On the other hand, Rows 3-5 in Table XVIII reveal that the SHSA layer achieves the highest mean AUC at 83.21%, surpassing the MHSA layer and the convolutional layer by 0.49% and 0.41%, respectively. However, it is 1.08% lower than when using only a single MHSA layer (Row 2). When the MHSA layer within the backbone is changed, although it is fixed during incremental sessions, our C-adapter and D-adapter are influenced slightly. All configurations still yield mean AUCs exceeding 82% and effectively prevent forgetting. These results demonstrate that our C-adapter and D-adapter effectively address the forgetting issue in the DIL task and operate with resilience to changes in the MHSA layers.

TABLE XIX
SENSITIVE STUDY OF HYPER-PARAMETER λ ON ORDER A-1. WE REPORT THE MEAN AUC(%) OF EACH TEST SET ON THE LAST SESSION.

	ChestXray14	Chexpert	MIMIC-CXR	All
	AUC	AUC	AUC	AUC
	Order A-1: ChestXray14 → Chexpert → MIMIC-CXR			
Baseline λ=0	78.01	88.32	78.45	81.59
λ=0.1	**78.33**	**95.27**	**79.27**	**84.29**
λ=0.3	78.12	89.86	79.03	82.33
λ=0.5	77.80	90.27	79.26	82.44
λ=1	77.75	94.44	79.10	83.76
λ=5	71.08	88.09	71.96	77.04

9) Influence of Hyper-parameter λ: We conduct extensive experiments on Order A-1 to evaluate the effect of hyper-parameter λ. The comparison results can be seen in Table XIX, detailing the average AUC achieved with $\lambda = 0, 0.1, 0.3, 0.5, 1, 5$ respectively. Compared to the Row 1 ($\lambda = 0$), applying the $L_{soft-con}$ mitigates the catastrophic forgetting significantly. Observations of intermediate variables show that incorporating $L_{soft-con}$ makes the domain centers obtained through clustering more distinguishable, enhancing the matching accuracy between image features and domain centers. Specifically, the mean AUCs of each session increase by 0.32%, 6.95%, and 0.82% when $\lambda = 0.1$ compared to $\lambda = 0$. We conduct more experiments of different λ and find the best performance is when $\lambda = 0.1$ (where the mean AUC of session 1 is 78.33%, and it is 95.27% and 79.27% of session 2 and session 3 respectively). Results for other values of λ, reported in the remaining rows of Table XIX, show lower performance levels.

V. Conclusion

In this paper, we focus on domain incremental learning on chest X-ray images. Due to the restrictions on medical data privacy and complex relationships of pathology labels, to solve this problem, we propose an architecture-based method named the CaD framework. Specifically, it contains two different modules, which are Cross-domain Invariant Feature Absorption module (CIFA) with a C-adapter and Domain-Specific Feature Retention module (DSFR) with the D-adapters respectively. The D-adapter captures each domain's unique knowledge and has an extra 2D convolutional layer. The C-adapter captures shared knowledge, and we follow an absorbing strategy to merge past knowledge into current knowledge. To model co-occurrence relationships between labels accurately, a multi-label contrastive loss is used in the incremental training process. Leveraging multiple publicly available mainstream datasets, we simulate domain incremental scenarios in real clinical settings by designing 3 different data streams. We also exchange the order of domains to verify the robustness. From extensive experimental results, our method outperforms existing state-of-the-art methods both on medical images and natural images.

References

[1] T. Franquet, "Imaging of pneumonia: trends and algorithms," *European Respiratory Journal*, vol. 18, no. 1, pp. 196–208, 2001.

[2] F. Wu, S. Zhao, B. Yu, Y.-M. Chen, W. Wang, Z.-G. Song, Y. Hu, Z.-W. Tao, J.-H. Tian, Y.-Y. Pei, *et al.*, "A new coronavirus associated with human respiratory disease in china," *Nature*, vol. 579, no. 7798, pp. 265–269, 2020.

[3] C. Huang, Y. Wang, X. Li, L. Ren, J. Zhao, Y. Hu, L. Zhang, G. Fan, J. Xu, X. Gu, *et al.*, "Clinical features of patients infected with 2019 novel coronavirus in wuhan, china," *The lancet*, vol. 395, no. 10223, pp. 497–506, 2020.

[4] W. H. Organization *et al.*, "Pneumonia of unknown cause-china. emergencies preparedness, response web site," 2020.

[5] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, *et al.*, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *arXiv preprint arXiv:1711.05225*, 2017.

[6] S. Taslimi, S. Taslimi, N. Fathi, M. Salehi, and M. H. Rohban, "Swinchex: Multi-label classification on chest x-ray images with transformers," *arXiv preprint arXiv:2206.04246*, 2022.

[7] E. Kim, S. Kim, M. Seo, and S. Yoon, "Xprotonet: diagnosis in chest radiography with global and local explanations," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15719–15728, 2021.

[8] Y. Fang, H. Zhang, J. Xie, M. Lin, L. Ying, P. Pang, and W. Ji, "Sensitivity of chest ct for covid-19: comparison to rt-pcr," *Radiology*, vol. 296, no. 2, pp. E115–E117, 2020.

[9] M. M. Hammer, "The evolution of covid-19: omicron and subvariants," 2023.

[10] L. Luo, L. Yu, H. Chen, Q. Liu, X. Wang, J. Xu, and P.-A. Heng, "Deep mining external imperfect data for chest x-ray disease screening," *IEEE transactions on medical imaging*, vol. 39, no. 11, pp. 3583–3594, 2020.

[11] K. Sanchez, C. Hinojosa, H. Arguello, D. Kouamé, O. Meyrignac, and A. Basarab, "Cx-dagan: Domain adaptation for pneumonia diagnosis on a small chest x-ray dataset," *IEEE Transactions on Medical Imaging*, vol. 41, no. 11, pp. 3278–3288, 2022.

[12] M. Lenga, H. Schulz, and A. Saalbach, "Continual learning for domain adaptation in chest x-ray classification," in *Medical Imaging with Deep Learning*, pp. 413–423, PMLR, 2020.

[13] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio, "An empirical investigation of catastrophic forgetting in gradient-based neural networks," *arXiv preprint arXiv:1312.6211*, 2013.

[14] R. Kemker, M. McClure, A. Abitino, T. Hayes, and C. Kanan, "Measuring catastrophic forgetting in neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

[15] A. Robins, "Catastrophic forgetting, rehearsal and pseudorehearsal," *Connection Science*, vol. 7, no. 2, pp. 123–146, 1995.

[16] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

[17] Y. Liu, B. Schiele, and Q. Sun, "Rmm: Reinforced memory management for class-incremental learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 3478–3490, 2021.

[18] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. K. Dokania, P. H. Torr, and M. Ranzato, "On tiny episodic memories in continual learning," *arXiv preprint arXiv:1902.10486*, 2019.

[19] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, "Dark experience for general continual learning: a strong, simple baseline," *Advances in neural information processing systems*, vol. 33, pp. 15920–15930, 2020.

[20] T. L. Hayes, K. Kafle, R. Shrestha, M. Acharya, and C. Kanan, "Remind your neural network to prevent catastrophic forgetting," in *European Conference on Computer Vision*, pp. 466–483, Springer, 2020.

[21] A. Iscen, J. Zhang, S. Lazebnik, and C. Schmid, "Memory-efficient incremental learning through feature adaptation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pp. 699–715, Springer, 2020.

[22] L. Qu, N. Balachandar, M. Zhang, and D. Rubin, "Handling data heterogeneity with generative replay in collaborative learning for medical imaging," *Medical image analysis*, vol. 78, p. 102424, 2022.

[23] W. Sun, Z. Cai, Y. Li, F. Liu, S. Fang, and G. Wang, "Security and privacy in the medical internet of things: a review," *Security and Communication Networks*, vol. 2018, pp. 1–9, 2018.

[24] C. Baweja, B. Glocker, and K. Kamnitsas, "Towards continual learning in medical imaging," *arXiv preprint arXiv:1811.02496*, 2018.

[25] H. Ravishankar, R. Venkataramani, S. Anamandra, P. Sudhakar, and P. Annangi, "Feature transformers: privacy preserving lifelong learners for medical imaging," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part IV 22*, pp. 347–355, Springer, 2019.

[26] J. Schwarz, W. Czarnecki, J. Luketina, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, and R. Hadsell, "Progress & compress: A scalable framework for continual learning," in *International conference on machine learning*, pp. 4528–4537, PMLR, 2018.

[27] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[28] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, and T. Pfister, "Learning to prompt for continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 139–149, 2022.

[29] Y. Wang, Z. Huang, and X. Hong, "S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 5682–5695, 2022.

[30] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2097–2106, 2017.

[31] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya, *et al.*, "Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 590–597, 2019.

[32] A. E. Johnson, T. J. Pollard, S. J. Berkowitz, N. R. Greenbaum, M. P. Lungren, C.-y. Deng, R. G. Mark, and S. Horng, "Mimic-cxr, a de-identified publicly available database of chest radiographs with free-text reports," *Scientific data*, vol. 6, no. 1, p. 317, 2019.

[33] H. Q. Nguyen, K. Lam, L. T. Le, H. H. Pham, D. Q. Tran, D. B. Nguyen, D. D. Le, C. M. Pham, H. T. Tong, D. H. Dinh, *et al.*, "Vindr-cxr: An open dataset of chest x-rays with radiologist's annotations," *Scientific Data*, vol. 9, no. 1, p. 429, 2022.

[34] D. Demner-Fushman, M. D. Kohli, M. B. Rosenman, S. E. Shooshan, L. Rodriguez, S. Antani, G. R. Thoma, and C. J. McDonald, "Preparing a collection of radiology examinations for distribution and retrieval," *Journal of the American Medical Informatics Association*, vol. 23, no. 2, pp. 304–310, 2016.

[35] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[36] F. Behrendt, D. Bhattacharya, J. Krüger, R. Opfer, and A. Schlaefer, "Data-efficient vision transformers for multi-label disease classification on chest radiographs," *Current Directions in Biomedical Engineering*, vol. 8, no. 1, pp. 34–37, 2022.

[37] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[38] B. Chen, J. Li, X. Guo, and G. Lu, "Dualchexnet: dual asymmetric feature learning for thoracic disease classification in chest x-rays," *Biomedical Signal Processing and Control*, vol. 53, p. 101554, 2019.

[39] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International conference on machine learning*, pp. 10347–10357, PMLR, 2021.

[40] D. Hou, Z. Zhao, and S. Hu, "Multi-label learning with visual-semantic embedded knowledge graph for diagnosis of radiology imaging," *IEEE Access*, vol. 9, pp. 15720–15730, 2021.

[41] M. Perkonigg, J. Hofmanninger, and G. Langs, "Continual active learning for efficient adaptation of machine learning models to changing image acquisition," in *International Conference on Information Processing in Medical Imaging*, pp. 649–660, Springer, 2021.

[42] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," *Advances in neural information processing systems*, vol. 30, 2017.

[43] G. M. Van De Ven, Z. Li, and A. S. Tolias, "Class-incremental learning with generative classifiers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3611–3620, 2021.

[44] O. Ostapenko, M. Puscas, T. Klein, P. Jahnichen, and M. Nabi, "Learning to remember: A synaptic plasticity driven framework for continual learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11321–11329, 2019.

[45] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro, "Learning to learn without forgetting by maximizing transfer and minimizing interference," *arXiv preprint arXiv:1810.11910*, 2018.

[46] S. Srivastava, M. Yaqub, K. Nandakumar, Z. Ge, and D. Mahapatra, "Continual domain incremental learning for chest x-ray classification in low-resource clinical settings," in *MICCAI Workshop on Domain Adaptation and Representation Transfer*, pp. 226–238, Springer, 2021.

[47] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[48] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," *Advances in neural information processing systems*, vol. 30, 2017.

[49] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 139–154, 2018.

[50] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *International conference on machine learning*, pp. 3987–3995, PMLR, 2017.

[51] J. Serra, D. Suris, M. Miron, and A. Karatzoglou, "Overcoming catastrophic forgetting with hard attention to the task," in *International conference on machine learning*, pp. 4548–4557, PMLR, 2018.

[52] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.

[53] S. Golkar, M. Kagan, and K. Cho, "Continual learning via neural pruning," *arXiv preprint arXiv:1903.04476*, 2019.

[54] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[55] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[56] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[57] R. Wang, X. Dai, *et al.*, "Contrastive learning-enhanced nearest neighbor mechanism for multi-label text classification," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 672–679, 2022.

[58] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[59] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE transactions on neural networks*, vol. 22, no. 2, pp. 199–210, 2010.

[60] W. Wang, H. Li, Z. Ding, F. Nie, J. Chen, X. Dong, and Z. Wang, "Rethinking maximum mean discrepancy for visual domain adaptation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 1, pp. 264–277, 2021.

[61] Z. Du, J. Li, H. Su, L. Zhu, and K. Lu, "Cross-domain gradient discrepancy minimization for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3937–3946, 2021.

[62] J. Huang and H. Qi, "Unsupervised domain adaptation with multi-kernel mmd," in *2021 40th Chinese Control Conference (CCC)*, pp. 8576–8581, IEEE, 2021.

[63] J. Li, E. Chen, Z. Ding, L. Zhu, K. Lu, and H. T. Shen, "Maximum density divergence for domain adaptation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 11, pp. 3918–3930, 2020.

[64] H. Yan, Z. Li, Q. Wang, P. Li, Y. Xu, and W. Zuo, "Weighted and class-specific maximum mean discrepancy for unsupervised domain adaptation," *IEEE Transactions on Multimedia*, vol. 22, no. 9, pp. 2420–2433, 2019.

[65] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5018–5027, 2017.

[66] J. Na, H. Jung, H. J. Chang, and W. Hwang, "Fixbi: Bridging domain spaces for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1094–1103, 2021.

[67] Z. Xiao, H. Wang, Y. Jin, L. Feng, G. Chen, F. Huang, and J. Zhao, "Spa: a graph spectral alignment perspective for domain adaptation," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[68] T. Westfechtel, H.-W. Yeh, D. Zhang, and T. Harada, "Gradual source domain expansion for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 1946–1955, 2024.

[69] K. Zhang, X. Liu, J. Shen, Z. Li, Y. Sang, X. Wu, Y. Zha, W. Liang, C. Wang, K. Wang, *et al.*, "Clinically applicable ai system for accurate diagnosis, quantitative measurements, and prognosis of covid-19 pneumonia using computed tomography," *Cell*, vol. 181, no. 6, pp. 1423–1433, 2020.

[70] P. Afshar, S. Heidarian, N. Enshaei, F. Naderkhani, M. J. Rafiee, A. Oikonomou, F. B. Fard, K. Samimi, K. N. Plataniotis, and A. Mohammadi, "Covid-ct-md, covid-19 computed tomography scan dataset applicable in machine learning and deep learning," *Scientific Data*, vol. 8, no. 1, p. 121, 2021.

[71] H. Ko, H. Chung, W. S. Kang, K. W. Kim, Y. Shin, S. J. Kang, J. H. Lee, Y. J. Kim, N. Y. Kim, H. Jung, *et al.*, "Covid-19 pneumonia diagnosis using a simple 2d deep learning framework with a single chest ct image: model development and validation," *Journal of medical Internet research*, vol. 22, no. 6, p. e19569, 2020.

[72] D. Kollias, A. Arsenos, L. Soukissian, and S. Kollias, "Mia-cov19d: Covid-19 detection through 3-d chest ct image analysis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 537–544, 2021.

[73] S. Serte and H. Demirel, "Deep learning for diagnosis of covid-19 using 3d ct scans," *Computers in biology and medicine*, vol. 132, p. 104306, 2021.

[74] W. Kang, L. Lin, B. Zhang, X. Shen, S. Wu, A. D. N. Initiative, *et al.*, "Multi-model and multi-slice ensemble learning architecture based on 2d convolutional neural networks for alzheimer's disease diagnosis," *Computers in Biology and Medicine*, vol. 136, p. 104678, 2021.

[75] A. Bustos, A. Pertusa, J.-M. Salinas, and M. De La Iglesia-Vaya, "Padchest: A large chest x-ray image dataset with multi-label annotated reports," *Medical image analysis*, vol. 66, p. 101797, 2020.

[76] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[77] E. B. Zaken, S. Ravfogel, and Y. Goldberg, "Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," *arXiv preprint arXiv:2106.10199*, 2021.

[78] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," *arXiv preprint arXiv:2101.00190*, 2021.

[79] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, and S.-N. Lim, "Visual prompt tuning," in *European Conference on Computer Vision*, pp. 709–727, Springer, 2022.