Evolving Parameterized Prompt Memory for Continual Learning

Muhammad Rifki Kurniawan[1], **Xiang Song**[1], **Zhiheng Ma**[3],
Yuhang He[2], **Yihong Gong**[1, 2], **Qi Yang**[4], **Xing Wei**[1 *]

[1]School of Software Engineering, Xi'an Jiaotong University
[2]College of Artificial Intelligence, Xi'an Jiaotong University
[3]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
[4]School of Computer Science and Technology, Xi'an Jiaotong University
{rifki259, songxiang} @stu.xjtu.edu.cn, zh.ma@siat.ac.cn, {heyuhang, ygong, yangqi, weixing} @mail.xjtu.edu.cn

Abstract

Recent studies have demonstrated the potency of leveraging prompts in Transformers for continual learning (CL). Nevertheless, employing a discrete key-prompt bottleneck can lead to selection mismatches and inappropriate prompt associations during testing. Furthermore, this approach hinders adaptive prompting due to the lack of *shareability* among nearly identical instances at more granular level. To address these challenges, we introduce the Evolving Parameterized Prompt Memory (EvoPrompt), a novel method involving adaptive and continuous prompting attached to pre-trained Vision Transformer (ViT), conditioned on specific instance. We formulate a continuous prompt function as a neural bottleneck and encode the collection of prompts on network weights. We establish a paired prompt memory system consisting of a stable *reference* and a flexible *working* prompt memory. Inspired by linear mode connectivity, we progressively fuse the working prompt memory and reference prompt memory during inter-task periods, resulting in continually evolved prompt memory. This fusion involves aligning functionally equivalent prompts using optimal transport and aggregating them in parameter space with an adjustable bias based on prompt node attribution. Additionally, to enhance backward compatibility, we propose compositional classifier initialization, which leverages prior prototypes from pre-trained models to guide the initialization of new classifiers in a subspace-aware manner. Comprehensive experiments validate that our approach achieves state-of-the-art performance in both class and domain incremental learning scenarios. Source code is available at https://github.com/MIV-XJTU/EvoPrompt.

Introduction

Despite deep networks have achieved remarkable performance on parallel multi-task learning (Misra et al. 2016), they mostly suffer from *catastrophic forgetting* (McCloskey and Cohen 1989) of past knowledge and are biased toward the recent task under sequential tasks setting. Thus, the key issue of CL methods is how to balance the flexibility and rigidity, which is referred to as the *stability-plasticity dilemma* (Mermillod, Bugaiska, and Bonin 2013). To meet this challenge, some methods replay the previous samples (Rebuffi et al. 2017), penalize significant network changes on either parameters (Kirkpatrick et al. 2017) or prior neural activations (Tao et al. 2020), or dynamically learn less-interference parameters (Douillard et al. 2022). However, training *tabula rasa* is no longer viable due to available pre-training, yet directly training pre-existing CL algorithms from pre-trained models yields inconsistent performance (Lee, Zhong, and Wang 2023).

Continual adaptation through the prompting of the pre-trained model, specifically Vision Transformer (Dosovitskiy et al. 2021), demonstrates a promising avenue. L2P (Wang et al. 2022c) builds a prompt pool to compose key-prompt pairs and exploits a discrete key-value bottleneck for prompt selection. DualPrompt (Wang et al. 2022b) complements L2P with a versatile G-Prompt attached on some earlier blocks. S-prompt (Wang, Huang, and Hong 2022) ensures that previous tasks prompts remain undisturbed by isolating the prompt learning process for each task. However, since they utilize a pool-based framework, the corresponding issue of suboptimality hinders the final performance as a result of inaccurate prompts selection at test-time (Wang et al. 2022b). Furthermore, aggressive discretization of query-key association means that non-identical classes should attach the same prompt if from the same task, avoiding prompt shareability among nearly identical instances at fine-grained representation. To bridge this gap, we propose a parameterized prompt memory scheme with incremental evolution in continual learning.

We redesign prompt parameterization as feed-forward networks (FFNs) with multilayer perceptron (MLP) bottleneck, depicted in Figure 1, where we introduce dual functional prompt memory composing *reference* prompt memory (RPM) and *working* prompt memory (WPM). The RPM generalizes all prompts so far, while the WPM is task-specific and adapts quickly as new tasks appear. To integrate both without forgetting, we draw inspiration from *linear mode connectivity* (Frankle et al. 2019), where there is a single basin with a low error landscape between different task solutions. Specifically, we introduce *incremental fusion* during the inter-task period to functionally align WPM with RPM, and then integrate both together in param-

[*]Corresponding author.

eter space. We formulate the alignment as an *optimal transport* (OT) problem and the integration as a linearly weighted aggregation adjusted by neuron attribution. This reformulation allows us to store all common prompts in a continually evolved unified representation, while the prompt attached during testing is adaptive and unique for each input instance.

Humans can effectively deduce unknown aspects of an uncertain event, such as calculating the likelihood that object B belongs to A. This capability is realized by employing valuable heuristic principles, specifically *anchoring-and-adjustment heuristic* (Tversky and Kahneman 1975). Where they exploit the available information at hand, using the nearest similar concept for anchoring and then predicting with small adjustments from the anchor. Inspired by this, we propose *compositional classifier initialization* (CCI), inferring the future classifiers from available old classifiers and prototypical relations between classes. Technically, we utilize *prior assumptions* about new and old task relations, represented by similarity-based attention between class mean embedding (prototype) inferred from pre-training, to aid in the initialization of future classifiers. Consequently, the novel classifier for new classes exists within the old classifier subspace, laying on common space, thereby mitigating backward incompatibility.

Therefore, our contributions are as follows: 1) We propose **Evo**lving Parameterized Memory **Prompt** (EvoPrompt), an effective CL method based on prompt parameterization learning, which learns a reference shareable prompt and incrementally fuse the newly learned working prompts into this reference prompt. This allows gradual memory evolution while also adaptive prompt generation conditioned on input query. 2) We explore the backward-compatible initialization strategy for future classifiers via compositional classifier initialization. 3) Our proposals surpass the state-of-the-art prompt-based CL methods by a large gap on both domain and class incremental tasks.

Related Work

Continual learning. Generally, continual learning methods address catastrophic forgetting by preserving previous task information. *Rehearsal-based* methods replay old samples while learning the current task by generating them through generative networks (Shin et al. 2017; Kemker and Kanan 2018), storing the data in raw input space (Rebuffi et al. 2017; Chaudhry et al. 2019; Wei et al. 2023) or deep embedding space (Zhu et al. 2021). *Regularization-based* methods penalize the networks by dynamically learning optimal solutions for current and past tasks by regularizing the important parameters of past tasks (Mitchell et al. 2015; Akyürek et al. 2022) or distilling the past networks response (Wu et al. 2019; Tao et al. 2020; Yu et al. 2020). *Architectural-based* methods learn independent networks for each task to alleviate catastrophic parameter interference, grouped into dynamic networks (Wang et al. 2022a; Yan, Xie, and He 2021) or find subnets from the whole networks (Aljundi, Chakravarty, and Tuytelaars 2017). Recently, some efforts adapting pre-trained networks for CL have achieved promising results. L2P (Wang et al. 2022c) proposes rehearsal-free prompts to inform the pre-trained ViT. They use the prompt-pool framework with a key-value bottleneck. DualPrompt (Wang et al. 2022b) complements the previous work by introducing a general prompt shared between tasks. S-Prompt (Wang, Huang, and Hong 2022) focuses on domain incremental learning through a training domain expert prompt on ViT and contrastive language-image pre-training (CLIP) (Radford et al. 2021). Our method aligns with the non-expandable architectural approach involving prompt-based CL that progressively evolves with the data stream.

Mode connectivity and model fusion. Understanding the connection between various neural training methods, often referred to as *mode connectivity*, offers good insight for CL. However, it is not necessarily connected by a linear path. According to (Frankle et al. 2019), a suitable training process can result in a flat basin with low error between multiple local minima, forming a linear path when the networks share the same initial parameters. The necessity also pertains to CL prompts, where the emergence of a linear mode depends on the use of a common initialization (Mirzadeh et al. 2021). If achieved, using a simple linear interpolation of the parameters can lead to generalizable minimum values, which in turn enables the merging of multiple models (Singh and Jaggi 2020; Ainsworth, Hayase, and Srinivasa 2023). This framework serves as our basis, involving rapid adaptation of working prompt memory using FFN at each stage, and gradually merging it with the reference prompt memory, producing prompt memory that perpetually evolves.

Proposed Method

Problem Formulation

The main objective of CL is to train a unified model over a sequence of tasks, where each task introduces new class or domain experiences. Formally, we have a set of discrete sequential tasks denoted $(\mathfrak{D}_1, ..., \mathfrak{D}_T)$, encompassing a total of T tasks. Each individual task t is composed of training data $\mathfrak{D}_t^{train}$ and a corresponding testing dataset $\mathfrak{D}_t^{test}$. The training dataset for the t-th task consists of $|\mathcal{X}^t|$ samples, where each sample i comprises an input image x_i^t paired with its corresponding label y_i^t. In class incremental learning (CIL), the testing dataset is constructed using test input images that cover all classes learned up to the current task t. While, domain incremental learning (DIL) involves conducting tests across all trained domains. In CIL, tasks have unique classes, while in DIL, classes stay constant while new domains are introduced with each task.

Reformulating Incremental Prompt Tuning

Here, we begin by providing a succinct overview of prompting ViTs and followed by detailed introduction of our continuous selection approach for prompting facilitated by FFN.

Transformer with deep prompt tuning. A pre-trained ViT (Dosovitskiy et al. 2021) can be regarded as the composition of a patch embedding f_ϑ and multiple blocks of Transformer $f_{\boldsymbol{\theta}}$ with the parameter sets ϑ and $\boldsymbol{\theta} = \{\theta_1, \theta_2, ..., \theta_B\}$, respectively. Where, $f_{\boldsymbol{\theta}}$ combine B sequential blocks and each block b contains a Multi-head Self-

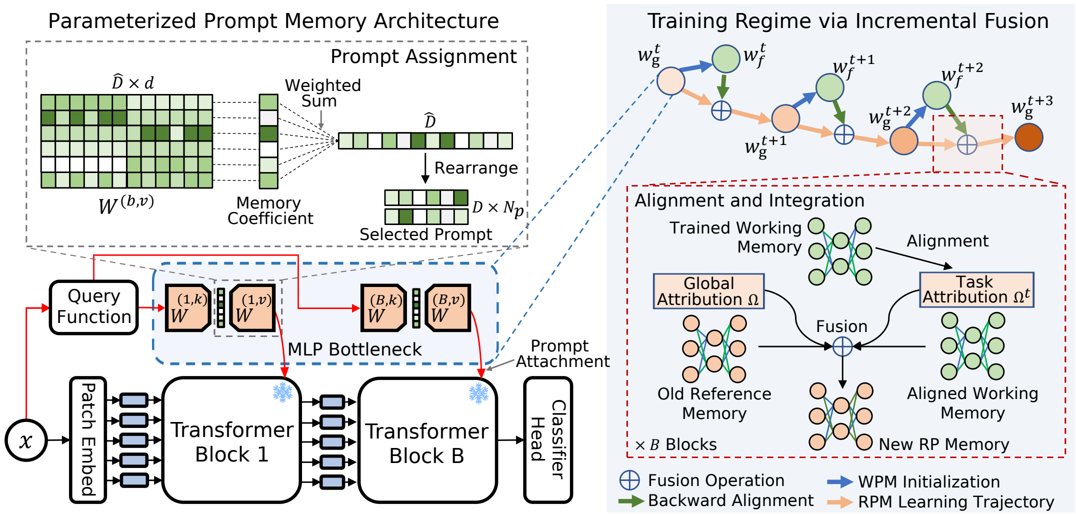


Figure 1: EvoPrompt core: memory prompt parameterization using FFN and training via incremental fusion with alignment. Given input x, the memory identifies patterns to determine *memory coefficient*, to weigh the value memory for the final prompt. At task transitions, the working prompt (green) is merged with the reference prompt (orange) to consolidate knowledge.

Attention (MSA) block. The entire parameter set of ViT is defined as $f = f_\vartheta \circ f_{\boldsymbol{\theta}}$. By passing an image $x \in \mathbb{R}^{3 \times w \times h}$ to f_ϑ, we can get image token $E_0 = f_\vartheta(x)$. Then, we define the image token on block b as $E_b \in \mathbb{R}^{N \times D}$, where N denotes the number of token embeddings and D is the dimension. Steering the representation towards the target task can be achieved by learning trainable context or prompts (Lester, Al-Rfou, and Constant 2021), facilitating lightweight adaptation without modifying the whole backbone. To do so, we prepend a learnable prefix prompt $p_b \in \mathbb{R}^{N_p \times D}$ using VPT-Deep (Jia et al. 2022) on the input token before each MSA block, where N_p is the prompt length. The modified input and corresponding block output are formulated as:

$$\begin{aligned} [h_1, z_1, E_1] &= f_{\theta_1}([p_1, z_0, E_0]), \\ [h_b, z_b, E_b] &= f_{\theta_b}([p_b, h_{b-1}, z_{b-1}, E_{b-1}]), b > 1 \end{aligned} \tag{1}$$

where θ_b denotes the network parameters of block b, $z_b \in \mathbb{R}^D$ is class token [cls], $[\cdot, \cdot]$ indicates tensor concatenation along token sequence length dimension, and $h_b \in \mathbb{N}^{N_p \times b \times D}$ denotes the prompt output on block b. As the block goes deeper, the prompt output will be extended, resulting in a final accumulated $B \times N_p$ length of prompt token.

Feed forward networks as prompt memory. Instead of formulating continual prompting as discrete selection (Wang et al. 2022b), we reformulated prompting by employing FFN, particularly a MLP bottleneck. It encodes the prompts in the neural weight space, which we call *prompt memory*. The proposed design yields multiple advantages: 1) predicting task identity is no longer necessary, 2) we can achieve end-to-end learning, thereby tightly connected to target objective, 3) semantic-aware prompting facilitate enhanced prompt shareability at a pattern level, instead of shared prompts based on task id that shared among semantically different classes. Our approach utilizes FFN to build a key-value memory. Specifically for an input image x, following the previous method (Wang et al. 2022c), i.e., taking the frozen extractor f as a query function, we get the query feature $q(x) = f(x)[0]$, $q(x) \in \mathbb{R}^D$, corresponding to [cls] token. At block b, the associated prompt is obtained as follows:

$$\begin{aligned} \boldsymbol{p}_b &= f_{\mathbf{W}^{(b)}}\left(q(x); \mathbf{W}^{(b)}\right), \\ &= \mathrm{ReLU}\left(q(x) \cdot \mathbf{W}^{(b,k)}\right) \cdot \mathbf{W}^{(b,v)}, \end{aligned} \tag{2}$$

where $\mathbf{W}^{(b,k)} \in \mathbb{R}^{D \times d}$ expresses the set of keys in matrices at block b, which can be interpreted as linear down-projection parameters, $\mathbf{W}^{(b,v)} \in \mathbb{R}^{d \times \hat{D}}$ is a group of values or upper-projection parameters, where $\hat{D} = D \times N_p$ and d that satisfies $d \ll D$ is the bottleneck dimension, with ReLU non-linear activation in between.

Eq. (2) closely resembles key-value neural memories (Sukhbaatar et al. 2015), where memory capacity is represented by the hidden dimension d. The memory coefficients (Geva et al. 2021), stemming from the ReLU activation, manifest as non-negative unnormalized weights extracted from hidden outputs. These coefficients are used to assign the corresponding prompts $\mathbf{W}^{(b,v)}$. The ReLU activation facilitates implicit non-negative sparse selection by discarding irrelevant memory given the query.

Prompt Evolution via Incremental Fusion

Reference and working prompt memory. We embrace dual memory paradigm, as regularly advocated in (Pham, Liu, and Hoi 2021; Arani, Sarfraz, and Zonooz 2022), by introducing two distinct memory mechanisms: *reference* prompt memory $\mathbf{W}_g$ for stable and broad knowledge and *working* prompt memory for adaptive and targeted knowledge $\mathbf{W}_f$. During the learning of task t, the WPM is the primary learner for assimilating new knowledge. This occurs through batch updates using standard Stochastic Gradient Descent (SGD), which minimizes the Cross-Entropy (CE) loss by tuning classifier head φ and $\mathbf{W}_f$:

$$\min_{\mathbf{W}_f,\varphi} -\sum_{x_i^t \in \mathcal{X}^t} y_i^t \log \sigma_s\left(f_\varphi\left(f\left(x_i^t, p_i\right)[0]\right)\right), \tag{3}$$

where $\sigma_s(\cdot)$ denotes Softmax function. During inter-task transition, the trained WPM is integrated into RPM, producing RPM that evolve continually, and followed by WPM initialization from the RPM for the next phase. This transfer is preferable because may promote mode connectivity between these dual memory (Frankle et al. 2019).

Alignment before aggregation. During the learning process, as WPM is adjusted to optimize for a new task, the network's functionality can potentially shift, leading to reduced direct match consistency with RPM. To counteract this, we align the WPM with the RPM, aiming to achieve functional compatibility before proceeding with fusion. This alignment ensures that the consolidation of prompt memory occurs within memory segments that are functionally comparable. Consequently, our objective is to determine a permutation matrix $\mathbf{P}_\ell$ at layer ℓ that identifies functionally equivalent parameters. While various existing techniques are viable for this purpose (Tatro et al. 2020; Kantorovich 2006), for the sake of simplicity, we opt to align the memory by employing optimal transportation (Singh and Jaggi 2020). Specifically, this involves finding an optimal mapping that projects the updated WPM onto the weight space of the functionally equivalent old RPM. This approach facilitates a meaningful and effective alignment, enhancing the compatibility between the WPM and RPM for subsequent fusion.

Given layer ℓ that has N_ℓ nodes, we define probability mass values of WPM α_ℓ and RPM β_ℓ, respectively. We initialize them based on node importance measured by normalized ℓ_1-norm of its incoming weights row vector. Our next step involves addressing a minimization problem to derive a permutation matrix $\mathbf{P}_\ell$ that serves as a projector and is represented by a transport map:

$$\begin{aligned} \mathbf{P}_\ell = &\min_{\mathbf{P}_\ell \in \mathbb{R}_+^{N_\ell \times N_\ell}} \operatorname{tr}\left(\mathbf{P}_\ell^T \mathbf{D}_\ell\right) = \operatorname{OT}\left(\alpha_\ell, \beta_\ell, \mathbf{D}_\ell\right), \\ &\text{s.t. } \mathbf{P}_\ell \mathbf{1}_\ell = \alpha_\ell,\ \mathbf{P}_\ell^T \mathbf{1}_n = \beta_\ell, \\ &\mathbf{D}_{\ell,ij} = \delta\left(\mathbf{W}_{f,\ell(i)}^t, \mathbf{W}_{g,\ell(j)}^t\right), \end{aligned} \tag{4}$$

in which $\operatorname{tr}(\cdot)$ specifies the Frobenius inner product. The ground cost, denoted as $\mathbf{D}_\ell$, characterizes the one-to-one mapping cost between nodes of the WPM and the RPM, where $\delta(\cdot)$ is Euclidean distance, and i, j refer to node index. When the transport map is in place, we multiply the working prompt weight by the current transportation matrix $\mathbf{P}_\ell$ to bring it closer to the reference prompt, and followed by memory evolution via momentum merging:

$$\hat{\mathbf{W}}_{f,\ell}^t \leftarrow \operatorname{diag}\left(\frac{1}{\beta_\ell}\right) \mathbf{P}_\ell \mathbf{W}_{f,\ell}^t \mathbf{P}_{\ell-1} \operatorname{diag}\left(\frac{1}{\beta_{\ell-1}}\right), \tag{5}$$

$$\mathbf{W}_{g,\ell}^{t+1} \leftarrow \lambda_\ell \hat{\mathbf{W}}_{f,\ell}^t + (1-\lambda_\ell)\mathbf{W}_{g,\ell}^t, \tag{6}$$

where $\hat{\mathbf{W}}_{f,\ell}^t$ is aligned WPM and λ_ℓ denotes fusion weight. Here, because inter-block memory is not directly connected, only value memory is subject to backward alignment with previous layers transport map $\mathbf{P}_{\ell-1}$. During fusion, we take into consideration the importance of each node while merging memory, and use the λ to determine the fusion weight of node parameter vectors.

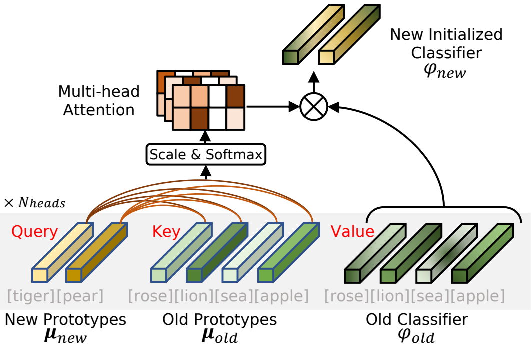

New Initialized Classifier φ_{new}
Multi-head Attention
Scale & Softmax
× Nheads
Query
Key
Value
[tiger][pear]
[rose][lion][sea][apple]
[rose][lion][sea][apple]
New Prototypes $\boldsymbol{\mu}_{new}$
Old Prototypes $\boldsymbol{\mu}_{old}$
Old Classifier φ_{old}

Figure 2: Compositional classifier initialization via multi-head prototypical attention (MPA).

Attribution-aware fusion. Instead of adopting a balanced merging approach (where λ_ℓ is set to 0.5), we embrace the concept of attribution-aware, merging to effectively navigate the stability-plasticity dilemma. To achieve this, we introduce anchor fusion weight λ_{base} establishing a base value set to 0.5. Subsequently, we fine-tune this factor using a adjustment factor λ_{adj} weighted by attribution change $\Delta\hat{\Omega}_\ell = \hat{\Omega}_\ell^t - \hat{\Omega}_\ell$. The adjustment factor is contingent on $\hat{\Omega}^t$ and $\hat{\Omega}$, both of which belong to the range $[0, 1]$. These values represent the min-max normalized *working* and *global* node attributions, respectively. Mathematically, this relationship can be expressed as:

$$\begin{aligned} \lambda_\ell &= \lambda_{\text{base},\ell} + \Delta\hat{\Omega}_\ell \lambda_{\text{adj},\ell}, \quad \lambda \in [0,1], \\ \lambda_{\text{adj},\ell} &= \begin{cases} \lambda_{\text{base},\ell}, & \text{if } \hat{\Omega}_\ell^t - \hat{\Omega}_\ell < 0 \\ 1 - \lambda_{\text{base},\ell}, & \text{if } \hat{\Omega}_\ell^t - \hat{\Omega}_\ell \geq 0. \end{cases} \end{aligned} \tag{7}$$

Intuitively, if the node vectors hold equal attribution on both WPM and RPM, the weight scheme enforces a balanced fusion at $\lambda_\ell \approx 0.5$, as $\Delta\hat{\Omega}_\ell$ is low. When less important nodes gain significance, the merging weight assigned to WPM becomes more pronounced, causing WPM to have more contribution to the final RPM. Once task t has been learned, the attribution $\Omega_{\ell(j)}$ of node j at layer ℓ is updated as follows:

$$\Omega_{\ell(j)}^t = \frac{1}{|\mathcal{X}^t|} \sum_{x_i^t \in \mathcal{X}^t} \operatorname{ReLU}\left(f_{n_{\ell(j)}}\left(x_i^t\right)\right), \forall i, \hat{y}_i = y_i, \tag{8}$$

$$\Omega_{\ell(j)} = \max\left(\Omega_{\ell(j)}, \Omega_{\ell(j)}^t\right), \tag{9}$$

where $f_{n_{\ell(j)}}$ is the output node $n_{\ell(j)}$ during prompting and $\hat{y}_i$ denotes the label prediction. Since, the attribution is saved at the node level, negligible amount of memory is required. Furthermore, it focuses solely on the correctly predicted labels and disregards the wrongly predicted ones, avoiding any detrimental effects arising from prediction errors.

Method	**Split CIFAR-100**							
	5 Steps		**10 Steps**		**20 Steps**		Avg	Avg
	Acc.(↑)	Forget.(↓)	Acc.(↑)	Forget.(↓)	Acc.(↑)	Forget.(↓)	Acc.(↑)	Forget.(↓)
FT-seq	73.17 ±0.75	2.95 ±0.56	62.77 ±2.30	20.73 ±2.05	55.97 ±2.95	32.74 ±2.97	63.97 (+0.00)	18.81 (−0.00)
LP-seq	71.69 ±0.61	**1.36 ±0.27**	66.90 ±0.53	13.08 ±0.32	60.98 ±0.74	21.27 ±1.20	66.52 (+2.55)	11.90 (−6.91)
NME-seq	78.30	7.70	78.33	**1.14**	78.33	**2.68**	78.32 (+14.35)	**3.84** (**−14.97**)
L2P	86.53 ±0.14	7.67 ±0.20	84.97 ±8.21	8.21 ±0.22	83.39 ±0.41	10.18 ±0.24	84.96 (+20.99)	8.69 −10.12)
DualPrompt	88.26 ±0.33	5.72 ±0.43	86.83 ±0.37	6.21 ±0.35	84.11 ±0.45	8.75 ±0.38	86.40 (+22.43)	6.89 (−11.92)
ESN	88.09 ±0.21	5.18 ±0.13	85.96 ±0.14	4.54 ±0.35	82.71 ±0.51	6.44 ±0.31	85.59 (+21.62)	5.39 (−13.42)
CODA-P-S	88.90 ±0.26	6.29 ±0.27	86.33 ±0.25	6.29 ±0.52	81.71 ±0.47	9.41 ±0.22	85.65 (+21.68)	7.33 (−11.48)
CODA-P	**89.16 ±0.26**	6.08 ±0.33	87.31 ±0.14	5.95 ±0.41	81.69 ±0.38	9.85 ±0.58	86.05 (+22.08)	7.29 (−11.52)
EvoPrompt-S	88.69 ±0.16	9.93 ±0.22	87.95 ±0.13	2.38 ±0.14	**84.98 ±0.36**	3.42 ±0.39	**87.20** (**+23.23**)	5.24 (−13.57)
EvoPrompt	88.97 ±0.41	10.12 ±0.35	**87.97 ±0.30**	2.60 ±0.42	84.64 ±0.14	3.98 ±0.24	87.19 (+23.22)	5.57 (−13.24)
Upper-bound†	90.85 ±0.12	-	90.85 ±0.12	-	90.85 ±0.12	-	90.85	-

Table 1: Benchmark on average accuracy and forgetting on various splits of Split CIFAR-100. The last two columns present the metrics mean from multiple steps scenario. The best and second best results are marked in bold and underscore, respectively.

Compositional Classifier Initialization

The key insight of the previous method (Wang et al. 2022c) revolves around the independent learning of task classifiers (Ahn et al. 2020). However, this approach gives rise to two primary issues: 1) lack of reverse compatibility, the new classifiers outputs is less comparable with past tasks classifiers, and 2) classifiers produced through this independent learning strategy tend to be less discriminative, as they lack a direct comparison with classifiers from previous tasks. Thus, we introduce *compositional classifier initialization* (CCI), illustrated on Figure 2, inspired by bias-adjustment heuristic (Tversky and Kahneman 1975), where people typically estimate the unknown by leveraging relevant available information. In this context, the unknown is a future classifiers and available knowledge is a learned classifiers and strong pre-trained embedding representation. Concretely, we compute class mean embedding from pre-trained, called *prototypes*, and find the inter-class prototypes relation, represented as attention, forming the *foundational relational presumption* between old and target tasks. Then, we use the obtained probability simplex presented by multi-head attention to linearly combine the previous classifiers to initialize future classifiers, thus introduce prior implicit bias. We define the class prototype of class c by simply mean features, $\mu_c = \frac{1}{|\mathcal{X}_c^t|}\sum_{x\in\mathcal{X}_c^t} f\left(x\right)[0]$.

At task t, we have old classifiers $\boldsymbol{\varphi}_{old}$, previous classes C_{old}, and the old prototypes $\boldsymbol{\mu}_{old}$. We employ multi-head attention to determine the relation between the new classes C_{new} and the previously learned classes C_{old}. The process involves using new classes prototypes μ_{new} as the query and the old prototypes $\boldsymbol{\mu}_{old}$ as the key. The similarity between the query and past prototypes is measured to derive relational attention. This obtained attention is then used as guidance to initialize a new class classifiers as follows:

$$\varphi_{new} = \mathrm{PA} = \mathrm{Softmax}\left(\frac{d\left(\boldsymbol{\mu}_{new}, \boldsymbol{\mu}_{old}\right)}{\tau}\right)\boldsymbol{\varphi}_{old}, \qquad (10)$$

where the φ_{new} is the initialized classifier of a new class n, $d\left(\cdot\right)$ presents cosine similarity, and PA is short of Prototypes Attention. This attention allows for subspace-aware initialization, ensuring that future classes reside within the convex subspace of prior classes. When, extended to k multi-head operation (denoted by MPA), we have:

$$\begin{aligned}\varphi_{new} &= \mathrm{MPA}\left(\boldsymbol{\mu}_{new}, \boldsymbol{\mu}_{old}, \boldsymbol{\varphi}_{old}\right)\\ &= \mathrm{Concat}\left(\mathrm{PA}_1\left(\boldsymbol{\mu}_{new,1}, \boldsymbol{\mu}_{old,1}, \boldsymbol{\varphi}_{old,1}\right),\right. \qquad (11)\\ &\qquad \left. ..., \mathrm{PA}_k\left(\boldsymbol{\mu}_{new,k}, \boldsymbol{\mu}_{old,k}, \boldsymbol{\varphi}_{old,k}\right)\right).\end{aligned}$$

We share a comparable motivation with subspace regularization (Akyürek et al. 2022), but our work centers on subspace-aware initialization techniques instead of adding regularization penalties during training. Classifier initialization is also explored through bidirectional projection (Zhou, Ye, and chuan Zhan 2021), which involves finding the projector using OT. However, our approach differs as we utilize *prior* from prototypes derived from foundational vision model and employ a multi-head attention mechanism based on a similarity measure as the projector.

Experiments

Benchmark Protocols

Datasets and evaluation. Our evaluation encompasses Split CIFAR-100 (Krizhevsky 2009) and Split ImageNet-R (Hendrycks et al. 2021) for CIL, and CORe50 (Lomonaco and Maltoni 2017) for DIL, maintaining original sequential class order. We decompose each task into incremental steps (5, 10, and 20) for CIL. We benchmark against state-of-the-art prompt-based methods, assessing average accuracy (Lopez-Paz and Ranzato 2017), forgetting (Chaudhry et al. 2018), and final test accuracy for CORe50 DIL. Supplementary materials provide detailed dataset information.

Training details. We utilize ViT-B/16 (Dosovitskiy et al. 2021) architecture pre-trained on ImageNet-1K, featuring 12 Self-Attention blocks and 768 channel dimensions. Training involves 224 input sizes and a batch size of 64. Our approach uses Adam optimizer (Kingma and Ba 2015) with a constant learning rate (lr) of 0.003 for 5 epochs on CORe50, lr

Method	Split ImageNet-R							
	5 Steps		10 Steps		20 Steps		Avg	Avg
	Acc.(↑)	Forget.(↓)	Acc.(↑)	Forget.(↓)	Acc.(↑)	Forget.(↓)	Acc.(↑)	Forget.(↓)
FT-seq	61.41 ±0.38	5.76 ±0.48	50.28 ±2.29	24.28 ±1.73	39.25 ±0.90	40.38 ±0.77	50.31 (+0.00)	23.48 (−0.00)
LP-seq	59.83 ±0.33	**1.50 ±0.41**	55.30 ±0.12	7.85 ±0.10	51.97 ±0.34	13.87 ±0.21	53.64 (+3.33)	7.74 (−15.74)
NME-seq	61.06	6.64	61.40	**0.76**	61.76	2.89	61.41 (+11.10)	**3.43 (−20.05)**
L2P	66.63 ±0.33	6.65 ±0.38	64.05 ±0.39	10.05 ±0.26	60.34 ±0.17	14.44 ±0.61	63.67 (+13.36)	10.38 (−13.10)
DualPrompt	71.06 ±0.35	4.19 ±0.25	69.71 ±0.25	5.44 ±0.12	66.26 ±0.46	8.74 ±0.33	69.01 (+18.70)	6.12 (−17.36)
ESN	73.42 ±0.40	3.79 ±0.55	71.07 ±0.29	4.99 ±0.49	64.77 ±0.71	6.65 ±1.24	69.75 (+19.44)	5.14 (−18.34)
CODA-P-S	73.80 ±0.40	5.56 ±0.64	71.95 ±0.41	5.92 ±0.35	69.67 ±0.35	6.23 ±0.40	71.81 (+21.50)	5.90 (−17.58)
CODA-P	73.77 ±0.48	6.60 ±0.52	72.42 ±0.40	6.26 ±0.61	70.18 ±0.43	5.53 ±0.21	72.12 (+21.81)	6.13 (−17.35)
EvoPrompt-S	76.79 ±0.23	9.84 ±0.15	76.22 ±0.16	2.33 ±0.24	**74.68 ±0.51**	2.70 ±0.19	75.90 (+25.59)	4.96 (−18.52)
EvoPrompt	**77.16 ±0.18**	9.89 ±0.30	**76.83 ±0.08**	2.78 ±0.06	74.41 ±0.23	**2.56 ±0.22**	**76.13 (+25.82)**	5.08 (−18.40)
Upper-bound†	79.13 ±0.18	-	79.13 ±0.18	-	79.13 ±0.18	-	79.13	-

Table 2: Benchmark results on average accuracy and forgetting metrics on various steps of Split ImageNet-R in CIL. The last two columns present the mean accuracy and forgetting from multiple steps scenario.

Method	Test Acc. (↑)	Δ Acc. (↑)
NME-seq	78.20	+00.00
EWC †	74.82 ±0.60	−3.38
LwF †	75.45 ±0.40	−2.75
L2P†	78.33 ±0.06	+0.13
S-iPrompts‡	83.13 ±0.51	+4.93
S-liPrompts‡	89.06 ±0.86	+10.86
ESN‡	91.80 ±0.31	+13.60
EvoPrompt-S	94.77 ±0.50	+16.57
EvoPrompt	**95.27 ±0.15**	**+17.07**
Upper-bound	91.32 ±0.23	-

Table 3: Results on CORe50 in DIL. The label ‡ and † are derived from source paper and (Wang et al. 2022c), each.

0.05 for 20 epochs on Split CIFAR-100 and 50 epochs on ImageNet-R. Unlike previous work that neglected the cost of the stored model (Yan, Xie, and He 2021; Douillard et al. 2022) and focused solely on the final parameter count (Wang et al. 2022b), we introduce a lighter variant, EvoPrompt-S, adhering to the *strict realistic* setting (Zhou et al. 2023), accommodating additional RPM storage requirements.

Evaluation Results

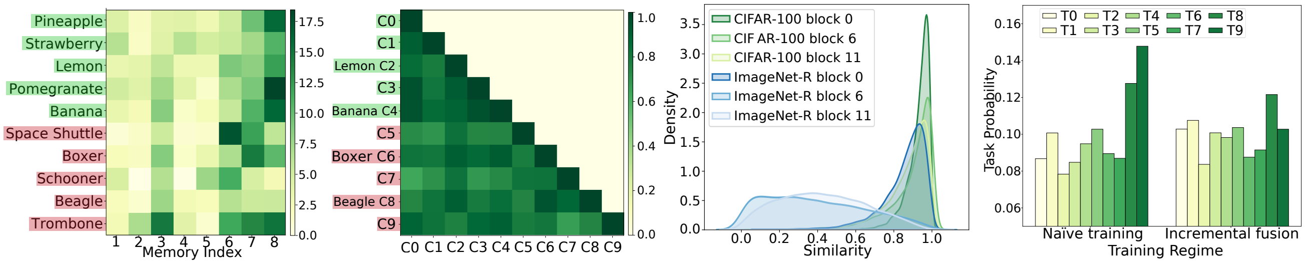
Quantitative results on CIL. Table 1 and 2 present evaluation results for Split CIFAR-100 and Split ImageNet-R, respectively. EvoPrompt significantly outperforms other methods on both datasets, including the latest prompt-based approaches. In CIFAR-100 split, EvoPrompt-S achieves an average accuracy of **87.20%** across multiple steps, close to our normal version (with greater prompt length), with a forgetting rate of only **5.24%**, lower than other methods. Compared to the leading SOTA, DualPrompt, EvoPrompt achieves **0.8%** higher accuracy (**86.40%→87.20%**).

In the more challenging Split ImageNet-R dataset, with higher resolution and greater diversity, our exceptional performance shines, achieving an average accuracy of **76.13%**. We outperform CODA-P by **4.01%** improvement and exhibit the lowest forgetting rate of **5.08%**, despite having significantly fewer parameters, with 5× and 13× smaller than CODA-P for EvoPrompt and EvoPrompt-S, respectively (see supplementary). Our approach particularly excels with a larger number of tasks, as we implicitly contrast with old classifiers. On the contrary, other prompt-based methods that address tasks in isolation experience performance degradation with fewer classes per task. This underscores the effectiveness of our approach in realistic continual learning scenarios with high-resolution datasets and a large number of tasks.

Quantitative results on DIL. Our results on DIL are reported in Table 3. Not only evaluating CL performance, CORe50 implicitly benchmarks the generalization capability to unfamiliar domains. The report clearly shows that EvoPrompt surpasses the other methods by a considerable gap. Specifically, our approach achieves an accuracy of **95.27%**, outperforming ESN and S-liPrompts by **3.47** and **6.21%** in terms of accuracy, respectively. Even our lighter version surpasses the S-liPrompts with CLIP by **5.71%**. Additionally, EvoPrompt demonstrated a **3.95%** improvement over the upper-bound. This is because the CORe50 test set includes new domains that are not present during training. The upper-bound tends to overfit to the training set, while our approach, involving prompt-based training without tuning the pre-trained backbone, exhibits better generalization.

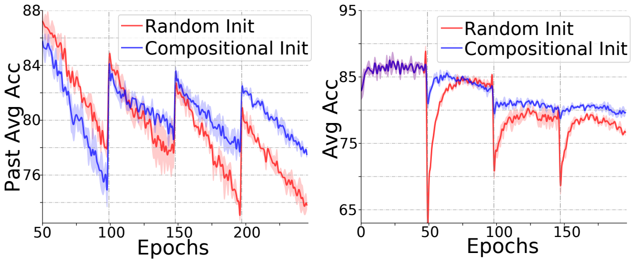
Analysis

Prompt assignment. To assess our prompting shareability and diversity, we compute prompt coefficients for distinct classes based on activated hidden nodes (Fig.3(a)) and interclass coefficient similarity (Fig.3(b)). Taking the "pineapple" class as an example, we select the top-5 most similar (green banner) and dissimilar classes (red banner), then compare their coefficients. Fig.3(a) reveals a clear pattern of semantic coherence in the top-5 similar and dissimilar classes. Similar classes exhibit higher coefficients, enhancing shareability (Fig.3(b)). Abstraction levels influence this



(a) Prompt coefficient (b) Coefficient similarity (c) Prompt assignment diversity (d) Prediction probability

Figure 3: Analysis of (a) prompt memory coefficient of top-5 most similar and dissimilar to class "pineapple", on intermediate block; (b) prompt coefficient similarity among classes in (a); (c) inter-instance prompt coefficient similarity distribution, on different level of abstraction; (d) model prediction probability on the utilization or absence of incremental fusion.



(a) Past average accuracy (b) Global average accuracy

Figure 4: Accuracy dynamic of initial four tasks on (a) past tasks accuracy and (b) global accuracy on all tasks so far.

Core Components	Avg. Acc(↑)	Forgetting(↓)
FT-Seq (Baseline)	50.28 ±2.29	24.28 ±1.73
+ FFN prompt	68.43 ±0.58	5.88 ±0.53
+ incremental fusion	72.44 ±0.57	1.05 ±0.23
+ attribution-aware	73.16 ±0.38	**0.89 ±0.53**
+ OT alignment	73.51 ±0.18	1.10 ±0.17
+ CCI	**76.83 ±0.08**	2.78 ±0.06

Table 4: The ablation studies for each component contribution evaluated on 10 steps Split ImageNet-R.

behavior, with the last layer showing greater uniformity. Simultaneously, our similarity density analysis in Fig. 3(c) confirms variable prompt selection before the penultimate layers, becoming more consistent at deeper levels. This variance could result from earlier-layer prompting compensating for feature suppression by the original pre-trained model. Dataset characteristics significantly impact diverse coefficients; higher diversity datasets like Split ImageNet-R yield more diverse coefficients than Split CIFAR-100. Our approach adeptly addresses diverse dataset challenges.

Incremental fusion and recency bias. Average probabilities for each task on Split ImageNet-R are calculated to analyze the bias, following (Buzzega et al. 2020). Unlike naïve training that has bias issues, our approach of integrating incremental fusion at the task level results in a notable improvement in the equilibrium of task probabilities, as illustrated in Fig. 3(d). This highlights the crucial role played by incremental fusion in mitigating recency bias.

Stability gap. A *stability gap* (Lange, van de Ven, and Tuytelaars 2023) arises when the learning system abruptly and temporarily forgets previously learned information at task transition. We confirmed our effectiveness by periodic evaluation of its accuracy on past classes and global encountered classes. As depicted in Figure 4, our method demonstrates stable performance, smooth transitions between tasks, and faster acquisition of current knowledge compared to random initialization. As learning progresses, our approach also experiences some incremental decrease in performance, but it exhibits significant improvements in learning stability by avoiding massive forgetting and performance degradation. The utilization of CCI imparts a bias towards prior knowledge, thereby curbing substantial losses and abrupt gradient changes during task transitions.

Ablation Study. As shown in Table 4, our simple approach using FFN parameterization effectively attains a competitive baseline performance of **68.43%** accuracy. By combining attribution-aware fusion and WPM alignment, a substantial gain of around ∼**1%** is achieved, highlighting the significance of this approach. Our final performance gains are largely influenced by the incorporation of prompt memory fusion and compositional classifier initialization to improve **4.01%** and **3.32%** accuracy, respectively.

Conclusion

This paper presents EvoPrompt, a prompt-based approach that employs continually evolved parameterized memory prompt with continuous bottleneck using FFN and attribution-aware incremental prompt fusion, which facilitates the sharing and adaptability during prompting. Maximizing learned knowledge is achieved through the introduction of compositional classifier initialization, enhancing both learning stability and backward compatibility. Our framework scales to multiple steps scenarios and datasets with high intra-diversity, such as Split ImageNet-R and CORe50, proving the generalization capability introduced by our proposed method. Comprehensive experiments exhibit superior performance compared to the state-of-the-art.

Acknowledgments

This work was supported by the National Key Research and Development Project of China (2020AAA0105600), the National Natural Science Foundation of China (62376209, U21B2048, 6230070870, 62206271), and the Shenzhen Key Technical Projects (JSGG20220831105801004, CJGJZD2022051714160501, JCYJ20220818101406014), and sponsored by the CAAI-MindSpore Open Fund, developed on OpenI Community.

References

Ahn, H.; Kwak, J.; Lim, S. F.; Bang, H.; Kim, H.; and Moon, T. 2020. SS-IL: Separated Softmax for Incremental Learning. *Proceedings of the IEEE International Conference on Computer Vision*, 824–833.

Ainsworth, S.; Hayase, J.; and Srinivasa, S. 2023. Git Re-Basin: Merging Models modulo Permutation Symmetries. In *International Conference on Learning Representations*.

Akyürek, A. F.; Akyürek, E.; Wijaya, D.; and Andreas, J. 2022. Subspace Regularizers for Few-Shot Class Incremental Learning. In *International Conference on Learning Representations*.

Aljundi, R.; Chakravarty, P.; and Tuytelaars, T. 2017. Expert Gate: Lifelong Learning with a Network of Experts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Arani, E.; Sarfraz, F.; and Zonooz, B. 2022. Learning Fast, Learning Slow: A General Continual Learning Method based on Complementary Learning System. In *International Conference on Learning Representations*.

Buzzega, P.; Boschini, M.; Porrello, A.; Abati, D.; and Calderara, S. 2020. Dark Experience for General Continual Learning: a Strong, Simple Baseline. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc.

Chaudhry, A.; Dokania, P. K.; Ajanthan, T.; and Torr, P. H. S. 2018. Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence. In *European Conference on Computer Vision*.

Chaudhry, A.; Rohrbach, M.; Elhoseiny, M.; Ajanthan, T.; Dokania, P. K.; Torr, P. H.; and Ranzato, M. 2019. Continual learning with tiny episodic memories.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

Douillard, A.; Ramé, A.; Couairon, G.; and 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*.

Frankle, J.; Dziugaite, G. K.; Roy, D. M.; and Carbin, M. 2019. Linear Mode Connectivity and the Lottery Ticket Hypothesis. In *International Conference on Machine Learning*.

Geva, M.; Schuster, R.; Berant, J.; and Levy, O. 2021. Transformer Feed-Forward Layers Are Key-Value Memories. In *Empirical Methods in Natural Language Processing*.

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 *Proceedings of the IEEE International Conference on Computer Vision*.

Jia, M.; Tang, L.; Chen, B.-C.; Cardie, C.; Belongie, S.; Hariharan, B.; and Lim, S.-N. 2022. Visual Prompt Tuning. In *European Conference on Computer Vision*.

Kantorovich, L. 2006. On the Translocation of Masses. In *Journal of Mathematical Sciences*, volume 133, 1381–1382.

Kemker, R.; and Kanan, C. 2018. FearNet: Brain-Inspired Model for Incremental Learning. In *International Conference on Learning Representations*.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; Hassabis, D.; Clopath, C.; Kumaran, D.; and Hadsell, R. 2017. Overcoming catastrophic forgetting in neural networks. In *Proceedings of the National Academy of Sciences*, volume 114, 3521–3526.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images.

Lange, M. D.; van de Ven, G. M.; and Tuytelaars, T. 2023. Continual evaluation for lifelong learning: Identifying the stability gap. In *International Conference on Learning Representations*.

Lee, K.-Y.; Zhong, Y.; and Wang, Y.-X. 2023. Do Pre-trained Models Benefit Equally in Continual Learning? In *IEEE/CVF Winter Conference on Applications of Computer Vision*, 6474–6482.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Lomonaco, V.; and Maltoni, D. 2017. CORe50: a New Dataset and Benchmark for Continuous Object Recognition. In *CoRL*.

Lopez-Paz, D.; and Ranzato, M. 2017. Gradient Episodic Memory for Continual Learning. In *Advances in Neural Information Processing Systems*.

McCloskey, M.; and Cohen, N. J. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*. Elsevier.

Mermillod, M.; Bugaiska, A.; and Bonin, P. 2013. The stability-plasticity dilemma: investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in Psychology*, 4.

Mirzadeh, S. I.; Farajtabar, M.; Gorur, D.; Pascanu, R.; and Ghasemzadeh, H. 2021. Linear Mode Connectivity in Multitask and Continual Learning. In *International Conference on Learning Representations*, volume abs/2010.04495.

Misra, I.; Shrivastava, A.; Gupta, A.; and Hebert, M. 2016. Cross-Stitch Networks for Multi-Task Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Mitchell, T. M.; Cohen, W. W.; Hruschka, E.; Talukdar, P. P.; Yang, B.; Betteridge, J.; Carlson, A.; Dalvi, B.; Gardner, M.; Kisiel, B.; Krishnamurthy, J.; Lao, N.; Mazaitis, K.; Mohamed, T.; Nakashole, N.; Platanios, E. A.; Ritter, A.; Samadi, M.; Settles, B.; Wang, R. C.; Wijaya, D.; Gupta, A. K.; Chen, X.; Saparov, A.; Greaves, M.; and Welling, J. 2015. Never-Ending Learning. volume 61, 103 – 115.

Pham, Q.; Liu, C.; and Hoi, S. 2021. DualNet: Continual Learning, Fast and Slow. In *Advances in Neural Information Processing Systems*, volume 34.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning*.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. iCaRL: Incremental Classifier and Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Shin, H.; Lee, J. K.; Kim, J.; and Kim, J. 2017. Continual Learning with Deep Generative Replay. In *Advances in Neural Information Processing Systems*. Curran Associates Inc.

Singh, S. P.; and Jaggi, M. 2020. Model fusion via optimal transport. In *Advances in Neural Information Processing Systems*, volume 33.

Sukhbaatar, S.; Szlam, A. D.; Weston, J.; and Fergus, R. 2015. End-To-End Memory Networks. In *Advances in Neural Information Processing Systems*.

Tao, X.; Chang, X.; Hong, X.; Wei, X.; and Gong, Y. 2020. Topology-Preserving Class-Incremental Learning. In *European Conference on Computer Vision*.

Tatro, N.; Chen, P.-Y.; Das, P.; Melnyk, I.; Sattigeri, P.; and Lai, R. 2020. Optimizing Mode Connectivity via Neuron Alignment. In *Advances in Neural Information Processing Systems*, volume 33.

Tversky, A.; and Kahneman, D. 1975. *Judgment under Uncertainty: Heuristics and Biases*, 141–162. Dordrecht: Springer Netherlands. ISBN 978-94-010-1834-0.

Wang, F. L.; Zhou, D.-W.; Ye, H.-J.; and Zhan, D.-C. 2022a. FOSTER: Feature Boosting and Compression for Class-Incremental Learning. In *European Conference on Computer Vision*.

Wang, Y.; Huang, Z.; and Hong, X. 2022. S-Prompts Learning with Pre-trained Transformers: An Occam's Razor for Domain Incremental Learning. In *Advances in Neural Information Processing Systems*.

Wang, Z.; Zhang, Z.; Ebrahimi, S.; Sun, R.; Zhang, H.; Lee, C.-Y.; Ren, X.; Su, G.; Perot, V.; Dy, J.; and Pfister, T. 2022b. DualPrompt: Complementary Prompting for Rehearsal-Free Continual Learning. In *European Conference on Computer Vision*. Springer Nature Switzerland.

Wang, Z.; Zhang, Z.; Lee, C.-Y.; Zhang, H.; Sun, R.; Ren, X.; Su, G.; Perot, V.; Dy, J.; and Pfister, T. 2022c. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Wei, X.; Cao, A.; Yang, F.; and Ma, Z. 2023. Sparse Parameterization for Epitomic Dataset Distillation. In *Advances in Neural Information Processing Systems*.

Wu, Y.; Chen, Y.; Wang, L.; Ye, Y.; Liu, Z.; Guo, Y.; and Fu, Y. 2019. Large Scale Incremental Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Yan, S.; Xie, J.; and He, X. 2021. DER: Dynamically Expandable Representation for Class Incremental Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society.

Yu, L.; Twardowski, B.; Liu, X.; Herranz, L.; Wang, K.; Cheng, Y.; Jui, S.; and Weijer, J. v. d. 2020. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Zhou, D.-W.; Wang, Q.-W.; Ye, H.-J.; and Zhan, D.-C. 2023. A Model or 603 Exemplars: Towards Memory-Efficient Class-Incremental Learning. In *International Conference on Learning Representations*.

Zhou, D.-W.; Ye, H.-J.; and chuan Zhan, D. 2021. Co-Transport for Class-Incremental Learning. In *Proceedings of the 29th ACM International Conference on Multimedia*.

Zhu, F.; Zhang, X.-Y.; Wang, C.; Yin, F.; and Liu, C.-L. 2021. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.