International Journal of Computer Vision
https://doi.org/10.1007/s11263-024-02134-3

# Compositional Prompting for Anti-Forgetting in Domain Incremental Learning

**Zichen Liu**[1] · **Yuxin Peng**[1] · **Jiahuan Zhou**[1]

Received: 20 September 2023 / Accepted: 28 May 2024


**Abstract**
Domain Incremental Learning (DIL) focuses on handling complex domain shifts of a continuous data stream for visual tasks such as image classification and image segmentation. In real life, severe domain gaps in DIL are generated from various sources such as data style shifts, data quality degradation, environment changes, and so on. The well-known catastrophic forgetting issue in DIL becomes even more critical when simultaneously considering multiple sources of domain shifts. In this paper, we propose a unified and effective paradigm named Compositional Prompting (C-Prompt) to mitigate the critical forgetting challenge in DIL for image classification tasks. Unlike a popular type of conventional DIL approaches that need to retain abundant exemplars from the old domains, our exemplar-free C-Prompt leverages a prompt-guided Batch-wise Exponential Moving Average (BEMA) strategy to adaptively consolidate learned knowledge without retaining any exemplars. A set of prompts shared across different domains is designed to estimate the knowledge shifts for automatically balancing knowledge acquisition and forgetting. To enhance the learning ability, our proposed C-Prompt explores a domain-specific pool of learnable prompts for each domain, and all the prompt pools are further exploited in a cross-domain compositional manner to facilitate inference. Since the latest prompting-based DIL methods aim to learn one individual prompt for each domain, they always suffer from critical performance degradation caused by the incorrect prediction of domain index during inference and the limited learning capacity by using a single prompt per domain. Instead, our C-Prompt can not only readily acquire domain-specific knowledge but also exploit domain-shared knowledge. Extensive experiments on various large-scale multi-domain benchmarks have demonstrated the superiority of our proposed C-Prompt compared with state-of-the-art methods. Code is available at https://github.com/zhoujiahuan1991/IJCV2024-C-Prompt.

**Keywords** Domain incremental learning · Continual learning · Prompt learning · Continual domain adaptation

## 1 Introduction

Over the past years, the widely-adopted deep learning paradigm, pre-training plus finetuning, has remarkably advanced the progress in many computer vision tasks, such as image classification (Radford et al., 2021; Lu et al., 2019), object detection (Yang et al., 2021; Li et al., 2022) and semantic segmentation (Hao et al., 2020; Zhu et al., 2021). Although these deep networks can achieve promising performance in the pre-trained domain, their discriminant and generalization abilities are severely limited when dealing with data from different domains. Such a phenomenon may become even worse when learning multiple domains in a sequence. Researches in the field of Domain Adaptation focus on transferring models to new data domains but struggle to maintain performance across all domains simultaneously. On the other hand, the *Domain Incremental Learning (DIL)* (Tang et al., 2021) field aims to continuously learn from newly emerging data domains while retaining knowledge from old domains, enabling models to handle all data domains concurrently. Most recent DIL research refers to handling the gaps caused by data style shifting across domains (Simon et al., 2022; Tang et al., 2021; Lin et al., 2022) (e.g., art, cartoon, photo, sketch, and so on). Nonetheless, various realistic factors

Communicated by Gunhee Kim.

✉ Jiahuan Zhou
jiahuanzhou@pku.edu.cn

Zichen Liu
lzc20180720@stu.pku.edu.cn

Yuxin Peng
pengyuxin@pku.edu.cn

[1] Wangxuan Institute of Computer Technology, Peking University, Beijing 100871, China

Published online: 26 June 2024

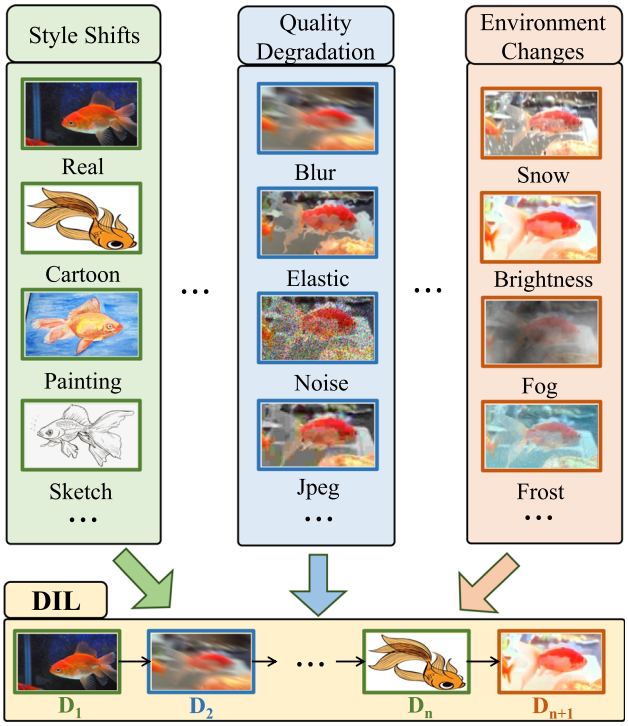


**Fig. 1** The domain gaps in DIL are usually caused by various sources including data style shifts, data quality degradation, realistic environment changes, and so on

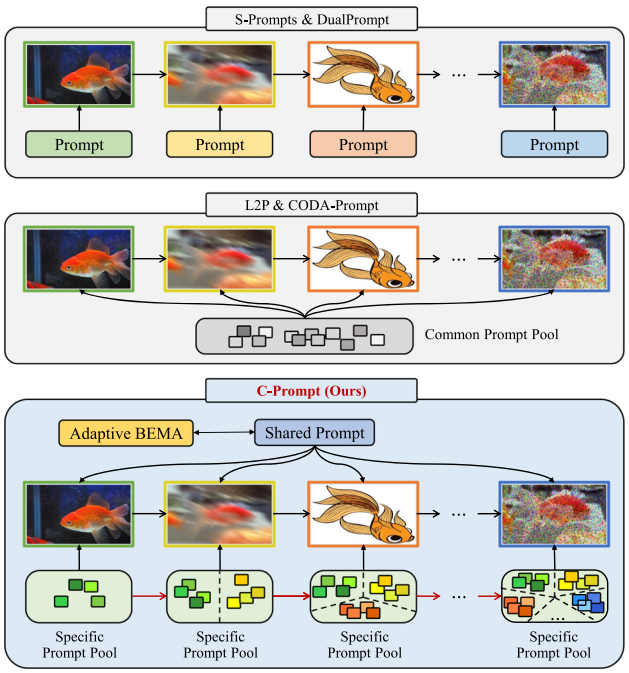


**Fig. 2** Compared with the latest prompting-based DIL methods, our C-Prompt can not only readily acquire domain-specific knowledge but also exploit domain-shared knowledge owing to the proposed adaptive BEMA-based compositional prompting model

including data quality degradation (e.g., blur, noise, and so on) and environment changes (e.g., snow, rain, haze, and so on) also inevitably result in severe domain gaps (Wang et al., 2022; Hendrycks & Dietterich, 2018) as shown in Fig. 1. The well-known catastrophic forgetting issue in DIL becomes even more critical when considering multiple sources of domain shifts simultaneously.

Generally, to mitigate the above catastrophic forgetting issue in DIL for image classification tasks, numerous methods (Tao et al., 2020; Buzzega et al., 2020; Cha et al., 2021) aim to retain a buffer of exemplars from old domains to perform either a rehearsal or a distillation when finetuning the whole model. However, storing exemplars not only introduces the cost of computation and storage but also readily violates privacy requirements in real-world applications such as medical diagnoses (Price & Cohen, 2019). Thus, various exemplar-free approaches (Garg et al., 2022; Simon et al., 2022; Wang et al., 2022) are proposed to constrain the model parameters or outputs during DIL to avoid extreme model drifting without any rehearsals. However, they always fail to achieve a proper trade-off between the knowledge acquisition of new domains and the forgetting of old domains in the long run.

Inspired by recent advances in visual prompt learning (Bahng et al., 2022; Zhou et al., 2022; Huang et al., 2023) which aims at adapting high-capacity deep models for downstream tasks effectively, several prompting-based IL methods Wang et al. 2022a, 2022b are proposed to leverage one individual prompt for each task to alleviate forgetting. However, when tackling DIL, their performance is severely limited by the incorrect selection of prompts during inference and the restricted capacity of using a single prompt per domain. Moreover, the above issues may even be aggravated in DIL due to the complicated sources of domain shifts.

Therefore, in this paper, we explore prompts in a novel domain-specific and compositional manner to facilitate DIL for image classification tasks. As shown in Fig. 2, instead of learning a single prompt for each domain Wang et al. 2022a, 2022b or a single prompt pool for all domains (Wang et al., 2022c; Smith et al., 2023), our proposed C-Prompt method maintains a series of domain-specific prompt pools consisting of multiple short-length prompts for each domain. For different learning samples, different compositions of prompts are automatically selected from the prompt pool to instruct learning, with the goal of better acquiring new knowledge from new domains. To further alleviate catastrophic forgetting in DIL, a globally shared prompt is learned using all samples across all domains as the surrogate of knowledge shifting across different learning stages. Our globally shared prompt can readily obtain a proper balance weight for the proposed Batch-wise Exponential Moving Average (BEMA) algorithm to adaptively mitigate forgetting without using any exemplars. The superiority of our method has been verified

by extensive experiments on various large-scale benchmarks against the state-of-the-art DIL approaches. Therefore, the main contributions of this work are three-fold:

- A novel domain-specific compositional prompt learning scheme is proposed that can well address the issues of incorrect domain prediction during inference and the limited learning capacity of the previous DIL methods.
- To tackle the catastrophic forgetting problem, a globally shared prompt is designed as the proxy of knowledge shifting in incremental learning.
- Extensive experiments on a more challenging DIL scenario that simultaneously considers the domain shifts caused by style changes, quality degradation, and environmental changes have verified the superiority of our proposed method.

## 2 Related Work

### 2.1 Incremental Learning

Based on different task scenarios, related works on incremental learning are mainly categorized into three types (Ven & Tolias, 2019): Task Incremental Learning (TIL) (Oren & Wolf, 2021; Kanakis et al., 2020), Class Incremental Learning (CIL) (Kirkpatrick et al., 2017; Li & Hoiem, 2017), and Domain Incremental Learning (DIL) (Volpi et al., 2021; Garg et al., 2022; Simon et al., 2022). TIL needs to acquire the task indexes in advance to determine a specific model for inference (Delange et al., 2021), which limits the effectiveness of the TIL methods in practical applications. CIL aims to incrementally learn new classes without knowing task information for inference. Unlike CIL, the goal of DIL is to continuously learn new knowledge from new data domains, while retaining knowledge from old data domains without knowing task information for inference.

In this paper, we focus on the challenging and practical DIL scenario. Several DIL methods rely on memory replay to overcome catastrophic forgetting by storing exemplars from old domains (Hayes et al., 2019; Buzzega et al., 2020; Cha et al., 2021). Recently, different rehearsal-free DIL approaches are studied by either regularizing important learning parameters (Garg et al., 2022; Simon et al., 2022; Wang et al., 2022; Tang et al., 2021; Fini et al., 2022; Wang et al., 2023) or dynamically modifying model architectures for new domains (Rusu et al., 2016; Kundu et al., 2020; Wang et al., 2020). Moreover, various CIL methods have been migrated and adapted for tackling DIL (Li & Hoiem, 2017; Pellegrini et al., 2020; Kirkpatrick et al., 2017; Hou et al., 2019; Rebuffi et al., 2017; Xie et al., 2022) but suffer from limited performances without specifically handling domain gaps. In summary, almost all the above DIL works focus on the domain shifts caused by a single source, but their performance in handling multi-source domain shifts has not been well investigated. Therefore, we propose a novel compositional prompting-guided DIL method that has demonstrated superior performance in simultaneously tackling a more challenging DIL scenario where the domain shifts are caused by multiple sources.

### 2.2 Domain Adaptation

A closely related research area to DIL is Domain Adaptation (DA) which aims to improve the performance of a model when there is a domain shifting between the source domain and target domain (Pan et al., 2010; Patel et al., 2015). DA techniques, including adversarial learning, image-to-image translation, cross-domain divergence minimization, and optimal transport, have been extensively explored (Wang & Breckon, 2020; Li et al., 2020; Lian et al., 2019). Self-training has emerged as a prominent trend in DA, leveraging labeled source data and pseudo-labeled target data to iteratively train a student model (Hoyer et al., 2022; Zou et al., 2018; Lian et al., 2019). In recent years, some efforts (Liang et al., 2020; Yang et al., 2021; Chen et al., 2020; Kundu et al., 2020; Agarwal et al., 2022) have gradually shifted towards addressing the more challenging and practical problem of Source-Free Domain Adaptation (SFDA), which involves adjusting models to adapt to new data domains without using source domain data. SHOT (Liang et al., 2020) utilizes a centroid-based label refinement for the self-training of model. G-SFDA (Yang et al., 2021) follow a similar strategy with further measures for refining pseudo-labels by encouraging consistent predictions between local neighbor samples. However, the aforementioned methods tend to focus only on the performance gain in the target domain regardless of the performance degradation in the source domain. Moreover, there is usually only one target domain considered in DA. In contrast, DIL focuses on alleviating catastrophic forgetting of old domains while continuously learning multiple new target domains in a sequence.

### 2.3 Prompt Learning

As a surging trend in natural language processing (NLP), manually designed prompts are leveraged to prepend instructions to the input text (Schick & Schütze, 2020; Shin et al., 2020). In some recent works such as Prompt Tune (Lester et al., 2021) and Prefix Tune (Li & Liang, 2021), the prompts are treated as learnable parameters which can be added to the pre-trained model. Consequently, the large-scale pre-trained models are no longer fine-tuned since only tuning the learnable prompts instead is enough for a more efficient adaptation to downstream tasks. In the field of incremental learning, few works are proposed to investigate prompting. Three latest methods, L2P (Wang et al., 2022c), DualPrompt (Wang et

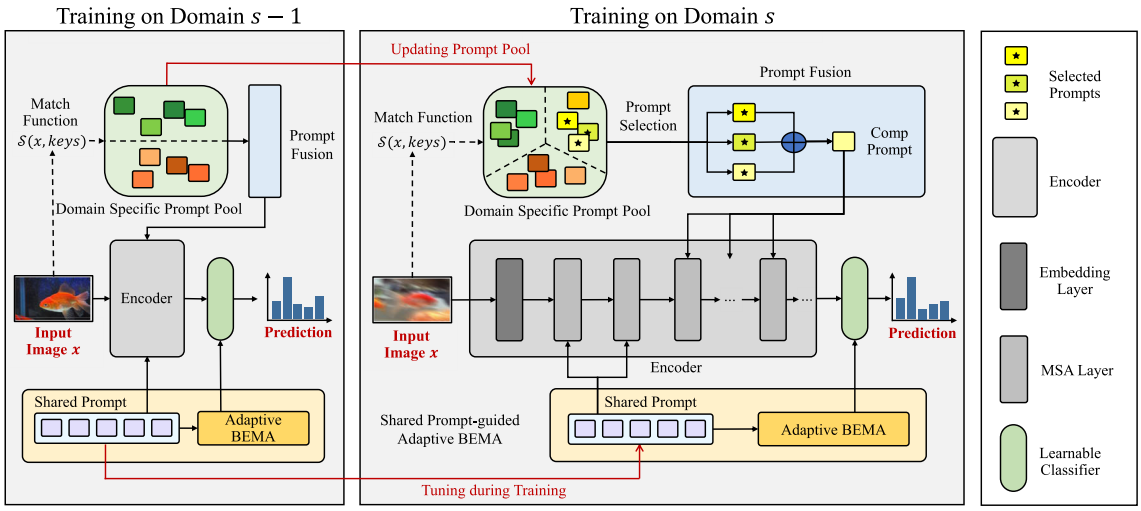


**Fig. 3** The overall pipeline of our proposed compositional prompting (C-Prompt) method. We keep the pre-trained ViT encoder fixed to retain the knowledge of the pre-trained model. During the training phase, a separate prompt pool is trained for each image domain to enhance the model's ability to acquire new knowledge. Additionally, we design the globally shared prompt to obtain a proper balance weight for the proposed Batch-wise Exponential Moving Average (BEMA) algorithm, to adaptively mitigate forgetting of the classifier

al., 2022a) and CODA-Prompt (Smith et al., 2023), aim at exploring prompt learning in the scenario of CIL rather than DIL. S-Prompts (Wang et al., 2022b) proposed to learn a specific prompt for each individual domain in DIL. In comparison, our proposed C-Prompt leverages a specific pool of learnable prompts for each individual domain to enhance knowledge acquisition meanwhile adopts a prompt-guided Batch-wise Exponential Moving Average (BEMA) strategy to adaptively consolidate learned knowledge without retaining any exemplars. In addition, Wang et al. (2022b) has to train and retain a separate classifier for each domain, which is not scalable if a large number of domains are trained sequentially. In contrast, our proposed C-Prompt method only needs to maintain a single classifier.

# 3 The Proposed Method

## 3.1 Problem Settings and Notations

In this paper, we focus on the challenging Domain Incremental Learning (DIL) scenario, where exists severe modal variations caused by various sources of domain shifts. The sequence of domains in DIL is denoted as $\mathcal{D} = \{\mathcal{D}_1, \cdots, \mathcal{D}_S\}$ where the $s$-th domain $\mathcal{D}_s = \{(\boldsymbol{x}_i^s, y_i^s)\}_{i=1}^{N_s}$ contains tuples of the input sample $\boldsymbol{x}_i^s \in \mathcal{X}$ and its corresponding label $y_i^s \in \mathcal{Y}$. The goal of DIL is to train a single model $f : \mathcal{X} \rightarrow \mathcal{Y}$ to predict the label $y = f(\boldsymbol{x}) \in \mathcal{Y}$ of the test sample $\boldsymbol{x}$ from arbitrary domains. When training on the $s$-th domain $\mathcal{D}_s$, the training data of previous $s$-1 domains $\{\mathcal{D}_i\}_{i=1}^{s-1}$ are not available.

In addition, the widely adopted assumption that the domain boundaries are clear and the domain changes abruptly during training (Wang et al. 2022a, 2022b, 2022c) is also followed here. Moreover, a pre-trained model, e.g., a vision transformer (ViT) (Dosovitskiy et al., 2020) on ImageNet, is used as the backbone model and kept frozen through the entire learning procedure as Wang et al. (2022a, 2022b, 2022c). However, unlike the rehearsal-based methods (Boschini et al., 2022; Rebuffi et al., 2017), we do not use any form of rehearsal buffers in our method.

## 3.2 Preliminary of Prompt Learning

As the latest learning paradigm, prompt learning initially emerged in the field of NLP (Schick & Schütze, 2020; Shin et al., 2020) but has rapidly spread to the computer vision area recently. A recent visual prompting technique, known as Prompt Tuning (PT) (Lester et al., 2021), proposes to freeze the model while learning the prompt parameters attached before the input token of a ViT to perform the downstream task.

Given an input sample $\boldsymbol{x} \in \mathbb{R}^{H \times W \times C}$ and a pre-trained ViT $f_{enc} = f_e \circ f_a$ where $f_e$ and $f_a$ are the pre-trained input embedding layer and self-attention layers separately. The embedding layer $f_e$ firstly projects $\boldsymbol{x}$ into a sequence-like output feature $\boldsymbol{h}_x$:

$$\boldsymbol{h}_x = f_e(\boldsymbol{x}), \tag{1}$$

where $\boldsymbol{h}_x \in \mathbb{R}^{L \times D}$ and $L$ is the token sequence length and $D$ is the embedding dimension. When solving downstream

tasks, the pre-trained backbone $f_{enc}$ is frozen, and the prompt parameters $\boldsymbol{p} \in \mathbb{R}^{L_p \times D}$ with token sequence length $L_p$ are learned. Then, the prompt parameters $\boldsymbol{p}$ is concatenated with $\boldsymbol{h}_x$ to form the final input embedding $\boldsymbol{h}_x^* = [\boldsymbol{p}; \boldsymbol{h}_x]$ into the self-attention layers $f_a$, the specific process is as follows:

$$\begin{aligned} f_a\left(\boldsymbol{h}_x^*\right) &= f_a\left(\left[\boldsymbol{p}; \boldsymbol{h}_x\right]\right) \\ &= \mathrm{MSA}\left(\boldsymbol{h}_Q, \left[\boldsymbol{p}_K; \boldsymbol{h}_K\right], \left[\boldsymbol{p}_V; \boldsymbol{h}_V\right]\right), \end{aligned} \tag{2}$$

where $\mathrm{MSA}(\cdot, \cdot, \cdot)$ is Multi-head self-attention layers, $\boldsymbol{p}_K$ and $\boldsymbol{p}_V$ are split from $\boldsymbol{p}$. Finally, the prediction $y$ of input image $\boldsymbol{x}$ is obtained by a trainable classifier $f_c$ as $y = f_c\left(f_a\left(\boldsymbol{h}_x^*\right)\right)$.

## 3.3 Compositional Prompting for Acquisition Enhancement

To tackle the challenging DIL task, we propose a novel compositional prompting method, named **C-Prompt** to address the dilemma of knowledge acquisition and forgetting as shown in Fig. 3. Our idea is motivated by the latest prompting-based methods in IL Wang et al. 2022a, 2022b which aim to learn a single separate prompt for each task but suffer from two main drawbacks in the DIL scenario. On the one hand, during inference, it's required to predict the domain information of the input sample so as to select the corresponding domain-specific prompt for inference. However, it is indeed difficult to predict an accurate domain index which will inevitably hinder model performance. On the other hand, using one single prompt is not powerful enough to capture the enriched intra-domain variations and handle the critical inter-domain gaps. Moreover, since the domain-specific prompt is fixed once the domain is trained, there is no way to explore the intrinsic discriminative information sharing among domains to further benefit learning which leads to sub-optimal learning performance.

### 3.3.1 Training on the *s*-th Domain

As shown in Algorithm 1 and Fig. 4, to address these drawbacks, our proposed C-Prompt maintains a domain-specific prompt pool $\mathcal{P}^s$ for the $s$-th domain during training. $\mathcal{P}^s$ consists of a number of $n$ short-length prompts as shown in Eq. (3):

$$\mathcal{P}^s = \{\left(\boldsymbol{p}_1^s, \boldsymbol{k}_1^s\right), \left(\boldsymbol{p}_2^s, \boldsymbol{k}_2^s\right), \cdots, \left(\boldsymbol{p}_n^s, \boldsymbol{k}_n^s\right)\}, \tag{3}$$

where $\left(\boldsymbol{p}_i^s, \boldsymbol{k}_i^s\right)$ is the tuple of the $i$-th prompt $\boldsymbol{p}_i^s \in \mathbb{R}^{L_p \times D}$ and its corresponding learnable key $\boldsymbol{k}_i^s \in \mathbb{R}^D$ for matching. $L_p$ and $D$ represent the token length and embedding dimension. For the training of the $s$-th domain, the prompts in $\mathcal{P}^s$ are automatically matched by each input sample $\boldsymbol{x}$ by utilizing a query mechanism to reduce the semantic gap between

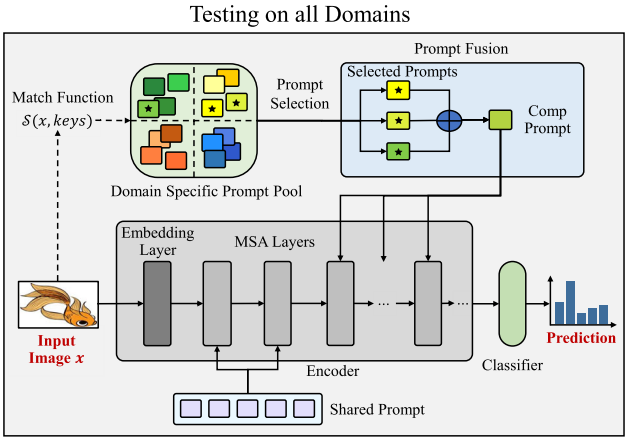


**Fig. 4** Illustration of the testing process of our method on all domains. For each image, the most relevant prompts are automatically selected from all domain prompt pools which are further combined as the Comp Prompt. The image itself, ComP Prompt, and Shared Prompt are jointly fed into the model for prediction

**Algorithm 1** Training on the $s$-th domain

**Input:** Data $\mathcal{D}_s = \{(\boldsymbol{x}_i^s, y_i^s)\}_{i=1}^{N_s}$, prompt pool $\mathcal{P}^s$, shared prompt $\boldsymbol{p_g}$, parameters of classifier and average classifier $\boldsymbol{\theta}, \boldsymbol{\theta^*}$, embedding layer $f_e$, self-attention layers $f_a$.
**Output:** Tuned prompt pool $\mathcal{P}^s$, shared prompt $\boldsymbol{p_g}$, parameters of classifier and average classifier $\boldsymbol{\theta}, \boldsymbol{\theta^*}$.

**while** $\mathcal{D}_s$ is not $\varnothing$ **do**
  Sample a batch $\mathcal{B} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{b}$ from $\mathcal{D}_s$
  $\mathcal{D}_s \Leftarrow \mathcal{D}_s \setminus \mathcal{B}$
  **for** $(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{B}$ **do**
    Get $\{\boldsymbol{p}_j\}_{j=1}^{k}$ based on Equation 6
    $\overline{\boldsymbol{p}}_{\boldsymbol{x}} \Leftarrow \sum_{j=1}^{k} \frac{\boldsymbol{p}_j}{k}$
    $\boldsymbol{h_x} \Leftarrow f_e(\boldsymbol{x})$
    $\boldsymbol{h_x^*} \Leftarrow \left[\boldsymbol{p_g}; \overline{\boldsymbol{p}}_{\boldsymbol{x}}; \boldsymbol{h_x}\right]$
    $P_{\boldsymbol{x}} \Leftarrow f_c\left(f_a\left(\boldsymbol{h_x^*}\right)\right)$
  **end for**
  Calculate loss based on Equation 8
  Optimize $\boldsymbol{\theta}$ , $\mathcal{P}^s$ and $\boldsymbol{p_g}$
  Calculate $\beta$ based on Equation 12
  $\boldsymbol{\theta^*} \Leftarrow \beta \cdot \boldsymbol{\theta^*} + (1 - \beta) \cdot \boldsymbol{\theta}$
**end while**

**Algorithm 2** Testing on all domains

**Input:** Test images $\mathbf{X}$ , $S$ domain-specific prompt pools $\mathcal{P} = \{\mathcal{P}^s\}_{s=1}^{S}$, shared prompt $\boldsymbol{p_g}$, average classifier $f_c$, embedding layer $f_e$ and self-attention layers $f_a$.
**Output:** Predictions of all test samples $\mathbf{P}$.

$\mathbf{P} \Leftarrow \varnothing$
**for** $\boldsymbol{x} \in \mathbf{X}$ **do**
  Get $\{\boldsymbol{p}_j\}_{j=1}^{k}$ from $\mathcal{P}$ based on Equation 6
  $\overline{\boldsymbol{p}}_{\boldsymbol{x}} \Leftarrow \sum_{j=1}^{k} \frac{\boldsymbol{p}_j}{k}$
  $\boldsymbol{h_x} \Leftarrow f_e(\boldsymbol{x})$
  $\boldsymbol{h_x^*} \Leftarrow \left[\boldsymbol{p_g}; \overline{\boldsymbol{p}}_{\boldsymbol{x}}; \boldsymbol{h_x}\right]$
  $\mathbf{P_x} \Leftarrow f_c\left(f_a\left(\boldsymbol{h_x^*}\right)\right)$
  $\mathbf{P} \Leftarrow \mathbf{P} \cup \mathbf{P_x}$
**end for**

$\boldsymbol{x}$ and $\boldsymbol{p}_i^s$. To do so, the whole frozen pre-trained model $f_{enc}$ is used as the query function $q(\cdot)$ to encode $\boldsymbol{x}$ into the same dimension $D$ as the keys:

$$q = f_{enc} : \mathbb{R}^{H\times W\times C} \rightarrow \mathbb{R}^D. \tag{4}$$

Then the matching score between $q(\boldsymbol{x})$ and the prompt keys $\boldsymbol{k}_i^s$ can be measured via a matching function $\mathcal{S}(\cdot,\cdot)$:

$$\begin{aligned}\mathcal{S}\left(q(\boldsymbol{x}), \boldsymbol{k}_i^s\right) &= 1 - \cos\left(q(\boldsymbol{x}), \boldsymbol{k}_i^s\right) \\ &= 1 - \frac{q(\boldsymbol{x})\cdot \boldsymbol{k}_i^s}{\|q(\boldsymbol{x})\|_2 \cdot \left\|\boldsymbol{k}_i^s\right\|_2}.\end{aligned} \tag{5}$$

Specifically, the query vectors $q(\boldsymbol{x}) \in \mathbb{R}^D$ obtained from the encoder are not the CLS token, but rather the first image token.

Thus, the top-$k$ matched prompts in $\mathcal{P}^s$ for the input $\boldsymbol{x}$ can be obtained via solving Eq. (6):

$$\mathcal{K}_{\boldsymbol{x}}^s = \underset{\{n_i\}_{i=1}^k \subseteq \mathbb{N}_n}{\arg\max} \sum_{i=1}^{k} \mathcal{S}\left(q(\boldsymbol{x}), \boldsymbol{k}_{n_i}^s\right). \tag{6}$$

where $\mathcal{K}_{\boldsymbol{x}}^s$ is a subset of top-$k$ keys selected for $\boldsymbol{x}$, and $\{n_i\}_{i=1}^k$ are $k$ indices from $\mathbb{N}_n$ denoting top-$k$ matched prompts. Here, $\mathbb{N}_n$ represents the set of integers from 1 to $n$. Accordingly, $\mathcal{P}_{\boldsymbol{x}}^s$ is a subset of matched prompts from $\mathcal{P}^s$. Once $\mathcal{P}_{\boldsymbol{x}}^s$ is determined, a compositional prompt $\overline{\boldsymbol{p}_{\boldsymbol{x}}^s}$ for $\boldsymbol{x}$ can be obtained via a simple yet effective linear combination function $g_c : \mathbb{R}^{L_p\times D\times k} \rightarrow \mathbb{R}^{L_p\times D}$:

$$\overline{\boldsymbol{p}_{\boldsymbol{x}}^s} = g_c\left(\boldsymbol{p}_{n_1}^s, \boldsymbol{p}_{n_2}^s, \cdots, \boldsymbol{p}_{n_k}^s\right) = \sum_{i=1}^{k} \frac{\boldsymbol{p}_{n_i}^s}{k}. \tag{7}$$

The compositional prompt $\overline{\boldsymbol{p}_{\boldsymbol{x}}^s}$ is further concatenated with the embedding result $\boldsymbol{h}_{\boldsymbol{x}}$ to form the final input $\boldsymbol{h}_{\boldsymbol{x}}^* = \left[\overline{\boldsymbol{p}_{\boldsymbol{x}}^s}; \boldsymbol{h}_{\boldsymbol{x}}\right]$ of the self-attention layers.

#### 3.3.2 Optimization Objective

During the training of $s$-th domain, the compositional prompt-added embedding $\boldsymbol{h}_{\boldsymbol{x}}^*$ is fed into the frozen self-attention backbone $f_a$ and a learnable classifier $f_c$. Moreover, the globally shared prompt $\boldsymbol{p}_{\boldsymbol{g}}$ is added to the MSA layers of $f_{enc}$. Finally, the prompt pool $\mathcal{P}^s$, the globally shared prompt $\boldsymbol{p}_{\boldsymbol{g}}$ and learnable classifier $f_c$ are jointly trained in an end-to-end fashion:

$$\min_{\mathcal{P}^s, \boldsymbol{p}_{\boldsymbol{g}}, f_c} \mathcal{L}\left(f_c\left(f_a\left(\boldsymbol{h}_{\boldsymbol{x}}^*\right)\right), y\right) + \lambda \sum \mathcal{S}\left(q(\boldsymbol{x}), \boldsymbol{k}_{n_i}^s\right), \tag{8}$$

where $\mathcal{L}$ is the softmax cross-entropy loss and $\lambda$ is a weighting parameter.

#### 3.3.3 Testing on All Domains

As shown in Algorithm 2, once all the domains $\mathcal{D}$ are incrementally learned, we will obtain $S$ domain-specific prompt pools $\mathcal{P} = \{\mathcal{P}^s\}_{s=1}^S$. Instead of tackling the difficult domain index prediction task as in Wang et al. 2022a, 2022b, we directly leverage all the $N = n \cdot S$ prompts in $\mathcal{P}$ as the common prompt pool for all the testing samples. The same matching strategy in Eq. (6) is adopted but prompts from different domains can be readily matched and further fused based on Eq. (7). Therefore, our prompt composition operation not only avoids adding too many prompts into inference to increase the computational cost but also encourages inter-domain knowledge transfer and aggregation in test time.

### 3.4 Globally Shared Prompting for Anti-Forgetting

In our proposed C-Prompt method, besides leveraging $\mathcal{P}$ to enhance knowledge acquisition, we further design a globally shared prompt $\boldsymbol{p}_{\boldsymbol{g}} \in \mathbb{R}^{L_c\times D}$ added to the first two multi-head self-attention (MSA) layers. Different from $\mathcal{P}$, the $\boldsymbol{p}_{\boldsymbol{g}}$ is designed to be shared by all the samples across all domains, and it is a direct and intuitive idea that the variations in the globally shared prompt could reflect the model's acquisition of new knowledge and forgetting of old knowledge. To empirically validate this notion, we conducted experiments where the model sequentially learned 15 different domains. We computed the Euclidean distance between the globally shared prompt of the $s$-th domain $\mathcal{D}_s$ and the $(s-1)$-th domain $\mathcal{D}_{s-1}$ to quantify the prompt's variation. Additionally, we compared the model's performance on the $\mathcal{D}_{s-1}$ before and after learning the $\mathcal{D}_{s-1}$, using the difference as a measure of knowledge forgetting. By plotting the scatter plot of the prompt variation and model knowledge forgetting (Fig. 5), we visually observed a strong correlation between the shared prompt's parameter changes and the model's knowledge forgetting. Hence, by calculating the shared prompt, we can estimate the degree of the model's forgetting of old knowledge, which enables a better balance between acquiring new knowledge and forgetting old knowledge when using EMA for classifier parameter fusion.

As shown in Fig. 3, based on the globally shared prompt $\boldsymbol{p}_{\boldsymbol{g}}$, we propose a novel batch-wise adaptive exponential moving average (BEMA) strategy to alleviate the catastrophic forgetting in DIL. The overall pipeline of BEMA is illustrated in Fig. 6. Denote $\{\boldsymbol{\theta}_i\}_{i=1}^B$ as the classifier parameters of $B$ consecutive batches in training and $\boldsymbol{\theta}_i^*$ as the historical average classifier parameters at the $t$-th batch where $\boldsymbol{\theta}_1^* = \boldsymbol{\theta}_1$. When $t \geqslant 2$, we calculate $\boldsymbol{\theta}_t^*$ by

$$\boldsymbol{\theta}_t^* = \beta_t \cdot \boldsymbol{\theta}_{t-1}^* + (1-\beta_t)\cdot \boldsymbol{\theta}_t, \tag{9}$$

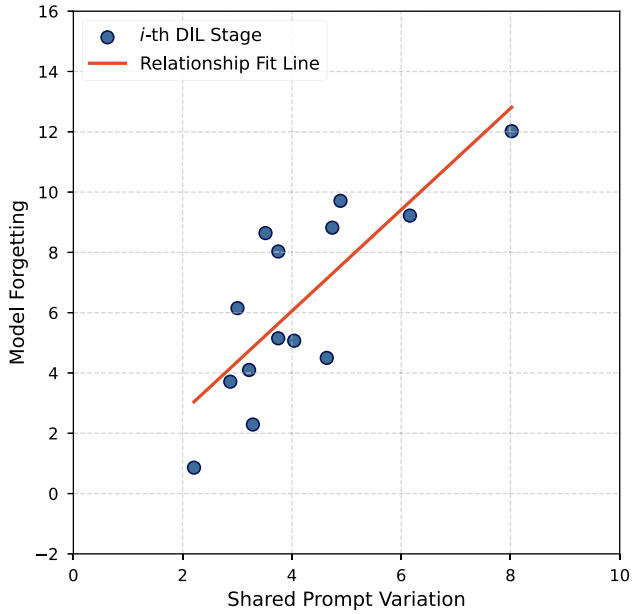


**Fig. 5** The scatter plot illustrates the relationship between the variation of the global prompt parameters and the model's knowledge forgetting in DIL. The experiments are conducted on the ImageNet-R dataset, sequentially training the model on 15 different domains

where $\beta_t$ is the adaptive weight of the $t$-th batch. Intuitively, the larger $\beta_t$ is, the smaller the weight of the newly learned classifier parameters is, the stronger the ability to alleviate forgetting and the worse the learning ability of new knowledge is, and vice versa.

To obtain an appropriate $\beta_t$ in Eq. (9) for different batches, we take the parameter updates of $\boldsymbol{p}_g$ from all training batches into consideration. Denote $\{\boldsymbol{p}_g^i\}_{i=1}^B$ as the historical shared prompts learned from $B$ consecutive batches, then the parameter changes between $\boldsymbol{p}_g^{t-1}$ and $\boldsymbol{p}_g^t$ can be measured as:

$$\mathcal{G}_t = \phi\left(\boldsymbol{p}_g^{t-1}, \boldsymbol{p}_g^t\right), \tag{10}$$

where $\phi(\cdot,\cdot)$ can be chosen as an Euclidean distance function for convenience. Thus, $\mathcal{G}_t^*$ can be calculated as the average of $\{\mathcal{G}_i\}_{i=1}^t$:

$$\mathcal{G}_t^* = \frac{1}{t-1}\sum_{i=2}^{t}\mathcal{G}_t, t \geqslant 2. \tag{11}$$

Then the adaptive weight $\beta_t$ can be obtained as:

$$\beta_t = 1 - \exp\left(\frac{\mathcal{G}_t}{\mathcal{G}_t^*} - \eta\right), \tag{12}$$

where $\eta$ is a hyperparameter to regularize the value of $\beta_t$.

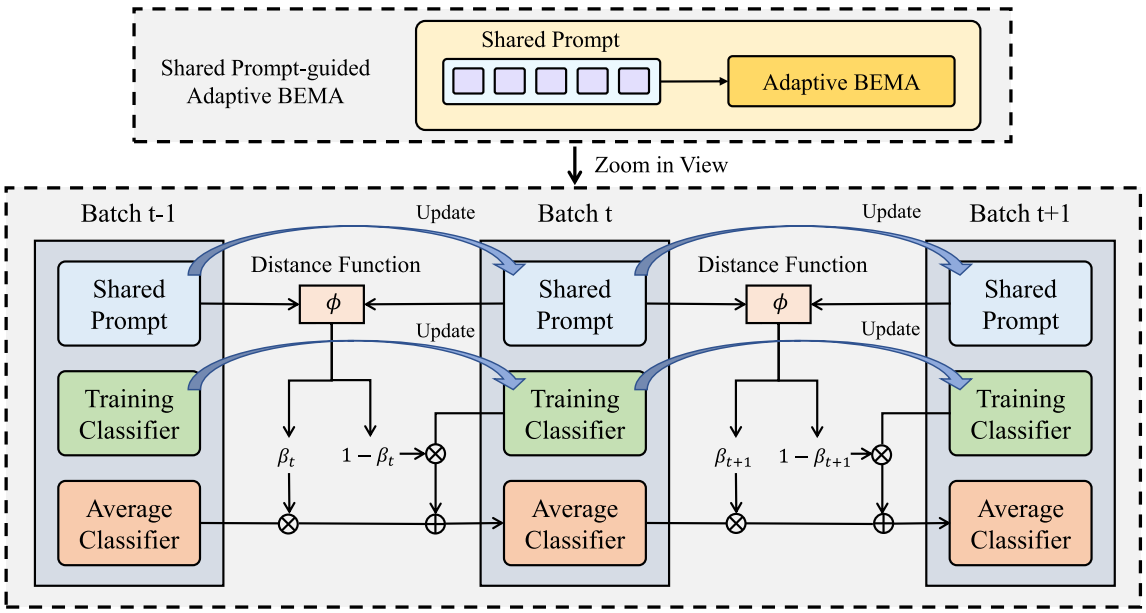


**Fig. 6** The pipeline of our proposed BEMA algorithm. During the training phase, The model's forgetting of old knowledge is measured by calculating the change in the shared prompt between two batches. Then the fusion weight $\beta_t$ between the Training Classifier and Average Classifier is adaptively generated to achieve a balance between knowledge acquisition and forgetting. In the testing phase, we use the Average Classifier to make predictions for the test samples

### 3.5 Comparison with Latest Prompting-Based DIL Methods

To further demonstrate the contribution of our method, we conducted a systematic comparison and analysis between C-Prompt and the latest prompting-based DIL methods. Our method differs from other prompting-based DIL methods mainly in four aspects: prompt design, training strategy of prompts, usage of prompts during testing, and the role of prompts in mitigating classifier forgetting. Therefore, we compare and analyze prompting-based DIL methods from these four aspects.

#### 3.5.1 S-Prompts & DualPrompt

*Method* Firstly, these methods design a separate set of prompts $p_t$ for each domain $D_t$. Secondly, during training on the $t$-th domain, only the corresponding prompt $p_t$ is trained. Thirdly, during testing, a set of prompts needs to be selected from $t$ prompts since the domain of the test sample is unknown. Finally, These methods do not have a design for mitigating classifier forgetting.

*Analysis* Firstly, there are differences between the data of each domain, and the limited learning ability of the model is due to learning only one set of prompts per domain. Secondly, as the accuracy of prompt selection decreases with the increase in the number of domains, the performance of the model decreases when there are many domains. Thirdly, since there is no design for mitigating classifier forgetting, the classifier experiences catastrophic forgetting, leading to a decrease in model performance.

#### 3.5.2 L2P & CODA-Prompt

*Method* Firstly, these methods design a shared prompt pool for all domains. Secondly, during training on each domain, prompts are freely selected from the entire prompt pool. Thirdly, during testing, prompts are selected from the entire pool. Finally, these methods also do not have a design for mitigating classifier forgetting.

*Analysis* Firstly, since the entire prompt pool is continuously updated during training, prompts learned in previous domains will be forgotten during later domain learning. Secondly, as these methods lack a design for mitigating classifier forgetting, the classifier also experiences catastrophic forgetting.

#### 3.5.3 Our C-Prompt

*Method* Firstly, we design a compositional prompt pool, where each domain has a separate prompt pool. Secondly, during training on the $t$-th domain, only the prompt pool of the $t$-th domain is updated. Thirdly, during testing, prompt matching and combination are performed uniformly across all domain prompt pools. Finally we also design a globally shared prompt for all domains, and its parameter changes reflect the degree of classifier forgetting, guiding the classifier to adaptively BEMA.

*Analysis* Firstly, our design of separate prompt pools for each domain ensures that the learned prompts are not updated after learning the corresponding domain to preserve old domain knowledge. Secondly, the design of freely selecting prompts from multiple domain prompt pools greatly enhances the diversity of our learned prompts, and during prediction, the model can adaptively select knowledge from different domains to assist in prediction. Thirdly, our BEMA can adaptively fuse based on the model's forgetting degree to better balance the learning of new knowledge and the retention of old knowledge.

## 4 Experiments

### 4.1 Experimental Settings

#### 4.1.1 Benchmarks and Protocols

All the experiments are conducted on four multi-domain benchmarks including DomainNet (Peng et al., 2019), ImageNet-R (Hendrycks et al., 2021), ImageNet-C (Hendrycks & Dietterich, 2018), and ImageNet-Mix.

- *DomainNet* (Peng et al., 2019). It is a large-scale dataset for domain adaptation and DIL which contains 345 classes and 586,575 images in total. These images are collected from 6 different style domains including Real, Quickdraw, Sketch, Painting, Infograph, and Clipart.
- *ImageNet-R* (Hendrycks et al., 2021). It is another widely-used multi-domain benchmark that contains a total of 30,000 images of 200 categories taken from ImageNet (Deng et al., 2009). All the images are split into 15 different style domains (e.g. art, cartoons, deviantart, graffiti, and so on). In our experiments, we divide the images in each domain of ImageNet-R into training and testing sets under a 7:3 ratio and train on all 15 different domains sequentially.
- *ImageNet-C* (Hendrycks & Dietterich, 2018). Unlike the above two datasets mainly focusing on image style variations, ImageNet-C is generated based on ImageNet (Deng et al., 2009) by collecting images from 1,000 categories of ImageNet which cover 15 diverse quality corruptions and environment changes covering noise, blur, weather changes, and so on. In experiments, we utilize 200 categories of ImageNet-C which are the same as ImageNet-R, and treat different corruption or environment types as different domains for DIL. The total 10,000 images of each

domain are further split into 7,000 for training and 3,000 for testing. For all three datasets, the evaluated methods are trained in all the domains in turn and then tested on all of them without knowing the domain information.

- *ImageNet-Mix*. To further simulate the critical multi-source domain shifting scenario in DIL, a mixed dataset ImageNet-Mix is built upon ImageNet-C and ImageNet-R which contains images from 200 common classes shared by both datasets. Thus, ImageNet-Mix contains a total of 30 domains which simultaneously involve different image styles, qualities, and environmental variations. When experimenting on ImageNet-Mix, we interchangeably train and evaluate the domains of ImageNet-R and ImageNet-C.

#### 4.1.2 Comparison Methods

For DIL, the **Upper-bound** performance that a method can reach is regarded as adopting supervised finetuning on all the data of all domains. Besides, **FT-Seq** is the sequential finetuning baseline aiming to train on all domains sequentially. EWC (Kirkpatrick et al., 2017) and LwF (Li & Hoiem, 2017) are two representative incremental learning baselines that are widely compared. In experiments, we accordingly transfer them to fit the DIL setting. Furthermore, the state-of-the-art regularization-based method ESN (Wang et al., 2023) and prompting-based DIL methods are compared including L2P (Wang et al., 2022c), DualPrompt (Wang et al., 2022a), S-Prompts (Wang et al., 2022b) and CODA-Prompt (Smith et al., 2023). For a fair comparison, we adopt the same pre-trained ViT (ViT-B/16) (Dosovitskiy et al., 2020) as the backbone for all five approaches and our proposed C-Prompt.

#### 4.1.3 Evaluation Metrics

Following previous works Wang et al. 2022b, 2022a,2022c, the Average Accuracy (Ave-ACC) and Average Forgetting are reported as the main evaluation metric for DIL. Specifically, after training on the $s$-th domain, the Average Accuracy $A_s$ is calculated as:

$$A_s = \frac{1}{s}\sum_{\tau=1}^{s} a_{s,\tau}, \tag{13}$$

where $a_{s,\tau}$ represents classification accuracy on $\tau$-th domain after training on $s$-th domain. The Average Forgetting $F_s$ is calculated as:

$$F_s = \frac{1}{s-1}\sum_{\tau=1}^{s-1} b_{s,\tau}, \tag{14}$$

$$b_{s,\tau} = \max_{i\in 1,2,\cdots,s-1} \{a_{i,\tau}\} - a_{s,\tau}. \tag{15}$$

#### 4.1.4 Implementation Details

In our experiments, we sorted domains by decreasing image counts, aligning with the setting in CaSSLe (Fini et al., 2022), to simulate a challenging DIL scenario for all methods which are consistent and fair.

To train C-Prompt, we leverage the Adam optimizer (Kingma & Ba, 2014) with default parameter settings, a batch size of 128, and a constant learning rate of 0.005 for all benchmarks. The training images are resized to $224\times224$ and normalized to the range of $[0, 1]$ to match the pre-trained model. Training too many epochs for a domain will result in catastrophic forgetting (Buzzega et al., 2020), hence we train every domain for 5 epochs. The hyperparameters $N$, $L_p$ and $k$ are the same as DualPrompt (Wang et al., 2022a) in our C-prompt, and we add one extra $\eta$ which is not sensitive as we have discussed in Sect. 5.4.

### 4.2 Comparison with State-of-the-Art

The overall results of the comparison methods on DomainNet, ImageNet-R, ImageNet-C, and ImageNet-Mix are reported in Table 1. As demonstrated, no matter whether DIL is conducted under style changes or quality variations, our C-Prompt can consistently outperform all the state-of-the-art (SOTA) DIL methods by a large margin on all four benchmarks. Compared with the second-best player, CODA-Prompt, the overall average performance improvement is 3.61% owing to the compositional prompt pool and BEMA utilization. Additionally, our C-Prompt achieve the best performance on the Average Forgetting metric, further confirming that our split prompt pool and BEMA design can better mitigate forgetting of old knowledge. Although another SOTA method, S-Prompts, proposes to retain a separate classifier for each domain, our C-Prompt still can beat S-Prompts by 1.31% on DomainNet by just using a single classifier for all domains. Moreover, S-Prompts performs badly in ImageNet-R, ImageNet-C, and ImageNet-Mix with a gap of 10–17% compared with our method because S-Prompts has to predict the domain index for each testing sample. As the number of domains increases, the accuracy of predicting the domain to which the sample belongs largely decreases, resulting in severe performance degradation in S-Prompts.

In addition, the per-domain performance details of DomainNet are reported in Table 2. Taking a closer look at the results, the performance of C-Prompt is close to the SOTA methods in the latter domains but significantly better than the SOTA methods in the former domains. This is mainly due to the design of compositional prompting and prompt-guided BEMA algorithm, which greatly improves the ability of our model to mitigate catastrophic forgetting. Noted that although FT-Seq achieves the best result in the last domain (Clipart), it suffers from severe performance drops on

**Table 1** The overall comparison results on four benchmarks

| Methods | Ave-ACC (↑) / Forgetting (↓) | | | | All |
|---|---|---|---|---|---|
| | DomainNet | ImageNet-R | ImageNet-C | ImageNet-Mix | |
| Upper-bound | 65.37/– | 70.78/– | 94.07/– | 79.08/– | 77.33/– |
| FT-Seq | 48.49/19.65 | *58.80*/8.41 | 67.47/24.73 | 54.17/20.94 | 57.23/18.43 |
| EWC (Kirkpatrick et al., 2017) | 42.77/6.52 | 37.93/7.34 | 31.79/10.84 | 28.46/8.92 | 35.24/8.41 |
| LwF (Li & Hoiem, 2017) | 45.01/7.23 | 54.85/6.39 | 22.51/15.23 | 40.04/6.83 | 40.60/8.92 |
| L2P (Wang et al., 2022c) | 48.27/5.25 | 54.66/6.03 | 68.12/5.82 | 56.35/4.92 | 56.85/5.51 |
| S-Prompts (Wang et al., 2022b) | *57.37*/2.31 | 45.05/1.89 | 69.22/2.94 | 56.49/2.41 | 57.03/2.39 |
| DualPrompt (Wang et al., 2022a) | 49.36/*1.44* | 57.66/*1.02* | 78.38/*1.46* | *63.09*/*1.39* | 62.12/*1.33* |
| ESN (Wang et al., 2023) | 40.62/5.23 | 46.27/5.28 | 64.18/6.34 | 53.25/6.29 | 51.08/5.79 |
| CODA-Prompt (Smith et al., 2023) | 54.40/1.83 | 58.41/1.34 | *78.82*/2.28 | 62.60/2.04 | *63.56*/1.87 |
| C-Prompt (Ours) | **58.68/1.34** | **62.55/0.75** | **81.16/1.25** | **66.30/0.92** | **67.17/1.07** |

Bold values indicate the highest result. Italic values indicate the second-highest result

**Table 2** The per-domain comparison results on DomainNet

| Methods | Real | Quickdraw | Painting | Sketch | Infograph | Clipart | Ave-ACC |
|---|---|---|---|---|---|---|---|
| Upper-bound | 79.09 | 68.03 | 66.26 | 65.68 | 37.21 | 75.95 | 65.37 |
| FT-Seq | 61.81 | 30.81 | 46.01 | 51.39 | 24.37 | 76.56 | 48.49 |
| EWC (Kirkpatrick et al., 2017) | 54.77 | 17.88 | 39.41 | 47.76 | 20.79 | 75.98 | 42.77 |
| LwF (Li & Hoiem, 2017) | 63.28 | 16.72 | 47.12 | 52.57 | 21.90 | 68.45 | 45.01 |
| L2P (Wang et al., 2022c) | 72.38 | 13.39 | 52.57 | 49.95 | 25.47 | 75.88 | 48.27 |
| S-Prompts (Wang et al., 2022b) | 80.29 | 41.19 | **65.87** | 55.37 | **37.43** | 64.08 | 57.37 |
| DualPrompt (Wang et al., 2022a) | 73.01 | 13.96 | 53.85 | 51.97 | 27.13 | 76.29 | 49.36 |
| ESN (Wang et al., 2023) | 66.40 | 10.12 | 47.93 | 48.87 | 22.23 | 78.14 | 40.62 |
| CODA-Prompt (Smith et al., 2023) | 74.58 | 23.36 | 56.91 | **59.21** | 31.61 | **80.71** | 54.40 |
| C-Prompt (Ours) | **83.91** | **45.46** | 65.23 | 58.12 | 30.98 | 68.36 | **58.68** |

Bold values indicate the highest result

previous domains due to the forgetting issue. A similar consequence can also be observed in Fig. 7 where the per-domain performance of ImageNet-R, ImageNet-C, and ImageNet-Mix is presented. Various visualization results of the test cases are shown in Fig. 8 to further demonstrate the superiority of our C-Prompt method. For the hard cases in which both the CODA-Prompt and DualPrompt predict incorrect classes, our C-Prompt can accurately handle them.

### 4.3 Comparison with Examplar-Based DIL Methods

To further validate the effectiveness of our method, we conducted experiments comparing it with exemplar-based DIL methods. Following the experimental settings of existing methods L2P and S-Prompts, we conducted experiments on the CORe50 (Lomonaco & Maltoni, 2017) dataset and compared with exemplar-based methods ER (Chaudhry et al., 2019), GDumb (Prabhu et al., 2020), BiC (Wu et al., 2019), DER++ (Buzzega et al., 2020), Co$^2$L (Cha et al., 2021) and DyTox (Douillard et al., 2022).

The experimental results are shown in Table 3. Without retaining exemplars, our method outperforms the second-best method by 2.18%. This is attributed to the design of our prompt pool and BEMA, allowing C-Prompt to retain knowledge from old domains while learning new knowledge. Although our method is not designed for scenarios that retain exemplars, when exemplars are retained, our method still achieves best performance, surpassing the second-best method by 1.79%. This is because retaining exemplars can slow down the model's forgetting of old domain knowledge, and our C-Prompt can further retain knowledge from old domains on this basis, resulting in higher accuracy.

### 4.4 Comparison with Parameter-Efficient Fine-Tuning (PEFT) Methods

Recently, some parameter-efficient fine-tuning (PEFT) methods were proposed to train a small number of parameters, allowing pre-trained models to adapt to downstream tasks efficiently. Considering that our C-Prompt is also a parameter-efficient method, we compare C-Prompt with

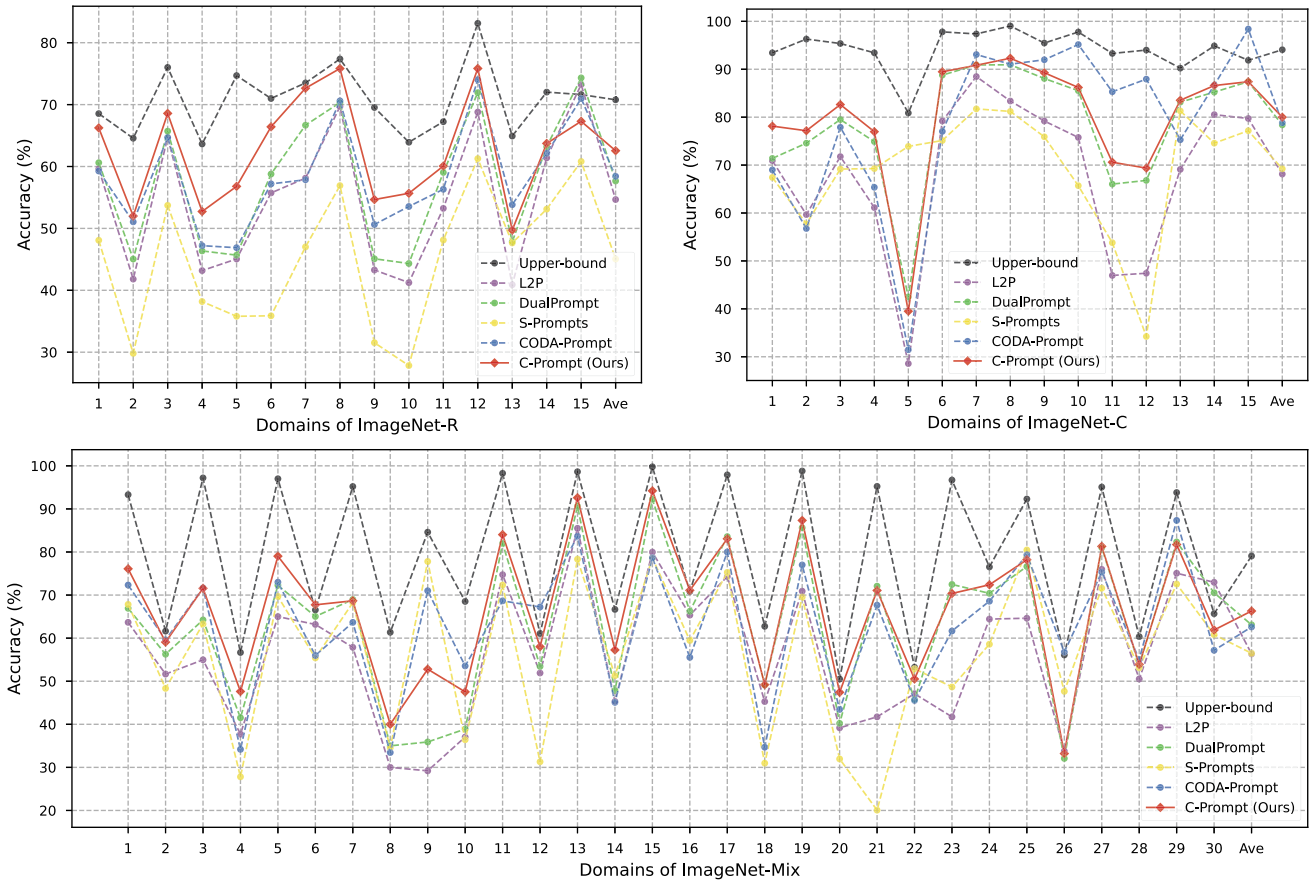


**Fig. 7** The per-domain comparison results on ImageNet-R, ImageNet-C and ImageNet-Mix

them in the DIL task. As shown in Table 4, while these methods have lower computational costs, they suffer from catastrophic forgetting and performance degradation when learning multiple different domains in a continuous manner. decreased by 8.2%, but improved by 5–29% in the former five domains, demonstrating the effectiveness of BEMA in retaining learned knowledge. Fortunately, the proposed compositional prompt can alleviate knowledge forgetting in the last domain to some extent.

## 5 Ablation Study and Analysis

### 5.1 The Influence of Different Components

To verify the effectiveness of different components in our proposed C-Prompt, ablation experiments are conducted on DomainNet and reported in Table 5. *ComP* represents the compositional prompting in Eq. (7) and *BEMA* denotes the proposed shared prompt-guided adaptive exponential moving average algorithm in Eq. (9). As demonstrated by the results, when neither component is used, the C-Prompt is degraded to a frozen pre-trained ViT model with a learnable classifier. Since the proposed compositional prompting and BEMA are complementary to each other, removing either one will result in severe performance degradation, but still outperform the ViT baseline. Note that when adding BEMA to the model, the performance in the last domain

### 5.2 Ablation on Compositional Prompting

#### 5.2.1 Why ComP Component is Needed?

As shown in Table 6, we conduct experiments where the backbone is fixed and only the classifier is learnable. We explore using KNN as the classifier, Fine-tuning the classifier, and using EWC and LwF for the classifier without the Comp component. However, the experimental results without Comp are inferior to those with Comp. This is because when the backbone is fixed to prevent forgetting, the learning capacity of the classifier is limited. In contrast, using the Comp module significantly enhances the model's learning capacity, leading to improved performance.

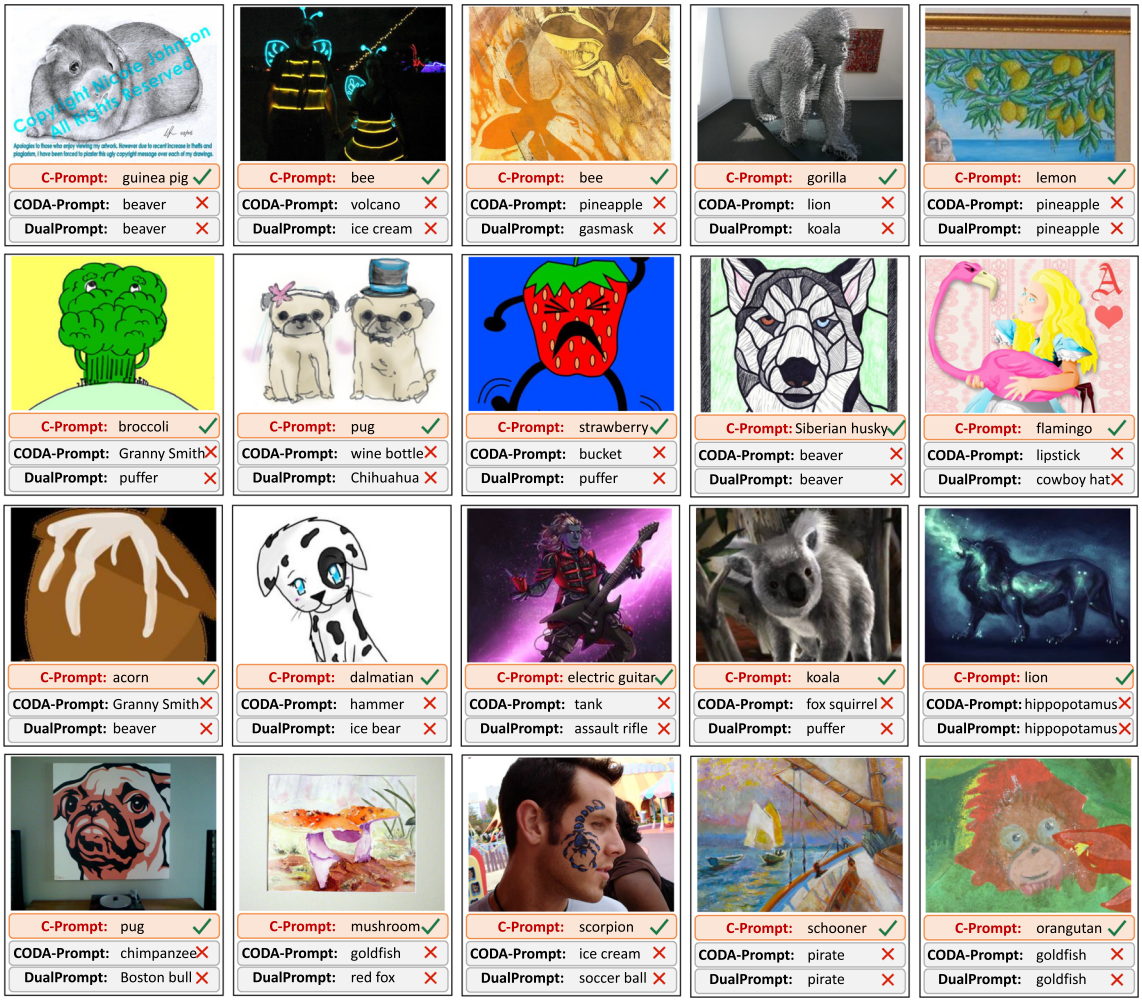


**Fig. 8** The visualization results of partial test samples in ImageNet-R. The predictions of CODA-Prompt, DualPrompt, and our proposed C-Prompt are presented for these samples

#### 5.2.2 The Influence of Different Forms of Prompting

We further explore the influence of different forms to utilize prompting. In Table 7, *Single* represents using only one single prompt pool for all domains and directly concatenating the selected prompts one by one without composing them, *Comp* represents fusing the selected prompts by Eq. (7), and *Pools* represents maintaining a specific prompt pool for each domain. Thus, *Pools* + *Comp* denotes our entire C-Prompt method. The results present that either fusing the selected prompts (55.74%) or learning multiple domain-specific prompt pools (57.53%) can all improve the performance since the former can readily utilize information from different prompts to complement each other and the latter is helpful to mitigate forgetting of the learned domains. Our whole C-Prompt can achieve the best performance (58.68%) by integrating all components.

#### 5.2.3 Selected Frequency of Prompts in the Prompt Pool

During experiments, we found that the prompts were not selected at the same frequency during training and testing. Figure 9 shows the selected frequency of each prompt in each domain-specific prompt pool on the DomainNet dataset during the final testing. Several prompts are frequently selected since they can capture the commonly shared information among most data samples. While, for some samples which exhibit specific information different from the others, a part of prompts can be learned to well handle them. As a result, these prompts are not frequently selected.

Additionally, we use T-SNE to visualize prompts. As shown in Fig. 10, our domain-specific prompts are generally diverse to learn the domain difference, while they also capture the domain commonality as the center cluster in the figure.

**Table 3** Comparison with exemplar-based DIL methods on the CORe50 dataset

| Methods | Buffer size | Ave-ACC |
|---|---|---|
| ER (Chaudhry et al., 2019) | 50/class | 80.10 |
| GDu mb (Prabhu et al., 2020) | | 74.92 |
| BiC (Wu et al., 2019) | | 79.28 |
| DER++ (Buzzega et al., 2020) | | 79.70 |
| Co$^2$L (Cha et al., 2021) | | 79.75 |
| DyTox (Douillard et al., 2022) | | 79.21 |
| L2P (Wang et al., 2022c) | | 81.07 |
| C-Prompt (ours) | | **82.86** |
| EWC (Kirkpatrick et al., 2017) | 0/class | 74.82 |
| LwF (Li & Hoiem, 2017) | | 75.45 |
| L2P (Wang et al., 2020) | | 78.33 |
| S-Prompts (Wang et al., 2022b) | | 83.13 |
| C-Prompt (ours) | | **85.31** |

Bold values indicate the highest result

**Table 4** Compare with other parameter-efficient fine-tuning (PEFT) methods on the ImageNet-R dataset

| Methods | Ave-ACC |
|---|---|
| BitFit (Zaken et al., 2021) | 53.08 |
| LoRA (Hu et al., 2021) | 57.96 |
| C-Prompt (ours) | **62.55** |

Bold values indicate the highest result

#### 5.2.4 The Influence of Different Designs in BEMA

Firstly, we study the influence of different update granularity for EMA. The granularity from coarse to fine is task-wise, epoch-wise, and batch-wise. The more fine-grained the granularity is, the more likely the model can retain more historical information during incremental learning, and thus the weaker the forgetting of old knowledge is. As shown in Table 9, our proposed batch-wise EMA strategy significantly outperforms the other two designs. This is because a batch-wise update is more suitable for handling severe data variations in the DIL scenario. Different choices of $\beta_t$ for adaptive knowledge consolidation are compared. Using the adaptive strategy defined in Eq. (12) performs better than the fixed $\beta_t$ and other widely-used adaptive ways.

**Table 6** Comparison of performance with and without ComP component on ImageNet-R

| Backbone | Classifier | Ave-ACC |
|---|---|---|
| Fixed ViT | KNN | 50.96 |
| | FT-C | 49.78 |
| | EWC-C | 51.09 |
| | LwF-C | 50.69 |
| ComP + Fixed ViT | FT-C | 56.58 |
| | BEMA (Ours) | **62.55** |

Bold value indicates the highest result
KNN means using the K-nearest neighbor as the classifier. FT-C, EWC-C, and LwF-C represent applying Fine-Tune, EWC, and LwF to the classifier respectively

### 5.3 Ablation on BEMA

#### 5.3.1 Why use BEMA for Classifier?

Due to the continuous updates of the classifier throughout the incremental learning process, if no anti-forgetting measures are taken, catastrophic forgetting will occur. We initially considered some classical and generic anti-forgetting methods, such as LwF and EWC. However, these methods have a significant drawback: they are unable to measure the differences between different domains and, therefore, cannot adaptively combine old and new knowledge. As mentioned in Sect. 3.4, to address this limitation, we devise the shared prompt-guided BEMA. As shown in Table 8, BEMA is capable of effectively balancing knowledge forgetting and acquisition, which enables BEMA to overcome the limitations of traditional anti-forgetting methods and offer a more robust solution for domain incremental learning tasks.

#### 5.3.2 The Influence of Different Designs in BEMA

Firstly, we study the influence of different update granularity for EMA. The granularity from coarse to fine is task-wise,

**Table 5** Ablation study about the influence of different components in C-Prompt

| Components | | DomainNet | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ComP | BEMA | Real | Quickdraw | Painting | Sketch | Infograph | Clipart | Ave-ACC |
| ✗ | ✗ | 71.75 | 10.41 | 50.05 | 40.93 | 23.26 | 71.20 | 44.60 |
| ✗ | ✓ | 72.90 | 13.55 | 52.22 | 47.98 | 24.59 | 76.12 | 47.89 |
| ✓ | ✗ | 72.84 | 16.77 | 53.23 | 50.14 | 25.72 | **76.56** | 49.21 |
| ✓ | ✓ | **83.91** | **45.46** | **65.23** | **58.12** | **30.98** | 68.36 | **58.68** |

✗ and ✓ represent without or with this component
Bold values indicate the highest result

**Table 7** Ablation study about the influence of different forms of prompting used in our proposed method

| Prompting forms | DomainNet | | | | | | |
|---|---|---|---|---|---|---|---|
| | Real | Quickdraw | Painting | Sketch | Infograph | Clipart | Ave-ACC |
| Single | 82.63 | 27.39 | 63.29 | 57.86 | 30.03 | 71.01 | 55.30 |
| Single + comp | 83.02 | 27.15 | 63.48 | 58.40 | 30.64 | **71.73** | 55.74 |
| Pools | 83.21 | 35.01 | 64.71 | **58.91** | **32.46** | 70.88 | 57.53 |
| Pools + comp | **83.91** | **45.46** | **65.23** | 58.12 | 30.98 | 68.36 | **58.68** |

Bold values indicate the highest result

**Table 8** The performance of using different anti-forgetting methods on ImageNet-R

| Methods | Ave-ACC |
|---|---|
| ComP + KNN | 50.96 |
| ComP + FT-C | 56.58 |
| ComP + EWC-C | 58.26 |
| ComP + LwF-C | 60.83 |
| ComP+BEMA (ours) | **62.55** |

Bold value indicates the highest result

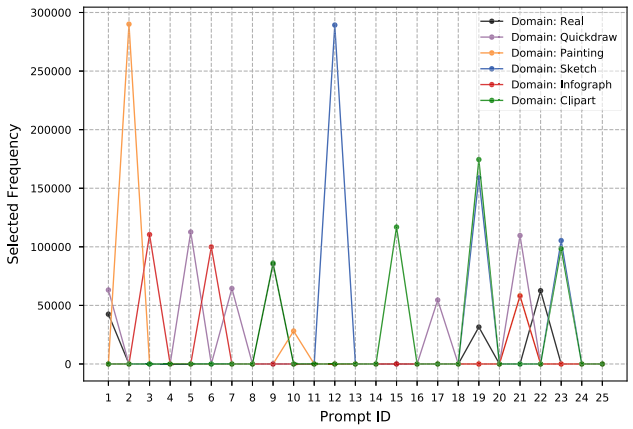


**Fig. 9** The selected frequency of prompts in the prompt pool on DomainNet

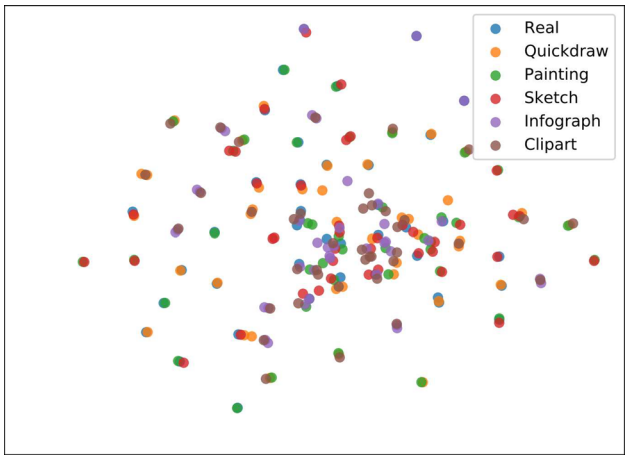


**Fig. 10** The T-SNE results of our domain-specific prompts. The different colored circles represent prompts from different domains

epoch-wise, and batch-wise. The more fine-grained the granularity is, the more likely the model can retain more historical information during incremental learning, and thus the weaker the forgetting of old knowledge is. As shown in Table 9, our proposed batchwise EMA strategy significantly outperforms the other two designs. This is because a batch-wise update is more suitable for handling severe data variations in the DIL scenario. Different choices of $\beta_t$ for adaptive knowledge consolidation are compared. Using the adaptive strategy defined in Eq. (12) performs better than the fixed $\beta_t$ and other widely-used adaptive ways.

**Table 9** Ablation study about the influence of different designs in the proposed BEMA algorithm

| BEMA designs | DomainNet | | | | | | |
|---|---|---|---|---|---|---|---|
| | Real | Quickdraw | Painting | Sketch | Infograph | Clipart | Ave-ACC |
| Task-wise | 84.31 | 20.22 | 62.11 | 55.05 | 29.70 | 68.93 | 53.39 |
| Epoch-wise | **84.37** | 23.82 | 63.18 | 56.32 | 30.43 | **69.23** | 54.56 |
| Batch-wise | 83.91 | **45.46** | **65.23** | **58.12** | **30.98** | 68.36 | **58.68** |
| $\beta_t = 0.9999$ | 83.22 | 36.33 | 63.86 | 57.53 | 30.17 | **69.53** | 56.77 |
| $\beta_t = t/(1+t)$ | **84.61** | 36.17 | 62.43 | 55.45 | 29.23 | 67.58 | 55.91 |
| $\beta_t = (t-1)/t$ | 84.35 | 39.61 | 63.92 | 57.18 | 30.14 | 68.86 | 57.34 |
| $\beta_t = t/(\mathcal{G}_t/\mathcal{G}_t^* + t)$ | 84.44 | 35.73 | 62.56 | 55.26 | 29.07 | 67.39 | 55.74 |
| $\beta_t$ in Eq. (12) | 83.91 | **45.46** | **65.23** | **58.12** | **30.98** | 68.36 | **58.68** |

Bold values indicate the highest result

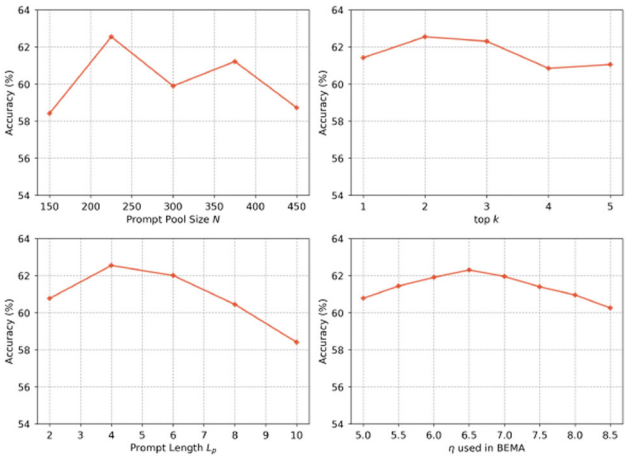


**Fig. 11** The influence of hyperparameters in C-Prompt

**Table 10** The performance of using BEMA to DualPrompt on DomainNet

| Methods | Ave-ACC | Quickdraw |
|---|---|---|
| DualPrompt | 49.36 | 13.96 |
| DualPrompt+BEMA | 57.84 | 36.15 |
| C-Prompt (ours) | **58.68** | **45.46** |

Bold values indicate the highest result

**Table 11** The performance of using different pre-trained backbone on ImageNet-R

| Pre-trained model | FT-Seq | C-Prompt |
|---|---|---|
| ViT-B/16 (Dosovitskiy et al., 2020) | 58.80 | **62.55** |
| MAE (He et al., 2022) | 49.76 | **56.09** |
| DINO (Caron et al., 2021) | 50.83 | **57.96** |

Bold values indicate the highest result achieved using the same backbone, i.e., the maximum value in each row

#### 5.3.3 Implement BEMA to DualPrompt

Due to the compatibility of our BEMA component with DualPrompt, as it only requires the shared prompt to adaptively mitigate classifier forgetting, we apply BEMA to DualPrompt to further demonstrate its effectiveness.

As shown in Table 10, DualPrompt with our BEMA can also largely improve the performance on the Quickdraw domain, and our C-Prompt performs even better. This demonstrates that our BEMA can effectively mitigate the forgetting problem and is more powerful in our C-Prompt than with DualPrompt.

**Table 12** Comparison of training time on DomainNet (seconds/epoch) with state-of-the-art

| Methods | Training time (s) |
|---|---|
| L2P (Wang et al., 2022c) | 973 |
| S-Prompts (Wang et al., 2022b) | 871 |
| DualPrompt (Wang et al., 2022a) | 928 |
| CODA-Prompt (Smith et al., 2023) | 992 |
| C-Prompt (Ours) | 989 |

### 5.4 Research on Generalization of C-Prompt

#### 5.4.1 Apply to more backbones

We conduct additional experiments using the pre-trained DINO (Caron et al., 2021) and MAE (He et al., 2022) as backbones on ImageNet-R. As shown in Table 11, for all three backbones, our C-prompt can largely and consistently improve the Ave-ACC performance over all domains.

#### 5.4.2 The Influence of Different Hyperparameters in C-Prompt

There are four key hyperparameters in C-Prompt including the prompt pool size $N$, the length of prompts $L_p$, the number of selected prompts $k$, and $\eta$ used in BEMA. Specifically, $L_p \times N$ determines the learning capacity of the model, $L_p$ determines the learning capacity of a single prompt, $n$ affects the diversity of the prompt pool, and $k$ affects the diversity of the final compositional prompt. Besides, $\eta$ is used to balance the learning of new knowledge and the retention of old knowledge. As shown in Fig. 11, the best combination of these hyperparameters can be achieved when $N = 225$, $L_p = 4$, $k = 2$, and $\eta = 6.5$.

### 5.5 Computation Cost of C-Prompt

#### 5.5.1 Comparison of Training Time with State-of-the-Art

In our experiments, our proposed C-Prompt achieves better performance than the SOTA methods on all four benchmarks. Moreover, we further investigate the time cost of C-Prompt compared to the SOTA incremental learning methods. The time cost (in seconds) of training one epoch on the first domain of DomainNet for the SOTA methods and our C-Prompt are reported in Table 12. Compared with L2P (Wang et al., 2022c), S-Prompts (Wang et al., 2022b) and DualPrompt (Wang et al., 2022a), our C-Prompt takes 1.64%, 13.55%, and 6.57% more time for training, respectively. Compared with CODA-Prompt (Smith et al., 2023), training time is almost the same. This is because after introducing the globally shared prompt-guided BEMA component, C-

**Table 13** Comparison of the number of parameters with the state-of-the-art

| Methods | Total params | Trainable params |
|---|---|---|
| L2P (Wang et al., 2022c) | 86,263,843 | 311,387 |
| S-Prompts (Wang et al., 2022b) | 92,964,094 | 1,637,910 |
| DualPrompt (Wang et al., 2022a) | 86,514,211 | 561,755 |
| CODA-Prompt (Smith et al., 2023) | 88,437,580 | 2,638,926 |
| C-Prompt (ours) | 89,180,505 | 3,381,851 |

**Total Params** and **Trainable Params** represent the numbers of total parameters (including the frozen and learnable parameters used) and the learnable parameters respectively

**Table 14** Comparison of iteration time with different numbers of prompts

| Prompts | Iteration time (s) |
|---|---|
| 10 | 2.03 |
| 30 | 2.06 |
| 90 | 2.09 |
| 150 | 2.11 |

Prompt needs to calculate the average results of the classifier at each batch, which has additional time overhead. Also, as we can see in Sect. 5.5.2 below, our method has more trainable parameters than the SOTA methods.

#### 5.5.2 Comparison of the Number of Parameters with State-of-the-Art

We also investigate the number of parameters to evaluate the memory overhead of C-Prompt during training. Table 13 shows the numbers of total parameters and trainable parameters of the SOTA methods and C-Prompt. Compared with L2P and DualPrompt, the total parameters of our model have only increased by 3.38% and 3.08% respectively, and the number of learnable parameters has risen by 6 to 10 times. Compared with S-Prompts, when the total parameter quantity of our model is reduced by 4.07%, the learnable parameters are doubled. Compared with CODA-Prompt, the trainable parameter quantity of our model is also increased by 28%. This is because our method leverages a more diverse prompt pool for prompt combination, which performs better on the DIL task with a more severe domain shift.

#### 5.5.3 Training Time Cost of Different Numbers of Prompts

As shown in Table 14, our time cost is almost the same as S-Prompts and is **not sensitive** to the prompt pool size. This is because, compared to the complex ViT structure, prompts are lightweight. Moreover, the prompt selection process is based on computing the cosine similarity between the query and the corresponding key in the prompt. This operation only requires one matrix multiplication to obtain the similarities and then sort the similarity results, making the additional computational overhead negligible and negligible.

## 6 Conclusion

The important and practical DIL task becomes even more challenging when multiple sources of domain shifts, such as style shifting, quality degradation, and environment changes, are simultaneously involved. Thus, we propose a novel paradigm called C-Prompt to mitigate the critical forgetting challenge caused by domain gaps in DIL. Unlike existing prompting-based methods, our proposed C-Prompt explores a set of globally shared prompts across all domains to estimate knowledge shifting for automatically balancing knowledge acquisition and forgetting. Moreover, a specific pool of learnable prompts for each domain is learned, and then all of them are further utilized in a compositional manner to enhance the learning capacity of models and facilitate model inference. Extensive experiments on various multi-domain datasets have thoroughly verified the superiority of our proposed C-Prompt against the state-of-the-art DIL methods.

**Acknowledgements** This work was supported by the National Natural Science Foundation of China (62376011, 61925201, 62132001).

**Data Availability** The datasets that support the results and analysis of the current study are available in the DomainNet http://ai.bu.edu/M3SDA/, ImageNet-R https://github.com/hendrycks/imagenet-r, ImageNet-C https://github.com/hendrycks/robustness and CORe50 https://vlomonaco.github.io/core50/ repositories.

## References

Agarwal, P., Paudel, D. P., Zaech, J. -N., & Van Gool, L. (2022). Unsupervised robust domain adaptation without source data. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, (pp. 2009–2018).

Bahng, H., Jahanian, A., Sankaranarayanan, S., & Isola, P. (2022). Exploring visual prompts for adapting large-scale models. arXiv preprint arXiv:2203.17274.

Boschini, M., Bonicelli, L., Buzzega, P., Porrello, A., & Calderara, S. (2022). Class-incremental continual learning into the extended der-verse. arXiv preprint arXiv:2201.00766.

Buzzega, P., Boschini, M., Porrello, A., Abati, D., & Calderara, S. (2020). Dark experience for general continual learning: A strong, simple baseline. *Advances in Neural Information Processing Systems, 33*, 15920–15930.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. In: *Proceedings of the IEEE/CVF international conference on computer vision*, (pp. 9650–9660).

Cha, H., Lee, J., & Shin, J. (2021). Co2l: Contrastive continual learning. In *ICCV*.

Chaudhry, A., Rohrbach, M., Elhoseiny, M., Ajanthan, T., Dokania, P. K., Torr, P. H., & Ranzato, M. (2019). On tiny episodic memories in continual learning. arXiv preprint arXiv:1902.10486.

Chen, C., Fu, Z., Chen, Z., Jin, S., Cheng, Z., Jin, X., & Hua, X.-S. (2020). Homm: Higher-order moment matching for unsupervised domain adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence, 34*, 3422–3429.

Delange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., & Tuytelaars, T. (2021). A continual learning survey: Defying forgetting in classification tasks. In *PAMI*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *CVPR*.

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

Douillard, A., Ramé, A., Couairon, G., & Cord, M. (2022). Dytox: Transformers for continual learning with dynamic token expansion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9285–9295.

Fini, E., Da Costa, V.G.T., Alameda-Pineda, X., Ricci, E., Alahari, K., & Mairal, J. (2022). Self-supervised models are continual learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, (pp. 9621–9630).

Garg, P., Saluja, R., Balasubramanian, V.N., Arora, C., Subramanian, A., & Jawahar, C. (2022). Multi-domain incremental learning for semantic segmentation. In *WACV*.

Hao, S., Zhou, Y., & Guo, Y. (2020). A brief survey on semantic segmentation with deep learning. *Neurocomputing, 406*, 302–321.

Hayes, T. L., Cahill, N. D., & Kanan, C. (2019). Memory efficient experience replay for streaming learning. In *ICRA*.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, (pp. 16000–16009).

Hendrycks, D., & Dietterich, T. G. (2018). Benchmarking neural network robustness to common corruptions and surface variations. arXiv preprint arXiv:1807.01697.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., & Guo, M., *et al.* (2021). The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*.

Hou, S., Pan, X., Loy, C.C., Wang, Z., & Lin, D. (2019). Learning a unified classifier incrementally via rebalancing. In *CVPR*.

Hoyer, L., Dai, D., & Van Gool, L. (2022). Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In *CVPR*.

Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.

Huang, Q., Dong, X., Chen, D., Zhang, W., Wang, F., Hua, G., & Yu, N. (2023). Diversity-aware meta visual prompting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, (pp. 10878–10887).

Kanakis, M., Bruggemann, D., Saha, S., Georgoulis, S., Obukhov, A., & Gool, L.V. (2020). Reparameterizing convolutions for incremental multi-task learning without task interference. In *ECCV*.

Kingma, D.P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., & Grabska-Barwinska, A., Hassabis, D. (2017). Overcoming catastrophic forgetting in neural networks. In *Proceedings of the National Academy of Sciences*.

Kundu, J. N., Venkat, N., & Babu, R. V., *et al.* (2020). Universal source-free domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, (pp. 4544–4553).

Kundu, J. N., Venkatesh, R.M., Venkat, N., Revanur, A., & Babu, R.V. (2020). Class-incremental domain adaptation. In *ECCV*.

Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. arXiv preprint arXiv:2104.08691.

Li, Z., & Hoiem, D. (2017). Learning without forgetting. *PAMI*.

Li, X.L., & Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. arXiv preprint arXiv:2101.00190.

Li, Y., Mao, H., Girshick, R., & He, K. (2022). Exploring plain vision transformer backbones for object detection. In *European conference on computer vision*, (pp. 280–296). Springer.

Lian, Q., Lv, F., Duan, L., & Gong, B. (2019). Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach. In *ICCV*.

Liang, J., Hu, D., & Feng, J. (2020). Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International conference on machine learning*, (pp. 6028–6039). PMLR.

Li, S., Liu, C., Lin, Q., Xie, B., Ding, Z., Huang, G., & Tang, J. (2020). Domain conditioned adaptation network. *Proceedings of the AAAI Conference on Artificial Intelligence, 34*, 11386–11393.

Lin, H., Zhang, Y., Qiu, Z., Niu, S., Gan, C., Liu, Y., & Tan, M. (2022). Prototype-guided continual adaptation for class-incremental unsupervised domain adaptation. In *ECCV*.

Lomonaco, V., & Maltoni, D. (2017). Core50: A new dataset and benchmark for continuous object recognition. In *Conference on robot learning*, (pp. 17–26). PMLR.

Lu, J., Batra, D., Parikh, D., & Lee, S. (2019). Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. Advances in Neural Information Processing System.

Oren, G., & Wolf, L. (2021). In defense of the learning without forgetting for task incremental learning. In *ICCV*

Pan, S. J., Tsang, I. W., Kwok, J. T., & Yang, Q. (2010). Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks, 22*, 199–210.

Patel, V. M., Gopalan, R., Li, R., & Chellappa, R. (2015). Visual domain adaptation: A survey of recent advances. *IEEE Signal Processing Magazine, 32*, 53–69.

Pellegrini, L., Graffieti, G., Lomonaco, V., & Maltoni, D. (2020). Latent replay for real-time continual learning. In *IROS*.

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., & Wang, B. (2019). Moment matching for multi-source domain adaptation. In *ICCV*.

Prabhu, A., Torr, P.H., & Dokania, P.K. (2020). Gdumb: A simple approach that questions our progress in continual learning. In *Computer vision–ECCV 2020: 16th European conference, Glasgow, UK*, Proceedings, Part II 16, (pp. 524–540). Springer.

Price, W., & Cohen, I. (2019). Privacy in the age of medical big data. *Nature Medicine, 25*, 37–43. https://doi.org/10.1038/s41591-018-0272-7

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., & Clark, J. (2021). Learning transferable visual models from natural language supervision. In *ICML*.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., & Lampert, C.H. (2017). icarl: Incremental classifier and representation learning. In *CVPR*.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., & Hadsell, R. (2016). Progressive neural networks. arXiv preprint arXiv:1606.04671.

Schick, T., & Schütze, H. (2020). Exploiting cloze questions for few shot text classification and natural language inference. arXiv preprint arXiv:2001.07676.

Shin, T., Razeghi, Y., Logan IV, R. L., Wallace, E., & Singh, S. (2020). Autoprompt: Eliciting knowledge from language models with automatically generated prompts. arXiv preprint arXiv:2010.15980.

Simon, C., Faraki, M., Tsai, Y.-H., Yu, X., Schulter, S., Suh, Y., Harandi, M., & Chandraker, M. (2022). On generalizing beyond domains in cross-domain continual learning. In *CVPR*.

Smith, J.S., Karlinsky, L., Gutta, V., Cascante-Bonilla, P., Kim, D., Arbelle, A., Panda, R., Feris, R., & Kira, Z. (2023). Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, (pp. 11909–11919).

Tang, S., Su, P., Chen, D., & Ouyang, W. (2021). Gradient regularized contrastive learning for continual domain adaptation. In *AAAI*.

Tao, X., Hong, X., Chang, X., & Gong, Y. (2020). Bi-objective continual learning: Learning 'new' while consolidating 'known'. In *AAAI*.

Ven, G.M., & Tolias, A.S. (2019). Three scenarios for continual learning. arXiv preprint arXiv:1904.07734.

Volpi, R., Larlus, D., & Rogez, G. (2021). Continual adaptation of visual representations via domain randomization and meta-learning. In *CVPR*.

Wang, Q., Fink, O., Van Gool, L., & Dai, D. (2022). Continual test-time domain adaptation. In *CVPR*.

Wang, Y., Huang, Z., & Hong, X. (2022). S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning. arXiv preprint arXiv:2207.12819.

Wang, Z., Jian, T., Chowdhury, K., Wang, Y., Dy, J., & Ioannidis, S. (2020). Learn-prune-share for lifelong learning. In *ICDM*.

Wang, Z., Zhang, Z., Ebrahimi, S., Sun, R., Zhang, H., Lee, C.-Y., Ren, X., Su, G., Perot, V., & Dy, J., Pfister, T. (2022). Dualprompt: Complementary prompting for rehearsal-free continual learning.

Wang, Z., Zhang, Z., Lee, C.-Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., & Pfister, T. (2022). Learning to prompt for continual learning. In *CVPR*.

Wang, Q., & Breckon, T. (2020). Unsupervised domain adaptation via structured prediction based selective pseudo-labeling. *Proceedings of the AAAI Conference on Artificial Intelligence, 34*, 6243–6250.

Wang, Y., Ma, Z., Huang, Z., Wang, Y., Su, Z., & Hong, X. (2023). Isolation and impartial aggregation: A paradigm of incremental learning without interference. *Proceedings of the AAAI Conference on Artificial Intelligence, 37*, 10209–10217.

Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., & Fu, Y. (2019). Large scale incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, (pp. 374–382).

Xie, J., Yan, S., & He, X. (2022). General incremental learning with domain-aware categorical representations. In *CVPR*.

Yang, S., Wang, Y., Van De Weijer, J., Herranz, L., & Jui, S. (2021). Generalized source-free domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, (pp. 8978–8987).

Yang, C., Wu, Z., Zhou, B., & Lin, S. (2021). Instance localization for self-supervised detection pretraining. In *CVPR*.

Zaken, E.B., Ravfogel, S., & Goldberg, Y. (2021). Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. arXiv preprint arXiv:2106.10199.

Zhou, K., Yang, J., Loy, C.C., & Liu, Z. (2022). Conditional prompt learning for vision-language models. In *CVPR*.

Zhu, Y., Zhang, Z., Wu, C., Zhang, Z., He, T., Zhang, H., Manmatha, R., Li, M., & Smola, A. (2021). Improving semantic segmentation via efficient self-training. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 46*(3), 1589–1602.

Zou, Y., Yu, Z., Kumar, B., & Wang, J. (2018). Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*.