This WACV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

MoP-CLIP: A Mixture of Prompt-Tuned CLIP Models for Domain Incremental Learning

Julien Nicolas
ETS Montreal

Florent Chiaroni
Thales Digital Solutions

Imtiaz Ziko
Thales Digital Solutions

Ola Ahmad
Thales Digital Solutions

Christian Desrosiers
ETS Montreal

Jose Dolz
ETS Montreal

Abstract

Despite the recent progress in incremental learning, addressing catastrophic forgetting under distributional drift is still an open and important problem. Indeed, while state-of-the-art domain incremental learning (DIL) methods perform satisfactorily within known domains, their performance largely degrades in the presence of novel domains. This limitation hampers their generalizability, and restricts their scalability to more realistic settings where train and test data are drawn from different distributions. To address these limitations, we present a novel DIL approach based on a mixture of prompt-tuned CLIP models (MoP-CLIP), which generalizes the paradigm of S-Prompting to handle both in-distribution and out-of-distribution data at inference. In particular, at the training stage we model the features distribution of every class in each domain, learning individual text and visual prompts to adapt to a given domain. At inference, the learned distributions allow us to identify whether a given test sample belongs to a known domain, selecting the correct prompt for the classification task, or from an unseen domain, leveraging a mixture of the prompt-tuned CLIP models. Our empirical evaluation reveals the poor performance of existing DIL methods under domain shift, and suggests that the proposed MoP-CLIP performs competitively in the standard DIL settings while outperforming state-of-the-art methods in OOD scenarios. These results demonstrate the superiority of MoP-CLIP , offering a robust and general solution to the problem of domain incremental learning.

1. Introduction

In machine learning, it is a common practice to assume that both training and test data follow the same underlying distribution. In real-world scenarios, however, this strong assumption is rarely met, leading to substantial performance degradation when the trained model is evaluated on test samples under a distributional drift. A simple solution to alleviate this issue is to train the model on the labeled samples from the new domain. However, when the learning is performed in a sequential manner on multiple domains, contemporary deep learning models tend to suffer from the phenomenon of *catastrophic forgetting*, wherein the acquired knowledge from previous domains is typically erased.

A simple strategy to address this issue consists in training different models, one per single domain. However, this approach is suboptimal, as all these models must be stored for future usage and the domain identity is not necessarily known at test time. To tackle the issue of forgetting learned knowledge, domain incremental learning (DIL) has recently emerged as an appealing alternative that alleviates the need to store multiple domain-specific networks. Among the different DIL approaches, rehearsal [2,3,18,35] and distillation-based [1, 17, 24] methods, which leverage a buffer of stored exemplars from old domains, dominate the literature. Nevertheless, from a privacy and storage standpoint, *exemplar-free* DIL approaches may offer a better solution in practical settings.

An appealing alternative to mitigate knowledge forgetting is prompt-learning, which is driving progress in a wide span of transfer learning problems [23, 49]. In this approach, domain-specific knowledge is preserved in the form of textual and visual prompts, alleviating the need of storing exemplars per domain. While some methods advocate for the joint learning of prompts across tasks [14, 41], the recent work in [39] instead favors the learning of the prompts independently, suggesting that this leads to the best performance per domain. This learning paradigm, referred to as S-Prompting [39], circumvents the issue of using expensive buffers by optimizing per-domain prompts, which are leveraged at testing time. In particular, centroids for each domain are obtained during training by applying K-Means on the training image features, which are generated with the fixed pre-trained transformer without using any prompts. Then, during inference, the standard KNN algorithm is used to identify the nearest centroid to the test image, whose associated domain prompt is added to the image tokens for classification. Despite the empirical performance gains observed by these approaches [14, 39, 41], a current limitation hampering their generalization is that they perform satisfactorily in *known* domains, but typically fail when *unseen* domains are presented (see Fig. 1). This is particularly important in real-world scenarios where training and testing data of the *a priori* same domain may present distributional drifts that degrade the model performance. In the case of S-Prompts [39], we argue that a potential reason behind this suboptimal performance stems from forcing the model to select a single domain (i.e., the closest one), which might be indeed far in the feature space.

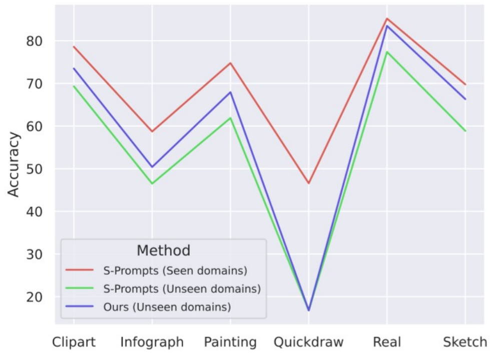

Figure 1. **Performance degradation under the presence of domain shift** between adaptation and testing samples, which shows that sota DIL approaches do not generalize well. We employ S-Prompts [39] as use-case. The red line represents the performance across each test domain, when all domains have been seen by the model. In contrast, the blue dotted line shows the performance of the same model when the test domain remains unknown, highlighting the performance degradation under distributional shift.

Motivated by these limitations, we introduce a novel *exemplar-free* DIL solution, based on prompt learning, which generalizes the recent S-liPrompts approach [39] for both in-distribution and out-of-distribution data. Specifically, our contributions can be summarized as follows:

- We first expose that existing state-of-the-art domain incremental learning approaches suffer in the presence of distributional shift between samples used for adaptation and testing, which hampers their generalization to unseen domains (Fig. 1).
- Based on these observations, we present a novel DIL strategy based on a mixture of prompt-tuned (MoP) CLIP models, generalizing the recent S-liPrompts approach [39] to work with both in-distribution and out-of-distribution data. In particular, the proposed approach learns class-wise features distributions for each domain, allowing to detect whether a given sample comes from a known domain.
- The proposed approach is *exemplar-free*, reducing the computational burden compared to conventional methods, and *agnostic to the sequence order*.
- Extensive experiments demonstrate that our approach performs at par with state-of-the-art DIL methods on known domains, while largely outperforming them under distributional drifts.

2. Related Work

Domain-Incremental learning (DIL) refers to continual learning scenarios in which the distribution of instances from fixed classes changes between domains. These real-world scenarios include, for example, the recognition of objects where new instances from varying environments appear in each new domain [28], or autonomous driving, where the car is exposed to ever-changing conditions. We focus on the domain-agnostic scenario, where the sample's domain remains unknown at inference time. The major challenge of this task is to find a good trade-off to adapt to the new instances distribution without deteriorating performance for samples of the previous distributions (i.e., alleviating *catastrophic forgetting*). The literature on this subject is abundant, where the main approaches are based on weight regularization [8, 22, 46], knowledge distillation in a teacher-student setting using current examples [26] or a memory buffer [9] and methods using or generating latent features [32, 37] or gradient examplars [9, 29, 31]. Nevertheless, these approaches require the use of *exemplars* from seen domains, which may result in storage, security and privacy issues. In contrast, the proposed approach only requires the storage of a single prototype per class and domain, largely alleviating these issues. Last, existing methods are tailored to perform optimally within known data distributions, i.e., the learned model is tested on the same domain used for adaptation. However, their out-of-domain generalization capabilities, i.e., the model encounters novel domains during testing, are rarely considered [39].

Prompt learning. Driven by the advances in Natural Learning Processing, prompt learning has emerged as an appealing learning strategy to adapt large scale pre-trained models to downstream tasks. While initial attempts to adapt language-vision models have centered on carefully designing handcrafted prompts [5], recent works focus on optimizing a task-specific continuous vector, which is optimized via gradients during fine-tuning [20, 30, 49, 50]. An underlying limitation of these approaches arises from the inherent

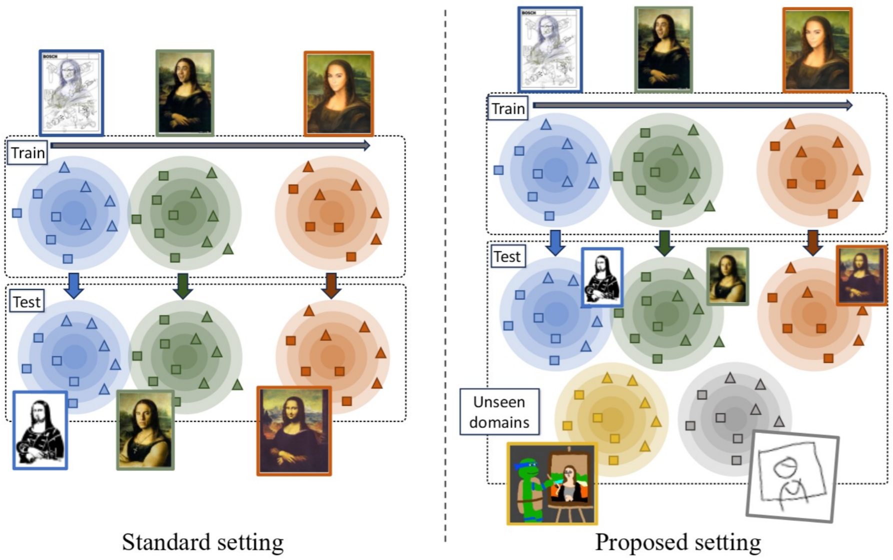

Figure 2. **Proposed generalization scenario for domain incremental learning** Standard problem (*left*): Only in-domain examples are encountered at test time. Addressed problem (*right*): Both in-domain and out-of-domain examples are presented at test time.

disparity between language and vision modalities, and thus fine-tuning only text prompts for visual recognition tasks may yield suboptimal performance. Motivated by this, visual prompt tuning (VPT) [19] was proposed as a powerful alternative to text prompting. In this approach, authors propose to optimize task-specific learnable prompts in either the input or visual embedding space. Following the satisfactory results achieved by VPT, fine-tuning visual prompts has gained popularity recently, particularly for adapting pre-trained models to novel unseen categories [10, 38, 43, 43].

Prompt tuning in domain incremental learning. This paradigm protects against catastrophic forgetting by optimizing a small set of learnable prompts. This contrasts with classical approaches which modify all the network parameters (or a subset),or store *exemplars* in a buffer. Despite the success observed in other tasks, the literature on prompt tuning for domain incremental learning remains underexplored, with just a handful works addressing this problem [4, 14, 39, 41]. For example, S-Prompts [39] learns in isolation a set of prompts per domain, and dynamically selects which set to use at test-time using a fixed key/value dictionary where the keys are computed with K-Means and the values represent the sets of prompts. L2P [41] uses an incrementally learnable key/value mechanism to select which prompts to prepend to the input image tokens at test-time, hence breaking the isolation between domains, which contrasts with our work, as it learns domain prompts independently. A main difference with these, and conventional DIL approaches, is that the proposed approach explicitly tackles the generability performance in domain incremental learning, while maintaining at par accuracy in known domains, which remains underexplored.

Domain generalization (DG) Existing literature on DG strongly relies on supervised knowledge from source domain data, regardless of whether it originates from a single domain [40] or multiple domains [11,44,47,48], which may not be realistic in continually changing scenarios, as knowledge comes in a sequential manner. Additionally, in scenarios involving distributional shifts, DG approaches primarily focus on the target domain, increasing the potential risk of catastrophic forgetting on previously learned domains [27].

3. Method

An overview of MoP-CLIP is illustrated in Fig. 3, which contains two phases: *i)* learning of in-distribution domain-specific visual and text prompts (sec. 3.2) and *ii)* selection of optimal prompts for a given test sample (sec. 3.3).

3.1. Problem definition

Let us denote as $\mathcal{S} = \{\mathcal{D}_s\}_{s=1}^{N}$ the sequence of datasets presented to the model in our incremental learning scenario, with N being the final number of domains. Each dataset is defined as $\mathcal{D}_s = \{\mathbf{x}_i^s, \mathbf{y}_i^s\}_{i=1}^{|\mathcal{D}_s|}$, where $\mathbf{x}_i \in \mathbb{R}^{W \times H \times C}$ represents an image of size $W \times H$ and C channels, and $\mathbf{y}_i \in \{0,1\}^K$ is its corresponding one-hot label for K target classes. In this setting, we have access to only one domain $\mathcal{D}_s$ at a time and storing samples from previous seen domains, commonly referred to as *exemplars*, is not allowed. Each time a new domain $\mathcal{D}_s$ becomes accessible, DIL aims to improve the model's performance on $\mathcal{D}_s$, while avoiding the loss of knowledge for past domains, $\mathcal{D}_{s-1}, \mathcal{D}_{s-2}, ...\mathcal{D}_1$.

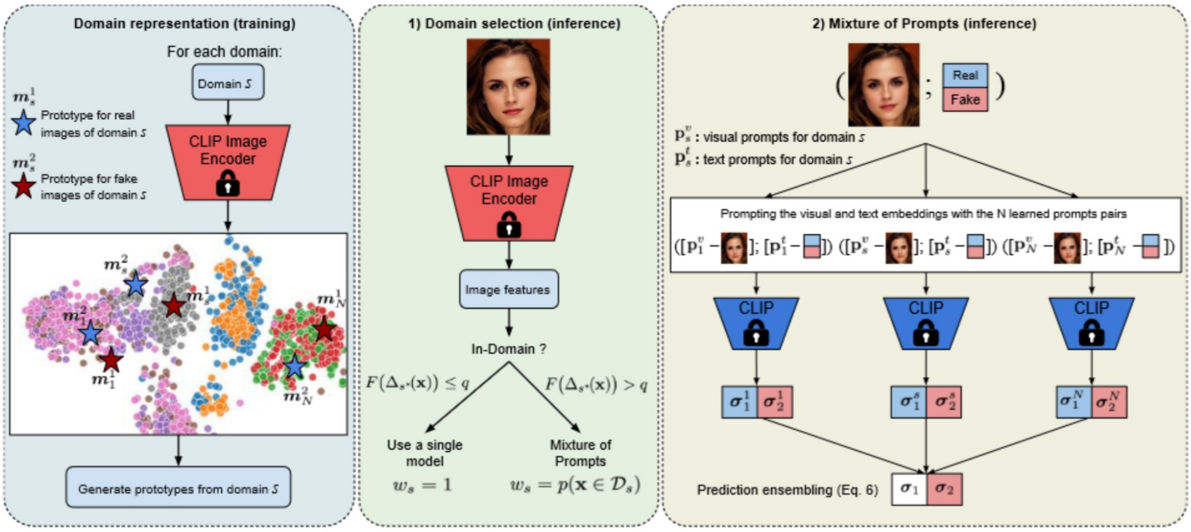

Figure 3. **Overview of MoP-CLIP.** The training phase (*left*): class-wise prototypes are identified from in-distribution domains. Inference (*middle* and *right*): domain selection and ensembling (Mixture of Prompts), respectively, for in-distribution and out-of-distribution samples. For simplicity, we depict the pipeline for 2 classes (Real *vs* Fake). However, the procedure for multiple classes (e.g., DomainNet or CoRE50) is exactly the same.

In the proposed setting, and in contrast to most existing literature on DIL, we assume that the model should also generalize well on unseen datasets, i.e., $\mathcal{D}_{s+1}, \mathcal{D}_{s+2}, ..., \mathcal{D}_{|\mathcal{D}_s|}$ (Fig. 2). In other words, our learning scenario leverages *backward transfer* to avoid catastrophic forgetting on seen domains, while optimizing *forward transfer* to facilitate knowledge transfer to new tasks/domains. Our motivation behind this bi-directional performance assessment relies on the realistic assumption that a distributional drift between training and testing data always exists.

3.2. Prompts Learning

Following the setting in [39], we define f_θ as the pre-trained vision transformer that generates a visual embedding $\mathbf{z}^v = f_\theta(\mathbf{x}_{\text{tok}}) \in \mathbb{R}^L$, where $\mathbf{x}_{\text{tok}} \in \mathbb{R}^{WH/R^2 \times M^v}$ corresponds to the image tokens (or patches), WH/R^2 is the number of tokens, R is the width/height of the (square) patch and M^v is the dimension of the image tokens embedding. We also define f_ϕ, a pre-trained text transformer that generates text embeddings of dimension M^t from class names tokens $\mathbf{c}_k$ for $k \in \{1, ..., K\}$. For each new domain $\mathcal{D}_s$ in the sequence $\mathcal{S}$, we can adapt the model by learning a visual prompt $\mathbf{p}_s^v \in \mathbb{R}^{L^v \times M^v}$ and a text prompt $\mathbf{p}_s^t \in \mathbb{R}^{L^t \times M^t}$, following [39]. In particular, these prompts are a set of continuous learnable parameters, where L^v, L^t are the visual and text prompt length. Thus, for the set of domains $\mathcal{S}$, we have a set of domain-specific visual and text prompts, denoted as $\mathcal{P}^v = \{\mathbf{p}_1^v, ..., \mathbf{p}_N^v\}$ and $\mathcal{P}^t = \{\mathbf{p}_1^t, ..., \mathbf{p}_N^t\}$. Now, with the domain-specific prompts, we can modify the embeddings that will be provided to the visual and text encoders, f_θ and f_ϕ. Concretely, for an image of domain s and class k, the input of the visual transformer is defined as $\tilde{\mathbf{x}}^v = [\mathbf{x}_{\text{tok}}, \mathbf{p}_s^v, \mathbf{x}_{\text{cls}}]$ with $\mathbf{x}_{\text{cls}}$ the classification token of the ViT. Similarly, the input of the text transformer is defined as $\tilde{\mathbf{c}}_k^t = [\mathbf{p}_s^t, \mathbf{c}_k]$. We then denote as $\tilde{\mathbf{z}}^v = f_\theta(\tilde{\mathbf{x}}^v)$ and $\tilde{\mathbf{z}}_k^t = f_\phi(\tilde{\mathbf{c}}_k^t)$ the embeddings of these inputs. The posterior probability of a given image $\mathbf{x}_i$ from $\mathcal{D}_s$ belonging to class k can be therefore defined as:

$$p(\mathbf{y}_k|\mathbf{x}, s) = \frac{e^{\cos(\tilde{\mathbf{z}}^v, \tilde{\mathbf{z}}_k^t)}}{\sum_{j=1}^{K} e^{\cos(\tilde{\mathbf{z}}^v, \tilde{\mathbf{z}}_j^t)}}, \tag{1}$$

where $\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \, \|\mathbf{b}\|}$ is the cosine similarity between vectors $\mathbf{a}$ and $\mathbf{b}$.

3.3. Inference

At test time, the domain of the images to classify remains unknown. In S-liPrompts [39], the domain s^* closest to a given test sample is selected by finding the minimum distance between the visual embeddings and prototypes computed with K-Means over the domains $\mathcal{S}$. This strategy is generally effective in finding the closest domain when $\mathbf{x} \in \mathcal{D}_s$ and $\mathcal{D}_s$ has been already presented to the model. In this setting, $p(\mathbf{y}_k|\mathbf{x}, s)$ yields satisfying predictions, as the domain of the sample $\mathbf{x}$ can be easily inferred and the scenario becomes a classification task under in-distribution data. Nevertheless, when the model has not been exposed to $\mathcal{D}_s$ during training or adaptation, the selection of an existing closest domain (other than $\mathcal{D}_s$) might not match with the real distribution of the new domain. In this case, the strategy used in S-liPrompts may actually move the test sample away from its original distribution. To overcome this issue, we propose to enhance the domain selection mechanism in two separate ways: *i)* dynamically allowing the model to select n close domains and *ii)* leveraging per-domain predictions in an ensembling scheme for samples of unseen domains.

To select the right prompt, we propose a strategy based on a set of class-specific prototypes for each domain, $\mathcal{E}_s = \{\boldsymbol{m}_s^k\}_{k=1}^K$, instead of prototypes obtained with K-Means as in [39]. Let $\mathcal{D}_s^k \subset \mathcal{D}_s$ be the samples of domain $\mathcal{D}_s$ belonging to the class k, we compute the prototype of class k for domain $\mathcal{D}_s$ by averaging the visual embeddings of examples in $\mathcal{D}_s^k$:

$$\boldsymbol{m}_s^k = \frac{1}{|\mathcal{D}_s^k|} \sum_{\{\boldsymbol{z}^v \,|\, \mathbf{x} \in \mathcal{D}_s^k\}} \boldsymbol{z}^v \tag{2}$$

Next, we present how these prototypes are used to select the domain and how they are leveraged in our approach.

i) Domain Selection. Given the class-specific prototypes, we select the domain s^* of a test example $\mathbf{x}$ as the one with the nearest prototype for any class:

$$s^* = \underset{1\leq s\leq N}{\operatorname{argmin}}\, \Delta_s(\mathbf{x}) \tag{3}$$

with

$$\Delta_s(\mathbf{x}) = \min_{\boldsymbol{m}_s^k \in \mathcal{E}_s} \|\mathbf{z}^v - \boldsymbol{m}_s^k\|_2. \tag{4}$$

As mentioned before, test examples may also come from an out-of-distribution (OOD) domain (i.e., not part of any domains encountered at training time). To determine if a given sample $\mathbf{x}$ is from a previously-seen domain or is OOD, we compare its distance to the closest prototype of the selected domain, $\Delta_{s^*}(\mathbf{x})$, with the distances of training samples from that domain. Let $\Psi_s^k = \big\{\|\mathbf{z}^v - \boldsymbol{m}_s^k\|_2 \,|\, \mathbf{x} \in \mathcal{D}_s^k\big\}$ be the set of distances for domain $\mathcal{D}_s$ and class k. During training, the distribution of distances for each domain $\mathcal{D}_s$ and class k is estimated from Ψ_s^k with a Gaussian of mean μ_s^k and standard deviation σ_s^k.

At test time, we find the class corresponding the nearest prototype for the selected domain, i.e., $k^* = \operatorname{argmin}_{1\leq k\leq K} \|\mathbf{z}^v - \boldsymbol{m}_{s^*}^k\|_2$. We then use the distribution $P = \mathcal{N}(\cdot\,; \mu_{s^*}^{k^*}, \sigma_{s^*}^{k^*})$ to determine whether $\Delta_{s^*}(\mathbf{x})$ is normal. Specifically, we classify a sample $\mathbf{x}$ as in-distribution if $F(\Delta_{s^*}(\mathbf{x})) \leq q$ where F is the cumulative distribution function of P, i.e., $F(x) = P(X \leq x)$ and q is a specified threshold.

Afterwards, if $\mathbf{x}$ is in-distribution, we use $p(\mathbf{y}_k \,|\, \mathbf{x}, s^*)$ to classify $\mathbf{x}$. Otherwise, $\mathbf{x}$ belongs to a new (unseen) domain. In such case, we propose the following ensembling technique to classify it.

ii) Ensembling If $\mathbf{x} \in \mathcal{D}_{s'}$ and $\mathcal{D}_{s'}$ has not been encountered during training, we model $\mathbf{z}^v$ as being part of a mixture of the known domains. In particular, we resort to a Gaussian mixture model to estimate the mixture weights ($w_s = p(s|\mathbf{x})$). While this could be done with L-dimensional covariance and mean vectors per domain (on the features), it does not perform well as L increases. We propose the following model:

$$\begin{aligned} w_s &= p(\mathbf{x} \in \mathcal{D}_s) \\ &= \frac{\mathcal{N}(\Delta_s(\mathbf{x}); \mu_s^{k^*}, \sigma_s^{k^*})}{\sum_j \mathcal{N}(\Delta_j(\mathbf{x}); \mu_j^{t^*}, \sigma_j^{t^*})}, \end{aligned} \tag{5}$$

where $t^* = \operatorname{argmin}_{1\leq k\leq K} \|\mathbf{z}^v - \boldsymbol{m}_j^k\|_2$. Note that the hypotheses done to reach the proposed model in eq. (5) are detailed in Supplemental Material. We then combine the predictions using the different prompts ($p(\mathbf{y}_k|\mathbf{x}, s)$) based on those weights:

$$p(\mathbf{y}_k|\mathbf{x}) = \sum_{s=1}^{N} p(\mathbf{y}_k|\mathbf{x}, s) \cdot w_s \tag{6}$$

4. Experiments

The experiments reported in this section validate empirically that MoP-CLIP yields competitive performance compared to state-of-the-art DIL when dealing with in-domain (ID) examples, while significantly outperforming these approaches in the presence of out-of-domain (OOD) examples. Furthermore, we perform a series of ablation experiments to better identify the impact of the key components of the proposed method.

4.1. Experimental setup

A. Datasets. To assess the performance of the proposed method, we resort to three popular DIL benchmarks which have been extensively used in the literature: CDDB-Hard [25], DomainNet [33], and CORe50 [28], whose details are given below:

CDDB Dataset [25] is a continual (incremental) deepfake detection benchmark, whose goal is to identify real and fake images across different domains. In particular, in the proposed work we employ the Hard setting as in [39], which is the most challenging track of CDDB. This dataset contains a total of 27,000 images across 5 different domains: GauGAN, BigGAN, WildDeepfake, WhichFaceReal, and SAN. We also use Glow, StarGAN and CycleGAN to evaluate OOD performance.

DomainNet [33] is a dataset for domain adaptation commonly used to benchmark DIL methods. It contains a total of 600,000 images across 6 different domains, each containing the same 345 classes. In particular, we use the experimental setup presented in CaSSLe [15].

CORe50 [28] is a dataset designed for continual object recognition. However, in this work we focus on its domain-incremental learning scenario. This setting is comprised of

11 distinct domains, each containing the same 50 object categories. From the 11 domains, 8 are composed of 120,000 images which are seen sequentially during training, whereas the remaining 3 domains compose the fixed unseen test set.

B. Comparison methods. We benchmark MoP-CLIP to several state-of-the-art DIL methods. These include **non-prompting** approaches (EWC [22], LwF [26], ER [9], GDumb [34], BiC [42], DER++ [6] and Co^2L [7]), **prompting-based** methods (L2P [41], DyTox [14] and S-liPrompts [39]) and a **self-supervised** learning method, CaSSLe [15], following the experimental set-up in [39]. For OOD experiments, we only evaluate those methods that are in direct competition with our approach, in terms of *exemplars* buffer use. In particular, we compare to the following methods, whose respective codes are publicly available: EWC[1], LwF[2], DyTox[3], L2P[4], and S-liprompts[5].

C. Evaluation metrics and protocol. o assess the performance of the proposed approach, we resort to standard metrics in the incremental learning literature. **In-domain setting:** On DomainNet and CDDB-Hard we follow the original work in [25] and employ the average classification accuracy (AA), as well as the average forgetting degree (AF), which is the mean of the popular backward transfer degradation (BWT). We formally define the average accuracy as $AA = \frac{1}{N}\sum_{i=1}^{N} A_{i,N}$ with $A_{i,N}$ the accuracy on domain i measured after having trained on N domains. This metric is computed at the end, i.e., after having seen all the domains, e.g., on CDDB: GauGAN $\rightarrow$ BigGAN$\rightarrow$ WildDeepfake$\rightarrow$ WhichFaceReal$\rightarrow$ SAN. Furthermore, the average forgetting degree on CDDB can be defined as $\frac{1}{N-1}\sum_{i=1}^{N-1} BWT_i$ with $BWT_i = \frac{1}{N-i-1}\sum_{j=i+1}^{N}(A_{i,j} - A_{i,i})$ as originally proposed in [25] (i.e., the forgetting degree is computed for each domain at each adaptation step, then averaged). **Out-of-domain setting:** We follow [28] to compute the AA on CORe50 on the fixed test set, which contains 3 hold-out splits that can be considered as OOD with respect to the training set. Furthermore, as in [39], we compute the AA on 3 unseen domains (Glow, StarGAN and CycleGAN) in CDDB-Hard. Last, as no independent hold-out subset of unseen domains exists for DomainNet, we propose using the Cumulative Accuracy on the unseen domains during the incremental learning of the model (i.e., average accuracy on the unseen domains averaged on all the steps), defined as follows: $CA = \frac{1}{N-1}\sum_{i=1}^{N-1}\frac{1}{N-j-1}\sum_{i=j}^{N} A_{i,j}$.

[1] https://github.com/G-U-N/PyCIL/
[2] https://github.com/G-U-N/PyCIL/
[3] https://github.com/arthurdouillard/dytox
[4] https://github.com/JH-LEE-KR/l2p-pytorch
[5] https://github.com/iamwangyabin/S-Prompts

D. Implementation details We use the same setting as [39], i.e. use ViT-B/16 [13] as our base image encoder and the text encoder of CLIP, both initialized by CLIP pretraining on ImageNet [36]. We follow [39] and use the same image encoder model as a backbone (i.e., ViT-B/16 [13] pretrained on ImageNet [36]) across all the compared methods, for a fair comparison. As suggested in [39], we use a more advanced backbone (i.e. ConViT pretrained on ImageNet [36]) on DyTox [14] as it underperforms a random model with ViT-B/16 as backbone. We empirically fix $q = 0.94$ for the 3 datasets, based on the ablation study in Figure 5, such that we do not deteriorate ID performance while improving OOD performance on CDDB-Hard. For EWC, LwF and CaSSLe, we use the same hyperparameters as in the original papers, whereas we keep the hyperparameters reported in [39] for DyTox, L2P and S-Prompts.

4.2. Results

In-domain distributions. We first evaluate the proposed approach in the standard DIL scenario where the testing samples are drawn from the same distribution as the training/adaptation images. These results, which are reported under the *Seen-Domains* columns of Tables 1 and 2, demonstrate that the proposed MoP-CLIP approach yields superior performance than existing *exemplar-free* methods. In particular, MoP-CLIP outperforms the very recent approaches DyTox [14] and L2P [41] by large margin, with improvement gains of around 20-30% in terms of average classification accuracy under the same storage conditions. Furthermore, the degree of knowledge forgetting is also largely reduced, going from -45.85 in DyTox to -0.79 in our approach. Furthermore, if storing exemplars is allowed, DyTox [14] significantly improves its performance, but still underperforms our approach yet incurring a non-negligible overhead. Last, it is noteworthy to highlight that the proposed approach reaches similar performance than S-liPrompts [39] in this scenario, with at par values in the CDDB-Hard dataset and remarkable performance gains in DomainNet. Note that this result is somehow expected, as our approach is a generalization of S-liPrompts for the OOD scenario, and differences in the in-distribution setting may come from the domain prompt selected.

An interesting observation is that prompting-based methods, which do not store exemplars from old tasks, typically outperform their buffer-storage counterparts. For example, S-liPrompts [39] and MoP-CLIP bring considerable improvements compared to LUCIR (between 6-8%) or iCaRL (ranging from 9 to 15%). We hypothesize that this phenomenon comes from the absence of interference between domains when doing the adaptation. In this scenario, the knowledge from previously learned domains remains isolated in the form of optimized domain prompts, and the only knowledge shared is derived from pre-trained transformers.

Table 1. **Results on CDDB-Hard for both ID and OOD scenarios.** Evaluation of existing state-of-the-art DIL methods in the standard *seen-domain* setting and more challenging *unseen-domain* scenario. For the unseen-domain experiments, we only reproduced the results for related (i.e., *exemplar-free*) methods. Best results are highlighted in **bold**.

Method	Prompt	Pretraining	Buffer size	Seen-Domains AA (↑)	Seen-Domains AF (↓)	Unseen-Domains AA (↑)
LRCIL $_{\text{IROS'20}}$ [32]	✗	ImageNet		76.39	-4.39	-
iCaRL $_{\text{WIFS'19}}$ [31]	✗	ImageNet	*100ex/class*	79.76	-8.73	-
LUCIR $_{\text{CVPR'19}}$ [18]	✗	ImageNet		82.53	-5.34	-
LRCIL $_{\text{IROS'20}}$ [32]	✗	ImageNet		74.01	-8.62	-
iCaRL $_{\text{WIFS'19}}$ [31]	✗	ImageNet	*50ex/class*	73.98	-14.50	-
LUCIR $_{\text{CVPR'19}}$ [18]	✗	ImageNet		80.77	-7.85	-
DyTox $_{\text{CVPR'22}}$ [14]	✓	ImageNet		86.21	-1.55	-
EWC $_{\text{PNAS'17}}$ [22]	✗	ImageNet		50.59	-42.62	-
LwF $_{\text{TPAMI'17}}$ [26]	✗	ImageNet		60.94	-13.53	50.05
DyTox $_{\text{CVPR'22}}$ [14]	✓	ImageNet	*No buffer*	51.27	-45.85	50.46
L2P $_{\text{CVPR'22}}$ [41]	✓	ImageNet		61.28	-9.23	57.34
S-liPrompts $_{\text{NeurIPS'22}}$ [39]	✓	CLIP		**88.65**	**-0.69**	76.79
MoP-CLIP (ours)	✓	CLIP		88.54	-0.79	**82.02**

Table 2. **Results on DomainNet for both ID (AA metric) and OOD (CA metric) scenarios**. Best values are highlighted in **bold**.

Method	Prompt	Pretraining	Buffer size	Seen Domains	Unseen Domains
DyTox $_{\text{CVPR'22}}$ [14]	✓	ImageNet	*50ex/class*	62.9	-
DyTox $_{\text{CVPR'22}}$ [14]	✓	ImageNet		13.5	4.2
LwF $_{\text{TPAMI'17}}$ [26]	✗	ImageNet		49.2	43.4
CaSSLe $_{\text{CVPR'22}}$ [15](SimCLR [12])	✗	ImageNet		48.1	45.4
CaSSLe $_{\text{CVPR'22}}$ [15](BYOL [16])	✗	ImageNet		52.9	48.7
CaSSLe $_{\text{CVPR'22}}$ [15](Barlow Twins [45])	✗	ImageNet	*No buffer*	51.4	47.6
CaSSLe $_{\text{CVPR'22}}$ [15](SupCon [21])	✗	ImageNet		54.2	50.5
L2P $_{\text{CVPR'22}}$ [41]	✓	ImageNet		40.1	25.5
S-liPrompts $_{\text{NeurIPS'22}}$ [39]	✓	CLIP		67.7	66.4
MoP-CLIP (Ours)	✓	CLIP		**69.7**	**67.0**

Table 3. **Results on CORe50.** Note that CORe50 already provides separate training and testing domains, and thus results can only be computed on the **OOD scenario**. Results are reported as the Acc metric, where the best values are highlighted in **bold**. In our method, we use the same q as in the other datasets, whereas * indicates that q is fixed based on the validation set of CORe50, as typically done in all the other approaches.

Method	Prompt	Pretraining	Buffer size	AA
GDumb $_{\text{ECCV'20}}$ [34]	✗	ImageNet		74.92
BiC $_{\text{CVPR'19}}$ [42]	✗	ImageNet		79.28
DER++ $_{\text{NeurIPS'20}}$ [6]	✗	ImageNet	*50ex/class*	79.70
Co^2L $_{\text{ICCV'21}}$ [7]	✗	ImageNet		79.75
DyTox $_{\text{CVPR'22}}$ [14]	✓	ImageNet		79.21
L2P $_{\text{CVPR'22}}$ [41]	✓	ImageNet		81.07
EWC $_{\text{PNAS'17}}$ [22]	✗	ImageNet		74.82
LwF $_{\text{TPAMI'17}}$ [26]	✗	ImageNet		75.45
L2P $_{\text{CVPR'22}}$ [41]	✓	ImageNet	*No buffer*	78.33
S-liPrompts $_{\text{NeurIPS'22}}$ [39]	✓	CLIP		89.06
MoP-CLIP (Ours)	✓	CLIP		**91.43**
MoP-CLIP (Ours)*	✓	CLIP		**92.29**

Performance under domain distributional shift. We now want to assess the benefits of the proposed approach when the testing dataset presents a distributional drift over the training data. In particular, we advocated that the proposed approach is a generalization of [39] to be able to handle samples coming from an unseen distribution. To support this claim, and to demonstrate the superiority of our approach on unseen domains, we resort to the OOD experiments, which are reported in the right-most columns of Tables 1 and 2, as well as Table 3. From these results, we can observe that excluding S-liPrompts, the performance gains brought by the proposed approach are substantial compared to other *exemplar-free* methods, ranging from 17% (EWC in CORe50) to 40% (L2P [41] in DomainNet). Even when comparing to state-of-the-art competitors that store exemplars (e.g., DyTox [14] or Co^2L [7] in CORe50), MoP-CLIP yields considerable improvements, ranging from 11% to nearly 17%. The clear superiority of our approach lies on the isolation of different domains during learning, which do not degenerate the generalization capabilities brought by the pre-trained transformers. Furthermore, when comparing the proposed MoP-CLIP to S-liPrompts [39], we observe that our method outperforms the latter by around 6%, 2% and 3% in CDDB-Hard, DomainNet and CORe50 benchmarks, respectively. These performance gains on OOD samples might likely come from the flexibility of MoP-CLIP in selecting a subset of similar domains for a given test sample, which allows the model to properly weight the contribution of each domain prompt. In contrast, S-liPrompts [39] forces the model to select only one domain from the seen domains, which impedes its scalability to novel distributions, as empirically shown in these results, as well as in Figure 1.

On the impact of the different components. The empirical study in Table 4 justifies the need of employing the proposed approach over the strong baseline S-liPrompts [39], as well as showcases the impact of each choice. In a practical scenario, it is unrealistic to assume that the test samples always follow the same distribution as the data used for adaptation. Furthermore, the domain of each sample typically remains unknown. Thus, to align with real-world conditions, we will consider the average of in-distribution and out-of-distribution performance as our metric of reference to evaluate the impact of the different choices. We can observe that in nearly all the cases, the use of an ensembling strategy results in consistent improvements over the single model predictions (considering same distances). An interesting observation is that distances related to the L_2-norm typically degrade the performance on ID samples. We observe that in this scenario, the distributions overlap considerably and $p(s|\mathbf{x})$ (derived from the Gaussian mixture) is too far from 1 for most ID samples, making the discrimination of samples by these distance measures difficult. Nevertheless, this behavior is reversed in the presence of OOD samples. In particular, our simplification assumes an isotropic Gaussian distribution of the points around the prototypes and therefore reduces the noise in the coordinate-wise variances (which can explain the performance degradation observed when using the Mahanalobis distance), replacing it with distance-wise variances. Thus, the proposed approach combines the best of both worlds, leading to the best average performance across all the configurations.

Strategy to select the domain prompts. As emphasized in Sec. 3.3, [39] uses K-Means over the features extracted

Table 4. **Impact of MoP-CLIP design choices.** *Maha* denotes the Mahanalobis distance, and GMM a Gaussian Mixture Model. Furthermore, *Hybrid* denotes the nature of our approach, which uses an ensembling for OOD samples and a single domain prompt for ID samples. Results on CDDB-Hard and DomainNet (DN) show the average accuracy (AA). Best results in **bold**.

Method	Ensembling	Distance	Seen Domains		Unseen Domains		Harm. Mean	
			CDDB	DN	CDDB	DN	CDDB	DN
S-liPrompts [39]	✗	L1	88.65	67.70	76.79	66.40	82.29	65.54
MoP-CLIP - no ens. (a)	✗	L2	**89.48**	69.52	76.95	66.58	82.74	68.02
-	✗	Maha	80.45	64.54	76.66	65.61	78.51	65.07
-	✗	L2-GMM	75.72	61.95	75.76	62.47	75.74	62.21
-	✓	Uniform	67.55	62.72	83.61	63.14	74.73	62.93
-	✓	L1	89.29	54.23	80.05	56.16	84.42	55.18
-	✓	L2	68.37	63.29	84.07	63.84	75.41	63.56
-	✓	Maha	80.48	61.97	77.56	55.38	78.99	58.49
MoP-CLIP - ens. (b)	✓	L2-GMM	72.51	64.21	**89.21**	64.42	80.00	64.31
MoP-CLIP (Proposed)	Hybrid	ID (a)/ OOD (b)	88.54	**69.67**	82.02	**67.05**	**85.21**	**68.33**

with a pre-trained ViT to compute the prototypes which are used to dynamically select which prompt to use at test time. While this strategy is memory efficient, it lacks flexibility, as the number of clusters needs to be adjusted according to the dataset employed. To alleviate this issue, we instead use class-wise prototypes as a *hyperparameter-free* alternative to compute representative prototypes. The effect of using either k-Means or class-prototypes is depicted in Fig. 4. From these results, we empirically observe that this choice improves performance in both in-distribution and out-of-distribution domains, leading to a higher average performance. Furthermore, it is noteworthy to mention that using class-wise prototypes makes the distribution of points around prototypes Gaussian, which explains the satisfactory performance of MoP-CLIP, particularly on samples from unseen domains.

Figure 4. **k-Means or class prototypes as domain centroids?** Ablation study that demonstrates the benefits of using class prototypes (our approach) rather than k-Means prototypes, as in [39].

How much trade-off is sufficient? The influence of the threshold q from our simple out-of-distribution criterion (Sec. 3.3) to select between seen and unseen domains is shown in Figure 5. As stressed earlier, we aim for a compromise between ID and OOD performance, in order to provide generalizable models. As target domains should remain unknown at inference, we selected a fixed q value that provided the optimal average performance across both settings. Nevertheless, these plots reveal two interesting findings. First, the average performance of the model is not very sensitive to the choice of q. For example, the performance of ID samples decreases as q decreases, whereas OOD performance improves. On the other hand, if q increases, the accuracy in the ID scenario increases, while it decreases for OOD samples. And second, if prior knowledge about the target domain is available –an assumption made by all existing DIL literature– the performance of MoP-CLIP is further increased, enlarging the gap with SOTA methods.

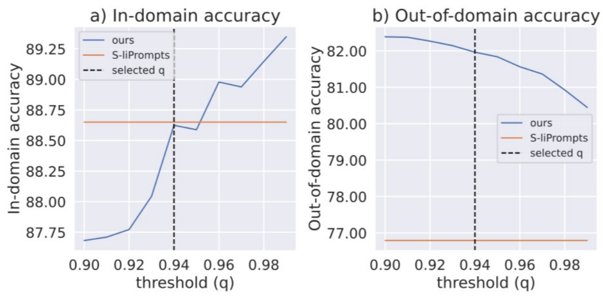

Figure 5. **A controllable trade-off between in-domain and out-of-domain prediction performances.** Impact of the threshold q (Sec. 3.3) on the accuracy, evaluated on CDDB-Hard.

5. Conclusion

Findings from this work reveal that existing literature on domain incremental learning suffers under the presence of distributional drift, hampering their scalability to practical scenarios. To overcome this issue, we have proposed a generalization of the recent S-ilPrompts [39] approach, that further handles out-of-distribution samples. In addition to outperforming current state-of-the-art, particularly in the unseen domain setting, our method brings several interesting benefits compared to most existing DIL method. First, MoP-CLIP is *exemplar-free*, eliminating the limitations of conventional DIL approaches in terms of storage and privacy. Furthermore, as prompts are learned independently on each domain, and the model parameters remain fixed during the adaptation, the performance of our approach is insensitive to the ordering of the seen domains. This contrasts with a whole body of the literature, where the choice of the sequence order can significantly impact the final performance. Our comprehensive evaluation shows the empirical gains provided by MoP-CLIP, pointing to visual prompt tuning as an appealing alternative for general domain incremental learning. Finally, we stress that while powerful, the proposed approach retains the spirit of S-ilPrompts [39], which advocates for a simple yet elegant method.

Potential Negative Impact: Language-vision models and prompt tuning heavily rely on pre-training data which may contain biases and reinforce existing societal prejudices.

References

[1] Hongjoon Ahn, Jihwan Kwak, Subin Lim, Hyeonsu Bang, Hyojun Kim, and Taesup Moon. Ss-il: Separated softmax for incremental learning. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 844–853, 2021. 1

[2] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32, 2019. 1

[3] Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8218–8227, 2021. 1

[4] Benjamin Bowman, Alessandro Achille, Luca Zancato, Matthew Trager, Pramuditha Perera, Giovanni Paolini, and Stefano Soatto. a-la-carte prompt tuning (apt): Combining distinct data via composable prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14984–14993, 2023. 3

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 2

[6] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *NeurIPS*, 2020. 6, 7

[7] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 9516–9525, 2021. 6, 7

[8] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, pages 532–547, 2018. 2

[9] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019. 2, 6

[10] Aochuan Chen, Yuguang Yao, Pin-Yu Chen, Yihua Zhang, and Sijia Liu. Understanding and improving visual prompting: A label-mapping perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19133–19143, 2023. 3

[11] Chaoqi Chen, Jiongcheng Li, Xiaoguang Han, Xiaoqing Liu, and Yizhou Yu. Compound domain generalization via metaknowledge encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7119–7129, 2022. 3

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 7

[13] 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. In *International Conference on Learning Representations*, 2021. 6

[14] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9285–9295, 2022. 1, 2, 3, 6, 7

[15] Enrico Fini, Victor G Turrisi Da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-supervised models are continual learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2022. 5, 6, 7

[16] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 7

[17] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Lifelong learning via progressive distillation and retrospection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 437–452, 2018. 1

[18] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 831–839, 2019. 1, 7

[19] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pages 709–727. Springer, 2022. 3

[20] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*, pages 105–124. Springer, 2022. 2

[21] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020. 7

[22] 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. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 2, 6, 7

[23] Teven Le Scao and Alexander M Rush. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, pages 2627–2636, 2021. 1

[24] Kibok Lee, Kimin Lee, Jinwoo Shin, and Honglak Lee. Overcoming catastrophic forgetting with unlabeled data in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 312–321, 2019. 1

[25] Chuqiao Li, Zhiwu Huang, Danda Pani Paudel, Yabin Wang, Mohamad Shahbazi, Xiaopeng Hong, and Luc Van Gool. A continual deepfake detection benchmark: Dataset, methods, and essentials. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1339–1349, 2023. 5, 6

[26] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 2, 6, 7

[27] Chenxi Liu, Lixu Wang, Lingjuan Lyu, Chen Sun, Xiao Wang, and Qi Zhu. Deja vu: Continual model generalization for unseen domains. In *International Conference on Learning Representations*, 2023. 3

[28] Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. *CoRR*, abs/1705.03550, 2017. 2, 5, 6

[29] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017. 2

[30] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5206–5215, 2022. 2

[31] Francesco Marra, Cristiano Saltori, Giulia Boato, and Luisa Verdoliva. Incremental learning for the detection and classification of gan-generated images. In *2019 IEEE international workshop on information forensics and security (WIFS)*, pages 1–6. IEEE, 2019. 2, 7

[32] Lorenzo Pellegrini, Gabriele Graffieti, Vincenzo Lomonaco, and Davide Maltoni. Latent replay for real-time continual learning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10203–10209. IEEE, 2020. 2, 7

[33] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019. 5

[34] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *ECCV*, 2020. 6, 7

[35] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019. 1

[36] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. 6

[37] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017. 2

[38] Kihyuk Sohn, Huiwen Chang, José Lezama, Luisa Polania, Han Zhang, Yuan Hao, Irfan Essa, and Lu Jiang. Visual prompt tuning for generative transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19840–19851, 2023. 3

[39] Yabin Wang, Zhiwu Huang, and Xiaopeng Hong. S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 1, 2, 3, 4, 5, 6, 7, 8, 13

[40] Zijian Wang, Yadan Luo, Ruihong Qiu, Zi Huang, and Mahsa Baktashmotlagh. Learning to diversify for single domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 834–843, 2021. 3

[41] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022. 1, 2, 3, 6, 7

[42] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *CVPR*, 2019. 6, 7

[43] Yinghui Xing, Qirui Wu, De Cheng, Shizhou Zhang, Guoqiang Liang, and Yanning Zhang. Class-aware visual prompt tuning for vision-language pre-trained model. *arXiv preprint arXiv:2208.08340*, 2022. 3

[44] Xufeng Yao, Yang Bai, Xinyun Zhang, Yuechen Zhang, Qi Sun, Ran Chen, Ruiyu Li, and Bei Yu. Pcl: Proxy-based contrastive learning for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7097–7107, 2022. 3

[45] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021. 7

[46] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017. 2

[47] Xingxuan Zhang, Yue He, Renzhe Xu, Han Yu, Zheyan Shen, and Peng Cui. Nico++: Towards better benchmarking for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16036–16047, 2023. 3

[48] Yabin Zhang, Minghan Li, Ruihuang Li, Kui Jia, and Lei Zhang. Exact feature distribution matching for arbitrary style transfer and domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8035–8045, 2022. 3

[49] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language mod-

els. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022. 1, 2

[50] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 2