Generating Instance-level Prompts for Rehearsal-free Continual Learning

Dahuin Jung[1,*], Dongyoon Han[2], Jihwan Bang[3], Hwanjun Song[4,*,†]
[1] Department of Electrical and Computer Engineering, Seoul National University, Seoul, Korea,
[2] NAVER AI Lab, [3] NAVER Cloud, [4] AWS AI Labs
annajung0625@snu.ac.kr, {dongyoon.han, jihwan.bang}@navercorp.com, hwanjuns@amazon.com

Abstract

We introduce Domain-Adaptive Prompt (DAP), a novel method for continual learning using Vision Transformers (ViT). Prompt-based continual learning has recently gained attention due to its rehearsal-free nature. Currently, the prompt pool, which is suggested by prompt-based continual learning, is key to effectively exploiting the frozen pre-trained ViT backbone in a sequence of tasks. However, we observe that the use of a prompt pool creates a domain scalability problem between pre-training and continual learning. This problem arises due to the inherent encoding of group-level instructions within the prompt pool. To address this problem, we propose DAP, a pool-free approach that generates a suitable prompt in an instance-level manner at inference time. We optimize an adaptive prompt generator that creates instance-specific fine-grained instructions required for each input, enabling enhanced model plasticity and reduced forgetting. Our experiments on seven datasets with varying degrees of domain similarity to ImageNet demonstrate the superiority of DAP over state-of-the-art prompt-based methods. Code is publicly available at https://github.com/naver-ai/dap-cl.

1. Introduction

Humans can learn and solve continuously emerging tasks by leveraging knowledge from past experiences. Inspired by it, continual learning (CL) methods aim at tackling a sequence of tasks using a single model without experiencing performance deterioration in previously learned tasks [1, 56, 59]. Typically, *rehearsal*-based methods [5, 7, 22, 42], motivated by the complementary learning systems of humans [35], store a data subset of past tasks to alleviate forgetting while acquiring new information. By maintaining a replay buffer of reasonable size, this approach has shown superiority over regularization [8, 25] and architecture-based methods [2, 33] in various settings. However, the rehearsal-based approach is reluctant to use when data privacy is concerned, or the memory budget is tight. In this regard, lately, *prompt*-based rehearsal-free CL methods, such as L2P [58] and DualPrompt [57], have been presented and proved to outperform existing rehearsal-based methods without relying on a replay buffer.

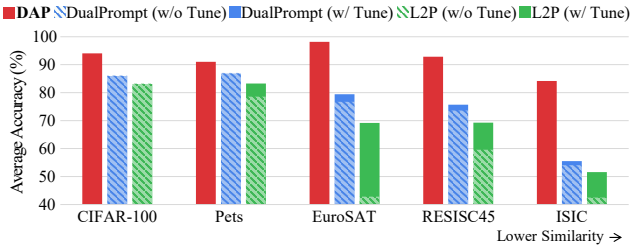
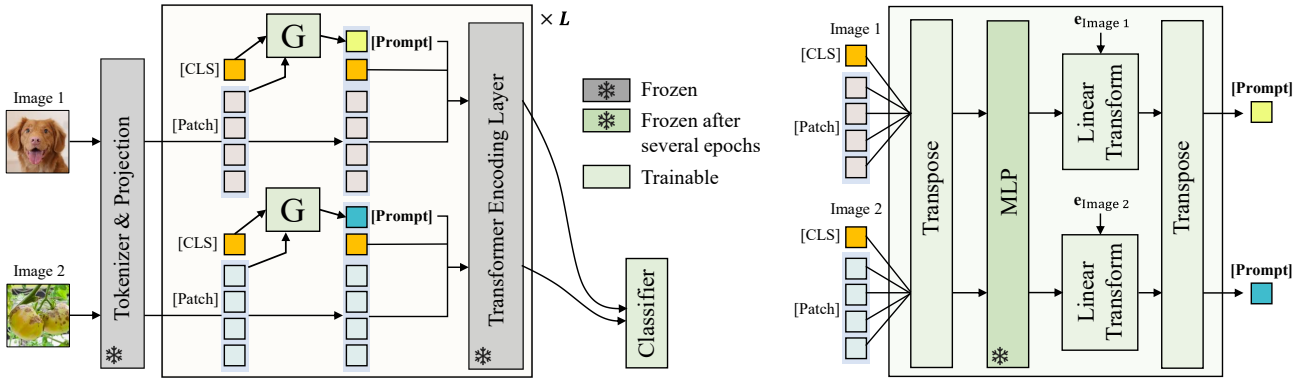


Figure 1. Average accuracy changes of prompt-based CL methods including DAP across five datasets with varying levels of domain similarity to ImageNet. The datasets are sorted with domain similarity in descending order from left to right.

Prompt-based learning was initially introduced in the field of natural language processing for effective transfer learning [31]. Instead of fine-tuning entire weights, it is able to condition an untouched (frozen) pre-trained backbone such that it performs well to a specific downstream task. That is, only small additional weights (prompts) are trained to adjust learned representations from a source task to a new target task. Recently, with the recent advent of Vision Transformers (ViT) [15], the notion of prompt-based learning has been adapted in CL [14, 47, 51, 55, 57, 58]. L2P and DualPrompt maintain a *prompt pool*, which is a constant number of prompts to learn shared prompts across tasks to mitigate forgetting as well as to benefit from previously learned task knowledge. Hence, the prompt pool is considered a set of instructions to tune the frozen backbone to adapt to a sequence of new tasks.

Currently, most pre-trained backbones frozen in prompt-based learning are assumed to be trained on a large-scale natural image collection like ImageNet; L2P and Dual-

*Work was done while working at NAVER AI Lab.
†Corresponding author.



(a) Domain-Adaptive Prompt (DAP). (b) Adaptive Prompt Generator (G in (a)).

Figure 2. Overview of DAP (left) and the architecture of the proposed adaptive prompt generator (G) (right).

Prompt also utilize the ImageNet-pretrained ViT as a frozen feature extractor. However, testing benchmarks in prior literature [57, 58] are still confined only to natural images, despite no assumptions about data domain in CL. Data with varying levels of domain similarity to natural images can be provided as target tasks in the CL setup, e.g., EuroSAT data of satellite images [19] and ISIC data of skin diseases [12]. As such, prompts should be designed to encode more domain-relevant knowledge and have the ability to finely instruct the model to properly leverage learned representations. Our findings in Figure 1 reveal that existing prompt-based CL methods exhibit two weaknesses concerning domain generalization, and thus they cannot be consistently effective across images with varying domains.

Their weaknesses arise since they rely on the prompt pool. First, they require *hyperparamter tuning* for the target domain in advance. The prompt pool can support varying levels of domain similarity by adjusting the pool size, and controlling the domain coverage of prompts. However, it is unrealistic to assume prior knowledge of the target domain in CL. Additionally, an expansion of the prompt pool leads to an increase in the memory budget. As shown in Figure 1, the two representative prompt-based CL methods show a significant performance degradation with a decrease in domain similarity. The best pool sizes (w/ Tune in Figure 1), which were found via a grid search for each data domain in advance, improve the accuracy of the model, but a large performance drop still appears. Second, since the number of prompts in the pool is typically much smaller than the number of total training instances, each prompt is forced to be optimized in a *group-level* fashion. This makes it difficult to provide delicate instructions to the model more suitable per data instance, possibly resulting in better domain generalization.

In this paper, we propose a *pool-free* prompt-based CL framework named **DAP**, supporting domain-adaptive CL using the frozen backbone pre-trained on ImageNet. As shown in Figure 2(a), without relying on the pool of a constant number of prompts, it adaptively generates a single prompt per instance from input tokens; hence, DAP does not need to choose learned prompts from the pool, unlike L2P and DualPrompt. The adaptive prompt encodes domain-relevant knowledge corresponding to the target task, delicately steering the frozen backbone's representation via the attention mechanism in ViT. Specifically as shown in Figure 2(b), a feed-forward network and a linear transformation are utilized to extract instance-specific information conditioned on a transposed input, creating the adaptive prompt in a timely manner. With the poolless instance-level prompts, DAP becomes versatile for various CL tasks without domain restrictions, even when applied to a large benchmark. Our main contributions are:

- This is the first study to pose and examine the domain scalability problem of the current prompt-based CL methods on benchmarks with varying levels of domain similarity to natural images.
- We propose a novel framework named DAP, which no longer relies on the prompt pool and generates each prompt in an instance-level manner, facilitating enhanced plasticity and reduced forgetting.
- We conduct extensive experiments on seven datasets with varying domains, including satellite, dermatology, and radiology images. DAP significantly outperforms the state-of-the-art prompt-based CL methods.

2. Related Work

Prompting for Transfer Learning. Originally, prompting [31] refers to adding heuristic language instructions to the input text to help a frozen pre-trained language model understand a downstream task. Typically, the design of a prompting function has been in a heuristic form, and GPT-3 [6] showed excellent generalization performance on transfer learning tasks using manually created prompts. How-

ever, recently, prompting methods such as Prompt Tuning [28] and Prefix Tuning [29] suggest the notion of learnable prompts in continuous space, and this technique becomes mainstream in prompt-based learning. The current prompt-based CL methods, L2P and DualPrompt, including DAP (ours), also belong to this direction. However, note that DAP is the only method that creates a prompt on-the-fly in an instance-level fashion using an adaptive prompt generator (in Figure 2(b)).

Continual Learning. For CL scenarios, there are multiple setups depending on which type of data is provided as the next task: class incremental learning [1, 5, 8, 26, 42, 57, 58] where classes are incremented from the same domain, domain incremental learning [11, 16, 48] where same classes from different domains, and cross-domain continual learning [46] where classes are incremented from different domains. We mainly focus on the class incremental learning, which is the most general setup [34] in the recent literature on the CL community.

Class incremental learning has established a couple of research directions. Several studies have been initially proposed based on regularization [1, 8, 26]. Although they succeeded to mitigate catastrophic forgetting by constraining the fast update of important parameters, it is difficult to preserve past knowledge completely when the number of incoming tasks increases. In this regard, a rehearsal-based approach [3, 5, 7, 42] maintains a replay buffer (episodic memory) which can store a small amount of data from the previous tasks. Active research has been made to decide which instances to store for high diversity and uncertainty in the buffer [3, 5, 7, 42]. However, this direction is not applicable to scenarios where data privacy or memory budget is concerned.

Currently, there is a growing research trend in CL based on prompting and Vision Transformers (ViTs). They leverage a frozen backbone pre-trained on ImageNet without maintaining the replay buffer. L2P [58] firstly introduces the notion of the prompt pool to tune the frozen ViT backbone for CL tasks. Then, inspired by complementary learning systems, DualPrompt [57] utilizes two different types of prompts, G-Prompt and E-Prompt, whose goals are learning task-invariant and task-specific knowledge, respectively. Although prompt-based CL research has recently been expanded to domain incremental [55] or multimodal [14, 20, 23, 51] learning, prompt-based CL is still in the early stage, and there were not enough studies or benchmarks to examine the influence of domain similarity between source data on pre-training and target data of CL. In particular, the weaknesses of assuming the prompt pool with a fixed size have yet to unveil. In this paper, we verify the limitation of current prompt-based CL methods and propose a pool-free prompt-generation approach to enable pre-trained ViT-backed CL robust against a domain shift.

Structurally, DAP exhibits certain similarities with hypernetworks-based CL [52]. However, while hypernetworks generate the parameters of the model, our adaptive prompt generator generates prompts. We explore the differences between them in Supp. H.

Cross-Domain Transfer. Learning with cross-domain is a more realistic scenario in which source and target domains are dissimilar. Domain generalization [53] and cross-domain few-shot learning [17, 37] are representative research fields that pre-train a model on source data and then adapt it to target data. In their scenarios, it is difficult to effectively transfer source information into the target domain because of the large domain gap between one another. In the CL tasks, there has been no such limitation to suffer from a cross-domain scenario. However, the prompt pool-based CL methods with the frozen ViT backbone yield a new challenge related to domain generalization in CL setups.

3. Prerequisites

3.1. Vision Transformer (ViT)

ViT generally consists of a single patch embedding layer, a stack of L transformer layers, and the classifier. Specifically, an input image x is split into d-dimensional patch tokens of size n through the patch embedding layer. Next, a trainable class token [CLS] is prepended to the patch tokens [PATCH] along the sequence length dimension, producing the initial input tokens $\mathbf{E_0} \in \mathbb{R}^{(n+1)\times d}$. Let $\mathbf{E}_l$ be the input tokens to the l-th transformer layer,

$$\mathbf{E}_l = [\text{CLS}; \text{PATCH}_1, \ldots, \text{PATCH}_n] \in \mathbb{R}^{(n+1)\times d}. \quad (1)$$

Then, it passes to the l-th transformer layer and produces the input $\mathbf{E}_{l+1}$ to the successor $(l+1)$-th layer as:

$$\begin{aligned} \mathbf{E}_{l+1} &= \texttt{LN}(\mathbf{E}_l^{'} + \mathbf{E}_l^{''}), \quad \text{where} \quad \mathbf{E}_l^{''} = \texttt{MLP}(\mathbf{E}_l^{'}) \\ &\text{and} \quad \mathbf{E}_l^{'} = \texttt{LN}\big(\texttt{MHSA}(\mathbf{E}_l, \mathbf{E}_l, \mathbf{E}_l)\big) \end{aligned} \quad (2)$$

and each transformer layer has a multi-head self-attention (MHSA) followed by a feed-forward network (MLP) with the skip connection (+) [18] and layer normalization (LN) [4]. As last, the classifier of a single feed-forward layer predicts class labels by mapping the class token [CLS] of the final transformer layer as:

$$\hat{y} = \text{Classifier}([\text{CLS}]_L), \quad (3)$$

where $\hat{y}$ is a predicted class probability distribution.

3.2. Continual Learning Protocol

Continual Learning (CL) solves a sequence of tasks with a single model. For this problem, a training benchmark $\mathcal{D} = \{\mathcal{D}_1, \cdots, \mathcal{D}_T\}$ denotes a sequence of tasks with size T, where each task $\mathcal{D}_t = \{(x, y)\}$ are sampled from a joint

data distribution of the input and label space $\mathcal{X}_t \times \mathcal{Y}_t$ at task t. Let the target model as $f : \mathcal{X} \rightarrow \mathcal{Y}$. As notations, we denote the patch embedding layer as f_p, the stack of transformer layers as f_b, and the classifier parameterized by θ as $f_{c,\theta}$ ($f = f_p \circ f_b \circ f_{c,\theta}$). Then, following DualPrompt [57], we consider the disjoint class-incremental learning setting, where task boundaries do not share classes, task identity is only given at training time, and utilize a ViT-Base model pre-trained on ImageNet as the frozen feature extractor; ViT-Base and ImageNet are the most commonly used backbone and source data [57, 58]. In our setup, the training data $\mathcal{D}$ can have varying levels of domain similarity to ImageNet, which is the source data used for pre-training.

3.3. Prompt-based Continual Learning

The modern prompt-based CL methods, L2P [58] and DualPrompt [57], leverage prompts to tune the frozen ImageNet-pretrained ViT-Base, where a single prompt P is defined as a sequence of d-dimensional tokens with sequence length p. Given a prompt pool $\mathcal{P} = \{P_1, P_2, \dots, P_M\}$ of pool size M, they select a fixed number of suitable prompts using different prompting functions from the pool and then prepend them to the input token $\mathbf{E}$ of Eq. (1) as:

$$\hat{\mathbf{E}} = [P_{s_1}; \cdots; , P_{s_o}; \mathbf{E}] \in \mathbb{R}^{\big((o \times p)+n+1\big) \times d}, \quad (4)$$

where s_i is the index of an i-th selected prompt from the pool, and o is the number of selections, which is commonly set to be $1-4$ in L2P and DualPrompt.

The selection of prompts from $\mathcal{P}$ is crucial for achieving excellent performance in CL. While L2P and DualPrompt employ similar prompting functions, DualPrompt further trains and utilizes the task-specific key. More specifically, DualPrompt maintains an auxiliary learnable token called the task-specific key k_t, which is encouraged to be closer to the [CLS] tokens of all training instances $\mathcal{D}_t$ belonging to the target task t using the pre-trained model. To optimize k_t, the matching loss [57] is formulated as:

$$\ell_{\text{matching}}(x, k_t) = S_C(f_b(f_p(x))[\text{CLS}], k_t), \quad x \in \mathcal{D}_t, \quad (5)$$

where S_C represents cosine similarity and $f_b(f_p(\cdot))[\text{CLS}]$ is the [CLS] token of the last transformer layer of the frozen pre-trained model (without prompts). They choose prompts for x based on the closest task-specific key k'_t by $\text{argmax}_{t \in \{1,\dots,T\}}$. With the selected prompts, $\mathbf{E}'$ of Eq. (2) is replaced as:

$$\mathbf{E}' = \texttt{LN}\big(\texttt{MHSA}(\mathbf{E}, \hat{\mathbf{E}}, \hat{\mathbf{E}})\big). \quad (6)$$

Thus, the information of prepended prompts is percolated to the patch tokens. The number of output tokens does not change since only the key and value inputs change in MHSA.

4. Domain-adaptive Prompt (DAP)

We propose Domain-Adaptive Prompt (DAP) to adapt the frozen pre-trained ViT for domain-adaptive CL. DAP injects a prompt generated by the adaptive prompt generator into all transformer layers and preserves the backbone frozen during training a sequence of tasks. Figure 2 overviews the proposed DAP framework.

4.1. Adaptive Prompt Generator

The motivation for proposing the adaptive prompt generator is the *limited scalability* of the prompt pool. In other words, limiting the number of available prompts to a predefined pool size forces each prompt to be optimized in a group-wise fashion because the pool size is much smaller than the number of total training instances. Especially, in realistic CL scenarios where there is a possibly large domain gap between source and target domains, prompts must encode fine-grained instructions to adjust the learned representations from the source data to the target domain more effectively. However, the prompt pool only allows encoding partial knowledge of the target domains constrained by the pool size. As a result, it makes essential to tune the pool-related hyperparameters in advance, which requires prior knowledge of the target domain. However, in any domain, expanding the pool size without limit is impractical when the memory budget is tight.

To pursue a domain-adaptive rehearsal-free CL framework, DAP generates prompts instantaneously using the adaptive prompt generator. The adaptive prompt generator (G) consists of input transpose ($\top$), LN, MLP, and a linear transformation layer (LT). It receives the input tokens $\mathbf{E} \in \mathbb{R}^{(n+1) \times d}$ and generates the adaptive prompt P^a, directly conditioning on the relation between the tokens as:

$$\begin{aligned} P^a &= \texttt{LT}\big(\texttt{MLP}\big(\texttt{LN}(\mathbf{E})^\top\big); \psi(e)\big)^\top \\ &= \big(\gamma_e \texttt{MLP}\big(\texttt{LN}(\mathbf{E})^\top\big) + \beta_e\big)^\top \in \mathbb{R}^{p \times d}, \end{aligned} \quad (7)$$

where ψ is a linear layer that predicts two types of affine transformation parameters [γ, β] and e is a conditional input embedding for ψ.

The primary objective of the adaptive prompt generator (Eq. (7)) is to create a prompt that contains instance-specific fine-grained instructions for each input. To encode the instance-level instructions required for a correct prediction, it is essential to examine the relationship across input patches rather than considering each patch individually [49]. For it, after normalization, we first transpose ($\top$) the input dimension from $(n+1) \times d$ to $d \times (n+1)$ before passing through MLP. Then, the generator's MLP encodes $(n+1)$-dimensional input into p-dimensional output, providing instance-level prompts considering the global information of each channel. Through the MLP layer, we generate a prompt that well holds instance-wise domain-related

knowledge to adapt to the target domain. To mitigate catastrophic forgetting, the MLP layer is frozen after several training epochs sufficient to adapt to the target domain.

Subsequently, we make use of the feature-wise transformation framework (LT) [40, 41, 43] to encode additional instructions that aid in prediction along with the created instance-level prompts. LT performs an affine transformation on the created instance-level prompts using two types of parameters: a scaling parameter $\gamma_e \in \mathbb{R}^d$ and another shifting parameter $\beta_e \in \mathbb{R}^d$ conditioned on e as $[\gamma_e, \beta_e] = \psi(e)$. For the conditional input embedding e, we utilize the task embedding obtained from k'_t of Eq. (5) to encode task-relevant instructions. Whereas DualPrompt uses k'_t to select a prompt in the prompt pool, we utilize it to embed supplementary instruction beneficial to prediction in conjunction with the created instance-level prompts. We empirically verify that DAP outperforms L2P and DualPrompt, as it is relatively robust to matching accuracy. The analysis is presented in Supp. E.

4.2. Optimization of DAP

As can be seen in Figure 2(a), DAP prepends the adaptive token P_l^a to the input tokens $\mathbf{E}_l$ in Eq. (1) of every transformer layer as:

$$\tilde{\mathbf{E}}_l = [P_l^a; \mathbf{E}_l] \in \mathbb{R}^{\left((1\times p)+n+1\right)\times d}, \quad 1 \le l \le L. \tag{8}$$

We train a separate generator G_l for each transformer layer. DAP uses a single adaptively generated prompt and, therefore, the number of prompts $o = 1$ unlike L2P and DualPrompt in Eq. (4). This simplicity is a strong advantage since a simple method often makes a significant impact and is widely accepted [21].

DAP is updated to minimize the following total loss:

$$\min_{f_{c,\theta}, G_{1:L,\phi}, r_t} \ell_{ce}(f_{c,\theta}(f_b(\tilde{\mathbf{E}}_{1:L})[\text{CLS}]), y) - \lambda \ell_{\text{matching}}(x, k_t), \tag{9}$$

where $(x, y) \in D_t$, ℓ_{ce} is the cross-entropy loss, and ℓ_{matching} is the matching loss described in Eq. (5). λ is a balancing term between the two losses (we use the fixed $\lambda = 0.1$ for simplicity). While the parameters of the ViT backbone are frozen, only the parameters ϕ of the adaptive prompt generator and θ of the classifier are updated.

5. Experiments

Evaluation Benchmarks. We conduct extensive experiments with seven datasets of the varying levels of domain similarity[1] to ImageNet [37, 45].

- **Natural Domain:** *Split CIFAR-100* is a typical benchmark widely used in the CL community. It is built by dividing the original CIFAR-100 [27] into 10 tasks with 10 disjoint classes each. Original Oxford-IIIT Pet data [38] consists of 37 pet categories with roughly 200 images per category. *Split Pets* is built by selecting 35 categories among 37 and splitting them into 7 tasks.
- **Aerial Domain:** EuroSAT [19] is a collection of satellite images of the landscapes. *Split EuroSAT* is built by splitting the original 10 classes into 5 tasks of 2 disjoint classes each. RESISC45 [10] contains 45 scene classes, each class having 700 images. *Split RESISC45* is built by splitting the 45 classes into 9 tasks of 5 disjoint classes each. The two aerial datasets are still color images of natural scenes but without perspective distortion.
- **Medical Domain:** CropDiseases [36] is a collection of diseased plant images, which contains natural images but is specialized in the medical and agriculture industries. ISIC2018 [12] and ChestX [54] are dermoscopy images of human skin lesions and X-Ray images on the human chest, which no longer represent natural images. CropDiseases, ISIC2018, and ChestX consist of 38, 7, and 8 categories, respectively. Considering task splitting and class imbalance, we chose 35, 6, and 6 categories and split them into 7, 3, and 2 tasks respectively for *Split CropDiseases*, *Split ISIC*, and *Split ChestX*.

The domain similarity with ImageNet decreases in the order of natural, aerial, and medical domains [37]. More analysis including data description is provided in Supp. A.

Methods. We compare DAP with two groups of *rehearsal-free* approaches, regularization-based methods including EWC [26] and LwF [30] and prompt-based methods including L2P [58] and DualPrompt [57]. For a fair comparison, all of the compared methods are started from the same ImageNet pre-trained ViT-Base[2] [15]. We also compare the settings favorable to the compared methods. We mark † in the method (e.g., L2P† in Table 1) when the method is favorably tuned for each target domain in advance by conducting the pool-related hyperparameter search. A detailed description of the compared methods and grid-search space of the favorable setting are provided in Supp. B and C. In addition, as a reference to readers, we provide the results of FT-seq, which is a default CL method only using fine-tuning in a sequential manner, and Upper-bound, which is a method to obtain the maximal performance in non-CL setup using fine-tuning. Refer to [57] for the comparison of prompt-based and rehearsal-based methods.

Evaluation Metrics. For evaluation, we repeat every experiment 3 times and report their average values with standard errors using three widely used CL metrics: average accuracy (Avg Acc ↑) [32] of the final average accuracy by the model, forgetting measure (Forgetting ↓) [9] of the ability to alleviate forgetting, and learning accuracy (Lrn Acc

[1] Domain similarity is inferred using Earth Mover's Distance [13, 37].

[2] We use the same pre-trained model by L2P [58] and DualPrompt [57].

Method	(a) Average Accuracy (↑)							Total Avg Acc (↑)
	Split CIFAR-100	Split Pets	Split EuroSAT	Split RESISC45	Split CropDiseases	Split ISIC	Split ChestX	
FT-seq	27.64±2.18	27.82±0.26	20.50±0.71	12.30±0.99	27.60±1.98	34.70±1.84	28.97±1.76	25.65
EWC [26]	59.60±1.27	59.40±0.14	47.40±1.13	61.75±1.48	73.30±5.09	44.35±1.77	30.65±5.02	53.78
LwF [30]	68.22±1.63	62.50±3.54	40.40±5.37	51.15±0.64	75.10±1.98	38.15±0.21	35.00±1.84	52.93
L2P [58]	83.07±1.12	78.34±0.92	42.58±0.90	59.37±2.09	53.26±0.25	42.27±3.98	32.51±0.90	55.91
L2P† [58]	83.07±1.12	83.25±1.04	69.17±8.62	69.28±0.21	59.73±1.41	51.60±6.11	39.48±4.41	65.08
DualPrompt [57]	85.93±0.82	86.85±0.76	76.48±1.75	73.35±1.14	80.84±0.58	53.81±5.19	45.28±4.61	71.79
DualPrompt† [57]	85.93±0.82	86.85±0.76	79.41±1.94	75.68±0.81	84.23±2.22	55.50±0.41	47.29±3.98	73.56
DAP	**94.05±1.19**	**91.02±0.44**	**98.18±0.56**	**92.84±1.68**	**97.88±0.89**	**84.18±2.54**	**52.50±3.57**	**87.24**
Upper-bound	94.20±0.80	91.83±1.92	99.03±0.55	96.54±1.56	99.56±0.36	84.65±3.18	52.98±2.23	88.40
Method	**(b) Forgetting Measure (↓)**							**Total Forgetting (↓)**
FT-seq	83.56±2.89	82.51±1.49	99.96±0.07	98.90±0.14	84.15±1.63	93.85±0.92	62.30±2.07	86.46
EWC [26]	24.65±0.07	8.85±0.35	2.30±1.56	20.10±2.12	9.30±3.25	5.50±0.71	4.30±4.67	10.71
LwF [30]	15.44±1.48	18.15±0.92	6.00±1.84	11.25±0.78	27.35±1.95	1.80±0.99	13.10±2.83	13.30
L2P [58]	7.45±0.14	15.01±1.10	47.03±1.10	31.98±4.83	25.75±5.76	43.33±3.08	25.20±7.46	27.96
L2P† [58]	7.45±0.14	10.45±1.10	12.47±6.05	11.76±0.18	12.78±2.83	15.25±1.22	12.57±4.24	11.82
DualPrompt [57]	5.60±0.62	8.38±0.74	12.78±1.23	12.03±1.03	7.85±2.88	22.15±2.60	4.10±0.28	10.41
DualPrompt† [57]	5.60±0.62	8.38±0.74	9.74±2.64	11.21±0.57	7.04±0.81	23.47±4.07	2.55±1.24	9.71
DAP	**2.28**±0.96	**1.21±0.45**	**0.61±0.53**	**6.24±1.89**	**1.71±1.02**	**0.72±0.42**	**1.97±1.54**	**2.11**
Method	**(c) Learning Accuracy (↑)**							**Total Lrn Acc (↑)**
FT-seq	98.30±2.17	98.06±0.33	99.96±0.06	99.44±0.06	99.42±0.17	95.99±0.56	59.20±1.15	92.91
EWC [26]	81.78±1.29	66.20±0.42	46.30±3.54	79.30±0.71	80.90±7.92	48.00±1.27	32.80±7.35	62.18
LwF [30]	82.05±0.07	76.00±1.27	45.20±3.82	61.05±0.07	89.55±2.05	39.30±0.85	41.60±0.42	62.11
L2P [58]	89.57±0.71	89.84±0.37	80.19±0.02	87.73±2.12	75.33±5.20	71.16±5.94	45.10±2.83	76.99
L2P†0 [58]	89.57±0.71	90.38±0.40	79.15±3.78	79.73±0.37	70.96±4.24	73.71±1.53	45.38±2.83	75.55
DualPrompt [57]	90.54±0.35	92.14±0.37	86.70±2.74	84.04±0.22	87.55±1.90	65.91±1.20	47.33±4.47	79.17
DualPrompt† [57]	90.54±0.35	92.14±0.37	87.12±0.04	86.18±1.05	90.18±1.58	71.15±3.13	48.81±3.36	80.87
DAP	**96.37±0.74**	**92.91±0.19**	**98.54±0.47**	**98.26±0.85**	**99.34±0.09**	**84.49±2.45**	**53.47±2.88**	**89.05**

Table 1. **Results on class-incremental learning on various data domains.** † indicates that the method is carefully tuned with pool-related hyperparameter searches on each data domain using their validation sets. Datasets are sorted with domain similarity to ImageNet in descending order from left to right.

↑) [44] of the ability to acquire new information well. Furthermore, to evaluate robustness against the domain gap between source and target data, we report the total average accuracy (Total Avg Acc ↑), which is the average of Avg Acc for all datasets with varying domain similarity. Refer to Supp. D for a detailed description of evaluation metrics.

We follow two evaluation setups in the literature [57, 58], namely batch- and instance-wise inference[3]. For the former, multiple testing instances with the same label constitute a batch, while for the latter, each testing instance is considered a batch. We report the results of the batch-wise setup as the main results following L2P and DualPrompt. The results of the instance-wise setup are presented in Supp. E.

Implementation. We train DAP using Adam [24] with β_1, β_2 of 0.9, learning rate of 0.01, and batch size of 128. We resize the input images to 224×224 resolution and normalize them between 0 and 1. Regarding hyperparameters, we set the prompt sequence length p to 10, the balancing term λ to 0.1, and, the task embedding size to 16 throughout all experiments. We use a single classifier head at inference. To ensure models converge, we train Split Pets, Split RESISC45, Split ISIC, and Split ChestX for 30 epochs per each task, and Split CIFAR-100, Split EuroSAT, and Split CropDiseases for 5 epochs per each task. `MLP` consists of a single linear layer. The hyperparameters of L2P and DualPrompt are set to the best ones for Split CIFAR-100 given in the original paper. The implementation details including the hyperparameter search are provided in Supp. C.

[3]https://github.com/google-research/l2p/tree/main/configs

Method	Total Avg Acc (↑)	Additional Parameters	
		Min.	Max.
L2P [58]	55.91	0.08M	0.16M
L2P† [58]	65.08	0.16M	0.39M
DualPrompt [57]	71.79	0.25M	0.33M
DualPrompt† [57]	73.56	0.33M	1.02M
DAP	87.24	0.36M	0.44M

Table 2. **Paramter efficiency comparison**. We compare the number of minimum and maximum additional parameters required for Table 1's results by the three prompt-based CL methods.

5.1. Main Results with Varying Domain Similarity

Average Accuracy. Table 1(a) compares the average accuracy of five different CL methods including DAP on seven datasets with varying domain similarity. As the domain similarity gets lower, the average accuracies of the two existing prompt-based methods, L2P and DualPrompt,

Ablated Components	Split CIFAR-100			Split EuroSAT			Split ISIC		
	Avg Acc (↑)	Forgetting (↓)	Lrn Acc (↑)	Avg Acc (↑)	Forgetting (↓)	Lrn Acc (↑)	Avg Acc (↑)	Forgetting (↓)	Lrn Acc (↑)
None	94.05±1.19	2.28±0.96	96.37±0.74	98.18±0.56	0.61±0.53	98.54±0.47	84.18±2.54	0.72±0.42	84.49±2.45
(A) w/o `LN` layer	94.01±1.58	2.46±1.25	95.96±0.74	96.39±1.35	1.26±0.42	97.00±1.35	79.98±1.02	1.30±0.87	80.02±0.46
(B) w/o input tranpose	78.47±1.66	8.50±2.01	85.53±0.80	46.64±1.85	37.65±2.55	79.76±1.02	60.28±3.02	15.30±1.97	71.33±2.16
(C) w/o `MLP` layer	87.30±1.27	6.48±1.70	92.50±0.64	79.95±2.97	7.24±1.19	85.48±1.93	46.30±4.45	22.21±3.33	64.33±2.91
(D) w/o `LT` layer	79.76±1.84	8.26±0.71	86.68±1.81	52.03±2.83	12.47±3.54	61.01±1.41	41.61±2.93	8.13±0.35	47.19±2.80

Table 3. **Ablation studies on the layers of the adaptive prompt generator (G).** The domain similarity with ImageNet decreases in order of Split CIFAR-100, Split EuroSAT, and Split ISIC. See text for detailed explanations.

Method	Split ImageNet-R			Split DomainNet		
	Avg Acc	Forgetting	Lrn Acc	Avg Acc	Forgetting	Lrn Acc
L2P	60.98±0.70	9.93±0.43	69.23±0.78	80.67±0.85	5.33±0.87	85.14±0.99
DualPrompt	68.97±2.87	4.66±2.15	72.85±2.27	81.89±0.63	5.21±1.17	87.27±1.80
DAP	**70.12±2.24**	**2.90±2.70**	**73.24±2.81**	**83.51±1.07**	**5.30±0.52**	**88.77±0.79**

Table 4. Results on benchmarks with a large number of classes.

drop much greater than that of the two regularization-based methods, EWC and LwF. That is, the methods of using a frozen backbone with the prompt pool are indeed vulnerable to a domain gap between source and target data. The pool-related hyperparameter tuning with prior domain knowledge improves their average accuracy by up to 26.6% (see L2P† and DualPrompt†), but still the performance is much worse than DAP without any domain-based tuning. Since the prompts by DAP can tune the frozen backbone to adapt even distant domains, its total average accuracy over all datasets is higher by 13.7–31.1% than the two prompt-based methods including their improved ones.

Forgetting Measure and Learning Accuracy. In Tables 1(b) and 1(c), we observe that DAP outperforms other methods not only on average accuracy but also on forgetting measure and learning accuracy. This is presumably because unlike the existing methods that select prompts from the optimized pool, DAP creates a suitable prompt on-the-fly based on the transposed input tokens, as in Eq. (7). Thus, DAP can achieve finer granularity in the prompt generation, enabling fine-grained encoding. As a result, except the FT-seq method, DAP attains the lowest forgetting measure and the highest learning accuracy in every dataset, explaining the high average accuracy of DAP in Table 1(a).

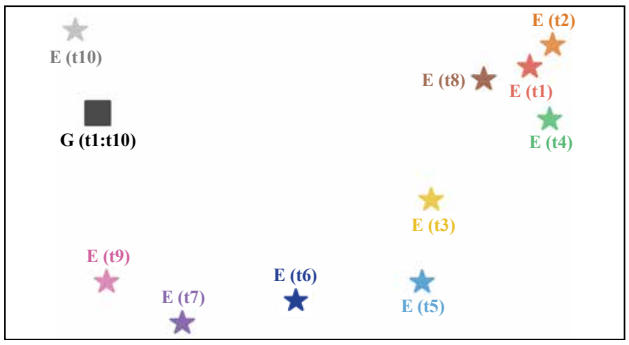
Parameter Efficiency. Table 2 summarizes the number of additional learnable parameters introduced by the three prompt-based methods including their total average accuracy. L2P and DualPrompt require a small amount of additional learnable parameters for prompts and classification heads (see the 1st and 3rd rows). However, the number of required parameters greatly increases by up to 3 times if the pool size increases by the pool-related hyperparameter tuning (see 2nd and 4th rows). Contrarily to them, DAP maintains a consistently small number of learnable parameters while achieving the best average accuracy. The change in the number of learnable parameters of DAP is attributed to an increase in the number of task embeddings and classification heads, which are independent of prompts.

ImageNet-R and DomainNet. Moreover, to demonstrate that DAP is not only resilient to domain shifts but also performs well on larger and longer benchmarks, we experiment with Split ImageNet-R [57] and Split DomainNet [39], and report the results in Table 4. Further analysis and results on a different number of sequences are presented in Supp. G.

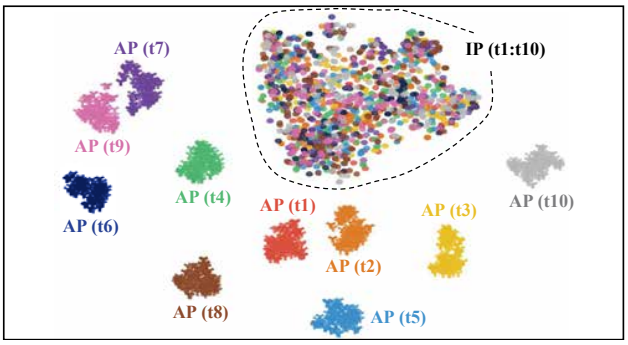
5.2. Component Ablation Study

The adaptive prompt generator consists of three layers and input transpose: (A) the normalization layer, `LN` in Eq. (7), for input tokens, (B) the normalized input tokens are transposed before passing through MLP layer, $\top$ in Eq. (7), (C) the MLP layer with the transposed input, `MLP` in Eq. (7), to generate an instance-specific prompt token, and (D) the linear transformation layer, `LT` in Eq. (7), to embed supplementary instructions for prediction. Table 3 removes each layer or input transpose from the canonical DAP, and report the results on Split CIFAR-100, Split EuroSAT, and Split ISIC, belonging to natural, aerial, and medical domains, respectively. For (B), a fixed random mapping is performed to match the dimension of the created adaptive prompt token with the dimension of other input tokens. For (C), instead of using the `MLP` layer, we initialize a fixed random prompt token and pass it through the `LT` layer.

First, (A) *the removal of the pre-activation* `LN` puts unnormalized patch tokens to the `MLP` and `LT` layers, so the overall performance shows a slight decline, however, the degree of degradation is quite significant when dealing with datasets with distant similarity to ImageNet. That is, the normalization and affine transformation of the `LN` layer seem to contribute to improving the adaptation ability to a specialized domain. Second, (B) *if the input tokens are not transposed*, the information used to train `MLP` is limited to each individual input token, rather than a combination of all tokens' each dimension. This means that `MLP` cannot encode instance-specific instructions that take into account the relationship between input tokens. Empirically, when input transpose is not used, all benchmarks suffer from a drastic performance drop. Third, (C) *the removal of* `MLP` disables the instance-level prompt generation. Therefore, the performance drop gets severe as the domain similarity decreases. On Split CIFAR-100 with high domain similarity, the degra-



(a) DualPrompt with G- and E-prompts.



(b) DAP with instance-wise (IP) and adaptive (AP) prompts.

Figure 3. **t-SNE plots on prompts**. For a single batch of each task (t1-t10) on Split CIFAR-100, we compare G-prompt (G) and E-prompts (E) selected by DualPrompt (left) vs. instance-wise (IP) and adaptive (AP) prompts generated by $\texttt{MLP}\big(\texttt{LN}(\mathbf{E})^\top\big)$ and $\texttt{LT}\big(\texttt{MLP}\big(\texttt{LN}(\mathbf{E})^\top\big)\big)$, respectively (right). Each point represents a prompt vector of d-dimension, and all the prompts are taken from the final model of each method.

dation is relatively small, while the degradation is significant on the rest two datasets with low similarity. These results empirically confirm that the MLP layer with the transposed input is a necessity for encoding instance-specific domain-related information. Particularly, when only LT is used without MLP, DAP is prone to vulnerable to catastrophic forgetting (see the Forgetting in Split EuroSAT and ISIC). Fourth, (D) *the removal of* LT disables the encoding of additional beneficial instructions for prediction based on input feature similarity. Thus, regardless of domain similarity with ImageNet, learning accuracy (Lrn Acc) is affected. As a result, the three layers and input transpose must be combined all together for pool-free prompt-based CL, which is effective for benchmarks with varying domains.

5.3. Hyperparameter Search

The sequence length p of a single prompt is a tunable hyperparameter that can largely affect DAP's performance. Regarding the sequence length, the p determines the capacity of the single prompt to encode knowledge for desired instructions. Figure 4 shows that DAP obtains consistently high average accuracy as long as the prompt length is greater than 10 for the three datasets with different domain similarities to ImageNet. Therefore, we set to fix $p = 10$ for all experiments.

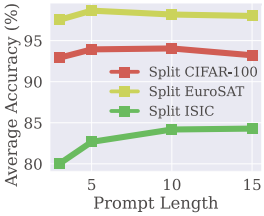


Figure 4. Ablation on p.

The performance of DAP is also influenced by the number of training epochs before freezing the MLP layer. However, we empirically find that the performance of DAP remains robust against variations in the number of training epochs. We explore the influence of the number of training epochs on MLP and report them in Supp. F. We also confirm that the performance remains consistent when the class order is randomly shuffled, which we report in Supp. F.

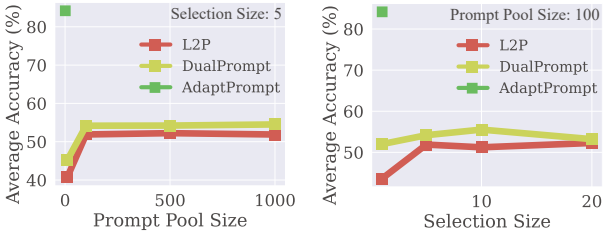


Figure 5. Scalability of L2P and DualPrompt on Split ISIC when expanding pool-related hyperparameters.

5.4. t-SNE Plots for Prompt Comparison

Because both DualPrompt and DAP utilize the task-specific key, we compare our generated prompts with the G- and E-prompts of DualPrompt using t-SNE [50] in Figure 3. Although the G- and E-prompts also learn task-invariant and task-relevant knowledge, respectively, they exhibit monotonous patterns with low diversity because they are chosen from the prompt pool of fixed size. DualPrompt encodes the group-wise instructions that are typically necessary for a given set of inputs belonging to a corresponding task. In contrast, using MLP conditioned on the transposed input, DAP first generates the necessary instruction capacity to more efficiently guide the pre-trained representation through instance-level prompting. Then, DAP incorporates task-relevant information that can improve plasticity along with the instance-level prompts exhibiting much higher diversity relying on each input instance.

6. Discussion: Scalability of Prompt Pool

One might argue that L2P and DualPrompt can be more robust against a large domain gap by simply increasing the prompt pool size and the number of selected prompts for a batch. This is partially true but not sufficient to obtain satisfactory performance. For a concrete answer, we test their

scalability with respect to the increased number of pool-related parameters, namely pool size and selection size. Figure 5 shows the scalability results using Split ISIC (medical) with a large domain gap from ImageNet (natural). There are some accuracy improvements before a certain size of pool and selection–i.e., 100 and 5 for each–but the accuracy remains almost the same regardless of further increases in their sizes. This is likely attributed to the key-value based query strategy of L2P and DualPrompt. With this strategy, the difficulty of finding proper prompts from the pool is prone to increase drastically as the pool and selection size increase. However, DAP takes patch tokens as input and generates the appropriate per-instance prompt, so even in Split ISIC, it achieve much higher performance only with a single prompt–i.e. the green square box.

7. Conclusion

Prompt-based CL, showing impressive performance without a replay buffer, has arisen in the field of CL. L2P and DualPrompt revise prompting to well adapt to the CL problem. However, this trend raises a different question, wondering about the scalability of the prompt-based CL methods on benchmarks with varying levels of domain similarity to ImageNet. To our knowledge, this is the first work that sorts out this curiosity and shows the limitations of the existing prompt-based CL methods on CL benchmarks with various domains. To overcome it, we propose DAP, a pool-free framework consisting of a prompt generator enabling pre-trained ViT-backed CL to be independent of domain reliability, while maintaining a significant performance.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A3B1077720) and the BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University in 2023. The authors would like to thank Prof. Sungroh Yoon at Seoul National University for his support and guidance.

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, pages 139–154, 2018. 1, 3

[2] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *CVPR*, pages 3366–3375, 2017. 1

[3] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *NeurIPS*, 32, 2019. 3

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 3

[5] Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *CVPR*, pages 8218–8227, 2021. 1, 3

[6] 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. *NeurIPS*, 33:1877–1901, 2020. 2

[7] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *NeurIPS*, volume 33, pages 15920–15930, 2020. 1, 3

[8] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*, pages 532–547, 2018. 1, 3

[9] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*, pages 532–547, 2018. 5

[10] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017. 5

[11] Nikhil Churamani, Ozgur Kara, and Hatice Gunes. Domain-incremental continual learning for mitigating bias in facial expression and action unit recognition. *IEEE Transactions on Affective Computing*, 2022. 3

[12] Noel Codella, Veronica Rotemberg, Philipp Tschandl, M Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*, 2019. 2, 5

[13] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *CVPR*, pages 4109–4118, 2018. 5

[14] Yi Dai, Hao Lang, Yinhe Zheng, Fei Huang, Luo Si, and Yongbin Li. Lifelong learning for question answering with hierarchical prompts. *arXiv preprint arXiv:2208.14602*, 2022. 1, 3

[15] 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 *ICLR*, 2021. 1, 5

[16] Nuwan Gunasekara, Heitor Gomes, Albert Bifet, and Bernhard Pfahringer. Adaptive online domain incremental continual learning. In *ICANN*, pages 491–502, 2022. 3

[17] Yunhui Guo, Noel C Codella, Leonid Karlinsky, James V Codella, John R Smith, Kate Saenko, Tajana Rosing, and

Rogerio Feris. A broader study of cross-domain few-shot learning. In *ECCV*, pages 124–141, 2020. 3

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3

[19] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. 2, 5

[20] Roei Herzig, Ofir Abramovich, Elad Ben-Avraham, Assaf Arbelle, Leonid Karlinsky, Ariel Shamir, Trevor Darrell, and Amir Globerson. Promptonomyvit: Multi-task prompt learning improves video transformers using synthetic scene data. *arXiv preprint arXiv:2212.04821*, 2022. 3

[21] IBM. "simple" threshold algorithm earns gödel prize. https://www.ibm.com/blogs/research/2014/05/simple-threshold-algorithm-earns-godel-prize/, 2014. Accessed: 2017-02-01. 5

[22] Dahuin Jung, Dongjin Lee, Sunwon Hong, Hyemi Jang, Ho Bae, and Sungroh Yoon. New insights for the stability-plasticity dilemma in online continual learning. In *ICLR*, 2023. 1

[23] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. *arXiv preprint arXiv:2210.03117*, 2022. 3

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[25] 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. 1

[26] 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. 3, 5, 6

[27] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[28] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 3

[29] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 3

[30] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017. 5, 6

[31] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021. 1, 2

[32] David Lopez-Paz, Robert Nishihara, Soumith Chintala, Bernhard Scholkopf, and Léon Bottou. Discovering causal signals in images. In *CVPR*, pages 6979–6987, 2017. 5

[33] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *CVPR*, pages 7765–7773, 2018. 1

[34] Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost van de Weijer. Class-incremental learning: survey and performance evaluation on image classification. *arXiv preprint arXiv:2010.15277*, 2020. 3

[35] James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3):419, 1995. 1

[36] Sharada P Mohanty, David P Hughes, and Marcel Salathé. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7:1419, 2016. 5

[37] Jaehoon Oh, Sungnyun Kim, Namgyu Ho, Jin-Hwa Kim, Hwanjun Song, and Se-Young Yun. Understanding cross-domain few-shot learning: An experimental study. *arXiv preprint arXiv:2202.01339*, 2022. 3, 5

[38] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, pages 3498–3505, 2012. 5

[39] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, pages 1406–1415, 2019. 7

[40] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *AAAI*, volume 32, 2018. 5

[41] Quang Pham, Chenghao Liu, Doyen Sahoo, and HOI Steven. Contextual transformation networks for online continual learning. In *ICLR*, 2020. 5

[42] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *ECCV*, pages 524–540, 2020. 1, 3

[43] James Requeima, Jonathan Gordon, John Bronskill, Sebastian Nowozin, and Richard E Turner. Fast and flexible multi-task classification using conditional neural adaptive processes. *NeurIPS*, 32, 2019. 5

[44] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *ICLR*, 2018. 6

[45] 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(3):211–252, 2015. 5

[46] Christian Simon, Masoud Faraki, Yi-Hsuan Tsai, Xiang Yu, Samuel Schulter, Yumin Suh, Mehrtash Harandi, and Manmohan Chandraker. On generalizing beyond domains in

cross-domain continual learning. In *CVPR*, pages 9265–9274, 2022. 3

[47] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *CVPR*, pages 11909–11919, 2023. 1

[48] Shikhar Srivastava, Mohammad Yaqub, Karthik Nandakumar, Zongyuan Ge, and Dwarikanath Mahapatra. Continual domain incremental learning for chest x-ray classification in low-resource clinical settings. In *Domain Adaptation and Representation Transfer, and Affordable Healthcare and AI for Resource Diverse Global Health*, pages 226–238. 2021. 3

[49] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *NeurIPS*, 34:24261–24272, 2021. 4

[50] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 8

[51] Andrés Villa, Juan León Alcázar, Motasem Alfarra, Kumail Alhamoud, Julio Hurtado, Fabian Caba Heilbron, Alvaro Soto, and Bernard Ghanem. Pivot: Prompting for video continual learning. *arXiv preprint arXiv:2212.04842*, 2022. 1, 3

[52] Johannes Von Oswald, Christian Henning, João Sacramento, and Benjamin F Grewe. Continual learning with hypernetworks. *arXiv preprint arXiv:1906.00695*, 2019. 3

[53] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 2022. 3

[54] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *CVPR*, pages 2097–2106, 2017. 5

[55] Yabin Wang, Zhiwu Huang, and Xiaopeng Hong. S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning. *arXiv preprint arXiv:2207.12819*, 2022. 1, 3

[56] Zhen Wang, Liu Liu, Yiqun Duan, Yajing Kong, and Dacheng Tao. Continual learning with lifelong vision transformer. In *CVPR*, pages 171–181, 2022. 1

[57] 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 arXiv:2204.04799*, 2022. 1, 2, 3, 4, 5, 6, 7

[58] 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*, pages 139–149, 2022. 1, 2, 3, 4, 5, 6

[59] Junting Zhang, Jie Zhang, Shalini Ghosh, Dawei Li, Serafettin Tasci, Larry Heck, Heming Zhang, and C-C Jay Kuo. Class-incremental learning via deep model consolidation. In *WACV*, pages 1131–1140, 2020. 1