Multimodal Parameter-Efficient Few-Shot Class Incremental Learning

Marco D'Alessandro[1*] Alberto Alonso[1*]
Enrique Calabrés[1,2] Mikel Galar[2]

[1]Neuraptic AI [2]Public University of Navarra
{marco.dalessandro, alberto.alonso, enrique.hernandez}@neuraptic.ai
mikel.galar@unavarra.es

Abstract

Few-Shot Class Incremental Learning (FSCIL) is a challenging continual learning task, where limited training examples are available during several learning sessions. To succeed in this task, it is necessary to avoid over-fitting new classes caused by biased distributions in the few-shot training sets. The general approach to address this issue involves enhancing the representational capability of a predefined backbone architecture by adding special modules for backward compatibility with older classes. However, this approach has not yet solved the dilemma of ensuring high classification accuracy over time while reducing the gap between the performance obtained on larger training sets and the smaller ones. In this work, we propose an alternative approach called Continual Parameter-Efficient CLIP (CPE-CLIP) to reduce the loss of information between different learning sessions. Instead of adapting additional modules to address information loss, we leverage the vast knowledge acquired by CLIP in large-scale pre-training and its effectiveness in generalizing to new concepts. Our approach is multimodal and parameter-efficient, relying on learnable prompts for both the language and vision encoders to enable transfer learning across sessions. We also introduce prompt regularization to improve performance and prevent forgetting. Our experimental results demonstrate that CPE-CLIP significantly improves FSCIL performance compared to state-of-the-art proposals while also drastically reducing the number of learnable parameters and training costs.

1. Introduction

Deploying ML systems in a dynamic environment requires accounting for continuous data streams arriving over time. This environment may experience shifts in data distribution or the addition of new classes. An ideal learning system must be able to learn new incoming classes while maintaining its discriminability over previously learned classes, thus avoiding catastrophic forgetting [31]. This continual learning problem formulation is known as Class-Incremental Learning (CIL), which requires dealing with the stability-plasticity dilemma [32, 14], *i.e.*, the trade-off between learning new classes and retaining old ones. In this work, we focus on a special case of CIL, named Few-Shot Class Incremental Learning (FSCIL, [44]), where only a few training examples are available at every learning session. Here, the additional challenge consists in avoiding over-fitting on new incoming classes caused by biased distributions in the few-shot training sets. This problem is particularly crucial in practical, real-world scenarios where data availability is limited. Examples of such scenarios include manufacturing settings [59, 2] and medical imaging [18]. In manufacturing, robots are deployed to carry out a diverse range of tasks, such as assembling or grasping objects. To perform these tasks, robots may need to adapt to new objects or materials, which may have a limited amount of training data available. In medical imaging the availability of data may also be limited due to the high costs of data collection and patient privacy, making it difficult to acquire new knowledge over time.

Recent research has focused on solving these problems through various approaches, such as meta-learning [57, 34], regularization techniques [30], or knowledge distillation [38, 6, 62]. These methods have shown promising results in achieving incremental learning over time with a limited amount of data available. The general approaches consist in enhancing the basic representational capability of a predefined backbone architecture by adding special modules to entail backward compatibility with older classes during learning sessions. These solutions are computationally expensive since they need a large number of iterations in each session to adapt the additional modules to new classes while maintaining backward compatibility. Despite the high computational cost, they still fail to efficiently reduce the gap between the performance obtained on larger training sets

[*]Equal contribution.

and the one obtained on smaller sets over time, which still remains an unsolved dilemma [67, 25].

In this work, we propose Continual Parameter-Efficient CLIP (CPE-CLIP) as an alternative approach to reduce the loss of information between different learning sessions. Inspired by the astounding continual learning performance obtained in a zero-shot setting [45], we use CLIP [36] as a starting point to build a continual learning system for FSCIL. Instead of relying on adapting additional modules to address information loss, we propose to adapt the CLIP architecture with lightweight learnable prompts for few-show image classification. In this way, we are able to take advantage of the vast amount of knowledge acquired by CLIP in large-scale pretraining and its inherent effectiveness in generalizing to new concepts. Notably, this is a Multimodal and Parameter-Efficient approach as it relies on learnable prompts, rather than finetunig, of both the language and vision encoders to allow transfer learning across sessions over time. We show that our approach significantly improves FSCIL performance compared to state-of-the-art proposals while also drastically reducing the number of learnable parameters and training costs. We also conduct extensive hyperparameter tuning and ablation studies to understand the functional properties of the multiple components of our model. Our main contributions can be summarized as follows:

- We propose a prompt-based approach to adapt the CLIP architecture for solving continual learning tasks in few-shot settings by reducing forgetting and supporting knowledge transfer over time, all while learning less than 0.3% of the total parameters.
- We combine two different prompt attachment methods with prompt regularization to smoothly transition to future tasks while maintaining constant performance over time.
- We achieve state-of-the-art performance on three popular benchmark datasets for FSCIL, and exceed previous state-of-the-art results by a great margin.

The rest of this paper is organized as follows. We discuss related works about our methodological approach and few-shot class-incremental learning in Section 2. Section 3 introduces the problem formulation. The proposed method is presented in Section 4. Moreover, we present our experimental results and final considerations in Sections 5 and 6, respectively.

2. Related Work

Our approach to the few-shot continual learning problem is related to several topics, so we introduce them separately.

2.1. Few-Shot Image Classification

Few-shot image classification aims to fit new unseen classes with an insufficient number of training examples [5, 50]. Several learning methods have been proposed for this purpose. For instance, in metric-based approaches, different network branches are built to classify images by calculating the distance between a query image from the test set and the training images in the few-shot training set [42, 48, 43]. Differently, in meta-learning, models are trained on a variety of learning tasks, such that they can solve new learning tasks using only a small number of training samples [11, 10, 57, 34]. A rather different and more recent perspective relies on pretrained multimodal vision-language models to classify images from labeled captions with minimal training examples [65, 64, 19]. In this approach, few-shot learning is based on the correct match between the query image and a text caption describing the category label. Our method can be seen as a continual learning adaptation of the latter approach.

2.2. Incremental Learning

Incremental learning deals with the problem of learning new information from non-stationary data streams [46, 28, 35]. According to the availability of task identifiers (IDs) over time, the problem formulation can pertain to either task- or class-incremental learning. There are several solutions in the literature that try to face these tasks by enabling learning of incoming information from new tasks while minimizing forgetting of previously acquired knowledge. In regularization-based methods, specific parameters are regularized for learned tasks in order to retain knowledge acquired on previous ones and avoid catastrophic forgetting [20, 24, 60]. Architecture-based methods assign an isolated portion of the backbone, or isolated parameters of additional branches, to each task [40, 58, 29, 51, 9]. In rehearsal-based methods, data from previously learned tasks are stored in a rehearsal buffer and used in the current task in addition to the current training set [3, 4, 33]. A more recent prompt-based rehearsal-free approach combines powerful pretrained backbones with learnable prompts that retain the knowledge acquired from the different tasks without modifying the weights of the main backbone, thus avoiding forgetting [52, 53, 41, 37]. Our method gets inspiration from the solutions proposed in the latter methods.

2.3. Few-Shot Class-Incremental Learning

FSCIL is a recent research topic proposed to tackle few-shots training inputs in a class-incremental setting [44], where task ID is not provided during evaluation. The general task is to initially learn from a number of base classes and then continuously update the model to represent new incoming classes. The main challenge of this setting is to avoid overfitting to new class few-shot samples. The first attempt to solve this issue proposed the *neural gas* (NG) network for representing knowledge [44], where feature space topologies were learned for different classes, and new classes were represented by growing and adapting the network's topology. In [61], authors decoupled learning representations and classifiers, by letting the latter be updated over time by means of a graph model propagating information between classifiers. Prototype modeling was also used to assign prototypes in the embedding space to reserve it for future incoming classes [62], or to use the average of new class embedding representations as a class prototype to replace classifiers [63]. Different methods addressed the problem by synthesizing features into a mixture of sub-spaces for incremental classes by using a VAE [7], or by adapting general deep learning architectures to enable a few parameters to be updated for every new set of novel incoming classes [30]. More recent approaches tried to combine features emerging from supervised and self-supervised models for boosting classifiers [1], or to calibrate distributions to avoid forgetting by retrieving distributions for old classes while estimating distributions for new classes [26].

2.4. Parameter-Efficient Learning

The most common way to adapt foundational large general-purpose pretrained models to downstream tasks is to finetune all the model parameters, which results in high computational costs and memory usage, and the need to store several copies of the finetuned model for different tasks. A lightweight alternative came from the parameter-efficient learning literature that proposed to update only a small number of extra parameters while keeping backbone parameters frozen [12, 22]. Several methods have been proposed to flexibly adapt pretrained backbones to different downstream tasks according to this logic. Adapter-tuning [15, 16] interleaves transformer layers with a feed-forward bottleneck module with skip-connection to adapt the layer's output before passing to the next layer. Prefix-tuning [23, 52, 17] prepends tunable prefix vectors as learnable embeddings to the keys and values of the multi-head attention layers in transformers [47]. In prompt-tuning [22, 53], a set of learnable embeddings is prepended to the input embeddings from the first layer, and the augmented input is then normally processed by the frozen transformer layers.

3. Problem Formulation

The FSCIL setting [44] can be defined as follows. We consider a stream of labelled training sets $D_0, D_1, \ldots, D_T$, where $D_t = \{(\boldsymbol{x}_{i,t}, y_{i,t})\}_{i=1}^{N_t^D}$, N_t^D is the number of training examples provided at session t, and T is the last incremental session. D_0 identifies the large-scale training set of base classes, and D_t is the few-shot training set of new classes, for $t > 0$. The base class dataset, D_0, is meant to have a sufficient number of training examples. On the contrary, insufficient training sets are provided for new classes. Consider the set of class labels C_t belonging to train set D_t. FSCIL has the following requirements: (1) classes don't overlap among sessions, $\forall t, \tau, t \neq \tau, C_t \cap C_\tau = \varnothing$, (2) base class set is bigger than new class sets, where $|C_0| > |C_t|$, and $N_0^D > N_t^D$, hold for $t > 0$, (3) new class sets have the same size, such that $\forall t, \tau,\ t \neq \tau,\ |C_t| = |C_\tau|$ and $N_t^D = N_\tau^D$ hold for $t, \tau > 0$. For the evaluation phase, the only requirement is that session-wise performances are computed by considering all the classes encountered up to the current session t. Consider the stream of labelled evaluation sets $E_0, E_1, \ldots, E_T$, and a model f, than the evaluation accuracy for session t can be computed as follows:

$$A_t = \frac{\sum_{(\boldsymbol{x}_i, y_i) \in E_0 \cup E_1 \cup \ldots E_t} [f(\boldsymbol{x}_i) = y_i]}{N_0^E + N_1^E + \ldots + N_t^E} \tag{1}$$

where N_τ^E is the number of evaluation examples for session τ, and $[\cdot]$ the indicator function.

4. Method

Our proposed method, CPE-CLIP, is summarized in Figure 1. The approach involves the use of the CLIP foundational vision-language model [36] as the primary building block of a continual learning system. CLIP is a neural network trained to align the modalities of language and vision, and to leverage the abundant supervision offered by natural language to reason about visual concepts. In our work, we exploit the capabilities of CLIP to cast image classification as a multimodal task, where text prompts (*e.g.* "a photo of a <category>") are used as query captions for the text encoder, and the matching of a test image to the caption serves as the classification criterion. Our approach learns prompts that are adapted to both the language and vision encoders [19]. By doing so, we maintain the knowledge acquired during pretraining by freezing the CLIP backbone while allowing the prompts to solve the continual learning task. To further enhance performance and avoid forgetting, we introduce prompt regularization.

Language Encoder. Here we learn language context prompts that are shared among all the classes of a given downstream task. Context prompts fulfill two purposes: (1) they prevent manually selecting inefficient prompts [65, 64] to provide the textual representation of an image, assuming the category label of that image is available, (2) they serve as stability parameters that learn *general* task-invariant properties shared among the session tasks. We then introduce L learnable tokens, $\boldsymbol{g}$, called G-Prompt (by following notation in Wang *et al.* [52]), such that $\boldsymbol{\Theta_g} \in \mathbb{R}^{L \times d^{\text{NLP}}}$, where d^{NLP} is the embedding dimension of CLIP language encoder. The input embeddings now follow the form $[g_1, g_2, \ldots, g_L, w]$ =

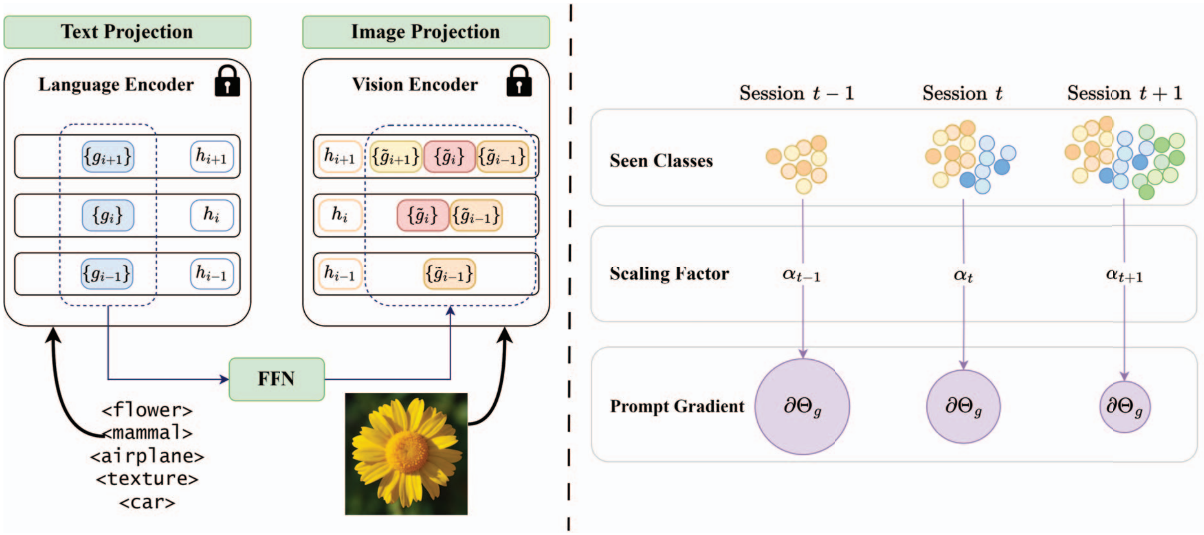

Figure 1. Summary of CPE-CLIP architecture and training process. The picture on the left describes the general structure of CPE-CLIP, where the G-Prompt contributes to generalizing task-invariant knowledge on the language encoder and is then projected to the vision encoder. Vision prompts are accumulated across subsequent layers, while a replacement strategy is used for the language encoder. The image on the right depicts the regularization process where an increase in the number of seen classes reduces parameter gradients by means of the scaling factor α_t, for a given session t.

$[\boldsymbol{g}, w]$, and w is the embedding for the category name of the input image. Let's define f_i^{NLP} as the i^{th} transformer layer of the language encoder, where $i = 1, 2, \ldots, K$ and K is the total number of layers. We admit the case where new learnable tokens can be further introduced in each language encoder layer, f_i^{NLP}, up to a certain depth D. In this way, different prompts are used independently across different layers to account for different levels of abstraction. The forward pass can then be described as:

$$[\_, h_1] = f_1^{\text{NLP}}([\boldsymbol{g}_1, w]) \tag{2}$$

for the first layer, and

$$[\_, h_i] = f_i^{\text{NLP}}([\boldsymbol{g}_{i-1}, h_{i-1}]) \qquad i = 2, 3, 4, \ldots, D \tag{3}$$

for subsequent layers up to layer D. Here, $[\cdot, \cdot]$ refers to vertical concatenation, h_i is the latent state output from layer i associated with the class category word embedding, and $\boldsymbol{g}_i$ represents the set of learnable prompts for layer i. In this phase, the output embeddings for the input prompt $\boldsymbol{g}_i$ are discarded for the next layer. Notably, when layer-specific prompts are introduced we have $\boldsymbol{\Theta_g} \in \mathbb{R}^{D \times L \times d^{\text{NLP}}}$. After D^{th} transformer layer, subsequent layers process previous output layers in a standard way until the final text representation:

$$[\boldsymbol{g}_i, h_i] = f_i^{\text{NLP}}([\boldsymbol{g}_{i-1}, h_{i-1}]) \qquad i = D+1, \ldots, K. \tag{4}$$

For simplicity, we refer to the last hidden state for the [EOS] token as h^{NLP}, which in the CLIP language encoder is used to represent the whole sentence. The hidden state is then projected to a lower dimensional space:

$$h_*^{NLP} = p^{\text{NLP}}(h^{\text{NLP}}), \tag{5}$$

where p^{NLP} is a linear projection layer, and h_*^{NLP} is the final low-dimensional vector for the text representation.

Vision Encoder. As for the language encoder, we conceive G-Prompt for the vision branch. Even in this case, prompts are concatenated with image patch embeddings across several layers of the hierarchy in order to interact with lower and higher-level image feature processing. We introduce L tokens, $\tilde{\boldsymbol{g}}$, such that $\tilde{\boldsymbol{g}} \in \mathbb{R}^{L \times d^{\text{CV}}}$, where d^{CV} is the embedding dimension of CLIP vision encoder, and $d^{\text{CV}} > d^{\text{NLP}}$. The input embeddings now follow the form $[c_1, c_2, \ldots, c_J, \tilde{g}_1, \tilde{g}_2, \ldots, \tilde{g}_L] = [\boldsymbol{c}, \tilde{\boldsymbol{g}}]$, where $\boldsymbol{c}$ is the embedded patches set of the input image plus the additional [CLS] token, and J the total number of embeddings. Let's now define f_i^{CV} as the i^{th} transformer layer of the vision encoder, where $i = 1, 2, \ldots, K$. Similar to Khattak *et al.* [19], we bridge the gap between language and vision prompts by explicitly expressing the latter as a function of the former. We introduce a learnable linear projection f^{PROJ}, $\boldsymbol{\Theta}_{f^{\text{PROJ}}} \in \mathbb{R}^{d^{\text{NLP}} \times d^{\text{CV}}}$, and constraint vision task-invariant prompts to be conditioned on language G-Prompt, such that $\tilde{\boldsymbol{g}}_i = f^{\text{PROJ}}(\boldsymbol{g}_i)$, and $\tilde{\boldsymbol{g}}_i$ is the set of prompts for vision encoder layer i. Although prompts in the vision branch are derived from language context prompts, we found it beneficial to use a different strategy for propagating prompts across the layers hierarchy. Here, we propose an *accumulation* method, as an alternative to the *replacement* method used in the language branch, where prompts in different layers are not independent anymore since they can interact with the processed output embeddings of prompts from previous layers. As usual, prompt accumulation takes place up to depth D. Formally, we describe the forward pass in the vision encoder as:

$$[h_1, \tilde{\boldsymbol{g}}_1] = f_1^{\text{CV}}([\boldsymbol{c}, \tilde{\boldsymbol{g}}_1]) \tag{6}$$

for the first layer, and

$$\begin{aligned} \tilde{\boldsymbol{g}}_i &= [\tilde{\boldsymbol{g}}_i, \tilde{\boldsymbol{g}}_{i-1}] \\ [h_i, \tilde{\boldsymbol{g}}_i] &= f_i^{\text{CV}}([\boldsymbol{c}, \tilde{\boldsymbol{g}}_i]) \end{aligned} \qquad i = 2, 3, 4, \ldots, D \tag{7}$$

for subsequent layers up to layer D. After D^{th} transformer layer, prompts are not accumulated anymore, and subsequent layers process previous output layers in a standard way until the final image representation:

$$[\tilde{\boldsymbol{g}}_i, h_i] = f_i^{\text{CV}}([\tilde{\boldsymbol{g}}_{i-1}, h_{i-1}]) \qquad i = D+1, \ldots, K \tag{8}$$

where now $\tilde{\boldsymbol{g}}_i$ is the final pooled prompt such that $\tilde{\boldsymbol{g}}_i \in \mathbb{R}^{LD \times d^{\text{CV}}}$. We refer to the last hidden state related to the [CLS] token as h^{CV}, which in the CLIP vision encoder is used to represent the whole image. The hidden state is then projected to a lower dimensional space:

$$h_*^{\text{CV}} = p^{\text{CV}}(h^{\text{CV}}) \tag{9}$$

where p^{CV} is a linear projection layer, and h_*^{CV} is the final low-dimensional vector for the image representation.

Multimodal Classification. The prediction probability for every given input image x to be classified as belonging to class z, $z = 1, 2, \ldots, Z$, is computed as:

$$p(y = z|x) = \frac{\exp[\rho(h_*^{\text{CV}}, h_{*z}^{\text{NLP}})]}{\sum_{j=1}^{Z} \exp[\rho(h_*^{\text{CV}}, h_{*j}^{\text{NLP}})]} \tag{10}$$

where ρ is the *cosine similarity*, h_*^{CV} the projected representation of image x, and h_{*z}^{NLP} is the projected representation of the sentence with the category name of z^{th} class in the training set.

Prompt Regularization. In FSCIL, base class training is crucial to initially tune the network to boost generalization to novel classes. In our case, the G-Prompt introduces a set of tokens to fulfill this purpose. Such tokens provide an efficient text representation that can be matched with an image in order to correctly classify the latter as belonging to the correct (semantic) category/label. However, the base class set provides a greater chance for generalization compared to session-related class sets, since it provides a greater number of classes and training examples. For this reason, we propose a mechanism to preserve knowledge proportionally to the number of classes encountered in different sessions. We define a scaling factor α_t for a given session t, $t = 1, 2, \ldots, T$, that affects the updating rate of G-Prompt parameters when training on session t:

$$\alpha_t = \frac{|C_t|}{\sum_{\tau=0}^{t} |C_\tau|}. \tag{11}$$

We then apply regularization as follows:

$$\frac{\partial \mathcal{L}_t}{\partial \boldsymbol{\Theta}_{\boldsymbol{g}}} \leftarrow \alpha_t \frac{\partial \mathcal{L}_t}{\partial \boldsymbol{\Theta}_{\boldsymbol{g}}} \tag{12}$$

and

$$\frac{\partial \mathcal{L}_t}{\partial \boldsymbol{\Theta}_{f^{\text{PROJ}}}} \leftarrow \alpha_t \frac{\partial \mathcal{L}_t}{\partial \boldsymbol{\Theta}_{f^{\text{PROJ}}}} \tag{13}$$

where $\mathcal{L}_t$ is the loss function for the classification task in session t. Such a regularization allows the G-Prompt, as well as the language-vision prompt projection, to be updated less consistently as the number of total seen classes increases.

5. Experiment

We evaluate CPE-CLIP on the three benchmarks [44] that provide the main baseline for model comparison in the FSCIL literature. Benchmarks include CIFAR100 [21], *mini*ImageNet [39], and CUB200-2011 [49].

5.1. Evaluation Benchmarks

For all the benchmarks we follow the split proposed by Tao *et al.* [44] since they provide the standard for all the proposals in the literature and ensure a fair model comparison. Benchmarks are described as follows:

- **CIFAR100** The dataset contains 60.000 32 × 32 RGB images from 100 classes. We use 60 classes as the base class set. The remaining 40 classes are split into 8 sessions where each session contains 5 new classes, and the few-shot training set consists of 5 examples per class (5-way 5-shot incremental task).

- ***mini*ImageNet** The dataset contains 60.000 84 × 84 RGB images. We use 100 classes as the base class set. The remaining 40 classes are split into 8 sessions of 5 few-shot training examples each (5-way 5-shot incremental task).

- **Caltech-UCSD Birds-200-2011 (CUB200)** The dataset contains 11.788 finegrained 224×224 RGB images from 200 classes of bird species. We use 100 classes as the base class set. The remaining 100 classes are partitioned into 10 sessions, or timestamps, where each session contains 10 new classes, and the few-shot training set consists of 5 examples per class (10-way 5-shot incremental task).

5.2. Implementation Details

We use the 16×16 patches OpenAI CLIP [36] version from the HuggingFace's Transformers library [54] as the starting backbone. Models and pipelines are built in PyTorch with the aid of Avalanche library [27]. We use the SGD optimizer with momentum, by setting learning rate to 0.00325, weight decay to $1e^{-5}$, and a cosine annealing with warmup, for all the benchmarks. For the base class training we set batch size to 32 and number of epochs to 3 for CIFAR100 and *mini*ImageNet, and batch size to 4 with 6 epochs for CUB200. For the new class session training sets we set batch size to 4 and number of epochs to 5 for all the benchmarks. All the experiments have been deployed on one single GeForce RTX 2080 Ti. For model comparison, we report the top-1 evaluation accuracy for base class and for every session since it is the standard practice in FSCIL. We also report the dropping rate (PD) metric, which measures the drop in accuracy in the last session w.r.t. the accuracy in base class session as a measure of forgetting, and the across-session average accuracy as a measure of overall performance.

5.3. Hyperparameter Tuning

We performed a hyperparameter tuning to select the best candidate model by varying the two hyperparameters affecting CPE-CLIP overall behavior on FSCIL. We used grid search to explore the following range of values: $L = [2, 4]$, $D = [1, 3, 6, 9, 12]$ in an exhaustive 2×5 search. For every configuration, we use the average across-session accuracy of 5 runs with random parameter initialization as the metric for model selection. Due to the computational burden of the exhaustive hyperparameter search, we focused on CUB200 benchmark only, since it conveys a special challenge for our CLIP-based method due to the fine-grained images associated with technical, non-common, text labels reflecting bird species. Results are shown in Table 1

	D				
L	1	3	6	9	12
2	67.48 (0.25)	68.63 (0.38)	68.54 (0.38)	69.49 (0.14)	**70.23** (0.31)
4	68.03 (0.26)	68.35 (0.07)	69.61 (0.45)	69.11 (0.36)	69.28 (0.28)

Table 1. Hyperparameter Tuning for L and D. Average across-session accuracy and standard errors (in brackets) for every configuration are reported.

The best model results in the hyperparameter set $L = 2$, $D = 12$. Therefore, we relied on this configuration for model comparison.

5.4. Comparison with state-of-the-art models

In this section, we show our main results on CIFAR100, *mini*ImageNet, and CUB200 benchmarks, shown in Table 2, 3, and 4, respectively, where CPE-CLIP is compared with the latest state-of-the-art FSCIL approaches [66, 1, 61, 55, 56, 67, 62, 63]. Models which are outperformed by a great margin by the most recent state-of-the-art methods were not included in the model comparison study.

According to these results, our model outperforms state-of-the-art models by a great margin. CPE-CLIP obtains the best classification accuracy in the base class session and reduces forgetting more efficiently, as shown by the PD metric, while maintaining stable high classification performances over time. CPE-CLIP shows superior abilities in reducing information loss when moving from training on a larger dataset, such as the base class set, to smaller datasets during learning sessions. It is worth mentioning that the other approaches included in the current model comparison primarily relied on ResNet [13] and ViT [8] as the main backbones, which were pretrained solely for the CUB200 benchmark. Furthermore, CPE-CLIP relies on CLIP which was not pretrained directly on popular foundational datasets such as ImageNet [39], differently from ResNet and ViT.

We have also conducted a comparison of the training time and number of learnable parameters for various models in our study. The comparison was meant to unerstand computational costs for completing the entire learning session stack. We only included models that guarantee reproducibility and have available hyperparameters. The results of this comparison are presented in Table 5. Overall, our findings indicate that CPE-CLIP significantly decreases computational costs without sacrificing performance.

5.5. Ablation Study

Here we analyze the importance of the relevant components in CPE-CLIP. For brevity, we only rely on the CUB200 benchmark. In particular, we focused on three ablated mod-

Method	Accuracy in each session (%)									Avg. ↑	PD ↓	∆ Avg.	∆ PD
	0	1	2	3	4	5	6	7	8				
CEC † [61]	73.07	68.88	65.26	61.19	58.09	55.57	53.22	51.34	49.14	59.53	23.93	+**23.89**	+**16.62**
SPPR ‡ [66]	76.33	72.33	67.33	63.33	59.00	55.33	53.00	50.33	47.33	60.47	29.00	+**22.95**	+**21.69**
CLOM † [67]	74.2	69.83	66.17	62.39	59.26	56.48	54.36	52.16	50.25	60.57	23.95	+**22.85**	+**16.64**
FeSSSS † [1]	75.35	70.81	66.70	62.73	59.62	56.45	54.33	52.10	50.23	60.92	25.12	+**22.50**	+**17.81**
MFS3 † [55]	73.42	69.85	66.44	62.81	59.78	56.94	55.04	53.00	51.07	60.93	22.35	+**22.49**	+**15.04**
PC † [56]	76.30	71.89	67.70	63.40	60.21	57.31	55.01	52.79	50.74	61.71	25.56	+**21.71**	+**18.25**
LIMIT † [63]	73.81	72.09	67.87	63.89	60.70	57.77	55.67	53.52	51.23	61.84	22.58	+**21.58**	+**15.27**
FACT † [62]	74.60	72.09	67.56	63.52	61.38	58.36	56.28	54.24	52.10	62.24	22.50	+**21.18**	+**15.19**
CLIP zero-shot	74.45	72.83	72.11	70.25	69.71	69.55	69.52	68.78	68.04	70.58	6.40	+12.84	−0.91
CPE-CLIP	**87.83**	**85.86**	**84.93**	**82.85**	**82.64**	**82.42**	**82.27**	**81.44**	**80.52**	**83.42**	**7.31**		

Table 2. CIFAR100 benchmark. ∆ PD represents the improvement for PD compared to other models. ∆ Avg. represents the improvement in across-session average accuracy compared to other models. † identifies the results taken from their respective papers, and ‡ shows the results approximated from the respective paper's figures since tabular results are unavailable.

Method	Accuracy in each session (%)									Avg. ↑	PD ↓	∆ Avg.	∆ PD
	0	1	2	3	4	5	6	7	8				
CEC † [61]	72.00	66.83	62.97	59.43	56.70	53.73	51.19	49.24	47.63	57.74	24.37	+**28.39**	+**16.91**
CLOM † [67]	73.08	68.09	64.16	60.41	57.41	54.29	51.54	49.37	48.00	58.52	25.08	+**27.61**	+**17.62**
LIMIT † [63]	72.32	68.47	64.30	60.78	57.95	55.07	52.70	50.72	49.19	59.05	23.13	+**27.08**	+**15.67**
PC † [56]	73.20	68.35	64.06	60.85	58.00	54.98	52.82	51.17	50.16	59.28	23.04	+**26.85**	+**15.58**
MFS3 † [55]	73.65	68.91	64.60	61.48	58.68	55.55	53.33	51.69	50.26	59.79	23.39	+**26.34**	+**15.93**
FACT † [62]	72.56	69.63	66.38	62.77	60.60	57.33	54.34	52.16	50.49	60.69	22.07	+**25.44**	+**14.61**
SPPR ‡ [66]	80.00	74.00	68.66	64.33	61.00	57.33	54.66	51.66	49.00	62.29	31.00	+**23.84**	+**23.54**
FeSSSS † [1]	81.50	77.04	72.92	69.56	67.27	64.34	62.07	60.55	58.87	68.24	22.63	+**17.89**	+**15.17**
CLIP zero-shot	77.13	76.49	75.31	77.30	75.35	75.28	73.92	74.18	73.17	75.35	3.96	+10.78	−3.50
CPE-CLIP	**90.23**	**89.56**	**87.42**	**86.80**	**86.51**	**85.08**	**83.43**	**83.38**	**82.77**	**86.13**	**7.46**		

Table 3. *mini*ImageNet benchmark. ∆ PD represents the improvement for PD compared to other models. ∆ Avg. represents the improvement in across-session average accuracy compared to other models. † identifies the results taken from their respective papers, and ‡ shows the results approximated from the respective paper's figures since tabular results are unavailable.

els. First of all, we consider the case where no accumulation strategy for prompt propagation is applied to the vision encoder. In this case, a standard replacement strategy is used, as for the language branch. Further, we focus on the contribution of projecting G-Prompt to the vision branch, by completely removing vision prompts. Finally, we consider the case where no regularization is applied so that G-Prompt updates consistently across sessions. Results are depicted in Figure 2.

The comparison between the main model and its ablated versions reveals noteworthy observations. Specifically, it appears that prompt regularization is a critical element for ensuring consistent performance over time by counteracting the influence of biased distributions of few-shot training examples in problematic sessions. Conversely, the exclusion of the vision prompt system does not appear to have a marked effect on the model's susceptibility to forgetting and information loss during sessions. However, performance generally deteriorates compared to the full model throughout each session, with a noticeable decline during the initial sessions, and ultimately converges in the later ones. It is noteworthy that a minimal difference in performance is observed between the total removal of prompts from the vision branch and the utilization of a prompt propagation replacement strategy. Overall, the results confirm the effectiveness of our prompt system in achieving the best performance in a continual learning setting. To summarize, prompt regularization allows to reduce information loss and forgetting by ensuring stable training over time, and the accumulation strategy for prompt propagation in the vision encoder provides better image representation ensuring a better text-image match and higher accuracy within specific sessions.

Method	Accuracy in each session (%)											Avg.↑	PD ↓	ΔAvg.	ΔPD
	0	1	2	3	4	5	6	7	8	9	10				
SPPR † [66]	68.68	61.85	57.43	52.68	50.19	46.88	44.65	43.07	40.17	39.63	37.33	49.32	31.35	+**21.47**	+**14.37**
PC † [56]	74.06	70.89	68.13	63.98	61.54	58.85	57.56	55.96	54.28	53.73	52.40	61.03	21.66	+**9.76**	+**4.68**
CEC † [61]	75.85	71.94	68.50	63.50	62.43	58.27	57.73	55.81	54.83	53.52	52.28	61.34	23.57	+**9.45**	+**6.59**
MFS3 † [55]	75.63	72.51	69.65	65.29	63.13	60.38	58.99	57.41	55.55	54.95	53.47	62.45	22.16	+**8.34**	+**5.18**
FeSSSS † [1]	79.60	73.46	70.32	66.38	63.97	59.63	58.19	57.56	55.01	54.31	52.98	62.85	26.62	+**7.94**	+**9.64**
FACT † [62]	75.90	73.23	70.84	66.13	65.56	62.15	61.74	59.83	58.41	57.89	56.94	64.42	18.96	+**6.37**	+**1.98**
LIMIT † [63]	75.89	73.55	71.99	68.14	67.42	63.61	62.40	61.35	59.91	58.66	57.41	65.50	18.48	+**5.29**	+**1.50**
CLOM † [67]	79.57	76.07	72.94	69.82	67.80	65.56	63.94	62.59	60.62	60.34	59.58	67.17	19.99	+**3.62**	+**3.01**
CLIP zero-shot	65.46	63.37	62.15	58.58	58.66	58.57	56.95	55.97	54.57	54.64	55.31	58.56	10.15	+12.23	−6.83
CPE-CLIP	**81.58**	**78.52**	**76.68**	**71.86**	**71.52**	**70.23**	**67.66**	**66.52**	**65.09**	**64.47**	**64.60**	**70.79**	**16.98**		

Table 4. CUB200 benchmark. Δ PD represents the improvement for PD compared to other models. Δ Avg. represents the improvement in across-session average accuracy compared to other models. † identifies the results taken from their respective papers.

	CIFAR100		*mini*ImageNet		CUB200	
Model	# params.	train. time	# params.	train. time	# params.	train. time
CEC [61]	295K	0.32	12.2M	1.41	12.3M	0.96
FACT [62]	280K	1.48	11.2M	7.30	11.3M	1.41
LIMIT [63]	295K	1.02	12.2M	2.00	12.3M	0.99
CLOM [67]	350K	0.24	14M	1.56	18.9M	0.35
CPE-CLIP	400K	0.69	400K	0.65	400K	0.27

Table 5. Training time (train. time), expressed in hours, and number of learnable parameters (# params.). Results are obtained by simulating models from their open-source training protocol. All the simulations were performed on the same GeForce RTX 2080 Ti.

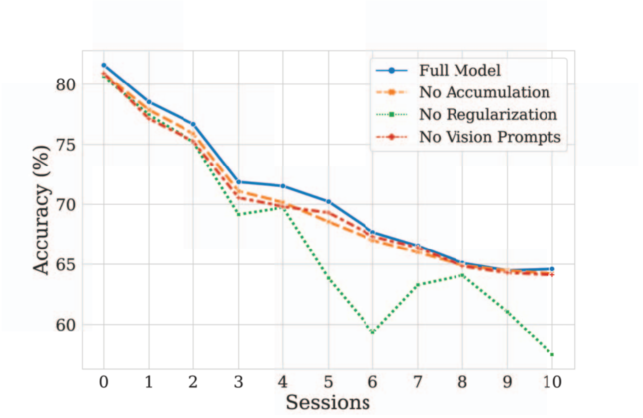

Figure 2. Ablation study depicting top-1 accuracy of 5-run simulations for the main model (Full Model), and three ablated versions where accumulation is removed from the vision branch (No Accumulation), no prompts are processed by the vision branch (No Vision Prompts), and no regularization is applied (No Regularization). Session 0 refers to base class.

6. Conclusions

Inspired by advances in few-shot image classification and parameter-efficient learning, we proposed a novel solution for solving the challenging task of Few-Shot Class Incremental Learning where a limited number of labeled data are available for each session. Our proposed CPE-CLIP effectively combined several technologies and modern ideas to conceive a multimodal few-shot continual learner that maintains high performances over time. We demonstrated that our proposed approach is capable of outperforming other approaches specifically designed for FSCIL, by relying on a small number of parameters and lower overall computational costs. CPE-CLIP introduces an accumulation strategy for prompt propagation that seems to be beneficial for enhancing image representation by ensuring the best classification accuracy. Prompt regularization ensures instead stable learning by reducing information loss over time.

Limitations. The CPE-CLIP architecture is built upon the CLIP framework as its underlying backbone. CLIP leverages text supervision to reason about visual concepts, which serves as a primary advantage for FSCIL. However, this also poses challenges when tackling tasks that lack image category labels, that are not readily processable by the CLIP vocabulary, or that are inherently ambiguous, leading to unreliable image-text matching. Additionally, the impact of regularization on a greater number of sessions has not been explored. Although decreasing the updating rate of G-Prompt parameters as more classes are seen seems crucial to avoid over-fitting, the scaling factor for the gradient approaches zero as sessions increase. In this case, the lack of proper G-Prompt parameters update can harm generalization on novel classes when unexpected distribution shifts occur.

References

[1] Touqeer Ahmad, Akshay Raj Dhamija, Steve Cruz, Ryan Rabinowitz, Chunchun Li, Mohsen Jafarzadeh, and Terrance E Boult. Few-shot class incremental learning leveraging self-supervised features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3900–3910, 2022.

[2] Ali Ayub and Alan R Wagner. Tell me what this is: Few-shot incremental object learning by a robot. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8344–8350. IEEE, 2020.

[3] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

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

[5] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

[6] Ali Cheraghian, Shafin Rahman, Pengfei Fang, Soumava Kumar Roy, Lars Petersson, and Mehrtash Harandi. Semantic-aware knowledge distillation for few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2534–2543, 2021.

[7] Ali Cheraghian, Shafin Rahman, Sameera Ramasinghe, Pengfei Fang, Christian Simon, Lars Petersson, and Mehrtash Harandi. Synthesized feature based few-shot class-incremental learning on a mixture of subspaces. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8661–8670, 2021.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[9] Sayna Ebrahimi, Franziska Meier, Roberto Calandra, Trevor Darrell, and Marcus Rohrbach. Adversarial continual learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 386–402. Springer, 2020.

[10] Thomas Elsken, Benedikt Staffler, Jan Hendrik Metzen, and Frank Hutter. Meta-learning of neural architectures for few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12365–12375, 2020.

[11] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[12] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*, 2021.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.

[15] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019.

[16] Neil Houlsby, Andreea Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. K-adapter: Infusing knowledge into pre-trained models with adapters. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 56–61, 2020.

[17] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual Prompt Tuning, July 2022. arXiv:2203.12119 [cs].

[18] Yifan Jiang, Han Chen, Hanseok Ko, and David K Han. Few-shot learning for ct scan based covid-19 diagnosis. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1045–1049. IEEE, 2021.

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

[20] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

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

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

[23] Xiang Li, Liunian Harold Li, Chi Li, Xiaodan Liang, and Li Dong. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5016–5027, 2021.

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

[25] Bin Liu, Yue Cao, Yutong Lin, Qi Li, Zheng Zhang, Mingsheng Long, and Han Hu. Negative margin matters: Understanding margin in few-shot classification. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 438–455. Springer, 2020.

[26] Binghao Liu, Boyu Yang, Lingxi Xie, Ren Wang, Qi Tian, and Qixiang Ye. Learnable distribution calibration for few-shot class-incremental learning. *arXiv preprint arXiv:2210.00232*, 2022.

[27] Vincenzo Lomonaco, Lorenzo Pellegrini, Andrea Cossu, Antonio Carta, Gabriele Graffieti, Tyler L. Hayes, Matthias De Lange, Marc Masana, Jary Pomponi, Gido van de Ven, Martin Mundt, Qi She, Keiland Cooper, Jeremy Forest, Eden Belouadah, Simone Calderara, German I. Parisi, Fabio Cuzzolin, Andreas Tolias, Simone Scardapane, Luca Antiga, Subutai Amhad, Adrian Popescu, Christopher Kanan, Joost van de Weijer, Tinne Tuytelaars, Davide Bacciu, and Davide Maltoni. Avalanche: an end-to-end library for continual learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2nd Continual Learning in Computer Vision Workshop, 2021.

[28] Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469:28–51, 2022.

[29] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018.

[30] Pratik Mazumder, Pravendra Singh, and Piyush Rai. Few-shot lifelong learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2337–2345, 2021.

[31] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[32] Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects, 2013.

[33] Quang Pham, Chenghao Liu, and Steven Hoi. Dualnet: Continual learning, fast and slow. *Advances in Neural Information Processing Systems*, 34:16131–16144, 2021.

[34] Marcin Przewięźlikowski, Przemysław Przybysz, Jacek Tabor, M Zięba, and Przemysław Spurek. Hypermaml: Few-shot adaptation of deep models with hypernetworks. *arXiv preprint arXiv:2205.15745*, 2022.

[35] Haoxuan Qu, Hossein Rahmani, Li Xu, Bryan Williams, and Jun Liu. Recent advances of continual learning in computer vision: An overview. *arXiv preprint arXiv:2109.11369*, 2021.

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[37] Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. Progressive prompts: Continual learning for language models. *arXiv preprint arXiv:2301.12314*, 2023.

[38] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

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

[40] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[41] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. *arXiv preprint arXiv:2211.13218*, 2022.

[42] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

[43] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018.

[44] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12183–12192, 2020.

[45] Vishal Thengane, Salman Khan, Munawar Hayat, and Fahad Khan. Clip model is an efficient continual learner. *arXiv preprint arXiv:2210.03114*, 2022.

[46] Gido M van de Ven, Tinne Tuytelaars, and Andreas S Tolias. Three types of incremental learning. *Nature Machine Intelligence*, pages 1–13, 2022.

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[48] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

[49] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[50] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.

[51] Zifeng Wang, Tong Jian, Kaushik Chowdhury, Yanzhi Wang, Jennifer Dy, and Stratis Ioannidis. Learn-prune-share for lifelong learning. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 641–650. IEEE, 2020.

[52] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pages 631–648. Springer, 2022.

[53] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022.

[54] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, Oct. 2020. Association for Computational Linguistics.

[55] Xinlei Xu, Saisai Niu, Zhe Wang, Wei Guo, Lihong Jing, and Hai Yang. Multi-feature space similarity supplement for few-shot class incremental learning. *Knowledge-Based Systems*, page 110394, 2023.

[56] Xinlei Xu, Zhe Wang, Zhiling Fu, Wei Guo, Ziqiu Chi, and Dongdong Li. Flexible few-shot class-incremental learning with prototype container. *Neural Computing and Applications*, pages 1–15, 2023.

[57] Li Yin, Juan M Perez-Rua, and Kevin J Liang. Sylph: A hypernetwork framework for incremental few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9035–9045, 2022.

[58] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017.

[59] Tianhe Yu, Chelsea Finn, Annie Xie, Sudeep Dasari, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot imitation from observing humans via domain-adaptive meta-learning. *arXiv preprint arXiv:1802.01557*, 2018.

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

[61] Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. Few-shot incremental learning with continually evolved classifiers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12455–12464, 2021.

[62] Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, Liang Ma, Shiliang Pu, and De-Chuan Zhan. Forward compatible few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9046–9056, 2022.

[63] Da-Wei Zhou, Han-Jia Ye, Liang Ma, Di Xie, Shiliang Pu, and De-Chuan Zhan. Few-shot class-incremental learning by sampling multi-phase tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[64] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022.

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

[66] Kai Zhu, Yang Cao, Wei Zhai, Jie Cheng, and Zheng-Jun Zha. Self-promoted prototype refinement for few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6801–6810, 2021.

[67] Yixiong Zou, Shanghang Zhang, Yuhua Li, and Ruixuan Li. Margin-based few-shot class-incremental learning with class-level overfitting mitigation. *arXiv preprint arXiv:2210.04524*, 2022.