AttriCLIP: A Non-Incremental Learner for Incremental Knowledge Learning

Runqi Wang[1,2*], Xiaoyue Duan[1*], Guoliang Kang[1,4], Jianzhuang Liu[2],
Shaohui Lin[3], Songcen Xu[2], Jinhu Lv[1,4], Baochang Zhang[1,4†]
[1]Beihang University [2]Huawei Noah's Ark Lab
[3]East China Normal University [4]Zhongguancun Laboratory, Beijing China

Abstract

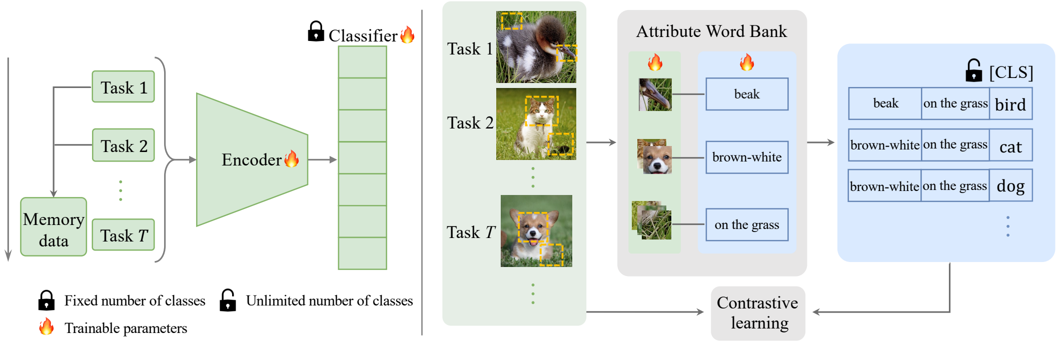
Continual learning aims to enable a model to incrementally learn knowledge from sequentially arrived data. Previous works adopt the conventional classification architecture, which consists of a feature extractor and a classifier. The feature extractor is shared across sequentially arrived tasks or classes, but one specific group of weights of the classifier corresponding to one new class should be incrementally expanded. Consequently, the parameters of a continual learner gradually increase. Moreover, as the classifier contains all historical arrived classes, a certain size of the memory is usually required to store rehearsal data to mitigate classifier bias and catastrophic forgetting. In this paper, we propose a non-incremental learner, named AttriCLIP, to incrementally extract knowledge of new classes or tasks. Specifically, AttriCLIP is built upon the pre-trained visual-language model CLIP. Its image encoder and text encoder are fixed to extract features from both images and text. Text consists of a category name and a fixed number of learnable parameters which are selected from our designed attribute word bank and serve as attributes. As we compute the visual and textual similarity for classification, AttriCLIP is a non-incremental learner. The attribute prompts, which encode the common knowledge useful for classification, can effectively mitigate the catastrophic forgetting and avoid constructing a replay memory. We evaluate our AttriCLIP and compare it with CLIP-based and previous state-of-the-art continual learning methods in realistic settings with domain-shift and long-sequence learning. The results show that our method performs favorably against previous state-of-the-arts. The implementation code will be available at https://gitee.com/mindspore/models/tree/master/research/cv/AttriCLIP.

1. Introduction

In recent years, deep neural networks have achieved remarkable progress in classification when all the classes (or tasks) are jointly trained. However, in real scenarios, the tasks or classes usually sequentially arrive. Continual learning [13, 21, 28] aims to train a model which incrementally expands its knowledge so as to deal with all the historical tasks or classes, behaving as if those tasks or classes are jointly trained. The conventional continual learning methods learn sequentially arrived tasks or classes with a shared model, as shown in Fig. 1(a). Such processing that fine-tunes the same model in sequence inevitably results in subsequent values of the parameters overwriting previous ones [20], which leads to catastrophic forgetting. Besides, the classification ability on historical data can be easily destroyed by current-stage learning. In the conventional continual learning methods, a classifier on top of the feature extractor is employed to perform recognition. As one group of weights in the classifier is responsible for the prediction of one specific class, the classifier needs to be expanded sequentially to make a continual learner able to recognize novel classes. Moreover, extra replay data is usually required to reduce the classifier bias and the catastrophic forgetting of learned features. It is still challenging if we expect a *non-incremental learner*, *i.e.*, the trainable parameters of the model do not incrementally increase and no replay data is needed to avoid the classifier bias and the catastrophic forgetting.

To address the above issues, this paper proposes a continual learning method named AttriCLIP , which adopts the frozen encoders of CLIP [19]. It is a typical visual-language model that conducts image classification by contrasting the features of images and their descriptive texts. In the face of increasing instances, we design a prompt tuning scheme for continual learning. As shown in Fig. 1(b), there are similar attributes in images with different categories, such as "a brown-white dog lying on the grass" and "a brown-white cat lying on the grass". They belong to different categories but both have "lying on the grass" and "brown-white" attributes, so the distance between these two images of different categories may be close in the feature space. Therefore, we selectively train different prompts based on the attributes of the images rather than the categories. In this way, there is

[*] Co-First Author.
[†] Corresponding Author.



(a) (b)

Figure 1. (a) Traditional framework for continual learning. The encoder and the classifier are trained by tasks in sequence, some of which even need extra memory data. In the framework, the model parameters of the current task are fine-tuned from the parameters trained by the previous last task and then are used for the classification of all seen tasks. The total number of categories the model can classify is fixed in the classifier. (b) Our proposed AttriCLIP for continual learning. AttriCLIP is based on CLIP, which classifies images by contrasting them with their descriptive texts. The trainable prompts are selected by the attributes of the current image from a prompt pool. The prompts are different if the attributes of the image are different. The trained prompts are concatenated with the class name of the image, which serve as a more accurate supervised signal for image classification than labels.

no problem of knowledge overwriting caused by sequential training of the same model with increasing tasks.

Specifically, an attribute word bank is constructed as shown in Fig 1(b), which consists of a set of (key, prompt) pairs. The keys represent the local features (attributes) of images and the prompts represent the descriptive words corresponding to the keys. Several prompts are selected according to the similarities between their keys and the input image. The selected prompts are trained to capture an accurate textual description of the attributes of the image. If images with different labels have similar attributes, it is also possible to select the same prompts, *i.e.*, the same prompts can be trained with images of different categories. Similarly, different prompts can be trained by the images of the same category. The trained prompts are concatenated with the class names and put into the text encoder to contrast with the image feature from the image encoder. This process makes our AttriCLIP distinct from all previous classifier-based framework as it serves as a non-incremental learner without the need to store replay data. The goal of our method is to select the existing attributes of the current image as the text description in the inference process, to classify the image. In addition, the structure of AttriCLIP can also avoid the problem of the increasing classifier parameters with the increase of the tasks and the problem of the inefficiency about memory data in the traditional continual learning methods.

The experimental setup of existing continual learning methods is idealized which divides one dataset into several tasks for continual learning. The model can set the output dimensionality of the classifier according to the total number of categories in the dataset. In practical applications, with the continuous accumulation of data, the total number of categories of samples usually cannot be obtained when the model is established. When the total number of categories exceeds the preset output dimensionality of the classifier, the model has to add parameters to classifier and requires the previous samples for fine-tuning, which greatly increases the training burden. Therefore, the continual learning approaches are required to have the ability to adapt to the categories of freely increasing data, *i.e.*, the model capacity should not have a category upper limit. In order to measure such ability of continual learning models, we propose a Cross-Datasets Continual Learning (CDCL) setup, which verifies the classification performance of the model on long-sequence domain-shift tasks. The contributions of this paper are summarized as follows:

- We establish AttriCLIP, which is a prompt tuning approach for continual learning based on CLIP. We train different prompts according to the attributes of images to avoid knowledge overwriting caused by training the same model in sequence of classes.
- AttriCLIP contrasts the images and their descriptive texts based on the learned attributes. This approach avoids the memory data requirement for fine-tuning the classifier of increasing size.
- In order to evaluate the performance of the model on long-sequence domain-shift tasks, we propose a Cross-Datasets Continual Learning (CDCL) experimental setup. AttriCLIP exhibits excellent performance and training efficiency on CDCL.

2. Related Work

Continual learning. The existing continual learning algorithms can be mainly divided into three categories: regularization-based, architecture-based, and rehearsal-based methods. Regularization-based methods [1, 8, 13] alleviate catastrophic forgetting to some extent by putting constraints on important parameters related to previous tasks without any memory replay. The core of architecture-based approaches is to assign independent parameters to different tasks, which can be achieved either by expanding a network [12, 22, 34], or by dividing the model into sub-networks [15, 25, 29]. However, most methods are not applicable to task-agnostic settings, where the task identity is unknown during inference. Even if the problem can be partially solved by recent methods [18, 32, 33], these methods are not lightweight enough. Rehearsal-based methods store the data of previous tasks in a so-called rehearsal buffer to train with the current task. Simple yet effective, these methods achieve impressive performance on challenging settings [2, 3]. However, the performance degrades as the buffer size reduces [3], and the approach to store data limits the application in privacy-sensitive scenarios [26].

Prompt tuning for continual learning. Recent continual learning works [30, 31] adopt visual prompt tuning to continual learning, which applies a small set of learnable parameters to the input, so as to provide additional instructions for the pre-trained model to be better transferred to downstream tasks [11]. L2P [31] first connects visual prompting with continual learning, and proposes to adapt the model to sequential tasks via a shared prompt pool. Inspired by the complementary learning systems, DualPrompt [30] proposed a different approach to append complementary visual prompts to the pre-trained backbone to learn task-invariant and task-specific instructions, further boosting the performance.

The prompts of L2P and DualPrompt are only attached to the image embeddings. Recent progress in vision-language models (*e.g.*, CLIP [19]) shows that language usually contains information complementary to vision. CLIP adopts a dual-stream architecture, which encodes image and text inputs into visual and textual representations in a joint embedding space. CLIP can generalize well across multiple downstream tasks. However, to the best of our knowledge, there is no work that takes text prompts into consideration in visual continual learning. We propose a CLIP-based prompt tuning method, which can continuously learn long-sequence tasks effectively and efficiently based on the learned attributes of images.

3. Methodology

3.1. Preliminaries

Continual learning formulation. Continual learning (CL) requires a model to continuously learn new knowledge from sequential tasks without forgetting the knowledge from previous ones. Consider a sequence of tasks $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_T\}$, where the t-th task $\mathcal{D}_t = \{(\mathbf{x}_i^t, y_i^t)\}_{i=1}^{n_t}$ contains n_t samples $\mathbf{x}_i^t$ and the corresponding labels y_i^t. During training on task $\mathcal{D}_t$, the access to data from $\{\mathcal{D}_1, \ldots, \mathcal{D}_{t-1}\}$ is unavailable or limited. In task-agnostic class-incremental learning, data $\mathcal{D}_t$ of different tasks arrives in sequence $t = \{1, \ldots, T\}$ without overlapping and the data arriving at different times comes from different classes. Besides, in task-agnostic setting, task identity is unknown at inference. Our AttriCLIP effectively tackles the settings above since it learns key attributes of images and allows unlimited number of output classes.

Prompt learning based on CLIP. CLIP [19] consists of an image encoder $f_\theta(\cdot)$ and a text encoder $g_\psi(\cdot)$. Specifically, the image $\mathbf{x} \in \mathbb{R}^{H\times W\times C}$ and the text $\mathbf{t} \in \mathbb{R}^D$ are fed into $f_\theta(\cdot)$ and $g_\psi(\cdot)$ respectively to obtain the image embedding $\mathbf{z} \in \mathbb{R}^D$ and the text embedding $\mathbf{w} \in \mathbb{R}^D$, where $\mathbf{t}$ is the input word token. In CLIP, $\mathbf{t}$ is obtained via one of the hand-crafted prompts which have a template like "a photo of a [CLS]", where [CLS] is the class name of the testing image. Thus, the probability of predicting the testing image $\mathbf{x}$ as the class y_i can be computed as:

$$p(y_i|\mathbf{x}) = \frac{e^{\langle \mathbf{z}, \mathbf{w}_{y_i}\rangle/\tau}}{\sum_{k=1}^{K} e^{\langle \mathbf{z}, \mathbf{w}_k\rangle/\tau}}, \tag{1}$$

where τ is a temperature parameter learned by CLIP, $\langle \cdot, \cdot \rangle$ denotes the cosine similarity, $\mathbf{w}_k$ is the embedding derived from $\mathbf{t}_k$ of the k-th class, and K is the total number of downstream dataset classes.

To bring about further improvements of CLIP's performance on downstream tasks, prompt learning has been proposed to replace the hand-crafted prompt templates with a set of continual learnable vectors $\mathbf{P}$. The knowledge of downstream data is encoded into these vectors to instruct the model to better perform downstream tasks. Specifically, CoOp [35] concatenates $\mathbf{P}$ with the embedding of a class name, formulating the text description $\mathbf{t}_k(\mathbf{P})$ of the k-th class as:

$$\mathbf{t}_k(\mathbf{P}) = [\mathbf{p}]_1[\mathbf{p}]_2 \ldots [\mathbf{p}]_M[\mathbf{CLS}]_k, \tag{2}$$

where each $[\mathbf{p}]_m \in \mathbb{R}^D$, $m \in \{1, \ldots, M\}$, is a learnable token of $\mathbf{P}$, and $\mathbf{P} \in \mathbb{R}^{D\times M}$ is shared among all classes. $[\mathbf{CLS}]_k$ is the text embedding of the k-th class name, which can also appear at the start and middle of the prompt. In this way, $\mathbf{w}_k$ in Eq. 1 is replaced by $g_\psi(\mathbf{t}_k(\mathbf{P}))$, and the probability of predicting the testing image $\mathbf{x}$ as the class y_i is computed as:

$$p(y_i|\mathbf{x}) = \frac{e^{\langle \mathbf{z}, g_\psi(\mathbf{t}_{y_i}(\mathbf{P}))\rangle/\tau}}{\sum_{k=1}^{K} e^{\langle \mathbf{z}, g_\psi(\mathbf{t}_k(\mathbf{P}))\rangle/\tau}}, \tag{3}$$

3.2. Framework of AttriCLIP

In CoOp, each class embedding corresponds to only one group of prompt vectors. However, images from the same class contain diverse attributes. Encoding these diverse attributes into the same group of prompts leads to catastrophic knowledge forgetting. Besides, the encoded knowledge in the prompts of CoOp cannot interact among different classes. However, the attributes of one class may help

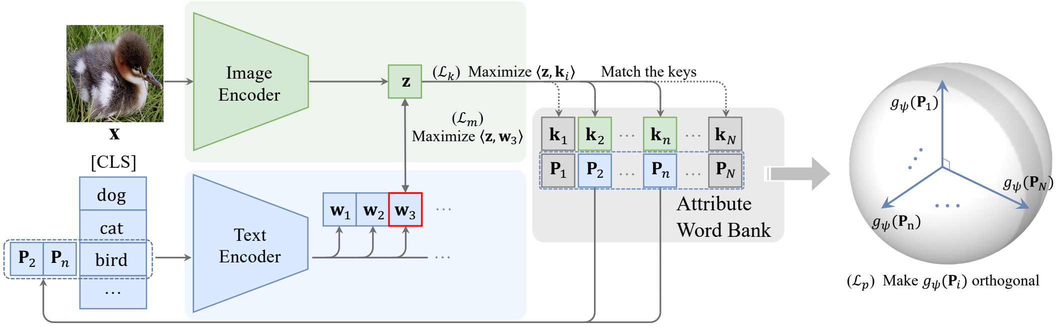

x
[CLS]
dog
cat
bird
...
$\mathbf{P}_2$
$\mathbf{P}_n$
Image Encoder
Text Encoder
z
$(\mathcal{L}_k)$ Maximize $\langle \mathbf{z}, \mathbf{k}_i \rangle$
Match the keys
$(\mathcal{L}_m)$ Maximize $\langle \mathbf{z}, \mathbf{w}_3 \rangle$
$\mathbf{w}_1$ $\mathbf{w}_2$ $\mathbf{w}_3$ ...
$\mathbf{k}_1$ $\mathbf{k}_2$ ... $\mathbf{k}_n$ ... $\mathbf{k}_N$
$\mathbf{P}_1$ $\mathbf{P}_2$ ... $\mathbf{P}_n$ ... $\mathbf{P}_N$
Attribute Word Bank
$g_\psi(\mathbf{P}_1)$
$g_\psi(\mathbf{P}_N)$
$g_\psi(\mathbf{P}_n)$
$(\mathcal{L}_p)$ Make $g_\psi(\mathbf{P}_i)$ orthogonal

Figure 2. Framework of AttriCLIP. The image keys $\mathbf{k}_i$ and the textual prompts $\mathbf{P}_i$ in the attribute word bank are trainable parameters. The blue and green boxer represent the image and text streams, respectively. The attribute word bank is optimized by three loss functions. $\mathcal{L}_m$ is the classification loss adopted to maximize the similarity between image feature $\mathbf{z}$ and the corresponding text features $\mathbf{w}$. $\mathcal{L}_k$ is designed to shorten the distance between the selected keys (*e.g.*, $\mathbf{k}_2$ and $\mathbf{k}_n$) and the image feature $\mathbf{z}$, so that the keys learn generalizable attributes. $\mathcal{L}_p$ makes the embeddings of the prompts $g_\psi(\mathbf{P}_i)$ orthogonal to increase the diversity of the prompts.

to identify another class with similar attributes. For example, given an image of a dog lying on the grass, the attribute "on the grass" in this image may also be found in images of other animals (*e.g.*, a cat lying on the grass). We believe that prompt tuning based on the image attributes can help the prompts learn the textual descriptions of these attributes, which generalize better among tasks.

Therefore, we propose AttriCLIP as shown in Fig. 2, which contains an attribute word bank to let the image itself decide which prompts to learn based on the attributes it has. Only a part of the prompts that are relevant to the current image attributes are selected and trained at a time. The attribute word bank stores visual and textual information, which consists of N (key, prompt) pairs:

$$\{\mathcal{K}, \mathcal{P}\} \triangleq \{(\mathbf{k}_1, \mathbf{P}_1), \ldots, (\mathbf{k}_N, \mathbf{P}_N)\}, \tag{4}$$

where $\{\mathcal{K}, \mathcal{P}\}$ denotes the attribute word bank, each $\mathbf{k}_i \in \mathbb{R}^D$ has the same dimensionality as the image embedding $\mathbf{z}$, and each $\mathbf{P}_i = [\mathbf{p}_i]_1 \ldots [\mathbf{p}_i]_M \in \mathbb{R}^{D \times M}$ is composed of M learnable vectors. Denoting the set of all keys as $\mathcal{K} = \{\mathbf{k}_i\}_{i=1}^N$ and the set of all prompts as $\mathcal{P} = \{\mathbf{P}_i\}_{i=1}^N$. $\mathcal{K}$ indicate the image attributes, and $\mathcal{P}$ indicate the prompt words. Ideally, we expect that the image itself can decide which prompts should be chosen based on the attributes it contains to guide the prediction. To this end, given an input image $\mathbf{x}_j$, we first obtain its image embedding $\mathbf{z}_j = f_\theta(\mathbf{x}_j)$, where j is the index of the image. Then, by scoring the match between $\mathbf{z}_j$ and $\mathbf{k}_i$ via a scoring function γ (*e.g.*, cosine distance), we select the top-C keys that match $\mathbf{z}_j$ most by:

$$\mathcal{K}_j = \text{Top-}C^{min}\{\gamma(\mathbf{z}_j, \mathbf{k}_{j_i})\}_{i=1}^N, \tag{5}$$

where Top-C^{min} denotes the operation of choosing the top-C minimal values for a set. $\mathcal{K}_j$ denotes the subset of top-C keys selected from $\mathcal{K}$ specifically for the j-th image. We then choose the corresponding prompts that are paired with these keys, denoted as $\mathcal{P}_j = \{\mathbf{P}_{j_i}\}_{i=1}^C$, where $\mathbf{P}_{j_i}$ is the i-th prompt selected specifically for $\mathbf{x}_j$. These prompts are attached to the class name embedding of $\mathbf{x}_j$ as illustrated in Fig. 2, and $\mathbf{t}_k(\mathbf{P})$ in Eq. 2 is then denoted as:

$$\mathbf{t}_k(\mathcal{P}_j) = \text{concat}(\mathbf{P}_{j_1}; \ldots; \mathbf{P}_{j_C}; [\text{CLS}]_k), \tag{6}$$

where $\text{concat}(\cdot)$ denotes concatenation. Therefore, given test image $\mathbf{x}_j$ and the prompts $\mathcal{P}_j$ selected according to the attributes of $\mathbf{x}_j$, the probability of predicting the image as class y_i is finally computed as:

$$p(y_i|\mathbf{x}_j) = \frac{e^{\langle \mathbf{z}, g_\psi(\mathbf{t}_{y_i}(\mathcal{P}_j)) \rangle / \tau}}{\sum_{k=1}^K e^{\langle \mathbf{z}, g_\psi(\mathbf{t}_k(\mathcal{P}_j)) \rangle / \tau}}. \tag{7}$$

From a high-level perspective, the proposed attribute word bank serves as a bridge between the output of the image encoder and the input of the text encoder. The keys are optimized to be close to the matched image embeddings, which contain rich high-level information, *i.e.*, image attributes. The prompts are optimized to include textual information related to the corresponding image attributes, so as to better guide the model predictions along with the class name embeddings. Since the proposed attribute word bank connects the image stream and the text stream, our prompts serve more as the textual descriptions of the image attributes compared with DualPrompt [30]. In addition, since the generalizable attributes are learned, the memory is no longer needed to fine-tune the classifier based on previous tasks, which makes AttriCLIP more efficient under the long sequence setting.

3.3. Optimization Objective of AttriCLIP

Based on Eq. 7, the image classification loss is formulated as:

$$\mathcal{L}_m = \mathbb{E}[-\log \frac{e^{\langle \mathbf{z}, g_\psi(\mathbf{t}_{y_i}(\mathcal{P}_j)) \rangle / \tau}}{\sum_{k=1}^K e^{\langle \mathbf{z}, g_\psi(\mathbf{t}_k(\mathcal{P}_j)) \rangle / \tau}}]. \tag{8}$$

In addition to $\mathcal{L}_m$, a matching loss is needed to pull the matched top-C keys $\mathcal{K}_j$ closer to the image embedding $\mathbf{z}_j$, so that the keys learn rich attributes from the samples. The matching loss adopted to optimize the keys is defined as:

$$\mathcal{L}_k = \sum_{i=1}^{C} \gamma(\mathbf{z}_j, \mathbf{k}_{j_i}). \tag{9}$$

We test three distance functions (*i.e.*, cosine distance [23], mean square error (MSE) [16] and triplet loss [5]) for γ, and find that the cosine distance works the best (see Sec. 4.4). Finally, in order to make the learned prompts more semantically diverse, we adopt a third loss to orthogonalize the embeddings of different prompts to increase the diversity of the prompts:

$$\mathcal{L}_p = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} |\langle g_\psi(\mathbf{P}_i), g_\psi(\mathbf{P}_j) \rangle|, \tag{10}$$

where $\langle \cdot, \cdot \rangle$ denotes the cosine similarity. In this way, the overall optimization objective is defined as:

$$\mathcal{L} = \mathcal{L}_m + \lambda_k \mathcal{L}_k + \lambda_p \mathcal{L}_p, \tag{11}$$

where λ_k and λ_p are balance factors. The keys are optimized by $\mathcal{L}_k$, and the prompts by $\mathcal{L}_m$ and $\mathcal{L}_p$.

4. Experiments

In this section, the implementation details are first described. We then compare the proposed AttriCLIP with other methods in the conventional class-incremental task-agnostic setting, and the proposed Cross-Datasets Continual Learning (CDCL) setting. Finally, we perform ablation studies to evaluate the effect of different components of AttriCLIP. We implement our model using the MindSpore Lite tool [17].

4.1. Implementation Details

Datasets. The experiments are conducted on CIFAR-100 [9] and ImageNet100 [4]. CIFAR100 consists of 60k images with a size of 32×32 from 100 classes, which are split into 10 tasks with 10 classes in each task. Each class consists of 500 training and 100 testing samples. ImageNet100, as a subset of ILSVRC2012 [10], contains samples sized 224×224 from 100 classes. Each class consists of about 1,300 training and 50 test samples. We split ImageNet100 into 10 tasks with 10 classes in each task. More details of ImageNet100 are provided in the supplementary.

Baselines. We compare the proposed AttriCLIP with existing CLIP-based methods (CoOp [35] and continual-CLIP [27]), prompt-based methods (DualPrompt [30]) and typical continual learning methods (LwF [13], iCaRL [21], DER [33], iTAML [20] and ARI [28]). The prompts of CoOp are trained in a sequence of tasks and store partial data from previous tasks in the memory for the prompts fine-tuning on subsequent tasks. The continual-CLIP evaluates a frozen pre-trained CLIP model in continual learning settings. iCaRL is a classic method of continual learning with memory data. ARI is the current state-of-the-art for non-prompt-based continual learning methods. We adopt ViT-L-14 [7] as the backbone for CoOp, continual-CLIP, DualPrompt and our AttriCLIP, and adopt ResNet [6] for other methods. All the methods are evaluated under the task-agnostic setting, and our proposed AttriCLIP does not need any memory, which makes the setting more practical and challenging.

Training details. We train the model for 10 epochs on each incremental task for all datasets. SGD is adopted as the optimizer with the initial learning rate set to 0.001 and following a cosine decay schedule. The weight decay is 0, the batch size is 32, and the loss weights λ_k and λ_p are 0.7 and 0.3 respectively. The prompt length $M = 12$, the number of attributes in the bank $N = 10$ and the number of selected attributes $C = 3$. The average results over 3 runs are reported for all methods.

4.2. Results of Class-Incremental Learning

In this section, we take the average accuracy [14], as the metric to measure the performance. The buffer size of data from previous tasks is denoted as Memory. We compare the proposed AttriCLIP with the prior arts on CIFAR100, and the results are reported in Table 1. From the results, we see that AttriCLIP achieves the best average accuracy compared with the recent state-of-the art methods such as ARI and Continual-CLIP. Besides, compared with previous CLIP-based methods (*i.e.*, CoOp and Continual-CLIP), AttriCLIP outperforms them by a large margin. Specifically, AttriCLIP outperforms CoOp by 13.8% without the need for any memory. It also outperforms Continual-CLIP by 14.7%. Note that “Upper-bound” in Table 1 denotes the standard supervised training on the data from all tasks, which is usually regarded as the upper bound of the performance one method can achieve. Compared with the upper-bound, the accuracy of AttriCLIP drops only 4.9%, demonstrating the effectiveness of learning image attributes for mitigating catastrophic forgetting.

We also compare our method with previous arts on ImageNet100, and report the results in Table 2. AttriCLIP still outperforms other memory-based methods without any memory needed. For example, AttriCLIP outperforms ARI, which was the state-of-the-art, by 4.0%. Compared with CLIP-based models, our method again outperforms CoOp and Continual-CLIP by 4.0% and 7.9% respectively. The accuracy of AttriCLIP decreases by 8.1% compared with the upper-bound. The results suggest that the effectiveness of our method over different datasets.

4.3. Results of Cross-Datasets Continual Learning

To simulate the practical setting where the model continuously learns long sequence tasks, we propose a new setting for evaluation, *i.e.*, Cross-Datasets Continual Learning

Table 1. Average accuracy [14] of different continual learning methods on CIFAR100 [9]. The accuracy of Task $t, t \in \{1, 2, \ldots, 10\}$ reported here is the test accuracy averaged over all the previous tasks (*i.e.*, Tasks $1, 2, \ldots, t$).

Method	Memory	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task9	Task10
LwF	0	89.3	70.1	54.3	45.8	39.8	36.1	31.7	28.9	24.4	23.9
iCaRL	2000	88.7	78.1	72.4	67.2	63.7	60.2	56.4	54.4	51.9	49.5
iTAML	2000	89.2	89.0	87.3	86.2	84.3	82.1	80.7	79.1	78.4	77.8
ARI	2000	88.6	86.9	85.8	84.6	83.1	81.8	81.6	81.0	80.2	80.9
CoOp	1000	95.8	90.7	85.2	83.4	80.8	75.8	74.7	71.7	71.3	67.6
Continual-CLIP	0	96.7	92.2	86.0	80.4	77.5	75.8	73.0	71.4	69.8	66.7
AttriCLIP	0	**97.8**	**93.7**	**91.0**	**87.5**	**84.7**	**82.5**	**82.3**	**81.9**	**81.7**	**81.4**
Upper-bound	-	-	-	-	-	-	-	-	-	-	**86.3**

Table 2. Average accuracy [14] of different continual learning methods on ImageNet100 [4]. The accuracy of Task $t, t \in \{1, 2, \ldots, 10\}$ reported here is the test accuracy averaged over all the previous tasks (*i.e.*, Tasks $1, 2, \ldots, t$).

Method	Memory	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task9	Task10
iCaRL	2000	82.1	80.6	75.5	70.1	68.1	65.8	62.5	61.3	60.7	59.5
DER	2000	81.7	80.6	76.0	72.1	74.4	71.8	70.5	68.3	67.3	66.7
ARI	2000	87.6	85.4	83.1	82.6	80.4	80.8	80.5	80.1	79.6	79.3
CoOp	1000	89.2	83.2	76.7	79.8	79.9	82.34	79.7	80.1	80.3	79.3
Continual-CLIP	0	93.3	87.6	83.1	81.7	80.5	80.2	79.3	78.5	76.9	75.4
AttriCLIP	0	**95.4**	**89.4**	**84.5**	**86.7**	**84.4**	**86.6**	**85.9**	**85.6**	**86.9**	**83.3**
Upper-bound	-	-	-	-	-	-	-	-	-	-	**91.4**

Table 3. Accuracy of different methods on CIFAR100. The models are either trained from scratch on CIFAR100 (CIFAR100), or fine-tuned on CIFAR100 after being continually trained from scratch on ImageNet100 (CIFAR100-I2C).

Method	Memory	CIFAR100	CIFAR100-I2C	FT
iCaRL-1	2000	49.5	49.7	+0.2
iCaRL-2	2000	49.1	46.5	-2.6
CoOp-1	1000	67.6	61.1	-6.5
CoOp-2	1000	67.6	59.0	-8.6
ARI-1	2000	80.9	74.5	-6.4
ARI-2	2000	79.7	59.9	-19.8
Continual-CLIP	0	66.7	66.7	0
DualPrompt-1	0	**86.5**	80.7	-5.8
DualPrompt-2	0	84.1	74.7	-9.4
AttriCLIP	0	81.4	**82.3**	**+0.9**

(CDCL). In CDCL, the model is continually trained on several datasets one by one in a sequential manner, and then the accuracy on each dataset is evaluated.

Learning attributes helps the model to generalize better to a new dataset. For comparison, we train two benchmarks for each method in Table 3: (1) CIFAR100 benchmark, where the model is trained from scratch on CIFAR100 under the same setting (10 tasks) as in Table 1, then evaluated on CIFAR100. (2) CIFAR100-I2C benchmark, where the model is first trained from scratch on ImageNet100 under the same setting (10 tasks) as in Table 2, then fine-tuned on CIFAR100 under the same setting (10 tasks) as in Table 1, and finally evaluated on CIFAR100. We define FT (Forward Transfer) in Table 3 as the accuracy on CIFAR100-I2C minus the accuracy on CIFAR100, which indicates the model's superior ability to transfer knowledge from the previous dataset to the new dataset.

Conventional continual learning methods adopt the typical classification architecture, which includes a classifier with a preset output dimensionality. However, in practical settings, the number of sequentially arriving classes is unlimited. Therefore, the classifier needs to be incrementally expanded to learn new tasks or classes. In this experiment, we use two schemes to expand the classifier for each classifier-based continual learning method: (1) *Method*-1, *i.e.*, increase the number of classifiers as the number of arriving classes increases; (2) *Method*-2, *i.e.*, directly increase the output dimensionality of a classifier so that it can handle more classes of data.

In our experiments, for *Method*-1, after training a classifier (with the output dimensionality as 100) on CIFAR100, we incrementally train another classifier (with the output dimensionality also as 100) on ImageNet100. These two classifiers share the same feature extractor. Besides, when training the classifier for ImageNet100, the data from CIFAR100 is unavailable in the memory bank for memory-based methods. On the other hand, for *Method*-2, we directly train one classifier but double its output dimensionality (*i.e.*, 200), so that it can jointly predict data from both CIFAR100 and ImageNet100. Besides, when training on ImageNet100, the data from CIFAR100 is available in the memory bank for memory-based methods. Note that the CoOp model does not have a classifier, so the only difference between *CoOp*-1 and *CoOp*-2 is whether the data from the previous dataset

Table 4. Accuracy of different methods on ImageNet100. The models are either trained from scratch on ImageNet100 (ImageNet100), or fine-tuned on CIFAR100 after being continually trained from scratch on ImageNet100 (ImageNet100-I2C).

Method	Memory	ImageNet100	ImageNet100-I2C	BT
iCaRL-1	2000	59.5	34.5	-25.0
iCaRL-2	2000	58.7	50.9	-7.8
CoOp-1	1000	79.3	57.6	-21.7
CoOp-2	1000	79.3	75.9	-3.4
ARI-1	2000	79.3	51.2	-28.1
ARI-2	2000	77.9	61.8	-16.1
Continual-CLIP	0	75.4	75.4	0
DualPrompt-1	0	**85.4**	63.6	-21.8
DualPrompt-2	0	81.9	77.8	-4.1
AttriCLIP	0	83.3	**90.3**	**+7.0**

Table 5. Comparison among different methods on ImageNet100 + CIFAR100 where each model is continually trained on ImageNet100 and CIFAR100 in sequence.

Method	Memory	CIFAR100+ ImageNet100
iCaRL-1	2000	30.7
iCaRL-2	2000	37.6
CoOp-1	1000	46.6
CoOp-2	1000	55.4
ARI-1	2000	32.5
ARI-2	2000	57.3
Continual-CLIP	0	54.9
DualPrompt-1	0	35.4
DualPrompt-2	0	67.1
AttriCLIP	0	**78.3**

is available in the memory bank.

We report the experimental results in Table 3. The results demonstrate that AttriCLIP is outperformed by *DualPrompt*-1 and *DualPrompt*-2 on CIFAR100. It is to be noted that the encoder of *DualPrompt*-1 and *DualPrompt*-2 is pre-trained on ImageNet-21k [6]. AttriCLIP still significantly exceeds the remaining methods. Specifically, AttriCLIP outperforms *CoOp*-1 and *CoOp*-2 by 13.8%. Moreover, we find that among all methods, AttriCLIP is the only one that effectively transfers the knowledge learned from ImageNet100 to improve the performance on CIFAR100 (*i.e.*, FT>0), and exceeds all other methods under the CIFAR100-I2C setting. This indicates that AttriCLIP effectively learns crucial image attributes into the prompts in the previous dataset, which helps it to generalize better to a new dataset.

Learning attributes helps the model NOT to forget the previous dataset. In Table 4, we test the accuracy of different continual learning methods on ImageNet100 also under two settings: (1) ImageNet100 benchmark, which is the same setting as in Table 2. (2) ImageNet100-I2C benchmark, which is the same as CIFAR100-I2C in training, but evaluating on ImageNet100. We define BT (Backward Transfer) as the accuracy on ImageNet100-I2C minus the accuracy on ImageNet100. The smaller the value of BT, the more knowledge from the previous dataset is forgotten after the model is trained on the new one. When BT>0, the model effectively transfers knowledge from the new dataset to improve the recognition performance on the previous dataset.

Table 6. Comparison of different loss functions for $\mathcal{L}_k$ on CIFAR100.

Loss function	Triplet loss	Cosine loss	MSE loss
Average acc.	80.22	**81.38**	80.81

Table 7. Average acc. of different loss weights λ_k on CIFAR100.

λ_k	0.1	0.3	0.5	0.7	0.9
Average acc.	80.28	80.30	81.11	**81.38**	80.86

As shown in Table 4, AttriCLIP is the only method which does not forget the knowledge from the previous dataset, and even improves the performance on the previous dataset (BT=+7.0%). This demonstrates that our method effectively learns generalizable attributes from the new dataset, which can help the model NOT to forget, or even consolidate previously learned knowledge.

In addition, *Method*s-1 in Table 4 forget previous knowledge more seriously than *Method*-2, which indicates that replaying the data from the previous dataset or training a classifier with large output dimensionality may help to mitigate catastrophic forgetting cross datasets. However, the output dimensionality of classifier cannot be expanded endlessly, and previously trained data is often unavailable in practical settings. This again highlights the advantage of our method.

Learning attributes helps the model evaluate on cross datasets. In Table 5, we train the models following the same setting as in Table 3, and finally evaluate their performances on both datasets. For a classifier-based *Method*-1, given one testing image, each of its two classifiers outputs a prediction vector of 100 dimensions. We simply choose the class with the highest score in these two output vectors as the prediction result. According to Table 5, AttriCLIP achieves the highest accuracy (78.3%) without the need for any memory, demonstrating the effectiveness of our method in the proposed CDCL setting.

4.4. Ablations

We conduct ablation studies on CIFAR100 following the same setting as in Table 1. The average accuracy over 10 tasks are reported.

Loss function L_k. We adopt three loss functions (*i.e.*, triplet loss, cosine distance loss, and MSE loss) for γ in Eq. 9. According to Table 6, the best result is obtained with cosine distance loss adopted for $\mathcal{L}_k$. We also vary the weight λ_k of $\mathcal{L}_k$ in Eq. 11. The result in Table 7 shows that the best performance (81.38%) is achieved with $\lambda_k = 0.7$.

Loss function L_p. We report the results with different

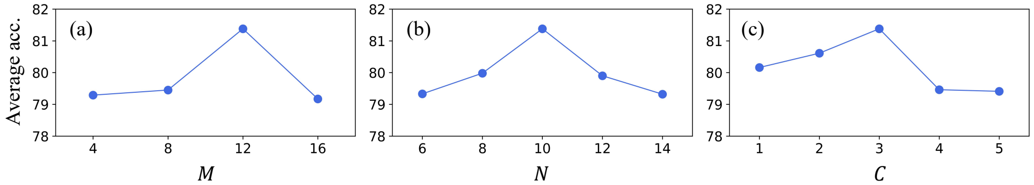


Figure 3. Ablation study of (a) the prompt length M, (b) the bank size N, and (c) the number of selected keys C on CIFAR100.

Table 8. Average acc. of different loss weights λ_p on CIFAR100.

λ_p	0	0.1	0.3	0.5	0.7	0.9
Average acc.	78.10	79.23	**81.38**	81.28	81.17	80.88

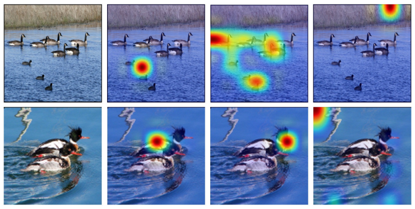


Figure 4. Visualization of the selected prompts of the same image using Grad-CAM [24].

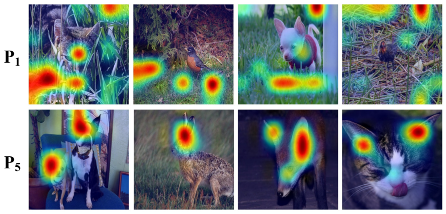


Figure 5. Visualization of the same prompts on different images using Grad-CAM [24].

loss weights λ_p of $\mathcal{L}_p$ in Table 8. Introducing $\mathcal{L}_p$ significantly improves the performance by increasing the diversity of prompts. The best result is obtained with $\lambda_p = 0.3$.

The length of prompts M**.** The result in Fig. 3(a) shows that the model achieves the best performance when $M = 12$. When the prompt length is too long, both the training efficiency and the computation budget will be increased.

Bank size N**.** By varying the number of (key, prompt) pairs in the attribute word bank, we find in Fig. 3(b) that the model achieves the best performance when $N = 10$.

The number C **of attributes selected.** We test the effect of different values of C in Eq. 5, and find in Fig. 3(c) that choosing too many keys and prompts to train at the same time affects the model performance. When $C = 3$, the model obtains optimal performance.

Visualization of prompts. To verify that different prompts do reflect different image attributes, we visualize the image contents corresponding to different prompts using Grad-CAM [24]. Specifically, given test image, several prompts are first selected based on the image attributes. Each selected prompt $\mathbf{P}_i$ is passed through the text encoder to obtain the prompt embedding $g_\psi(\mathbf{P}_i)$. The prompt embedding and the image feature $\mathbf{z}$ are then used to calculate $\mathcal{L}_m$, which is adopted to highlight the corresponding image contents using Grad-CAM.

In Fig. 4, the image contents in different columns correspond to different prompts. From Fig. 4, it can be seen that for the same image, different prompts do reflect different regions in the image, demonstrating the diversity of the learned prompts.

To verify whether the learned prompts do reflect image attributes with high-level semantics, we visualize the content of the same prompts (*e.g.*, $\mathbf{P}_1$ and $\mathbf{P}_5$) on different images in Fig. 5. It can be seen that $\mathbf{P}_1$ mainly captures the background of the images (*e.g.*, the grass), while $\mathbf{P}_5$ focuses more on the foreground (*e.g.*, the ears of the animals). This demonstrates that the prompts effectively learn key attributes which can generalize across images, thus improving the performance in continual learning.

5. Conclusion

We propose a novel continual learning method, named AttriCLIP, which can incrementally learn knowledge without incrementally increasing model parameters or constructing extra memory to store replay data. Our framework is based on the pretrained visual-language model CLIP. We fix both the image and the text encoders, only updating the text prompts to adapt to sequentially arrived tasks or classes. We design a module named attribute word bank to store attributes of images and their descriptive words. Experiments show that our method performs favourably against vanilla CLIP, typical prompt learning methods and previous state-of-the-arts, especially in the long-sequence and cross-domain settings. We believe our work paves the way for more practical continual learning, where we need to consider the incremental knowledge in a long task sequence or face the common domain shift.

Acknowledgements

This work was supported by National Natural Science Foundation of China under Grant 62076016, 62141604 and 62102151, Beijing Natural Science Foundation L223024 and Shanghai Sailing Program (21YF1411200). We gratefully acknowledge the support of MindSpore [17], CANN (Compute Architecture for Neural Networks) and Ascend AI Processor used for this research.

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, 2018. 2

[2] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *NeurIPS*, 2020. 3

[3] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In *ICCV*, 2021. 3

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5, 6

[5] Xingping Dong and Jianbing Shen. Triplet loss in siamese network for object tracking. In *ECCV*, 2018. 5

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint*, 2020. 5, 7

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[8] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. In *NAS*, 2017. 2

[9] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. In *Citeseer*, 2009. 5, 6

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 5

[11] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, 2021. 3

[12] Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *ICML*, 2019. 3

[13] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017. 1, 2, 5

[14] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *NeurIPS*, 2017. 5, 6

[15] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *CVPR*, 2018. 3

[16] Hans Marmolin. Subjective mse measures. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(3):486–489, 1986. 5

[17] Mindspore. https://www.mindspore.cn/. 5, 8

[18] Quang Pham, Chenghao Liu, and Steven Hoi. Dualnet: Continual learning, fast and slow. In *NeurIPS*, 2021. 3

[19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 3

[20] Jathushan Rajasegaran, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Mubarak Shah. itaml: An incremental task-agnostic meta-learning approach. In *CVPR*, 2020. 1, 5

[21] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017. 1, 5

[22] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint*, 2016. 3

[23] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *ICLR*, 2019. 5

[24] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017. 8

[25] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *ICML*, 2018. 3

[26] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *ACM SIGSAC*, 2015. 3

[27] Vishal Thengane, Salman Khan, Munawar Hayat, and Fahad Khan. Clip model is an efficient continual learner. *arXiv preprint*, 2022. 5

[28] Runqi Wang, Yuxiang Bao, Baochang Zhang, Jianzhuang Liu, Wentao Zhu, and Guodong Guo. Anti-retroactive interference for lifelong learning. In *ECCV*, 2022. 1, 5

[29] Zifeng Wang, Tong Jian, Kaushik Chowdhury, Yanzhi Wang, Jennifer Dy, and Stratis Ioannidis. Learn-prune-share for lifelong learning. In *ICDM*, 2020. 3

[30] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. *arXiv preprint*, 2022. 3, 4, 5

[31] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *CVPR*, 2022. 3

[32] Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition. In *NeurIPS*, 2020. 3

[33] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *CVPR*, 2021. 3, 5

[34] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint*, 2017. 3

[35] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. In *International Journal of Computer Vision*, pages 1–12. Springer, 2022. 3, 5