University of Modena and Reggio Emilia
XXXVII cycle of the International Doctorate School in
Information and Communication Technologies

A Deep Dive into the Latent Space of Continual Learners

Emanuele Frascaroli

Supervisor: Prof. Simone Calderara
Ph.D. Course Coordinator: Prof. Luigi Rovati

Modena, 2025

Review committee:

Prof. Emanuele Rodolà, Sapienza University of Rome
Prof. Elisa Ricci, University of Trento

Abstract

With advancements in technology and increasing computational power, the field of Deep Learning has experienced significant growth over the past decade. Deep Neural Networks (DNNs) have achieved remarkable performance across various tasks, setting state-of-the-art benchmarks in image, video, audio, and text domains. These models excel at learning complex patterns from data, requiring a substantial amount of labeled examples to reach their full potential. However, a key difference between humans and DNNs is the ability to remember. While humans can continuously learn and adapt to new tasks without forgetting past knowledge, DNNs tend to overwrite previously learned information when trained on new data. This phenomenon, known as *catastrophic forgetting*, hinders the continuous learning capability of these models. Consequently, updating a model with new information often necessitates expensive retraining on all data, which is not always feasible in real-world scenarios.

The field of Continual Learning (CL) aims to address this issue by developing strategies that enable models to retain past knowledge while learning new tasks. The CL literature has expanded significantly in recent years, with various approaches involving rehearsal, regularization, distillation, and architectural modifications. Despite these advancements, the gap between continuous learning and joint training persists, and the problem of catastrophic forgetting remains unsolved.

This thesis begins by examining the latent space - the internal data representations - of a model during the continuous learning process, and studying how it evolves over time. This analysis led to the development of several strategies to mitigate forgetting by acting on the latent space of DNNs, contributing to the CL literature. Two of these methods, named CaSpeR-IL and CLER, introduce new regularization terms in the loss function of existing CL models. CaSpeR-IL leverages spectral geometry techniques to constrain class representations, enhancing clustering behavior. CLER explores how invariant and equivariant self-supervised approaches impact the latent space of a model, exploiting their benefits to prevent forgetting. With the advent of Vision Transformers (ViTs), a new method named SCAD is proposed to adapt these architectures to new tasks through distillation between the model's internal layers.

Finally, this thesis investigates the impact of pretrained Vision-Language

Models (VLMs), such as CLIP, on a CL scenario. These models align the latent space of images and their corresponding captions, enabling zero-shot learning on classification tasks. This work presents two innovative methods to adapt VLMs to a CL scenario by leveraging the model's internal representations. STAR-Prompt employs a two-level prompting approach to balance stability (the capacity to remember past knowledge) and plasticity (the ability to learn new tasks). CGIL utilizes Variational Autoencoders to perform generative replay in the embedding space of CLIP, demonstrating state-of-the-art performance on both standard CL benchmarks and new scenarios that test the model's zero-shot capabilities.

Sommario

Con i progressi della tecnologia e l'aumento della potenza di calcolo, il campo del Deep Learning ha sperimentato una crescita significativa nell'ultimo decennio. Le reti neurali hanno raggiunto prestazioni notevoli in vari compiti, stabilendo benchmark all'avanguardia nei domini delle immagini, video, audio e testo. Questi modelli eccellono nell'apprendimento di pattern complessi dai dati, nonostante richiedano una quantità sostanziale di esempi etichettati per raggiungere il loro pieno potenziale. Tuttavia, una differenza chiave tra gli esseri umani e le reti è la capacità di ricordare. Mentre gli esseri umani possono apprendere e adattarsi continuamente a nuovi compiti senza dimenticare le conoscenze passate, le reti neurali tendono a sovrascrivere le informazioni precedentemente apprese quando vengono addestrate su nuovi dati. Questo fenomeno, noto come *catastrophic forgetting*, ostacola la capacità di apprendimento continuo di questi modelli. Di conseguenza, l'aggiornamento di un modello con nuove informazioni spesso richiede un costoso riaddestramento su tutti i dati, non sempre fattibile in scenari reali.

Il campo del Continual Learning (CL) mira a risolvere questo problema sviluppando strategie che consentano ai modelli di mantenere le conoscenze passate mentre apprendono nuovi compiti. La letteratura sul CL si è espansa significativamente negli ultimi anni, con vari approcci che coinvolgono il salvataggio di pochi dati, la regolarizzazione, la distillazione e le modifiche architetturali. Nonostante questi progressi, il divario tra apprendimento continuo e addestramento congiunto persiste, e il problema del dimenticare il passato rimane irrisolto.

Questa tesi nasce dall'idea di analizzare lo spazio latente - le rappresentazioni interne dei dati - di un modello durante il processo di apprendimento continuo, e studiare come evolve nel tempo. Questa analisi ha portato allo sviluppo di diverse strategie per mitigare il forgetting agendo sullo spazio latente delle reti neurali, contribuendo alla letteratura esistente. Due di questi metodi, denominati CaSpeR-IL e CLER, introducono nuovi termini di regolarizzazione nei modelli di CL esistenti. CaSpeR-IL sfrutta tecniche di geometria spettrale per vincolare le rappresentazioni delle classi. CLER esplora come gli approcci auto-supervisionati invarianti ed equivarianti impattano sullo spazio latente di un modello, sfruttando i loro benefici per prevenire il forgetting. Con l'avvento dei Vision Transformers (ViT), viene proposto un nuovo metodo denominato SCAD per adattare queste

architetture a nuovi compiti attraverso la distillazione di conoscenza tra i livelli interni del modello.

Infine, questa tesi indaga l'impatto dei Vision-Language Models (VLMs) pre-addestrati, come CLIP, in uno scenario di CL. Questi modelli allineano lo spazio latente delle immagini e delle loro corrispondenti didascalie, abilitando capacità zero-shot nei compiti di classificazione. Questo lavoro presenta due metodi innovativi per adattare i VLMs a uno scenario di CL sfruttando le rappresentazioni interne del modello. STAR-Prompt impiega un approccio di prompting a due livelli per bilanciare la stabilità (la capacità di ricordare le conoscenze passate) e la plasticità (la capacità di apprendere nuovi compiti). CGIL utilizza Variational Autoencoders per generare dati sintetici nello spazio latente di CLIP, dimostrando prestazioni all'avanguardia sia su benchmark standard che su nuovi scenari che testano le capacità zero-shot del modello.

Contents

Part I

Introduction

Chapter 1

Research Statement

Nowadays, the field of Artificial Intelligence (AI) is experiencing a renaissance and becoming increasingly pervasive in daily life. Its recent success in various applications and its ability to interact with everyday users have made it a hot topic in both the industrial and academic worlds. This success is mainly due to the development of Deep Learning (DL) techniques, specifically the advancements in Deep Neural Networks (DNNs). These models have shown remarkable performance on tasks involving large amounts of data, such as image classification, speech recognition, natural language processing, *etc.* Despite their success, DNNs still have various limitations. One such limitation is their capacity to remember. While humans can learn different tasks sequentially without forgetting, such as learning to recognize dogs after having learned to recognize cats, DNNs tend to overwrite previously learned tasks and, to be effective, they need to be trained on all tasks simultaneously (*e.g.* by seeing dogs and cats together). This is known as the *catastrophic forgetting* problem. The field of Continual Learning (CL) has emerged to address this issue, aiming to develop models that can learn continuously without forgetting previous knowledge.

This thesis aims to investigate the CL scenario and contribute to the development of new ways to tackle the forgetting phenomenon. In particular, this work starts with a question: *How does the latent space of a DNN evolve when learning new tasks sequentially?* The latent space of a DNN is the internal representation of the input data, and it can indicate how the model is learning and storing information. This work investigates the evolution of the latent space of different models in the context of CL. This analysis led to the development of new strategies to mitigate forgetting. Along with experimental results and ablation studies, this thesis demonstrates the significance of DNNs internal representations as a critical factor in advancing the field of CL.

Chapter 2

Contributions and Organization

This thesis covers various studies on the CL scenario, focusing on the evolution of the latent space of models while learning new tasks sequentially. Its contributions are organized as follows:

- **Part I** introduces the content of this thesis. Specifically:
  - **Chapter 1** presents the research statement and the aims of this work.
  - **Chapter 2** describes the structure of this thesis.
  - **Chapter 3** provides the necessary background on CL, the catastrophic forgetting problem, common benchmarks, and the main methods from the existing literature.
- **Part II** presents different studies on the latent space of existing CL models. These analyses led to the development of new strategies to regularize the evolution of internal representations, aiming to mitigate forgetting.
  - **Chapter 4** uses concepts from inverse spectral geometry to analyze the latent space, demonstrating that spectral properties can identify interference when learning new tasks. This analysis led to the development of Continual Spectral Regularizer for Incremental Learning (CaSpeR-IL), a regularization method that constrains the spectral properties of the evolving latent space to reduce forgetting.
  - **Chapter 5** studies how invariant and equivariant self-supervised losses affect the latent space, and presents Continual Learning via Equivariant Regularization (CLER), a regularizer that leverages equivariant losses to stabilize the latent space while learning new tasks.
- **Part III** and **Chapter 6** tackle the more challenging scenario of Multi-Label Continual Learning (MLCL) by proposing Selective Class Attention

Distillation (SCAD). This method leverages the knowledge distillation paradigm over the internal representations of a pretrained model to filter and retain knowledge relevant to previous tasks, thereby reducing forgetting.

- **Part IV** examines how the latent space of large pretrained Vision-Language Models (VLMs) can enhance the performance of CL models, achieving impressive results in preventing forgetting.
  - **Chapter 7** presents Semantic Two-Tier Additive Residual Prompting (STAR-Prompt), a method that utilizes the latent space of VLMs to aid an image classifier. It adopts a two-stage prompting strategy to finetune both models for current tasks while preserving previous knowledge.
  - **Chapter 8** explores various Parameter-Efficient Fine-Tuning (PEFT) techniques to train a VLM and proposes Continual Generative training for Incremental prompt-Learning (CGIL), a method that adapts the latent space of a VLM to new tasks while preventing forgetting. Experiments show that this method also enhances the model's zero-shot performance on unseen data.
- **Part V** summarizes the main findings of this thesis, emphasizing the critical role of DNNs latent space in the CL scenario.

Chapter 3

Technical Background

3.1 Deep Learning

The field of Artificial Intelligence (AI) was born in the 20^{th} century to develop algorithms able to perform tasks that require human intelligence. Today, this field is a broad discipline that includes several subfields. Machine Learning (ML) is a subset of AI that aims to give computers human abilities without being explicitly programmed to do so. In practice, ML algorithms learn patterns from data to make predictions or decisions. While the mathematical foundations of the perceptron were laid in the nineteen-sixties, Artificial Neural Networks (ANNs) only became popular in the last two decades. This is mainly due to two factors: (*i*) the availability of large datasets and annotations and (*ii*) the increase in computational power, specifically the improvements of Graphics Processing Units (GPUs). The conjunction of these factors allowed the development of Deep Neural Networks (DNNs), neural networks with many layers that can learn complex patterns from data. This gave rise to the field of Deep Learning (DL), where the increasing availability of data led DNNs to outperform traditional ML algorithms almost everywhere. In 2012, deep Convolutional Neural Networks (CNNs)[1] won the ImageNet Large Scale Visual Recognition Challenge, a competition that evaluates the performance of algorithms in image classification tasks. A few years later, with the takeover of ResNet [61] on vision benchmarks, the residual connection was introduced and became a standard in every DNN. In 2016, thanks to the advancements of Reinforcement Learning, AlphaGo [121] defeated the world champion in the game of Go. In 2017, the Transformer [130] architecture revolutionised the field of Natural Language Processing. Despite their attention mechanism being designed for sequence-shaped data, it has later been adapted to other domains. Nowadays, large DNNs have become popular even among common users, thanks to their effectiveness in tasks useful for everyday life.

[1] Specifically, the CNN called AlexNet [84].

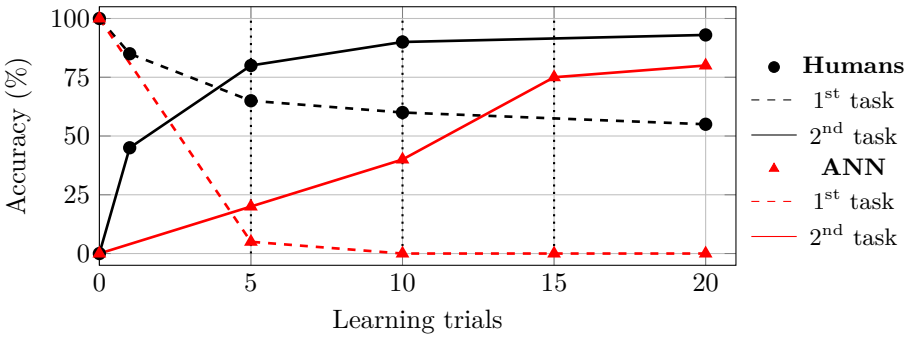


Figure 3.1: Comparison of human and ANN performance on a two-task learning experiment, based on data from *Barnes and Underwood* [11] and *McCloskey and Cohen* [96]. While human subjects exhibit gradual forgetting of associations from the 1st task during the learning of the 2nd task, ANNs experience a significantly faster forgetting of the earlier task, hence called *catastrophic*.

3.2 Catastrophic Forgetting

DL systems have demonstrated unparalleled capabilities in performing intricate tasks at levels often surpassing human performance. As AI applications proliferate, their transformative potential becomes increasingly evident in fields such as healthcare, autonomous systems, and natural language processing. However, a fundamental limitation of ANNs contrasts with their otherwise extraordinary achievements: their inability to retain previously learned information when trained sequentially on new tasks. Unlike the human brain, which can adapt to new tasks while retaining prior knowledge (*e.g.* learning to drive a car after having learned to ride a bicycle), ANNs often exhibit dramatic performance degradation on earlier tasks when exposed to new data distributions. This phenomenon is known as catastrophic forgetting, to highlight the difference with the gradual forgetting observed in human memory (Fig. 3.1). The root cause lies in the structure and optimization of neural networks. These models rely on a shared set of parameters, which are greedily adjusted to minimize error on the data they are exposed to through backpropagation. This process overwrites information critical for prior tasks, leading to abrupt and severe forgetting. While initially documented in shallow networks [96, 111], this limitation persists in modern DNNs [55], presenting a significant challenge in dynamic environments where data evolves over time.

3.3 Continual Learning

Catastrophic forgetting has far-reaching consequences for the lifecycle of AI systems. It hinders their ability to adapt to new data streams without extensive

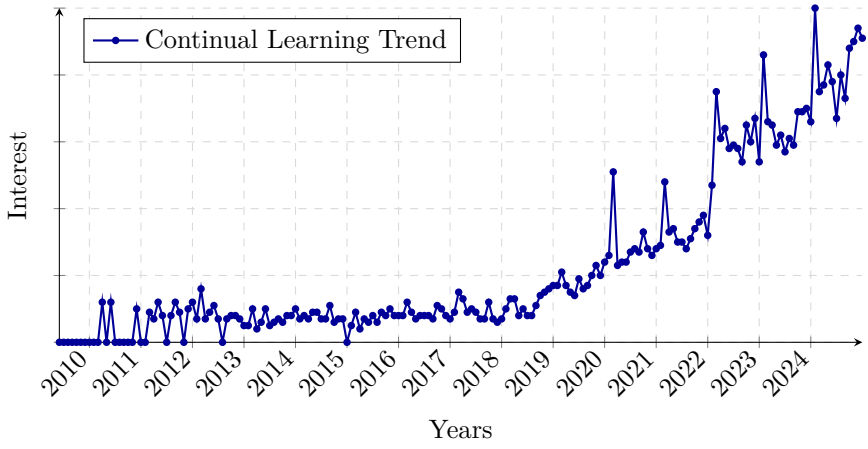


Figure 3.2: Google Trends data showing the increasing interest in Continual Learning over the past decade.

retraining, making their deployment in real-world, continually changing scenarios both costly and inefficient. To address this limitation, the field of Continual Learning (CL) has emerged as a promising research area aimed at enabling neural networks to learn continuously, acquiring new knowledge while preserving prior information. Inspired by the adaptability of human cognition, CL seeks to achieve a balance between *stability*, the retention of previously acquired knowledge, and *plasticity*, the capacity to learn and adapt to new tasks. Unlike traditional machine learning paradigms that assume access to static datasets, CL frameworks operate under conditions where data arrives sequentially, without the possibility of revisiting earlier tasks. This paradigm shift requires innovative approaches to ensure that models remain both efficient and resilient in the face of changing data distributions.

3.3.1 Standard Scenarios

The catastrophic forgetting problem can be observed in a variety of ML tasks, like segmentation [20, 143], detection [104, 74], generation [148], and captioning [41]. However, the majority of CL literature focuses on continual classification problems. This choice allows researchers to have a clear and standard problem definition, making it easier to compare different methods. As this thesis is focused solely on classification tasks, in line with the majority of CL literature, the notation CL will be always used to indicate *continual learning classification* problems unless otherwise stated.

Despite the single task of classification, CL scenarios can vary significantly, and many methods could not be directly compared due to different experimental conditions. To address this issue, the community has started to define standard

settings, benchmarks, and metrics. The first taxonomy of possible experimental setups was proposed by *Van de Ven et al.* [128, 129], in which a supervised classification problem is split into tasks observed sequentially. Each task is composed of a portion of the overall dataset, and the learner cannot access the data from previous tasks. The model is notified when the task changes, allowing it to take specific steps before moving on to the next task. Due to different approaches to splitting the data, the authors defined three main scenarios:

- **Domain-Incremental Learning (Domain-IL)**: the tasks are defined by different input data distributions. The model is faced with all possible classes at each task, but the input data changes. At test time, the model is evaluated on data from all tasks, without knowing the task identity of the input. These experiments rely on specialized datasets, either created by applying task-specific transformations to existing classification datasets or by utilizing datasets that inherently span multiple domains (*e.g.* DomainNet [102], a dataset originally designed for Domain Adaptation Problems).

- **Task-Incremental Learning (Task-IL)**: the model is presented with distinct, non-overlapping classification tasks, *i.e.* where each task has its own set of classes. During evaluation, the specific task of data is provided to the model, allowing it to learn a task-conditioned classification function. This approach simplifies the separation of knowledge across tasks and often results in optimal performance, making Task-IL one of the most straightforward scenarios [49, 7]. Nonetheless, this setting remains valuable for assessing the degree of forgetting in the model, as it eliminates classifier bias caused by imbalanced data presentation [140].

- **Class-Incremental Learning (Class-IL)**: the data is split into tasks the same way as in Task-IL, but with a key difference: the task identity is not provided during testing. This requires the model to learn a classification function that simultaneously identifies the source task and the corresponding class. This scenario is recognized as the most challenging and significant of the three [49], hence, it is widely adopted for the main evaluations in the CL literature.

These scenarios have been widely adopted in the CL literature, providing a common ground for comparing different methods. However, they have been also criticized for their limitations in modeling realistic incremental applications [6, 40]. For this reason, novel scenarios have been proposed by the recent CL literature, with the aim of bridging the gap between the theoretical assumptions of the standard scenarios and the real-world applications.

As this thesis primarily focuses on the Class-IL scenario, other settings will be introduced only if they are considered in the experiments.

3.4 State of the Art

The increasing interest in CL is reflected in the growing number of publications and research activities in the field. This surge has led to the development of a wide range of CL methods, each offering unique solutions to the catastrophic forgetting problem. These methods can be broadly categorized into three main families: *Architecture-based*, *Regularization-based*, and *Rehearsal-based* approaches [49, 39]. Each family leverages distinct strategies to mitigate forgetting. In this section, an overview of the most popular CL methods is provided, as they represent the state of the art in the field and are used for comparison in the experimental evaluations of this thesis.

3.4.1 Architectural Methods

Architectural solutions are techniques designed to mitigate catastrophic forgetting by allocating distinct sets of model parameters to handle different tasks. These methods often rely on modular or multi-headed architectures, where each task is assigned its own dedicated network components or subnetworks. By isolating task-specific parameters, architectural approaches prevent interference between tasks, thereby preserving previously learned knowledge. While highly effective in retaining past knowledge, these methods can be memory-intensive, as their parameter requirements often grow with the number of tasks, and typically require prior knowledge of task identities to work effectively. Consequently, they are best suited for scenarios like Task-IL, where task labels are available during testing. For this reason, as this thesis focuses on the more challenging Class-IL scenario, architectural methods will not be considered in the experimental evaluations.

Progressive Neural Networks (PNN) [115] represents a prototypical architectural approach, dedicating a separate instance of the backbone network to each task. To enable knowledge transfer between tasks, it incorporates specialized adaptation layers that act as connections between successive backbones. Although this design inherently prevents forgetting, it suffers from a significant drawback—its memory usage scales linearly with the number of tasks, making it resource-intensive.

3.4.2 Regularization Methods

Regularization-based approaches aim to prevent catastrophic forgetting by imposing constraints that preserve important information from previously learned tasks. Instead of allocating separate parameters for each task, these approaches regularize the update of model weights to ensure that changes do not significantly alter previously learned representations. These methods often are computationally efficient and scalable, making them well-suited for scenarios where the number of tasks can increase significantly. However, despite regularization methods being suitable for all CL scenarios, their performance may degrade as the number of tasks increases, particularly in highly complex or diverse datasets.

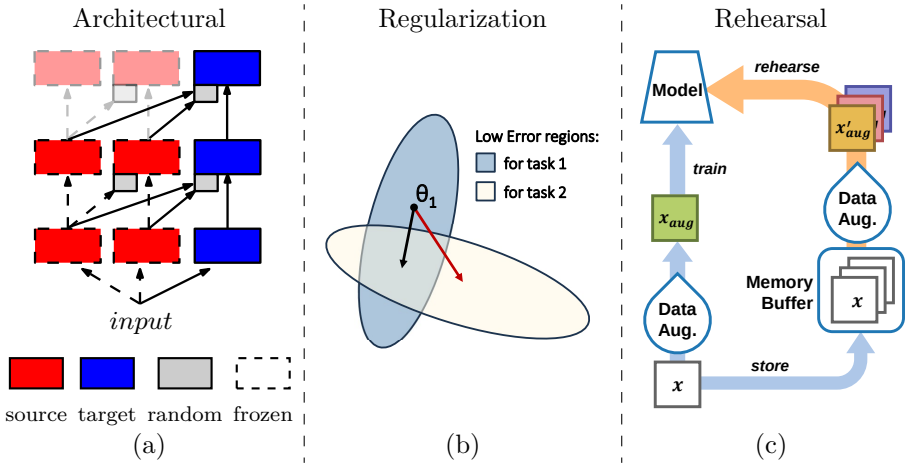


Figure 3.3: **Topology of CL methods**. (a) An example of an architectural method, PNN [115]. (b) A regularization-based method, EWC [80]. (c) A rehearsal standard approach, image taken from [18].

- **Elastic Weight Consolidation (EWC)** [80] is a pioneering regularization-based method that constrains the model's parameter updates to prevent catastrophic forgetting. It operates by identifying parameters that are critical for prior tasks using the Fisher Information Matrix, which measures the sensitivity of the model's output to changes in its parameters. The updates to these important parameters are then constrained by adding a quadratic penalty to the loss function, discouraging significant deviations from their original values. This approach allows the model to adapt to new tasks while retaining performance on older ones. As computing the Fisher Information Matrix can be cumbersome and memory-intensive, different approximations of this method have been proposed that are more efficient without sacrificing performance, such as **online Elastic Weight Consolidation (oEWC)** [116].

- **Learning without Forgetting (LwF)** [89] addresses catastrophic forgetting by preserving knowledge from previous tasks through distillation. It maintains the responses of the model trained on old tasks (*the teacher*) and uses them as a form of supervision while training the current model (*the student*) on new tasks. This approach enables the model to integrate new information while approximating the behavior of earlier tasks. Despite its simplicity and data efficiency, LwF may struggle in cases with highly disjoint tasks, where output distributions between tasks differ significantly. A noteworthy variation of this method is **MultiClass LwF (LwF.MC)** [112], which adopts a slightly different loss function and is designed to better operate in the Class-IL setting.

3.4.3 Rehearsal Methods

Rehearsal-based (or *replay-based*) methods address catastrophic forgetting by storing a subset of previously encountered data and replaying it during training. These approaches maintain a memory buffer containing representative samples from past tasks, which are then interleaved with data from the current task to reinforce prior knowledge and prevent its degradation. By revisiting earlier examples, the model retains past information while adapting to new inputs, effectively mitigating interference between tasks. Rehearsal methods can vary in their memory management techniques, how they select samples to store in the buffer, and in the different strategies of replaying the data. Additionally, techniques such as generative replay eliminate the need for explicit storage by training generative models to synthesize past data. While rehearsal methods are highly effective, their reliance on memory storage prevents their adoption in scenarios where retaining raw data is restricted, such as in privacy-sensitive applications.

- **Experience Replay (ER)** [111, 113] is one of the earliest and most intuitive rehearsal-based methods. It maintains a small memory buffer that stores samples from previously encountered training data. During subsequent tasks, data from the memory buffer is interleaved with new data, enabling joint optimization and preserving knowledge of past tasks. Typically, the buffer is populated using the *reservoir sampling* algorithm, which operates online and ensures that each data point in the stream has an equal probability of being included in memory. Despite its simplicity, ER remains a strong baseline and serves as the foundation for numerous approaches in the CL literature.

- **Incremental Classifier and Representation Learning (iCaRL)** [112] is a popular method that combines rehearsal and distillation techniques to address catastrophic forgetting. It maintains a memory buffer containing a limited number of exemplars from past tasks, selected using a herding strategy to ensure representativeness. During training, it employs knowledge distillation to retain prior knowledge by aligning outputs for old examples while simultaneously learning new classes. Additionally, the final predictions of the model leverage a prototype-based *nearest-mean-of-exemplars* classifier, exploiting the class representations from the memory buffer. This approach allows iCaRL to achieve robust performance on complex benchmarks even with small memory buffers.

- **Greedy Sampler and Dumb Learner (GDumb)** [107] focuses on optimizing performance using a fixed memory buffer. Instead of continuously training on incoming data, it focuses solely on collecting and storing samples within the buffer. Before evaluation, GDumb trains the backbone network from scratch using only the data stored in memory. Despite its simplicity, this approach frequently outperforms more advanced continual learning

methods. However, its effectiveness diminishes when the memory buffer is too small relative to the dataset size.

- **Pooled Outputs Distillation Network (PODNet)** [45] uses a representation distillation strategy to align feature maps from previous tasks with those of the current task. Moreover, this model learns multiple representations for each class, fitting multi-modal distributions to improve classification.
- **Dark Experience Replay (DER)** [17] combines experience replay with knowledge distillation. It stores a subset of past data in a memory buffer, along with their logits, during training. These soft predictions are then treated as a supervisory signal to enforce consistency between current and past predictions, mitigating catastrophic forgetting. Its advanced version, **Dark Experience Replay++ (DER++)**, mixes the *dark* distillation strategy with standard replay, achieving strong performance across many CL benchmarks.
- **Continual Prototype Evolution (CoPE)** [40] presents a learning approach based on slowly evolving class prototypes within a shared latent space. It adopts a classifier based on class prototypes, whose careful update scheme allows for learning incrementally while avoiding sudden disruptions in the model representations.
- **Supervised Contrastive Replay (SCR)** [94] combines experience replay with supervised contrastive learning. Similarly to iCaRL, it replaces the classifier with a nearest-neighbor classifier, effectively addressing recency bias and eliminating the need for structural changes when new classes are introduced. SCR then encourages embeddings of the same class to cluster closely while separating those of different classes, thanks to the Supervised Contrastive Loss [76].
- **DualNet** [106] combines a *fast learner* for rapid, task-specific learning and a *slow learner* for general, task-agnostic representations using **Self-Supervised Learning (SSL)**. This dual-backbone architecture decouples incremental classification from the learning of a transferable representation, allowing the model to adapt to new tasks while preserving past knowledge.
- **Experience Replay with Asymmetric Cross-Entropy (ER-ACE)** [19] builds upon standard ER by introducing the **Asymmetric Cross-Entropy (ACE)** loss function. This approach addresses class imbalances by treating old and new samples asymmetrically, preventing the model from overfitting to newly learned classes. This ensures a balanced learning process, preserving past knowledge and reducing interference between tasks.
- **Cross-Space Clustering and Controlled Transfer (CSCCT)** [10] leverages a two-stage training process to address catastrophic forgetting.

It first clusters the feature space to identify task-specific subspaces (*cross-space clustering*). Then, it transfers knowledge between tasks by leveraging semantic similarities between old and new classes (*controlled transfer*), to promote positive forward transfer while minimizing negative interference. This approach does not represent a standalone method, but rather a plug-and-play component that can be integrated into existing CL frameworks to enhance their performance.

- **eXtended Dark Experience Replay (X-DER)** [14] extends the DER method by addressing its shortcomings. It introduces a novel memory update mechanism that revises the knowledge stored in the buffer to incorporate new information about past data. Additionally, X-DER prepares the model for learning unseen classes by setting expectations for future tasks. This approach enhances the model's ability to adapt to new data distributions and improves the performance on challenging CL benchmarks.

- **Online Prototype Learning (OnPro)** [139] utilizes class prototypes and a contrastive SSL objective to effectively learn discriminative features for both new and previously encountered classes. It also incorporates a memory buffer management strategy that prioritizes samples from classes prone to misclassification, enhancing representation learning and retention.

3.4.4 Pretrain-Exploiting Methods

In the recent literature, a new category of methods arose. These methods leverage the pretraining of a model on a large amount of data and adapt it to new tasks in a continual learning scenario. These approaches are particularly effective since the knowledge of the pretrained model is already optimized for a wide range of tasks. As standard CL models start to learn from scratch, they are often outperformed by these methods. However, many standard methods are designed to allow different backbones, hence, they can also start with the same pretrained model, providing a fair comparison and an initial baseline. Most recent pretrain-exploiting methods leverage the **Vision Transformer (ViT)** architecture and **Parameter-Efficient Fine-Tuning (PEFT)** techniques, in particular *prompt-based* strategies. These techniques will be better explained in the next chapters of this thesis.

- **Learning to Prompt (L2P)** [138] is the pioneering prompting approach that utilizes learnable prompts to guide a pretrained model. A pool of small, trainable prompt parameters is maintained and dynamically selected based on input instances to condition the model appropriately for each task. The model can also be equipped with a rehearsal buffer, but it achieves competitive performance even without it.

- **DualPrompt** [137] introduces a hierarchy of general and task-specific prompts: the former capture task-invariant knowledge, while the latter

encode task-specific information. These prompts condition the backbone model through a *prefix-tuning* strategy, guiding the learning process for new tasks while preserving prior knowledge, and eliminating the need for rehearsal buffers.

- **COntinual Decomposed Attention-based Prompting (CODA-Prompt)** [124] allocates a set of prompts for each task, training only the ones linked to the current task. Additionally, the method employs attention vectors with learnable values to filter the queries, and conditions the model with a weighted sum of the prompts.

- **Slow Learner with Classifier Alignment (SLCA)** [149] addresses the progressive overfitting problem observed when fine-tuning pretrained models. The method involves selectively reducing the learning rate for the representation layer (termed *slow Learner*) to maintain generalizability and applying *classifier alignment* to adjust the classification, by modeling class-wise distributions and aligning classifiers in a post-hoc manner. This can effectively prevent forgetting and reduce the interference between tasks on the classification layer.

- **AttriCLIP** [135] leverage a pretrained Vision-Language Model (VLM) without expanding model parameters or requiring rehearsal memory. It represents each class with a combination of the class name and a set of learnable *attribute* prompts, selected from a trained attribute word bank.

- **PromptFusion** [27] addresses the stability-plasticity dilemma by separating the model into two modules: the *stabilizer*, which preserves prior knowledge, and the *booster*, which enhances the model's adaptability to new tasks. The two modules employ different backbones and distinct prompt tuning strategies, allowing the model to balance stability and plasticity effectively.

- **Zero-Shot Continual Learning (ZSCL)** [151] fine-tunes a pretrained VLM aiming to preserve its zero-shot capabilities. The method uses feature distillation with a reference dataset to maintain semantic diversity and parameter averaging to prevent drastic weight shifts.

- **Mixture-of-Experts Adapters (MoE-Adapters)** [145] leverages novel PEFT techniques and the Mixture-of-Experts [70, 118] paradigm to fine-tune a pretrained VLM. The model learns adapters as sparse experts and incrementally incorporates task-specific routers to select the corresponding experts. The authors also implemented the Distribution Discriminative Auto-Selector (DDAS) module, which automatically routes out-of-distribution inputs to the original VLM, maintaining the model's zero-shot capabilities.

3.5 Continual Learning Benchmarks

This section provides an overview of the most commonly used benchmarks in the CL literature. It begins with an introduction to the standard scientific notation used to describe CL scenarios, followed by a description of the evaluation metrics employed to assess CL methods. Next, it reviews the most popular datasets adopted in the literature and concludes with a discussion of the essential baselines required for any CL evaluation.

3.5.1 Notation

In CL, a model f_θ with parameters θ is trained sequentially on a series of T tasks, denoted as $\mathcal{T}_1, \ldots, \mathcal{T}_T$. Each task consists of pairs of input data with the associated label $\mathcal{T}_i = (\mathcal{X}_i, \mathcal{Y}_i) = \{(x^i_{(j)}, y^i_{(j)})\}_{j=1}^{|\mathcal{T}_i|}$. While the data points within each task are assumed to be *i.i.d.*, the overall training process violates this assumption due to shifts in data distribution between tasks. Moreover, in Task-IL and Class-IL scenarios every task has its own set of classes, meaning that $\mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset$ for $i \neq j$.

The goal of CL is to minimize the cumulative risk across all tasks:

$$\mathcal{L}_{\text{CL}} \triangleq \sum_{i=1}^{T} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{T}_i} \big[\mathcal{L}(f_\theta(x), y)\big], \tag{3.1}$$

where $\mathcal{L}$ represents the classification loss (*e.g.*, categorical cross-entropy) based on the model's prediction $f_\theta(x)$ and the ground truth y.

While the objective is to find the optimal parameters $\theta^* = \arg\min_\theta \mathcal{L}_{\text{CL}}$, the learner can only access data from one task at a time and cannot jointly optimize the model across all data.

3.5.2 Metrics

Since CL experiments involve learning across multiple tasks, various evaluation metrics have been introduced in the literature to capture different aspects of the learning dynamics and better understand the model's behavior. Let a_i^t represent the model's accuracy on the i$^{\text{th}}$ task after training on task $\mathcal{T}_t$. Based on this notation, the following metrics have been defined:

- **Final Average Accuracy (FAA)**, also referred to as ***Last* Accuracy** or Final Accuracy (FA), evaluates the model's final performance on the entire classification problem after incrementally learning all T tasks:

$$\text{FAA} \triangleq \sum_{i=1}^{T} a_i^T, \tag{3.2}$$

where the $\sum\!\!\!\!\!\!-$ symbol denotes the arithmetic mean operation. This metric offers a concise summary of the trade-off between learning tasks incrementally and training them jointly, enabling direct comparisons with the *i.i.d.* baseline (see Section 3.5.4). Due to its simplicity and effectiveness, FAA is widely used in the literature [91, 39, 7] and serves as the primary performance indicator in this thesis.

- **Final Average Incremental Accuracy (FAIA)** [112, 67], takes into account the accuracy at the end of each task, thus providing an indicator of the performance during the whole training sequence:

$$\text{FAIA} \triangleq \sum_{t=1}^{T} \sum_{i=1}^{t} a_i^t. \tag{3.3}$$

  This metric usually yields slightly higher values than FAA, as during the first few tasks forgetting is less pronounced, and for short sequences FAIA may not be a good indicator of the model's performance against forgetting.

- **Final Average Forgetting (FAF)** [22, 24, 23], commonly referred to as *forgetting*, measures the average performance degradation occurring on past tasks, as the difference between the peak accuracy obtained on that task and its last value:

$$\text{FAF} \triangleq \sum_{i=1}^{T-1} f_i, \quad \text{s.t.} \quad f_i = \left(\max_{l \in \{i,\dots,T-1\}} a_i^l \right) - a_i^T. \tag{3.4}$$

  This measure can yield a negative value if the model improves its accuracy on past tasks over time, a phenomenon referred to as positive *backward transfer*. However, this metric does not account for models that may exhibit low accuracy on a task immediately after learning it. As a result, it is possible to observe zero or negative forgetting, even in cases where the model never properly learned the task.

- **Final Average Adjusted Forgetting (FAAF)** is an adjusted version of FAF that was first proposed in the original paper for the approach presented in Chapter 5. This variant aims to facilitate the comparison between unevenly performing approaches by focusing on performance degradation alone and excluding backward transfer:

$$\begin{aligned} \text{FAAF} &\triangleq \sum_{i=1}^{T-1} \left[\frac{a_i^* - a_i^T}{a_i^*} \right]^+, \\ \text{s.t.} \quad a_i^* &= \max_{l \in \{i,\dots,T-1\}} a_i^l, \quad \forall i \in \{1, \dots, T-1\}, \end{aligned} \tag{3.5}$$

  where $[\cdot]^+$ indicates lower-bound clipping to zero. While FAF is upper-bounded by a model's peak accuracy, FAAF varies in $[0, 100]$: 100 denotes a method that retains no accuracy on previous tasks (maximum forgetting), and 0 one that has no performance decrease on past tasks.

3.5.3 Datasets

Designing an experimental setup for CL research requires tasks with shifting data distributions. In particular, for Class-IL and Task-IL scenarios, tasks should feature disjoint sets of classes. This is typically achieved by selecting an existing dataset and partitioning it into tasks, where each task includes a distinct subset of classes. The most commonly used datasets in CL research are:

- **Sequential MNIST (S-MNIST)** [128] is obtained by splitting the popular MNIST image classification dataset [86] into 5 tasks with 2 classes each. The dataset consists of 28×28 grayscale images representing handwritten digits, and each task includes approximately 6000 training images and 1000 test images. Although S-MNIST is a pioneering and simple benchmark, it has faced criticism for not being fully representative of modern computer vision tasks [142, 49].

- **Rotated MNIST (R-MNIST)** [91] is a Domain-IL classification benchmark obtained by applying random rotations to MNIST [86] images, with an angle sampled in the $[0, \pi)$ interval. Each task is associated with a single angle, including approximately 60000 training images and 10000 test images, in which the model should learn to recognize the digits. This dataset is designed to evaluate the model's ability to adapt to changes in the input data distribution, rather than adding new classes.

- **Sequential CIFAR-10 (S-CIF10)** [147] is a benchmark constructed from the CIFAR-10 dataset[83], which features 32×32 RGB images of natural objects such as animals and vehicles. It is organized into 5 tasks, each containing 2 classes. For each class, there are 5000 training images and 1000 test images.

- **Split CIFAR-100 (S-CIF100)** [147, 112, 25] is obtained by splitting the CIFAR-100 dataset [83] into 10 consecutive tasks, each comprising of 10 classes. The dataset consists of 32×32 RGB images depicting a diverse range of objects, including animals, vehicles, and household items. Each class features 500 training images and 100 test images.

- **Sequential *mini*ImageNet (S-*mini*Img)** [25, 46, 43] is obtained from *mini*ImageNet [131], a 100-class subset of the popular ImageNet dataset [42]. split in 20 classification tasks of 5 classes each. Each class presents 500 and 100 training and test 84×84 RGB images respectively.

- **Split Imagenet-R (S-Imagenet-R)** [137] is derived form the ImageNet-Rendition dataset [65], splitting the 224×224 RGB images into 10 tasks of 20 classes each. These images depict various artistic renditions, including art, cartoons, graffiti, embroidery, origami, paintings, sculptures, sketches, tattoos, toys, and video game representations of the original ImageNet classes, with a total of 24000 and 6000 training and test images respectively.

- **Split CUB-200 (S-CUB)** [47] derives from a sequential split of the Caltech-UCSD Birds-200 fine-grained visual classification dataset [133]. The dataset is divided into 10 tasks, each containing 20 classes, with only 30 training and test images per class. The images are 224×224 RGB pictures of birds in various poses and environments, and each class is associated with a unique bird species.

- **Split Cars-196 (S-Cars)** [149] is obtained by splitting the Stanford Cars dataset [82] into 10 tasks: the first 9 tasks contain 20 classes each, while the last task includes the remaining 16 classes. Each class corresponds to a specific car make, model, and year, with a total of 8144 training and 8041 test 224×224 RGB images.

- **Split EuroSAT (S-EuroSAT)** [75] is derived from the EuroSAT dataset [63, 64] and consists of 5 tasks, each containing 2 classes. The dataset comprises 27000 64×64 RGB images of land use and land cover classes, obtained from Sentinel-2 satellite imagery.

- **Split RESISC45 (S-RESISC)** [75] is a benchmark constructed from the dataset of Remote Sensing Image Scene Classification (RESISC). It contains 256×256 images, covering 45 scene classes with 700 images in each class, split into 9 balanced tasks. These images contain varying spatial resolution, ranging from 20cm to more than 30m per pixel.

- **Split CropDiseases (S-Crop)** [75] is derived from the PlantVillage dataset [68] regarding healthy/infected leaves of crops plants. It contains 54000 images of plant leaves, with resolutions ranging from 256×256 to 512×512 pixels. From the original 38 classes, based on species and disease combinations, the classes with the lowest number of examples ("*Potato healthy*", "*Apple Cedar*" and "*Peach healthy*") were excluded. The remaining 35 classes are split into 7 tasks, each containing 5 classes.

- **Split ISIC (S-ISIC)** [75] is obtained from the ISIC 2018 dataset [33], which contains 10015 high-quality images of dermoscopic images aimed at advancing automated skin lesion analysis, particularly for melanoma detection. To balance the number of images per class, the most frequent class "*Melanocytic nevus*" was excluded. The remaining 6 classes are split into 3 tasks of 2 classes each.

- **Split ChestX (S-ChestX)** [75] derives from the ChestX-ray8 dataset [136], which contains frontal-view X-ray images about patients chests with common thoracic disease. The 2 tasks of 3 classes each split is composed of the images without overlapping diseases belonging to the classes "*Cardiomegaly*", "*Consolidation*", "*Edema*", "*Fibrosis*", "*Pleural Thickening*" and "*Pneumothorax*".

3.5.4 Baselines

In CL experiments, two essential baselines serve as fundamental references for evaluating the performance of proposed methods. These baselines represent opposite ends of the learning spectrum, providing insights into the challenges of catastrophic forgetting and the ideal upper-bound performance.

Fine-tuning involves sequentially training a model on tasks one at a time, without any access to data from previous tasks. This approach simulates a naive learning process, where the model adapts to each new task but risks overwriting prior knowledge, resulting in catastrophic forgetting. As a result, fine-tuning often highlights the limitations of DNNs in retaining information over time, making it an important lower-bound baseline.

Joint training, on the other hand, represents the ideal scenario where all tasks are learned simultaneously. By training the model on the combined dataset, joint training eliminates the problem of forgetting, offering a viable upper-bound for performance, the maximum achievable performance when all data is available. Although unrealistic, this upper limit can theoretically be overcome by CL methods able to fully exploit the learning sequence achieving positive forward transfer and zero forgetting.

Together, these baselines provide a comprehensive framework for evaluating CL methods, helping to measure how closely new approaches can approximate the upper-bound performance of joint training while overcoming the limitations of simply fine-tuning the backbone.

Part II

Latent Space Regularization

Chapter 4

Spectral Geometry

4.1 Motivation

In the 'train from scratch' CL paradigm, the literature has increasingly favored the employment of *rehearsal* methods. These methods perform better than *architectural* and *regularization* approaches in the Class-IL scenario. However, while rehearsal methods effectively allow learners to track the joint distribution of all seen classes, their limited buffer size introduces various overfitting issues. Such problems include divergent gradients for new classes [19, 16], deteriorating decision surfaces [13], and predictive bias accumulation for current classes [140, 3]. Addressing these limitations is the focus of several recent studies.

This chapter aims to analyze the geometry of the latent space of a rehearsal-based CL model and, with the aid of spectral geometry, proposes a novel regularization term to improve performance. Specifically, this chapter investigates the changes occurring in the model's latent space as tasks progress. Given the Riemannian nature of the latent space of DNNs [8], spectral geometry is adopted to study its evolution. Observations reveal that the learner struggles to separate the latent projections of replay examples belonging to different classes, making the downstream classifier prone to interference whenever input distributions change and representations are perturbed. Consequently, this chapter introduces a loss term to endow the model's latent space with a cohesive structure without constraining individual coordinates. As illustrated in Fig. 4.1, the proposed approach, called CaSpeR-IL, leverages graph-spectral theory to promote well-separated latent embeddings. The proposal reduces the intrinsic bias of rehearsal methods by enforcing a desirable property on the model's latent space through a geometrically motivated regularization term. It can be seamlessly combined with any rehearsal-based CL method to improve accuracy and robustness against forgetting.

A strain of recent CL approaches similarly conditions the model's representation to facilitate clustering by means of a contrastive regularization object-

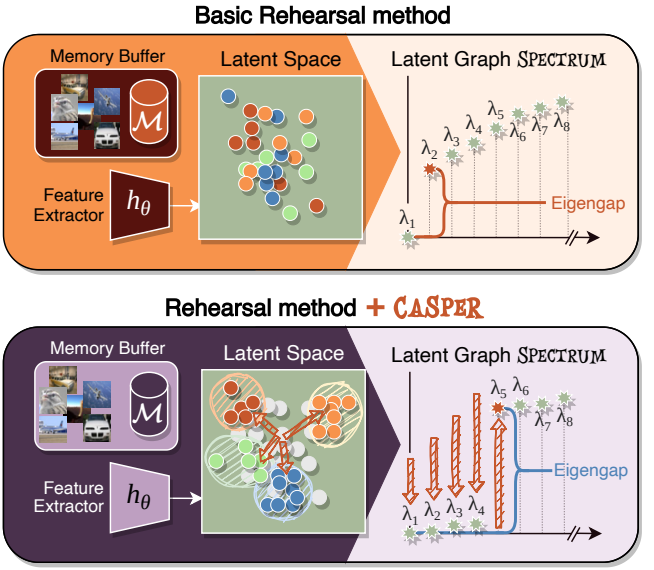
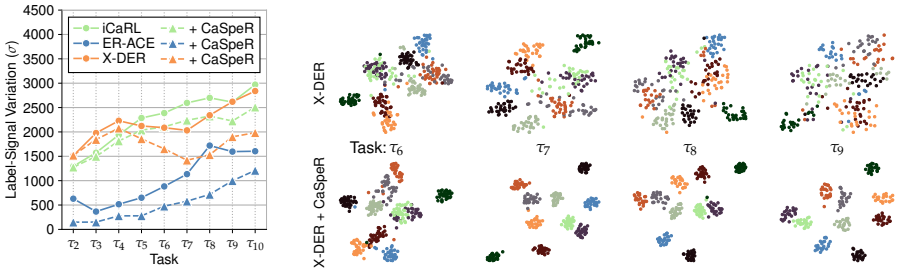


Figure 4.1: An overview of the proposed CaSpeR-IL regularizer. Rehearsal-based CL methods struggle to separate the latent-space projections of replay data points. Continual Spectral Regularizer for Incremental Learning (CaSpeR-IL) acts on the spectrum of the latent geometry graph to induce a partitioning behavior by maximizing the *eigengap* for the number of seen classes.

ive [94, 9, 10]. In the experimental section, this chapter compares the proposed approach against representatives of this family of methods. The results provide insights into how distinct formulations of similar clustering objectives lead to the emergence of different characteristics in latent space geometry.
In summary, the main contributions of this chapter include:

- studying interference in rehearsal CL models by investigating the geometry of their latent space.
- proposing CaSpeR-IL, a simple geometrically motivated loss term that induces the continual learner to produce well-organized latent embeddings.
- validating the proposal by combining it with several state-of-the-art (SOTA) rehearsal-based CL approaches, demonstrating its effectiveness across a wide range of evaluations.
- comparing CaSpeR-IL against recent contrastive-based incremental strategies, highlighting its superior synergy with CL models.
- presenting additional studies to further investigate the geometric properties conferred by the method on the model's latent space.



(a) (b)

Figure 4.2: How CL alters a model's latent space. (a) A quantitative evaluation measured as Label-Signal Variation (σ) within the LGG for buffer data points (*lower is better*); (b) TSNE embedding of the features computed by X-DER for buffered examples in later tasks (top). Interference between classes is visibly reduced if CaSpeR-IL is applied (bottom). All experiments are carried out on S-CIF100, (a) with buffer size 500, (b) with 2000.

4.2 Analysis of changing Latent Space Geometry

How does the latent space change with a novel task on the input stream?

Rehearsal-based CL methods store a pool of examples from previous tasks in a buffer B with fixed size m. To answer the aforementioned question, the graph $\mathcal{G}$ is computed over the latent-space projection of the replay examples collected by the CL model after training on $\mathcal{T}_i$ ($i \in \{2, ...T\}$)[1]. To measure the sparsity of the latent space with respect to class representations, the Label-Signal Variation σ [85] is computed on the adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{m \times m}$ of $\mathcal{G}$:

$$\sigma \triangleq \sum_{i=1}^{m} \sum_{j=1}^{m} \mathbb{1}_{y_i \neq y_j} a_{i,j} \,, \tag{4.1}$$

where $\mathbb{1}_{\cdot}$ is the indicator function[2], $a_{i,j}$ is the (i,j)-th element of $\boldsymbol{A}$, and y_i is the class label of the i-th example. The Label-Signal Variation σ represents the sum of the adjacency matrix elements corresponding to different-class pairs of examples. A lower σ indicates a more structured latent space, where examples from different classes are less entangled. In Fig. 4.2a several rehearsal CL methods are evaluated, showing a steadily increasing σ. This trend suggests that examples from distinct classes become more entangled in later tasks. A qualitative observation, using a TSNE embedding of the points in B shown in Fig. 4.2b, further highlights this effect, as distances between different-class examples decrease over time. Both evaluations demonstrate improvement when CaSpeR-IL is applied to the evaluated methods.

[1] See Section 4.3 for a detailed description of this procedure.

[2] The indicator function $\mathbb{1}_{\cdot}$ is equal to 1 if the condition $\cdot$ is true, and 0 otherwise.

4.3 Continual Spectral Regularizer

The method is built upon the eigendecomposition of the Laplace operator on a graph, situating it within the broader area of spectral graph theory. Specifically, it can be regarded as an *inverse* spectral technique, as it prescribes the general behavior of certain eigenvalues and seeks a graph whose Laplacian spectrum aligns with this behavior. In the geometry processing area, such approaches are referred to as *isospectralization* techniques. These techniques have been applied in diverse areas, including deformable shape matching [35], shape exploration and reconstruction [95], shape modeling [99], and adversarial attacks on shapes [110]. Unlike these methods, CaSpeR-IL operates on a single graph, as opposed to pairs of 3D meshes, and does not rely on a specific input spectrum as a target. Instead, it emphasizes maximizing the gap between nearby eigenvalues, regardless of their precise values. Since the graph represents a discretization of the latent space of a CL model, this regularization has significant implications for the learning process.

CaSpeR-IL builds upon the fact that the latent spaces of neural models contain structures that are informative of the data space on which they are trained [117]. This structure can be enforced through loss regularizers; for example, in [34], a minimum-distortion criterion is applied to the latent space of a VAE for a shape generation task. Following a similar line of thought, a geometric term is proposed to regularize the latent representations of a CL model. The approach is rooted in spectral geometry, motivated by the pursuit of a **compact representation** characterized by **isometry invariance**. As shown in [8], the latent space of DNNs can be modeled as a Riemannian manifold whose *extrinsic* embedding is encoded in the latent vectors. Being extrinsic, these vectors represent absolute coordinates encoding only one possible realization of the data manifold, among infinitely many possible isometries. Each isometry (*e.g.* a rotation by 45°) would encode the **same latent space**, but the **latent vectors would change**, a property that may lead to overfitting and lack of generalization. By leveraging spectral geometry, the method instead relies on *intrinsic* quantities that fully encode the latent space and are isometry-invariant.

The regularizer is based on the graph-theoretic formulation of clustering, aiming to partition the vertices of $\mathcal{G}$ into well-separated subgraphs with high internal connectivity. A body of results from spectral graph theory, dating back to [26, 122, 119], explains the gap occurring between neighboring Laplacian eigenvalues as a quantitative measure of graph partitioning. The proposed method draws on these results but transforms the *forward* problem of computing the optimal partitioning of a given graph into the *inverse* problem of seeking a graph with the desired partitioning. In Algorithm 1 the reader can find the steps to compute CaSpeR-IL's loss.

Building the graph. The parameters θ of a model f_θ include both the weights of the feature extractor and the classifier, θ^f and θ^c respectively. Examples in B are forwarded through the network, and their features are used to build a

Algorithm 1 CaSpeR-IL Loss Computation. The *BalancedSampling* function extract b examples from the buffer belonging to p different classes uniformly.

Require: Memory buffer B of saved samples, with size m; batch-size b; hyperparameter of number of classes to sample p

$\boldsymbol{x}^b \leftarrow \text{BalancedSampling}(B, p)$ ▷ Data extraction from buffer

$\boldsymbol{z}^b \leftarrow f_\theta^f(\boldsymbol{x}^b)$ ▷ Get features

$\boldsymbol{\mathcal{D}} = \{D_{i,j} = \|z_i^b - z_j^b\| \text{ with } i, j = 1, ..., m\}$ ▷ Calculate distance matrix

$\boldsymbol{A} \leftarrow k\text{-NN}(\mathcal{D})$ ▷ Retrieve adjacency matrix

$\boldsymbol{D} \leftarrow \text{diag}(\sum_i^m a_{1,i}, \sum_i^m a_{2,i}, ..., \sum_i^m a_{b,i})$ ▷ Calculate degree matrix

$\boldsymbol{L} \leftarrow \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$ ▷ Compute normalized laplacian

$\boldsymbol{\lambda} \leftarrow \text{Eigenvalues}(\boldsymbol{L})$ ▷ Retrieve eigenvalues

$\ell_{\text{CaSpeR}} \leftarrow -\lambda_{p+1} + \sum_{j=1}^{p} \lambda_j$ ▷ Casper loss

Output ℓ_{CaSpeR}

k-Nearest Neighbors (k-NN) graph $\mathcal{G}$; following [85], this is referred to as the **Latent Geometry Graph (LGG)**. Specifically, after identifying the k nearest neighbors of each point, the resulting adjacency matrix is symmetrized. As a result, the number of neighbors may exceed k.

Spectral Regularizer. Let $\boldsymbol{A}$ denote the adjacency matrix of $\mathcal{G}$. The degree matrix $\boldsymbol{D}$ is computed, and the normalized Laplacian is defined as: $\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$, where $\boldsymbol{I}$ represents the identity matrix. The eigenvalues $\boldsymbol{\lambda}$ of $\boldsymbol{L}$ are then computed and sorted in ascending order. Let g represent the number of different classes within the buffer. The regularizing loss is calculated as:

$$\ell_{\text{CaSpeR}} \triangleq -\lambda_{g+1} + \sum_{j=1}^{g} \lambda_j \,. \tag{4.2}$$

The proposed loss term is weighted using the hyperparameter ρ and optimized with gradient descent. This formulation increases the eigengap $\lambda_{g+1} - \lambda_g$ while minimizing the first g eigenvalues. Since the number of eigenvalues near zero corresponds to the number of loosely connected partitions within the graph [87], the loss indirectly encourages data points to be clustered without strict supervision.

Efficient Batch Operation. The application of CaSpeR-IL involves the computationally demanding task of constructing the entire LGG $\mathcal{G}$ at each forward step by processing all available replay examples in the buffer B, which is typically orders of magnitude larger than a batch of input examples. To address this challenge, an efficient approximation of the initial objective is proposed. Instead of operating directly on $\mathcal{G}$, a random sub-graph $\mathcal{G}_\text{p} \subset \mathcal{G}$ is sampled, spanning only p out of the g classes represented in the memory buffer. Since $\mathcal{G}_\text{p}$ still contains a substantial number of nodes, an additional sub-sampling step is applied to extract $\mathcal{G}_\text{p}^t \subset \mathcal{G}_\text{p}$, a smaller graph with t exemplars for each class. By repeating these random samplings during each forward step, a Monte Carlo approximation

of Eq. (4.2) is optimized:

$$\ell^*_{\text{CaSpeR}} \triangleq \mathop{\mathbb{E}}_{\mathcal{G}_\text{p}\subset\mathcal{G}} \left[\mathop{\mathbb{E}}_{\mathcal{G}^t_\text{p}\subset\mathcal{G}_\text{p}} \left[-\lambda^{\mathcal{G}^t_\text{p}}_{p+1} + \sum_{j=1}^{p} \lambda^{\mathcal{G}^t_\text{p}}_j \right] \right], \tag{4.3}$$

where the $\boldsymbol{\lambda}^{\mathcal{G}^t_\text{p}}$ represents the eigenvalues of the Laplacian of $\mathcal{G}^t_\text{p}$. The eigengap is enforced at p, given that each $\mathcal{G}^t_\text{p}$ is constructed with samples from p communities within $\mathcal{G}$. In practice, b samples are extracted from the buffer, maintaining b equal to the batch size to ensure balance [17, 18]. This results in $t = \frac{b}{p}$ samples from each of the p randomly chosen classes.

Hyperparameters. Although CaSpeR-IL introduces three hyperparameters, the degrees of freedom to grid-search them effectively reduce to two: the regularization weight ρ, which is independent, and the combination of p, the number of classes sampled from the buffer, and k, the number of nearest-neighbors for the adjacency matrix. Specifically, the product of p and k should be equal to (or slightly lower than) the number of extracted samples, which in the experiments matches the batch size.

4.4 Experiments

4.4.1 Benchmarks

Settings. The effectiveness of the proposed method is assessed using the Class-IL classification protocol [128], where the model makes predictions without task information, a benchmark recognized as both realistic and challenging [49, 7]. Results are also reported for the Task-IL and Domain-IL protocols, demonstrating that CaSpeR-IL enhances CL baselines within these scenarios as well.

Datasets. Experiments for Class-IL and Task-IL are conducted on three commonly used image datasets, where the classes from each main dataset are split into disjoint sets for sequential training: **Sequential CIFAR-10 (S-CIF10)** (5 tasks of 2 classes each), **Split CIFAR-100 (S-CIF100)** (10 tasks of 10 classes each), and **Sequential *mini*ImageNet (S-*mini*Img)** (20 tasks of 5 classes each). For Domain-IL, the dataset **Rotated MNIST (R-MNIST)** (20 tasks with changing input distribution) is employed. Details of these datasets are provided in Section 3.5.3.

Metrics. Model performance is primarily quantified using **Final Average Accuracy (FAA)**, reported as a percentage value. To evaluate the severity of performance degradation caused by catastrophic forgetting, the **Final Average Adjusted Forgetting (FAAF)** metric is adopted. Explanations of these metrics are provided in Section 3.5.2.

Benchmarked models. To evaluate the benefit of CaSpeR-IL, it is applied on top of several SOTA rehearsal-based methods: **ER-ACE** [19], **iCaRL** [112],

Class-IL	**S-CIF10**		**S-CIF100**		**S-*mini*Img**	
Joint (UB)	87.1 (–)		63.1 (–)		52.8 (–)	
Finetune (LB)	19.5 (100.0)		8.4 (100.0)		3.9 (100.0)	
Buffer Size	500	1000	500	2000	2000	5000
ER-ACE	66.1 (21.8)	71.7 (14.9)	35.0 (51.4)	46.5 (34.6)	22.0 (49.0)	27.3 (30.0)
+ **CaSpeR-IL**	69.6 (20.6)	73.8 (14.1)	36.7 (46.6)	47.9 (33.9)	23.4 (47.9)	29.2 (28.4)
iCaRL	52.7 (22.7)	62.9 (21.6)	39.6 (32.7)	40.5 (31.2)	19.4 (36.9)	20.2 (33.2)
+ **CaSpeR-IL**	55.7 (20.6)	64.0 (21.1)	40.9 (32.3)	41.8 (25.6)	20.5 (35.9)	21.4 (32.3)
DER++	67.4 (26.8)	71.2 (25.1)	28.0 (57.6)	43.3 (34.9)	20.9 (74.5)	28.6 (61.0)
+ **CaSpeR-IL**	69.1 (26.2)	73.1 (23.4)	32.2 (53.4)	47.0 (30.1)	22.6 (71.0)	30.0 (57.6)
X-DER	63.2 (15.0)	65.7 (12.3)	35.9 (44.5)	46.4 (23.6)	24.8 (44.7)	31.0 (30.1)
+ **CaSpeR-IL**	65.6 (14.4)	67.8 (10.7)	38.2 (43.9)	48.1 (18.5)	26.2 (41.7)	31.6 (28.7)
PODNet	37.2 (40.5)	46.0 (39.5)	30.2 (54.5)	32.1 (46.7)	16.8 (52.3)	20.8 (46.5)
+ **CaSpeR-IL**	39.9 (39.5)	47.4 (38.9)	32.3 (48.3)	38.6 (35.6)	18.1 (50.3)	23.6 (45.1)

Table 4.1: Class-IL results – FAA (FAAF) – for SOTA rehearsal CL methods, with and without CaSpeR-IL.

Task-IL	**S-CIF100**		**S-*mini*Img**	
Joint (UB)	88.81 (–)		87.39 (–)	
Finetune (LB)	30.10 (62.84)		24.05 (67.37)	
Buffer Size	500	2000	2000	5000
ER-ACE	73.86 (10.73)	80.69 (5.37)	69.05 (13.72)	72.78 (8.93)
+ **CaSpeR-IL**	75.14 (4.91)	81.57 (4.93)	69.59 (13.05)	74.14 (8.12)
iCaRL	78.38 (5.38)	78.47 (4.91)	70.35 (3.92)	70.44 (2.68)
+ **CaSpeR-IL**	79.31 (4.61)	79.43 (3.41)	71.19 (3.67)	71.93 (3.65)
DER++	70.55 (11.12)	78.60 (5.96)	69.78 (13.37)	73.81 (8.59)
+ **CaSpeR-IL**	73.25 (9.49)	80.78 (3.04)	70.97 (11.75)	75.18 (7.93)
X-DER	77.28 (2.43)	82.55 (0.92)	74.32 (4.95)	77.70 (3.71)
+ **CaSpeR-IL**	78.26 (5.47)	83.77 (0.27)	75.99 (3.88)	78.71 (2.32)
PODNet	67.37 (19.76)	69.63 (15.16)	60.60 (14.00)	66.15 (10.71)
+ **CaSpeR-IL**	70.81 (15.26)	71.90 (11.32)	64.84 (10.01)	70.85 (7.99)

Table 4.2: Task-IL results – FAA (FAAF) – for SOTA rehearsal CL methods, with and without CaSpeR-IL.

DER++ [17], **X-DER**[3] [14] and **PODNet** [45]. An explanation of these methods is provided in Section 3.4. The performance of the upper bound training on all classes jointly in an offline manner (*Joint*) and the lower bound training on each task sequentially without any method to prevent forgetting (*Finetune*) are also included.

Implementation details. The experiments adopt a batch size of 64 examples, with the same number used for sampling data from the memory buffer, always extracting 64 data points. For experiments on S-CIF10, S-CIF100, and R-MNIST, training is conducted for 20 epochs per task without a learning rate scheduler. For S-*mini*Img, training is conducted for 50 epochs per task, applying a learning rate decay of 0.1 at epochs 35 and 45. These choices remain consistent within each method to ensure fairness in comparisons. The same backbone is employed for all experiments on the same dataset. Specifically, ResNet18 [61] is used for S-CIF10, S-CIF100, and R-MNIST, while EfficientNet-B2 [126] is used for S-*mini*Img. The best hyperparameters for each model-dataset configuration are identified through grid search.

4.4.2 Main Results

A breakdown of Class-IL results is presented in Table 4.1. CaSpeR-IL demonstrates consistent improvements in FAA across all evaluated methods and datasets. The steady reduction in FAAF confirms that the adopted regularization effectively addresses catastrophic forgetting.

Improvements in accuracy do not always scale with increases in memory buffer size. This behavior contrasts with the typical patterns observed in replay regularization terms [21, 25]. This effect is attributed to the distinctively geometric approach employed; as spectral properties of graphs are robust with respect to coarsening [73], CaSpeR-IL remains effective even with a smaller data pool.

In Task-IL and Domain-IL (results presented in Table 4.2 and Table 4.3, respectively), the improvements are less pronounced compared to Class-IL. Existing methods already perform strongly in these less challenging scenarios. Nevertheless, CaSpeR-IL provides consistent improvements, demonstrating its ability to consolidate task-specific knowledge in Task-IL and mitigate biases introduced by new data distributions in Domain-IL.

4.4.3 Comparison with Contrastive Learning.

The thorough study in [60] interprets contrastive learning as a parametric form of spectral clustering on the input augmentation graph, which points to a link with the proposed approach. Given the similarity between CaSpeR-IL's goal and contrastive objectives, Supervised Contrastive Replay (SCR) [76] and Cross-Space Clustering and Controlled Transfer (CSCCT) [10], two existing CL contrastive baselines described in Section 3.4, are compared against the proposed regularizer.

[3]Specifically, the more effective baseline based on a Regular Polytope Classifier [105].

Domain-IL	**Rotated MNIST**	
Joint (UB)	95.76	
Finetune (LB)	67.66	
Buffer Size	200	500
ER-ACE	83.45	86.86
+ **CaSpeR-IL**	86.19	88.11
DER++	89.91	92.00
+ **CaSpeR-IL**	90.96	93.21

Table 4.3: FAA values on R-MNIST, with only the approaches compatible with Domain-IL.

The methods were evaluated on the Class-IL Split CIFAR-100 benchmark. SCR operates as a standalone model extending Experience Replay, whereas CSCCT functions as a module that can be integrated into existing CL methods. Consequently, CSCCT is regarded as a direct competitor implemented upon the different baselines. Despite similarities to CaSpeR-IL, CSCCT requires a past model snapshot during training and incorporates both streaming and memory data within its loss terms.

Results are presented in Table 4.4. To further investigate the effects of different CL approaches on the latent space, the average variance of same-class projections at the end of training was measured.

The findings indicate that SCR exhibits higher variance in the latent space compared to other baselines. In contrast, both CSCCT and CaSpeR-IL effectively reduce the latent-space variance of the models they are applied to. The results reveal an intriguing behavior: although CSCCT often achieves minimal variance, the accuracy improvement associated with CaSpeR-IL remains higher. Two potential explanations are suggested for these observations: *i)* intra-class variance in latent space does not directly correlate with accuracy; *ii)* while directly constraining individual coordinates may limit the model's ability to reorganize data points, spectral geometry may serve as a softer clustering approach, allowing the flexibility required to organize the latent space into more optimal structures for classification tasks.

4.4.4 Additional Experiments

As highlighted in the literature [7, 49, 17], comparing results from different CL studies is non-trivial due to the influence of even subtle changes in the experimental setting, which should always be consistent across all evaluated methods to ensure fairness. For this reason, experiments were conducted from scratch, applying the same conditions to all evaluated methods, rather than relying on pre-existing results. Additional experiments with different training

Class-IL	**Split CIFAR-100**			
Buffer Size	500		2000	
	FAA	Variance	FAA	Variance
SCR	31.18	2.2111	43.39	4.4439
ER-ACE	34.99	0.5313	46.52	0.5769
+ **CaSpeR-IL**	**36.70**	0.4926	**47.85**	0.5478
+ **CSCCT**	34.93	**0.3931**	45.91	**0.4290**
iCaRL	39.56	0.8381	40.47	0.8248
+ **CaSpeR-IL**	**40.57**	**0.8289**	**41.83**	**0.8057**
+ **CSCCT**	39.36	0.9167	40.87	1.0392
DER++	28.01	0.1283	43.27	0.1209
+ **CaSpeR-IL**	**32.16**	0.0964	**46.95**	0.1012
+ **CSCCT**	30.17	**0.0552**	44.27	**0.0857**
X-DER	35.89	0.2265	46.37	0.2523
+ **CaSpeR-IL**	**38.23**	0.2065	**48.11**	**0.2207**
+ **CSCCT**	36.23	**0.1974**	45.51	0.2242
PODNet	30.16	0.4229	32.12	0.7366
+ **CaSpeR-IL**	**32.27**	0.4197	**38.64**	0.5700
+ **CSCCT**	30.78	**0.1809**	33.59	**0.2577**

Table 4.4: Comparison with contrastive baselines. FAA and the average variance of same-class projections in the latent space are reported.

configurations are presented to further validate the effectiveness of CaSpeR-IL.

Longer training-phase. The setting described in [14] for S-CIF100 is replicated: batch size is set at 32, training lasts for 50 epochs per task, and a learning rate decay of 0.1 is applied at epochs 35 and 45. Results, shown in Table 4.5, demonstrate that CaSpeR-IL effectively improves different baselines regardless of the experimental setting.

Different batch-sizes. In Table 4.6, additional results for S-CIF100 (Class-IL) indicate that the effectiveness of CaSpeR-IL is not compromised by the reduced batch size (compared to 64 used in Section 4.4.2).

4.4.5 Continual Semi-Supervised Learning

In a supervised CL setting, CaSpeR-IL is applied to buffer data points to encourage the separation of all previously encountered classes in the latent space. The proposed approach, however, does not impose strict supervision requirements, as it does not depend on the labels attached to each node in the LGG. Instead, it only requires knowledge of the total number of classes g that

Class-IL	**S-CIF100**	
Buffer Size	500	2000
ER-ACE	37.12	49.20
+ **CaSpeR-IL**	38.40	50.38
iCaRL	47.47	52.68
+ **CaSpeR-IL**	48.81	54.44
DER++	36.28	50.66
+ **CaSpeR-IL**	40.44	52.22
X-DER	46.96	55.85
+ **CaSpeR-IL**	48.43	56.40
PODNet	36.44	43.97
+ **CaSpeR-IL**	40.63	46.80

Table 4.5: Class-IL FAA values on S-CIF100, with 50 epochs per task and batch size 32 (as for [14]).

Class-IL	**S-CIF100**	
Batch Size	16	32
ER-ACE	35.70	36.47
+ **CaSpeR-IL**	36.96	38.63
iCaRL	43.12	43.99
+ **CaSpeR-IL**	44.22	45.20
DER++	28.91	31.14
+ **CaSpeR-IL**	31.93	33.20
X-DER	33.02	39.19
+ **CaSpeR-IL**	35.42	40.46
PODNet	34.29	35.22
+ **CaSpeR-IL**	36.15	37.02

Table 4.6: Class-IL results – FAA – in S-CIF100 for different batch size configurations (20 epochs per task). Buffer size 500.

must be clustered (Eq. (4.2)). Assuming that classes are equally distributed in the data, the *balanced sampling* strategy (explained in Section 4.3) can be approximated through random sampling from the memory buffer. Consequently, CaSpeR-IL does not rely on the availability of annotations for each example, enabling its application to semi-supervised scenarios to improve accuracy and convergence.

[16] introduces **Continual Semi-Supervised Learning (CSemiSL)**, a new CL experimental benchmark in which only a fraction of the examples in the input stream are associated with annotations. Results for an experiment conducted on S-CIF100 in the CSemiSL setting, with only 0.8% or 5% annotated labels, are reported in Table 4.7. Typical CL methods operating under this scenario either discard a significant amount of data (ER-ACE), resulting in substantially reduced performance compared to the fully-supervised case, or utilize the in-training model to annotate unlabeled samples (*pseudo-labeling*, PsER-ACE). The latter approach may backfire if the supervision provided is insufficient for the learner to produce reliable responses, as observed in cases with 0.8% labels.

To leverage unlabeled exemplars, CaSpeR-IL is also applied to data points from the input stream by setting k equal to the number of classes in a given task. This leads to an overall improvement of the tested models and, in particular, mitigates failure cases where PseudoER-ACE is applied with limited annotated data. These results indicate that CaSpeR-IL effectively reduces the impact of noisy labels produced by *pseudo-labeling*.

While the proposed regularizer can operate without full supervision to a moderate extent, it still depends on the availability of supervised training signals.

	CSemiSL	
Labels %	0.8%	5%
ER-ACE	8.46	11.87
+ **CaSpeR-IL**	8.55	14.16
PsER-ACE	2.31	16.35
+ **CaSpeR-IL**	9.69	17.42
CCIC [16]	11.5†	19.5†
+ **CaSpeR-IL**	12.22	20.32

Table 4.7: Class-IL FAA values on S-CIF100, with reduced amount of annotations. Buffer size 2000.

Class-IL	w/o CaSpeR-IL		w/ CaSpeR-IL	
***k*-NN Clsf**	5-NN	11-NN	5-NN	11-NN
ER-ACE	43.73	44.41	46.75+3.02	47.29+2.88
iCaRL	34.86	37.78	36.00+1.14	38.33+0.55
DER++	44.21	44.24	45.75+1.54	46.00+1.76
X-DER	43.44	44.62	49.47+6.03	49.49+4.87
PODNet	21.11	22.60	27.88+6.77	28.94+6.34

Table 4.8: Class-IL FAA values of k-NN classifiers trained on top of the latent representations of replay data points. Results on S-CIF100 for Buffer Size 2000.

Extending the applicability of geometric-based constraints to unsupervised or self-supervised CL scenarios remains an area for future investigation.

4.5 Model Analysis

4.5.1 *k*-NN classification

To further assess whether CaSpeR-IL effectively separates the latent embeddings for examples of different classes, the accuracy of k-NN-classifiers [141] trained on top of the latent representations produced by the methods in Section 4.4.2 is evaluated. Results for 5-NN and 11-NN classifiers, using the final buffer B as a support set, are reported in Table 4.8. The findings indicate that CaSpeR-IL consistently demonstrates beneficial effects in this classification approach, further validating its capability to disentangle representations of different classes.

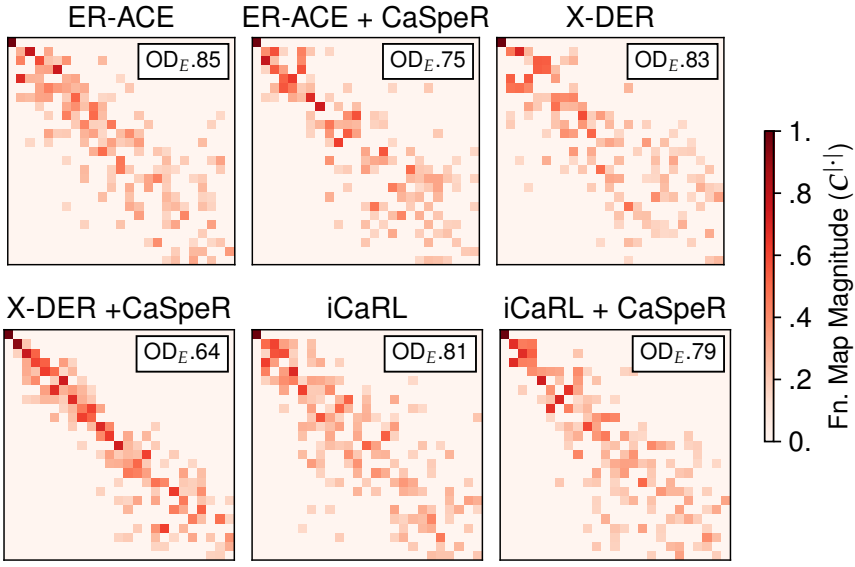


Figure 4.3: For several rehearsal methods with and without CaSpeR-IL, the functional map magnitude matrices $\boldsymbol{C}^{|\cdot|}$ between the LGGs $\mathcal{G}^{\tau_5}$ and $\mathcal{G}^{\tau_{10}}$, computed on the test set of $\tau_1, ..., \tau_5$ after training up to τ_5 and τ_{10} respectively (S-CIF100 - buffer size 2000). The closer $\boldsymbol{C}^{|\cdot|}$ to the diagonal, the less geometric distortion between $\mathcal{G}^{\tau_5}$ and $\mathcal{G}^{\tau_{10}}$. There are reported only the first 25 rows and columns of $\boldsymbol{C}^{|\cdot|}$, focusing on low-frequency correspondences [101], and a $\boldsymbol{C}^{|\cdot|} > 0.15$ threshold is applied to increase clarity.

4.5.2 Latent Space Consistency

To provide further insights into the dynamics of the latent space in the evaluated models, the emergence of distortions in the LGG is analyzed. Given a continual learning model, the focus is on comparing $\mathcal{G}^{\tau_5}$ and $\mathcal{G}^{\tau_{10}}$, the LGGs produced after training on τ_5 and τ_{10}, respectively, computed on the test set of tasks $\tau_1, ..., \tau_5$.

The comparison between $\mathcal{G}^{\tau_5}$ and $\mathcal{G}^{\tau_{10}}$ can be better understood in terms of the node-to-node bijection $T : \mathcal{G}^{\tau_5} \rightarrow \mathcal{G}^{\tau_{10}}$, which can be represented as a functional map matrix $\boldsymbol{C}$ [101] with elements $c_{i,j} \triangleq \langle \boldsymbol{\phi}_i^{\mathcal{G}^{\tau_5}}, \boldsymbol{\phi}_j^{\mathcal{G}^{\tau_{10}}} \circ T \rangle$, where $\boldsymbol{\phi}_i^{\mathcal{G}^{\tau_5}}$ is the i-th Laplacian eigenvector of $\mathcal{G}^{\tau_5}$ (similarly for $\mathcal{G}^{\tau_{10}}$), and $\circ$ denotes the standard function composition. In other words, the matrix $\boldsymbol{C}$ encodes the similarity between the Laplacian eigenspaces of the two graphs. In an ideal scenario where the latent space is subject to no modification between τ_5 and τ_{10} *w.r.t.* previously learned classes, T is an *isomorphism* and $\boldsymbol{C}$ is a diagonal matrix [101]. In a practical scenario, T is only approximately isomorphic, and, the better the approximation, the more $\boldsymbol{C}$ is sparse and funnel-shaped.

In Fig. 4.3 are reported $\boldsymbol{C}^{|\cdot|} \triangleq \text{abs}(\boldsymbol{C})$ for ER-ACE, iCaRL and X-DER on S-

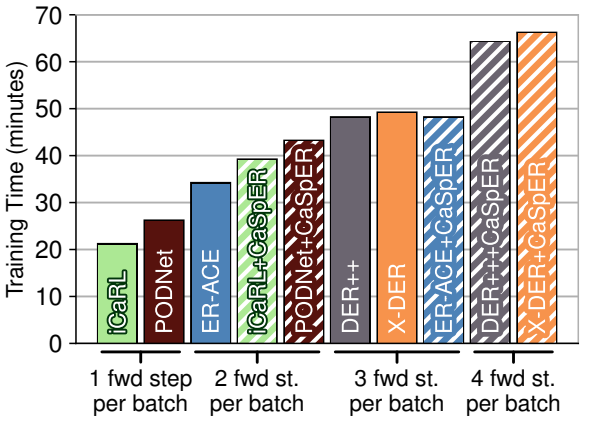


Figure 4.4: Wallclock training time for the approaches evaluated in Section 4.4 benchmarked on an identical hardware setup (GPU NVIDIA V100). Training time grows linearly *w.r.t.* the number of per-batch forward steps.

CIF100, both with and without CaSpeR-IL. It can be observed that the methods that benefit from the proposed regularizer display a tighter functional map matrix. This indicates that the partitioning behavior promoted by CaSpeR-IL leads to reduced interference, as the portion of the LGG that refers to previously learned classes remains geometrically consistent in later tasks. To quantify the similarity of each $\boldsymbol{C}^{|\cdot|}$ matrix to the identity, its off-diagonal energy OD_E [114] is reported, computed as the sum of the elements outside of the main diagonal divided by the Frobenius norm. CaSpeR-IL produces a clear decrease in OD_E, signifying an increase in the diagonality of the functional matrices.

4.5.3 Training Time

As outlined in Section 4.3, CaSpeR-IL involves non-negligible computations that could be perceived as having a significant impact on the overall training time. This section examines the efficiency of the approximated procedure by comparing the wall-clock time between models with and without CaSpeR-IL under identical run conditions.

In Fig. 4.4, the overall training time for the approaches evaluated in Section 4.4.2 is reported, specifically for S-CIF100, benchmarked on an identical hardware setup. The findings indicate that the overhead introduced by CaSpeR-IL is primarily driven by the additional forward step required for the corresponding batch through the model. For X-DER, which already performs three forward steps, a 33% increase in compute time is observed. For ER-ACE, which performs two steps, the increase is approximately 43%. Such overhead appears comparable to that of other CL solutions. For example, ER-ACE combined with CaSpeR-IL

results in a compute time roughly equivalent to X-DER and DER++.

4.6 Conclusions

This chapter examined how rehearsal-based CL models progressively reshape their latent-space representations when encountering new tasks. The investigation revealed that feature projections of previously learned classes tend to collapse toward each other, thus increasing the susceptibility to forgetting. To mitigate this limitation, a novel *spectral* regularizer, named **CaSpeR-IL**, was introduced. By promoting well-separated data clusters in the latent space, CaSpeR-IL improves performance across several standard CL benchmarks. Crucially, CaSpeR-IL differs from coordinate-wise contrastive constraints, operating instead through isometry-invariant quantities derived from the Laplacian spectra of graphs built on top of latent features. Experiments consistently showed that augmenting rehearsal-based CL approaches with CaSpeR-IL delivers more robust representations that exhibit reduced forgetting.

Overall, these findings underscore that leveraging the intrinsic geometry of the latent space is a promising direction for tackling catastrophic forgetting, potentially informing further research on isospectral techniques and spectral geometry approaches for continual learning.

Chapter 5

Online Equivariant Learning

5.1 Motivation

This chapter focuses on the problem of Online Continual Learning (OCL) [5, 23, 19, 91, 7]. In many practical scenarios, a CL model cannot afford multiple passes over the same training samples. Such a constraint is captured by the OCL setting, where each data point in the stream is encountered only once. This challenging yet realistic paradigm reflects real-world applications, where data typically arrives in a single pass and must be learned online. Many studies in CL often assume a relaxed training setup that allows iterating over each task for multiple epochs; under OCL, this assumption no longer holds, highlighting the need for methods that achieve robust performance while processing data on-the-fly.

This thesis focuses on studying the latent space of CL models, with a particular interest in methods that can shape the internal representations to be more robust and adaptable over time. In this context, SSL methods have gained considerable attention due to their capacity to learn flexible and transferable latent-space representations without requiring manual annotations [28, 31, 146]. In CL, many studies have explored the potential of SSL to facilitate knowledge transfer across tasks and reduce catastrophic forgetting [106, 21, 50, 92, 139].

SSL methods can be broadly divided into two classes, according to the type of transformations they exploit:

- **Invariance-based approaches** require the model to become insensitive to specific transformations (*e.g.*, random crops, flips, color jitter), effectively collapsing augmented views of the same input into similar representations [28, 76, 146].

- **Equivariance-based approaches** aim to preserve and reflect transformations in the output representation, rather than discarding them [54, 53]. The model is encouraged to recognize the transformation applied to the

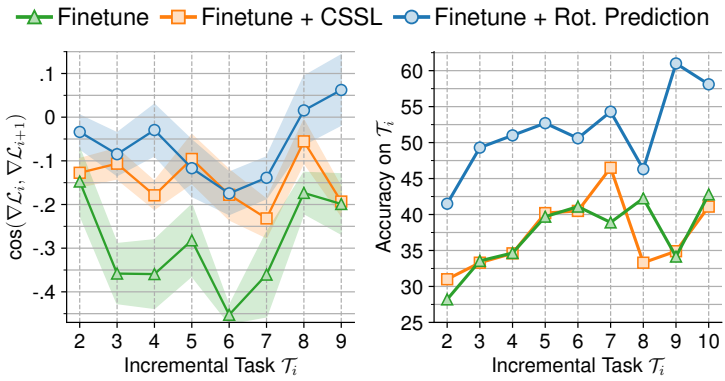
input, such as rotations or jigsaw puzzles, which provide a strong learning signal for representation learning. These additional objectives are often referred to as *pretext tasks*.

In the CL literature, these techniques often rely on contrastive learning [21, 50, 106, 10, 94]. Despite the success of Contrastive Self-Supervised Learning (CSSL) in CL, these methods exhibit notable drawbacks in the OCL setting. First, contrastive objectives typically require large batch sizes and multiple epochs per task to converge [28], in conflict with the single-pass constraint of OCL. Second, when trained with limited replay memory, these methods face a heightened risk of overfitting [13].

Some works in CL have begun exploring rotation tasks to increase batch size in contrastive objectives [59, 139], but the full potential of equivariance for mitigating forgetting remains underexplored. However, recent research suggests that the equivariant paradigm may offer advantages in online scenarios by facilitating faster adaptation and more distributed features [53, 2, 38]. Equivariant pretext tasks, such as four-fold rotation prediction, encourage the network to encode transformation-specific information, potentially yielding latent representations that are rich, redundant, and more resilient to abrupt distribution shifts. Given the limited availability of data in OCL, such tasks can serve as a powerful alternative to contrastive approaches, driving a more sample-efficient training process.

Building on these observations, this chapter introduces **Continual Learning via Equivariant Regularization (CLER)**, an online regularizer that can be easily combined with existing state-of-the-art CL approaches, leading to a generalized improvement in performance. CLER applies an equivariant pretext task to mitigate forgetting in a single-pass streaming environment and, unlike contrastive objectives, is designed for fast convergence and improved sample efficiency, making it particularly suitable for OCL. The main contributions are as follows:

- A novel regularizer for OCL, CLER, which leveraging an equivariant prediction task, helps maintain robust and transferable latent representations.
- Several variants of equivariant objectives are evaluated, each demonstrating consistent performance gains.
- Compatibility with state-of-the-art CL methods, considering both memory-based and memory-free approaches.
- An in-depth study on how CLER facilitates the emergence of distributed and redundant representations, identifying this as a crucial factor for enhancing model robustness in continual learning scenarios.



(a) (b)

Figure 5.1: Effects of SSL in an OCL scenario (S-CIF100) comparing a Finetuning baseline with no additional regularization (*green*), with a Contrastive SSL auxiliary objective (*orange*) and with an equivariant rotation prediction pretext task (*blue*). (a) Cosine similarity between the gradients induced on the model by consecutive tasks $\mathcal{T}_i$ and $\mathcal{T}_{i+1}$ after training on task $\mathcal{T}_i$. (b) Accuracy on task $\mathcal{T}_i$ after training on $\mathcal{T}_i$.

5.2 Invariance *vs.* Equivariance in CL

This chapter tries to demonstrate the benefits of equivariant self-supervised tasks for the OCL scenario. Fig. 5.1 illustrates a preliminary comparison of invariance- and equivariance-based SSL objectives in an OCL setting, using Split CIFAR-100 (S-CIF100) as a benchmark. The baseline (*green*) applies Finetuning without any specialized technique against forgetting. Next, a CSSL regularizer is introduced (*orange*), specifically following the Barlow Twins framework[1] [146]. Finally, an equivariant task (*blue*) is employed as an auxiliary objective, specifically, a four-fold rotation prediction. Panel (a) reports the cosine similarity between gradients of consecutive tasks, providing a measure of *interference* between different tasks in an online scenario. Panel (b) shows the task accuracy achieved immediately after training.

Although both contrastive and equivariant SSL strategies reduce interference *w.r.t.* Finetuning, the rotation prediction pretext task consistently induces more favorable gradient alignments. Furthermore, rotation prediction yields a more noticeable improvement in task-specific accuracy, suggesting that a single pass over the data is insufficient for purely contrastive methods to learn downstream task representations effectively. By preserving meaningful transformation information, equivariant SSL can more efficiently adapt the latent space, even under the tight constraints posed by OCL.

[1] A recent invariance-guided approach that has also shown its merit in CL [106, 50].

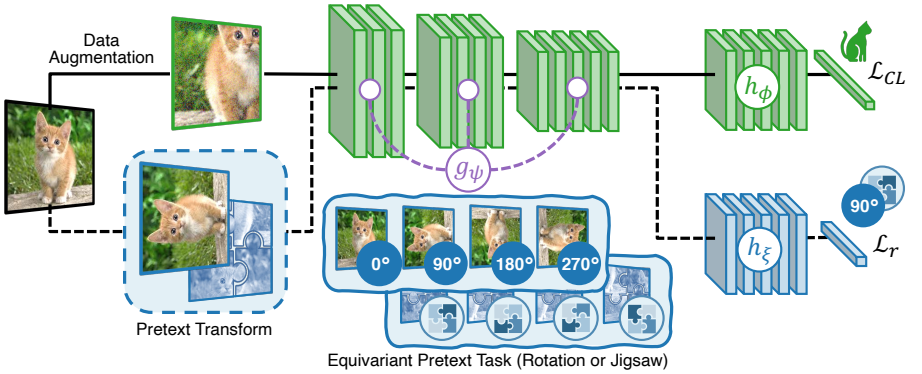


Figure 5.2: Overview of CLER. Two versions of the input image are fed into the in-training model: *i)* standard data augmentation is used to train the classification head (*green*); *ii)* a rotated image provides self-supervision on the rotation prediction head (*blue*).

5.3 CL via Equivariant Regularization

Motivated by the preliminary analysis in Section 5.2, a novel regularization term, **Continual Learning via Equivariant Regularization (CLER)**, is introduced. This term leverages equivariant self-supervised tasks to enhance the robustness of CL models in an online setting.

Definition. Let $\mathcal{A} = \{\mathcal{A}_i\}_{i=1}^{K}$ be a family of input transforms $\mathcal{A}_i : \mathcal{X} \to \mathcal{X}$ (*e.g.*, rotations, jigsaw puzzle). Then, a function f_θ is said to be *equivariant w.r.t.* $\mathcal{A}$ if there exists a mapping $\mathcal{M}_\mathcal{A}$ such that:

$$f_\theta(\mathcal{A}(\mathbf{x})) = \mathcal{M}_\mathcal{A}(f_\theta(\mathbf{x})), \quad \forall \mathbf{x} \in \mathcal{X}. \tag{5.1}$$

Following Eq. (5.1), each input exemplar is transformed using a randomly selected $\mathcal{A}_k$, and the in-training model is tasked with recognizing the transformation by predicting the correct label $k \in \mathcal{Y}_\mathcal{A} = \{1, \dots, K\}$. To achieve this, f_θ is rewritten as $h_\phi \circ g_\psi$, where g_ψ represents the early part of the network devoted to features extraction, and h_ϕ encompasses the latter part of the model, including the final classification head for the CL task. Additionally, h_ξ is introduced as a separate sub-network that follows the same structure as h_ϕ and is designed to project the representation $g_\psi(\cdot)$ onto the set $\mathcal{Y}_\mathcal{A}$.

For illustrative purposes, Fig. 5.2 considers the popular choice of $\mathcal{A}$ as the set of non-distorting image rotations $\{\text{Rot}_{0^\circ}, \text{Rot}_{90^\circ}, \text{Rot}_{180^\circ}, \text{Rot}_{270^\circ}\}$ [53, 54], making the pretext task a four-way classification problem[2]. The resulting approach, referred to as CLER, introduces a regularization term $\mathcal{L}_r$ that can be applied directly to any CL method. Let $\mathbf{x} \in \mathbf{B}_\text{in} \cup \mathbf{B}_\text{buf}$ represent a sample from either

[2]However, this approach can easily be extended to other choices of transformations, as shown in Section 5.5.2.

the input or replay batch, the regularization term is defined as:

$$\mathcal{L}_r = \lambda_r \cdot \mathop{\mathbb{E}}_{\substack{\mathbf{x}\sim\mathbf{B}_{\text{in}}\cup\mathbf{B}_{\text{buf}} \\ k\sim\mathcal{Y}_{\mathcal{A}}}} \left[\text{CE}\left(h_\xi(g_\psi(\mathcal{A}_k(\mathbf{x}))), k \right) \right], \tag{5.2}$$

where CE represents the cross-entropy loss, and λ_r is a scalar hyper-parameter controlling the strength of the regularization.

The label space $\mathcal{Y}_{\mathcal{A}}$ of the pretext task remains constant over time. Consequently, the objective of CLER can be compared to classification problems where only the data-generating distribution changes (similar to Domain-IL).

Equivariance & invariance. While the learning objective in Eq. (5.2) promotes sensitivity to the selected set of transformations, solving the CL task enforces the model to become invariant *w.r.t.* data augmentations. To prevent conflicts between the two objectives, Eq. (5.2) is computed exclusively on non-augmented inputs.

5.4 Experiments

5.4.1 Benchmarks

Settings. The evaluation focuses on the online Class-Incremental Learning (oCIL) setting, where the model is asked to gradually learn to solve all tasks, with no information regarding the task identifier (Task-ID). This scenario forces the learner to build a single-headed classifier and is usually depicted as the most challenging [49, 128, 7]. In the online learning scenario, the learner will then experience each task *only once* (single epoch).

Datasets. Experiments are conducted on two popular benchmarks for Class-IL, constraining data availability to a single pass: **Split CIFAR-100 (S-CIF100)** (10 tasks of 10 classes each) and **Sequential *mini*ImageNet (S-*mini*Img)** (20 tasks of 5 classes each). Details of these datasets are provided in Section 3.5.3.

Metrics. Model performance is primarily quantified using **Final Average Accuracy (FAA)**, reported as a percentage value. To evaluate the severity of performance degradation caused by catastrophic forgetting, the **Final Average Adjusted Forgetting (FAAF)** metric is adopted. Explanations of these metrics are provided in Section 3.5.2.

Benchmarked models. The proposed CLER is applied on top of several CL methods, including: **ER-ACE** [19], **X-DER** [14], **CoPE** [40], **DualNet** [106], and **OnPro** [139]. An explanation of these methods is provided in Section 3.4. To highlight the difference with invariance-based contrastive objectives, a regularization term based on CSSL losses is considered as a direct competitor to CLER. This regularizer is not applied to DualNet and OnPro, as these methods already include a contrastive loss. Additionally, the performance of a model

trained on the sequence of tasks with no countermeasure against forgetting is reported (*Finetune*), as well as two models trained jointly on all tasks: one for a single epoch (*Joint-online*) and one for 10 and 20 epochs on S-CIF100 and S-*mini*Img, respectively (*Joint-offline*).

Implementation details. All models are trained using Stochastic Gradient Descent (SGD) with a fixed batch size of 10 on both the input stream and the replay buffer. For a broader evaluation, all models are benchmarked with two sizes for the memory buffer: 500 and 2000 for S-CIF100 and 2000 and 8000 for S-*mini*Img. ResNet18 [61] is adopted as the backbone for all experiments. In DualNet, following the original implementation, the fast learner is constructed as a feed-forward network with the same number of convolutional layers as the number of residual blocks in the slow learner (ResNet18).

Regardless of the underlying CL method, the feature extractor g_ψ and the classification heads h_ϕ and h_ξ are defined by splitting the ResNet backbone at the second-last residual block. The classification heads h_ϕ and h_ξ are comprised of the last residual block, followed by a linear projection onto the respective sets of classes $\mathcal{Y} = \cup_{i=1}^{T} \mathcal{Y}_i$ and $\mathcal{Y}_\mathcal{A}$.
Each experiment is repeated 3 times, and the mean FAA and FAAF are reported.

5.4.2 Comparison with the State-Of-The-Art

The results of the evaluation on S-CIF100 and S-*mini*Img are presented in Table 5.1. The proposed equivariant loss term is compared against an alternative approach that promotes invariance using a CSSL objective. For the latter, the result obtained is the best among five different regularizers: MoCoV2 [30], SimSiam [31], Barlow Twins [146], SimCLR [28], and BYOL [56]. For CLER, the best result is chosen from either a 2×2 jigsaw puzzle (predicting one of 24 permutations) or a 4-fold rotation prediction. A detailed evaluation of various pretext tasks is provided in Section 5.5.2.

The presented evidence strongly supports the initial claims, with CLER improving the SOTA methods across all benchmarks. Specifically, improvements are observed across the board regarding FAA, while FAAF indicates stronger resistance to forgetting. Interestingly, the effect of the proposed regularization is consistent regardless of buffer size, with an average FAA improvement of 3.37 and 3.52 on S-CIF100 and 2.77 and 3.29 on S-*mini*Img. In contrast, DualNet appears to suffer the most from the lack of representation in the buffer. This behavior may be attributed to the fact that the slow learner is trained only on buffer data, limiting the quality of its learned representation. However, results on the challenging S-*mini*Img dataset, when combined with CLER, suggest that leveraging *equivariant* SSL mitigates such effects by enabling the fast learner to develop better representations during OCL.

On the other hand, CSSL delivers mixed results. While it provides slight performance gains over the baselines in most scenarios, it appears to negatively impact performance in others. This observation suggests that the limited training

oCIL	**S-CIF100**		**S-*mini*Img**	
Joint-offline	69.47 (–)		63.31 (–)	
Joint-online	23.14 (–)		10.68 (–)	
Finetune	7.00 (100)		3.21 (100)	
Buffer Size	500	2000	2000	8000
ER-ACE [19]	20.17 (38.75)	26.95 (23.69)	15.03 (35.01)	16.07 (37.94)
+ CSSL	21.54 (34.73)	28.49 (18.37)	17.89 (34.93)	18.96 (32.03)
+ CLER	**24.53** (33.76)	**30.89** (20.24)	**18.28** (28.84)	**20.13** (27.95)
X-DER [14]	25.80 (39.54)	30.44 (31.52)	17.51 (34.25)	18.01 (50.84)
+ CSSL	26.28 (34.19)	32.10 (27.33)	19.77 (36.47)	20.50 (36.34)
+ CLER	**29.35** (35.56)	**34.57** (29.71)	**21.26** (34.07)	**21.71** (34.76)
CoPE [40]	19.98 (75.32)	34.09 (46.39)	22.67 (57.96)	24.54 (55.09)
+ CSSL	20.06 (73.88)	32.02 (47.87)	20.71 (59.32)	22.66 (58.86)
+ CLER	**26.15** (69.28)	**38.48** (45.50)	**25.91** (57.73)	**26.76** (52.69)
DualNet [106]	11.09 (92.42)	19.93 (73.44)	16.21 (80.35)	25.33 (59.60)
+ CLER	**11.89** (89.97)	**20.88** (73.02)	**18.66** (72.74)	**30.90** (52.14)
OnPro [139]	17.20 (21.23)	22.24 (11.47)	11.10 (32.06)	13.21 (21.52)
+ CLER	**19.18** (20.23)	**26.41** (11.38)	**12.27** (31.00)	**14.13** (26.77)

Table 5.1: **FAA** (↑) and **FAAF** (↓) on the **oCIL** setting. The best result among Barlow Twins, SimSiam, SimCLR, BYOL, and MoCoV2 is reported for CSSL, and the best between 2×2 jigsaw puzzle and 4-fold rotation prediction is presented for CLER.

iterations in OCL may not be sufficient for CSSL to transfer useful knowledge, potentially hindering incremental learning.

5.5 Model Analysis

In the remainder, the various contributions of CLER are deeply studied, to gather further insights on its overall effect on the CL tasks.

5.5.1 Effects of equivariance and redundancy in CL

For an in-depth analysis of the effects induced on the backbone, three additional experiments are conducted on ER-ACE with and without CLER, drawing inspiration from the Network Pruning literature [98]. The objective is to examine how the information carried by the learned features is distributed across the parameters of the backbone.

Importance and redundancy. First, each parameter's contribution to the overall loss after training on S-CIF100 is quantified by computing the *importance*

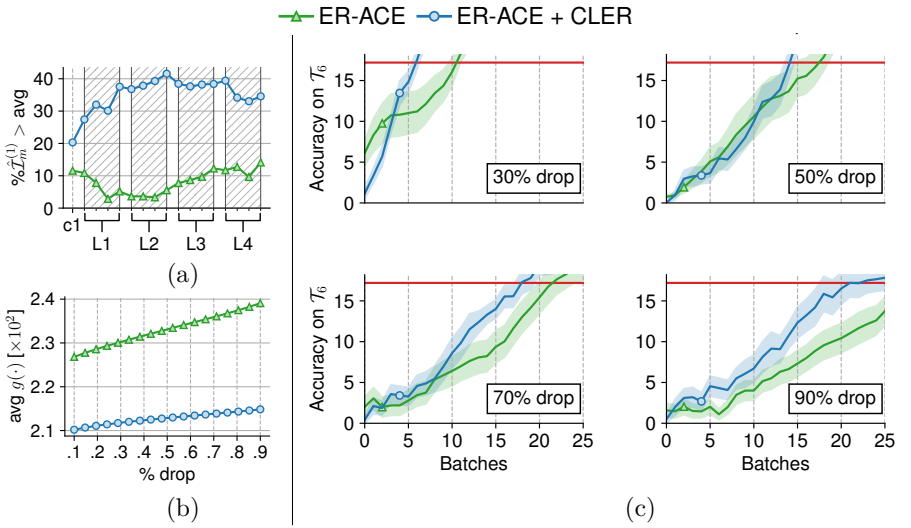


Figure 5.3: For S-CIF100, a structural analysis on ER-ACE with and without CLER. (a) percentage of neurons in each conv. layer with importance score $\hat{\mathcal{I}}_m^{(1)}$ over the layer average; (b) within-layer similarity score g after Geometric Median pruning. (c) after task $\mathcal{T}_6$, a random percentage of conv. filters is dropped and the model is tested after training on a few batches from the same task. The *red* line indicates the pre-drop accuracy of ER-ACE and serves as a target value.

measure $\hat{\mathcal{I}}_m^{(1)}$ proposed in [98]. In Fig. 5.3a, the focus is placed on the convolutional layers, and the proportion of parameters whose importance score exceeds the layer's average is reported to provide a compact per-layer evaluation.

Additionally, Geometric Median pruning [62] is performed on the model, discarding filters $\mathcal{F}_d$ that are the most redundant – *i.e.*, those most similar to others in the same layer. Fig. 5.3b reports the average within-layer similarity g for the discarded kernels:

$$g(\mathcal{F}_d) = \frac{1}{F} \sum_{j=1}^{F} |\mathcal{F}_d - \mathcal{F}_j|, \tag{5.3}$$

where F represents the total number of filters in the considered layer.

The results indicate that CLER encourages the model to fit the learned task with dense configurations of parameters (higher $\hat{\mathcal{I}}_m^{(1)}$ in Fig. 5.3a) that are also more similar to each other (lower g in Fig. 5.3b). This behavior may be linked to the performance increase reported in Section 5.4.2: as task-specific knowledge does not rely solely on a few parameters but is instead more distributed, subsequent weight updates are less likely to completely erase previously acquired knowledge. Furthermore, the higher rate of important parameters, combined with higher redundancy, suggests that filters erased due to forgetting may be

Buffer size	500	2000
ER-ACE + CLER	$+4.36^{\text{JS}}$	$+3.94^{\text{JS}}$
CoPE + CLER	$+6.17^{\text{JS}}$	$+4.39^{\text{JS}}$
ER-ACE + CLER	$+1.23^{\text{REL}}$	$+1.11^{\text{REL}}$
CoPE + CLER	$+2.33^{\text{REL}}$	$+1.19^{\text{REL}}$
ER-ACE + CLER	$+1.75^{\text{VF}}$	$+1.11^{\text{VF}}$
CoPE + CLER	$+2.50^{\text{VF}}$	$+1.85^{\text{VF}}$
ER-ACE + CLER	$+4.42^{\text{JS+R}}$	$+4.43^{\text{JS+R}}$
CoPE + CLER	$+7.26^{\text{JS+R}}$	$+6.01^{\text{JS+R}}$

Table 5.2: Performance **gain** of 2×2 jigsaw ($^{\text{JS}}$) compared to other choices of pretext tasks: relative patch location prediction (REL), vertical flip prediction (VF), and the combination of 4 way rotation and 2×2 jigsaw (JS+R).

restored as needed by leveraging redundant groups of parameters.

Recovery. To further investigate these findings, an additional evaluation probes the dynamics of learning with CLER. After training on the 6^{th} task of S-CIF100, a portion of the convolutional filters in the models is randomly dropped, and retraining is performed using only the cross-entropy loss on a few batches from the same task. The accuracy after each batch is reported in Fig. 5.3c. The results indicate that the distributed importance induced by the training objective leads to a higher initial drop in accuracy for CLER. However, the approach rapidly recovers performance, achieving the target pre-drop accuracy in fewer steps compared to the baseline.

5.5.2 On the choice of an equivariant pretext task

The work presented in this chapter focuses on two pretext tasks that represent the most popular choices for image classification [38, 54, 81]. Nevertheless, to further validate the effectiveness of CLER, Table 5.2 provides the performance gain *w.r.t.* the baseline on S-CIF100 for other choices of pretext tasks:

- Relative patch position prediction (REL): a task based on jigsaw puzzles, where the goal is to predict the position of an image patch given a reference one.
- Vertical flip prediction (VF), which can be seen as a subset of 4-fold rotation prediction.
- Rotation & jigsaw (JS+R), a combination of both 4-fold rotation prediction and 2×2 jigsaw tasks, for a total of 96 classes to predict.

As shown in Table 5.2, all choices lead to a consistent improvement, suggesting that pretext tasks based on equivariance can be effectively integrated to enhance

AUROC %	ER-ACE	X-DER
Baseline	68.60	72.34
+ CLER	71.48+2.88	75.53+3.19
+ CSSL	70.75+2.15	72.43+0.09

Table 5.3: **Misclassification Detection** score (↑) of backbones after an OCL training on S-CIF100 with different strategies. CLER is able to reach the best performance for both the rehearsal-based baselines.

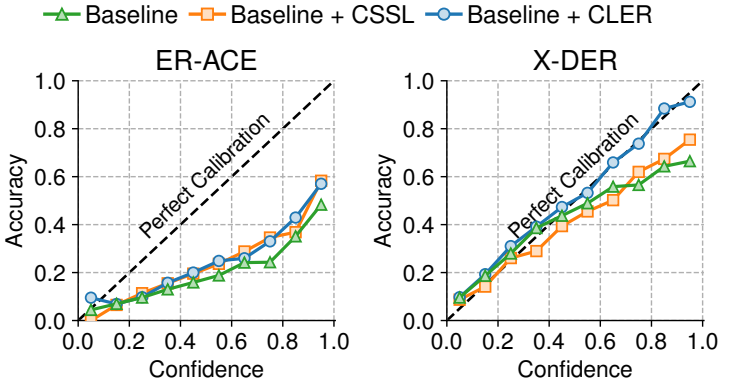


Figure 5.4: Effect of standard CSSL and CLER on the calibration of two rehearsal-based models. While in ER-ACE both self-supervised approaches improve the network scores, X-DER with CLER achieves almost perfect calibration.

performance. Notably, the best results are achieved with the combined JS+R task.

5.5.3 Assessing the Trustworthiness of OCL Models

The increasing range of applications of ANNs has raised significant concerns about the reliability of their decisions [48]. This concern has motivated a line of research focused on evaluating and improving the trustworthiness of ANNs. Related to CLER, prior work [4] highlights that pretext tasks can aid in assessing the confidence of model predictions. Building on these insights, this investigation examines whether equivariance-based regularization can similarly enhance trustworthiness. Such a property is particularly relevant in online settings, where misclassified examples are relatively common, and proper detection of out-of-distribution examples could be leveraged to enhance strategies for populating rehearsal memory buffers.

To address these aspects, a set of experiments is conducted on S-CIF100,

	ER-ACE [19]	**+ CSSL**	**+ CLER**
Epochs	**Buffer size** 500		
1 (OCL)	20.17 (38.75)	20.89 (36.03)	**25.08** (32.84)
5	32.47 (47.70)	33.53 (46.29)	**34.88** (45.52)
20	37.38 (46.79)	37.78 (50.55)	**39.35** (46.84)
50	37.94 (51.49)	39.61 (43.75)	**41.27** (46.78)
Epochs	**Buffer size** 2000		
1 (OCL)	26.95 (23.69)	27.80 (21.12)	**30.89** (20.24)
5	42.35 (27.49)	43.62 (27.11)	**45.67** (24.92)
20	48.03 (33.33)	49.16 (31.86)	**50.27** (31.20)
50	49.05 (33.91)	50.66 (34.48)	**52.17** (32.56)

Table 5.4: Performance comparison (FAA) for Equivariant- and Contrastive-based SSL objectives in a **multi-epoch setting**, evaluated on S-CIF100. CLER delivers a general improvement over CSSL even when the online constraint is relaxed.

focusing on misclassification detection and calibration [58] for ER-ACE and X-DER, both with and without CLER. Confidence is computed as the Maximum Softmax Probability (MSP), following [37]. Results for misclassification detection, reported in Table 5.3, demonstrate that CLER delivers improved performance. Additionally, Fig. 5.4 illustrates the calibration levels of the two approaches: for ER-ACE, both invariance- and equivariance-based tasks result in more reliable confidence intervals, while for X-DER, CLER achieves near-ideal calibration.

5.5.4 Is CLER's advantage tied to (online) CL?

Applicability to the multi-epoch CL. While the evaluation primarily focuses on the OCL scenario, the proposed approach may also be beneficial in less restrictive environments that allow for multiple iterations. Such settings simulate realistic low-latency scenarios where the objective is to develop algorithms capable of rapidly adapting to changing data streams while retaining past knowledge. Results from this evaluation on the S-CIF100 benchmark are summarized in Table 5.4, comparing ER-ACE with either CLER regularization or Barlow Twins.

As anticipated, an increasing number of epochs enables the model to fully leverage the knowledge from the data stream. However, since CSSL tasks typically require many iterations to converge, CLER remains a more effective choice for preventing forgetting while enhancing the base model's representation.

Non-incremental scenarios. The consistently improved performance of baseline methods when combined with CLER might suggest that SSL regularization is broadly effective and not specifically relevant to CL. To investigate this further, both CSSL and CLER regularization are applied to the multi-epoch

Method	CIFAR-100	*mini*ImageNet
Joint-offline	69.85±1.43	62.42±1.13
+ CSSL	70.24±0.47	63.10±0.61
+ CLER	70.92^{JS}±0.74	63.11^{JS}±0.16
Joint-online	23.14±0.74	10.68±0.67
+ CSSL	23.16±0.82	13.79±0.79
+ CLER	28.38^{JS}±1.82	14.77^{JS}±0.78

Table 5.5: Accuracy of **Joint** methods with CSSL and CLER. The epochs are set to 30 and 50 for CIFAR-100 and *mini*ImageNet respectively (Joint-offline).

Joint upper bound (Joint-offline), with results presented in Table 5.5. This evaluation demonstrates that, given sufficient epochs and full convergence, the inclusion of additional SSL terms does not significantly impact the achieved accuracy. Thus, the effectiveness of CLER in Continual Learning scenarios appears to stem from factors beyond standard SSL regularization, as discussed in Section 5.5.1.

To complement this result, the proposed regularization is also applied to single-epoch Joint training (Joint-online). In this context, CLER proves effective and outperforms CSSL. Consistent with the findings in Fig. 5.1, this result confirms that SSL facilitates learner convergence with limited data points and that the equivariant approach adopted by CLER demonstrates greater sample efficiency compared to conventional CSSL methods.

In conclusion, the findings suggest that CLER's effectiveness extends beyond SSL regularization and is inherently tied to CL. Due to its enhanced sample efficiency, the equivariant approach employed by CLER is particularly effective with fewer epochs, making it well-suited for the OCL setting.

5.5.5 Applicability to Data-Free Continual Learning

The SOTA competitors used to validate CLER in Section 5.4 belong to the rehearsal-based family of CL methods. These methods represent the most commonly adopted approach in the challenging oCIL scenario when training from scratch, as the performance of other classes of methods is often severely compromised [93, 19, 57, 150]. However, a line of research has raised concerns regarding the adoption of replay, citing potential privacy issues [123, 51]. This has motivated the exploration of *Data-Free* Class-IL, which refers to multi-epoch Class-Incremental Learning without the ability to store data in a memory buffer.

To demonstrate the flexibility of the proposed approach, its application is further evaluated on two data-free methods: the model inversion-based Relation-Guided Representation Learning (R-DFCIL) [51] and the distillation-based Multi-Class Learning without Forgetting (LwF.MC) [112]. Results presented in Table 5.6 illustrate that CLER delivers consistent performance improvements, highlighting

Method	S-CIF100	S-*mini*Img
LwF.MC [112]	36.15 (49.78)	20.75 (63.67)
+ CLER	**37.07** (49.37)	**21.64** (62.79)
R-DFCIL [51]	34.98 (54.59)	13.15 (83.47)
+ CLER	**36.74** (52.31)	**18.80** (75.43)

Table 5.6: FAA of data-free methods (*without buffer*) with and without CLER. 30 and 50 epochs on S-CIF100, S-*mini*Img respectively.

its effectiveness even in the absence of replay data.

5.6 Conclusions

This chapter presented CLER, an *equivariant* regularization strategy designed for Online Continual Learning. By leveraging pretext tasks such as rotation prediction or jigsaw puzzles, CLER preserves transformation-specific information rather than discarding it, as is typical in contrastive approaches. Experiments on multiple rehearsal-based and data-free CL methods demonstrated the following:

- **Faster adaptation and reduced forgetting**. Equivariant objectives produced more favorable gradient alignments, enabling quicker convergence and better retention compared to purely contrastive SSL.
- **Distributed and robust features**. Pruning studies highlighted that CLER fosters a more redundant representation, enhancing resilience to catastrophic forgetting and facilitating rapid recovery.
- **Improved trustworthiness**. Misclassification detection and calibration metrics indicated more reliable confidence scores under equivariant regularization.
- **General applicability.** While particularly effective in single-pass OCL, CLER also yielded gains in multi-epoch CL settings, suggesting broad relevance wherever large batch sizes and prolonged training cannot be guaranteed.

Overall, these findings underscore the value of equivariant tasks for fostering **robust latent representations** in challenging online continual learning scenarios.

Part III

Distillation of Pretrained Models

Chapter 6

Selective Attention Filtering

6.1 Motivation

Recent developments in Deep Learning, driven by the introduction of Transformer-based architectures [130], have led to a shift in CL research toward exploiting pretrained Vision Transformer (ViT) backbones. Among these approaches, a prominent category relies on ***prompting*** [138, 137, 124], in which a set of trainable tokens called *prompts* is concatenated with the input to the ViT, while the backbone typically remains frozen. Different strategies for prompt selection have been proposed to encode task-specific information within the prompts, which are updated during training. This mechanism proves effective in CL, enabling the model to recall past tasks and often reaching performance levels that approximate an offline upper bound, where the entire dataset remains accessible.

However, most CL benchmarks focus on single-label datasets, in which each image corresponds to a single ground-truth label. Such scenarios may only approximate real-world conditions, where multiple labels are typically associated with the same sample (*e.g.*, a person's age, gender, and nationality). Despite the prevalence of multi-label situations in practical applications, relatively few studies have examined these contexts in an incremental learning fashion [1, 77].

This chapter argues that modern state-of-the-art prompting CL methods, which excel at mitigating forgetting in single-label settings, encounter substantial difficulties when extended to multi-label data. To address this issue, **Multi-Label Continual Learning (MLCL)** is investigated as a novel research avenue where catastrophic forgetting remains insufficiently addressed. The contributions are summarized as follows:

- Two MLCL benchmarks are employed: IIRC CIFAR-100 (IIRC) [1], which appends superclass labels to the CIFAR-100 dataset [83], and Incremental WebVision (Incr. WebVision), a new real-world scenario extracted from the WebVision dataset [88].

- Modern prompting-based CL methods are evaluated in MLCL scenarios, revealing that they do not naturally extend to these challenges.

- Popular rehearsal-based CL baselines are adapted to a pretrained ViT backbone, demonstrating that these straightforward combinations are resilient under MLCL conditions.

- A new framework, **Selective Class Attention Distillation (SCAD)**, is introduced to address the unique challenges of MLCL. Inspired by works highlighting the importance of pretraining for mitigating forgetting [109, 15], this method leverages a frozen pretrained teacher and transfers only pertinent features to an incrementally trained student, effectively distilling knowledge from the teacher's latent space into the student network.

Experiments indicate that SCAD achieves superior performance on both proposed MLCL benchmarks, showing that selectively filtering information from a pretrained latent space can effectively alleviate catastrophic forgetting in multi-label incremental settings.

6.2 Multi-Label Continual Learning

In a multi-label setting, an image classification dataset $\mathcal{D}$ consists of pairs $(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^{|\mathcal{D}|}$, where $\mathbf{x}_i \in \mathbb{R}^{3\times H\times W}$ is an RGB image and $\mathbf{y}_i \in \mathbb{R}^{|\mathcal{Y}|}$ is the multi-hot label vector. Specifically, each element $y_{i,j}$ is 1 if the sample belongs to the j^{th} class, and 0 otherwise. The set $\mathcal{Y}$ encompasses all the possible labels.

In the CL scenario, following Section 3.5.1, a model f_θ is trained on a sequence of T tasks, where each task $\mathcal{T}_t = (\mathcal{X}_t, \mathcal{Y}_t), t = 1, ..., T$. The main differences of MLCL are the following:

- Labels $\mathcal{Y}$ are multi-hot vectors and, as in Class-IL and Task-IL scenarios, annotatios are partitioned over tasks $\mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset$ for $i \neq j$.

- While classes are partitioned into tasks and do not overlap, the same sample can belong to multiple tasks, as it can be associated with labels belonging to different tasks. In this case, it will be presented to the model twice but with only the ground-truth labels of the current task, filtering out the others.

- The evaluation metric is the **Precision-Weighted Jaccard Similarity (PWJS)**, explained in Section 6.4.1. It replaces the single class *accuracy* to compute standard CL metrics such as FAA (Section 3.5.2).

6.3 Selective Class Attention Distillation

Pretraining and forgetting. Prior studies [109, 97] have highlighted that large-scale pretrained models often exhibit stronger resilience to catastrophic forgetting. However, according to [15], continuously training on an evolving stream of tasks tends to move a model away from its original pretrained configuration, thereby exacerbating the forgetting process. In response, this chapter investigates strategies for maintaining closer alignment with the pretrained initialization throughout incremental training. Following insights from several knowledge-transfer approaches [134, 15, 90, 144], the proposed method, **SCAD**, employs two pretrained models. One network, the *teacher*, remains frozen, while the other, the *student*, undergoes parameter updates during training.

Backbone. Both teacher and student networks are instantiated as pretrained Vision Transformers (ViTs) [44], reflecting the growing adoption of these architectures in CL. In this model, an input image $\mathbf{x} \in \mathbb{R}^{3\times H\times W}$ is divided into S non-overlapping patches, each flattened and passed through a patch embedding layer to form a sequence of S tokens. A special *class token*, used for classification, is then prepended to the sequence. The resulting sequence $\mathbf{x}_p \in \mathbb{R}^{(S+1)\times D}$ (where D is the embedding dimension) progresses through a series of transformer encoder blocks, each containing Multi-head Self-Attention (MSA) [130] and feed-forward layers. The final class token extracted from the output is used for prediction. By virtue of MSA, each head can capture different local and global dependencies among the image patches, thereby enabling the model to learn a variety of relationships within the input.

Knowledge Transfer. Recent research has extensively focused on developing efficient strategies for distilling knowledge among ViTs [29, 144, 134], serving as inspiration for the proposed approach. By feeding an input through the model, intermediate representations can be obtained, denoted as $F^l \in \mathbb{R}^{(S+1)\times D}$, where $l \in 1, ..., L$ represents the index of the transformer block under consideration, and L is the total number of blocks. At this stage, the correlation $\mathcal{R}^l$ between the tokens of these representations is computed:

$$\mathcal{R}^l = \frac{F^l}{||F^l||_2^2} \cdot \left(\frac{F^l}{||F^l||_2^2}\right)^{\mathrm{T}}, \tag{6.1}$$

where T denotes the transpose operation and $\mathcal{R}^l \in \mathbb{R}^{(S+1)\times(S+1)}$. The attention vector $\mathcal{R}^{l,1}$, the first row of the correlation matrix, captures the relationships between the class token and all other tokens. The distance between the teacher's vector $\mathcal{R}_\tau^{l,1}$ and the corresponding vector $\mathcal{R}_s^{l,1}$ of the student can be computed:

$$D^l = \mathcal{R}_\tau^{l,1} - \mathcal{R}_s^{l,1}. \tag{6.2}$$

Minimizing this distance for every transformer block ensures that the student model maintains the relationships between the class token and the other tokens aligned to those learned by the teacher model.

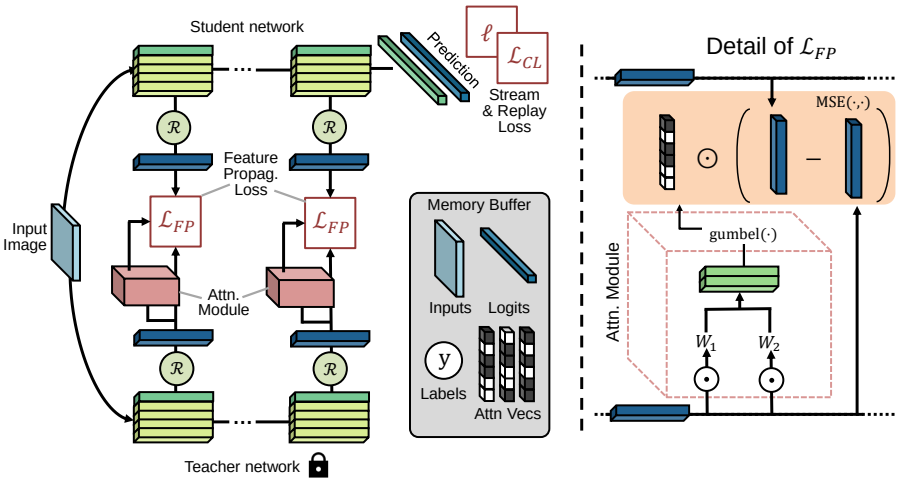


Figure 6.1: SCAD involves two pretrained ViTs: one is frozen, referred to as the *teacher*, while the other, referred to as the *student*, is updated during training. An image is provided as input to both the teacher and the student, and intermediate representations are extracted. From these representations, using Eq. (6.1) and the indexing indicated in Eq. (6.2), the attention vector that contains the relationships between the class token and other tokens is computed. These operations are summarized in the image by the operator $\mathcal{R}$. The attention vector is used by the attention modules (*adapters*) to compute binary vectors useful for filtering knowledge transfer. The feature propagation loss $\mathcal{L}_{\text{FP}}$ maintains alignment between the attention vectors of the teacher and the student models.

Adapter Networks. While this distillation technique effectively mitigates the issue of forgetting the pretraining state, it also introduces the risk of making the model too rigid, potentially limiting its adaptability to new tasks. To address this concern, additional learnable components, referred to as **adapters**, are introduced. These adapters are small Multi-Layer Perceptrons (MLPs) designed to prevent the transfer of irrelevant information from the teacher model to the student model. To achieve this objective, the attention vectors $\mathcal{R}_\tau^{l,1}$ generated by the teacher model are fed into these adapter modules, which project the input into vectors $\in \mathbb{R}^{(S+1)\times 2}$. Binary Gumbel-Softmax sampling is then applied on the last dimension, resulting in a binary vector $\mathbb{M}(\cdot)$. These vectors act as masks, filtering only the relevant information from the teacher model. The binary masks are then multiplied by the respective distance vectors $\mathcal{D}$, derived from Eq. (6.2). The final objective function for this process is defined as:

$$\mathcal{L}_{\text{FP}} = \sum_{l \in L} ||\mathbb{M}(\mathcal{R}_\tau^{l,1}) \odot \mathcal{D}^l||_2^2, \tag{6.3}$$

where $\odot$ is the element-wise multiplication. The pseudocode in Algorithm 2

Algorithm 2 SCAD - distillation loss

Require: Intermediate representations F_τ, F_s for each transformer blok l, adapters functions g per block, with $l \in 1, \ldots, L$,

$loss_{\text{FP}} \leftarrow 0$
for l **in** $1, \ldots, L$ **do**
  $norm\_F_\tau \leftarrow F_\tau^l / ||F_\tau^l||_2^2$
  $norm\_F_s \leftarrow F_s^l / ||F_s^l||_2^2$
  $\mathcal{R}_\tau^l \leftarrow norm\_F_\tau \cdot norm\_F_\tau^{\text{T}}$
  $\mathcal{R}_s^l \leftarrow norm\_F_s \cdot norm\_F_s^{\text{T}}$
  $attn\_vec_\tau \leftarrow \mathcal{R}_\tau^{l,1}$
  $attn\_vec_s \leftarrow \mathcal{R}_s^{l,1}$
  $bin\_mask \leftarrow gumbel(g^l(attn\_vec_\tau))$
  $dist \leftarrow bin\_mask \odot (attn\_vec_\tau - attn\_vec_s)^2$
  $loss_{\text{FP}} \leftarrow loss_{\text{FP}} + mean(dist)$
end for
$\mathcal{L}_{\text{FP}} \leftarrow \frac{1}{L} loss_{\text{FP}}$

outlines the sequence of operations to obtain the loss.

Rehearsal. To further enhance the model's ability to retain knowledge from past tasks, a rehearsal mechanism is incorporated, as replay strategies have been shown to be highly effective in addressing the forgetting problem [49, 7]. During training, a fixed-sized memory buffer $\mathcal{B}$ is fed with a subset of the examples seen by the model, sampled using the popular reservoir sampling strategy [132]. Specifically, both the input images x, the associated labels y, the logits h obtained by feeding the examples to the student network, and the binary masks m generated by the adapters are stored in the buffer. Inspired by the vast literature on rehearsal methods, the total replay loss $\mathcal{L}_{\text{r}}$ using data sampled from the buffer is composed of different components: *i)* the classification loss [111, 113]; *ii)* the distillation loss [17], which enforces similarity between the model's logits and the stored responses; and *iii)* the adapter mask replay loss, which ensures that the binary masks generated by the adapters are consistent with the stored masks. The final loss function is given by:

$$\mathcal{L}_{\text{r}} = \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{y},\mathbf{h},\mathbf{m})\sim\mathcal{B}} \Bigg[\text{BCE}(f_\theta(\mathbf{x}), \mathbf{y}) + \text{MSE}(f_\theta(\mathbf{x}), \mathbf{h}) + \text{BCE}\left(\mathbb{M}(\mathcal{R}_\tau(\mathbf{x})), \mathbf{m}\right) \Bigg], \tag{6.4}$$

where BCE denotes the binary cross entropy loss and MSE is the mean squared error. Each term in the loss function, including $\mathcal{L}_{\text{FP}}$, is weighted by a hyperparameter to regulate its contribution.

Finally, inspired by [19], the Asymmetric Cross-Entropy (ACE) technique is adopted, which computes the classification loss on the stream examples considering only the labels within the current batch. This approach helps in reducing abrupt changes in model representations.

6.4 Experiments

6.4.1 Benchmarks

Settings. The experimental evaluation is conducted in the more challenging Class-IL scenario, where the model is trained on a sequence of tasks with disjoint sets of classes, and the task identity remains unknown at test time.

Datasets. The evaluation utilizes two benchmarks: IIRC, a recently introduced dataset for multilabel incremental evaluation, and a novel benchmark, Incr. WebVision, specifically designed for this study.

IIRC CIFAR-100 (IIRC) [1]. The IIRC benchmark is derived from the CIFAR-100 dataset [83]. To each sample are assigned two labels: a superclass and a subclass label. The dataset is organized into 22 tasks, where the first task contains only superclass labels, while the subsequent tasks gradually introduce subclass labels. The authors defined two settings for this benchmark: *complete information*, where all labels for each sample are available, and *incomplete information*, where only the label of the current task is provided. The evaluation in this chapter focuses on the incomplete setup, identified by the authors as the more challenging scenario.

Incremental WebVision. The **WebVision** dataset [88] includes over 2.4 million web images sourced from the Internet, each accompanied by metadata such as a title, description, and tags. Using these tags, a challenging real-world incremental multi-label benchmark, referred to as **Incremental WebVision**, is created by following these steps:

1. Starting from a pool of $816,814$ tags, those with fewer than 1000 occurrences across the dataset are discarded.

2. Individual clusters of consistently co-occurring tags are manually reviewed, and semantically overlapping tags are merged. Feature similarity in CLIP-space [108] is used as an indicator of semantic proximity. This process results in a curated list of approximately $350,000$ images with 300 tags, serving as examples and (multi-)labels for the benchmark.

3. The dataset is divided into 6 tasks, each containing 50 unique new labels. The task's dataset is composed of approximately $58,000$ images associated with at least one tag in the corresponding set of labels.

4. 5000 images from each task are reserved for testing, and the remaining $53,000$ are used for training.

The resulting dataset portrays a particularly challenging and realistic scenario, as annotations in WebVision are known to contain significant noise [88].

Metrics. As stated in [125], conventional metrics fail to capture the varying levels of correctness in multi-label predictions. To address this limitation, building on the IIRC framework, the **Precision-Weighted Jaccard Similarity (PWJS)** is adopted as evaluation metric, defined as:

$$R_i^t = \sum_{j=1}^{|\mathcal{T}_t|} \frac{y_j^t \cap f_\theta^t(x_j)}{y_j^t \cup f_\theta^t(x_j)} \times \frac{y_j^t \cap f_\theta^t(x_j)}{f_\theta^t(x_j)}, \tag{6.5}$$

where R_i^t is the performance of a model on task i after being trained on tasks $\mathcal{T}_t$, f_θ^t is the model trained until the t^{th} task, x_j is the j^{th} sample from the task and y_j^t is the corresponding ground-truth labels, filtering out the labels that belong to future tasks $y_j^t \cap \mathcal{Y}_k = \emptyset$ with $k = t+1, \ldots, T$.

Since the evaluation involves a sequence of tasks, the final performance is summarized by adapting the popular Final Average Accuracy (FAA) metric to the multi-label setting, resulting in the definition of the **Final Average PWJS (FA-PWJS)**:

$$\text{FA-PWJS} = \sum_{i=1}^{T} R_i^T. \tag{6.6}$$

All experiments are conducted multiple times using the same pretrained weights. The average FA-PWJS and standard deviation are reported for all evaluated approaches.

Competitors. To evaluate the effectiveness of the proposed approach, a comprehensive comparison is conducted with the following state-of-the-art CL methods, adapted to the multi-label setting: **ER** [111], **ER-ACE** [19], **L2P** [138], and **CODA-Prompt** [124]. Descriptions of these methods are provided in Section 3.4. To further contextualize the results, additional baselines are included: **Finetuning**, representing a lower bound where no continual learning technique is applied during training, and **Joint** training, an upper-bound approach where the model is trained on the entire dataset without incremental learning.

To strengthen the validation of the proposed approach, a variant referred to as $\textbf{SCAD}_{\textbf{w/o FP}}$ is introduced. This variant excludes the distillation loss $\mathcal{L}_{\text{FP}}$ during training, relying solely on rehearsal with the first two terms in Eq. (6.4), along with the ACE loss for classification.

Implementation details. The same network architecture, a pretrained ViT-B/16, is used for all benchmarked models. SGD is employed as the optimizer for ER, ER-ACE, L2P, and SCAD, with a learning rate of 0.03. For CODA-Prompt, Adam is used with a learning rate of 0.001, as specified in its original paper [124]. Gradient norm clipping with a maximum norm value of 1.0 is applied to all models except CODA-Prompt, as this adjustment was observed to enhance performance. To ensure fairness, all methods are trained for the same number of epochs and with the same batch size.

FA-PWJS	IIRC		Incr. WebVision	
Joint (UB)	86.83 ±0.22		25.12 ±0.01	
Finetuning (LB)	0.24 ±0.01		18.81 ±0.59	
L2P [138]	3.76 ±0.01		0.31 ±0.25	
CODA-Prompt [124]	3.79 ±0.01		15.05 ±1.45	
Buffer Size	500	2000	2000	5000
ER [111, 113]	21.04 ±1.81	38.55 ±0.91	17.76 ±0.18	21.63 ±0.18
ER-ACE [19]	41.83 ±1.75	50.66 ±0.87	7.75 ±1.78	16.26 ±0.34
SCAD$_{w/o\ \text{FP}}$	41.66 ±2.55	61.82 ±0.09	8.97 ±0.01	17.03 ±0.01
SCAD	**45.23** ±0.03	**66.58** ±0.02	**20.02** ±0.35	**22.17** ±0.60

Table 6.1: Final Average PWJS (and standard deviation) for the IIRC and Incr. WebVision benchmarks. The lower part of the table includes methods requiring the memory buffer for rehearsal.

6.4.2 Results

The performance disparity between the two benchmarks is evident from the results presented in Table 6.1. Across all methods, there is a notable struggle when dealing with the WebVision dataset, probably due to the high presence of annotation noise in the dataset. Prompting-based methods, which usually excel in single-label tasks, show limited effectiveness in both benchmarks under multi-label conditions, indicating that prompting alone does not suffice to preserve performance when multiple labels per sample are involved.

In contrast, rehearsal-based approaches consistently achieve superior results. Interestingly, the ACE loss improve the performance on IIRC but degrade it on Incr. WebVision. This suggests that the ACE loss is beneficial when the dataset is less noisy, as in the IIRC benchmark.

SCAD achieves the best overall performance across all configurations, indicating that combining rehearsal with distillation from a pretrained latent space is particularly advantageous. The SCAD$_{w/o\ \text{FP}}$variant, which excludes the distillation loss, performs worse than the full SCAD model, demonstrating the importance of the distillation process in preserving knowledge from the teacher.

6.5 Model Analysis

6.5.1 Contribution of selective distillation

To evaluate the impact of incorporating adapter neural networks on model performance, an experiment was conducted by removing the adapters and performing distillation using the entire attention vectors. The results, presented in Table 6.2,

	Binary masks	
	✓	✗
FA-PWJS	66.58 ±0.02	63.68 ±0.01

Table 6.2: FA-PWJS achieved by SCAD on the IIRC benchmark with and without the use of adapters to filter the knowledge transfer.

are based on the IIRC benchmark with a buffer size of 2000. The findings indicate that filtering attention vectors through the binary masks generated by the adapters leads to improved performance. These results suggest that adapter networks effectively select and propagate only the relevant information from the teacher's attention vector to the student network.

6.5.2 Quantitatively measuring catastrophic forgetting in Multi-Label Continual Learning

While FA-PWJS provides a clear snapshot of a model's accuracy at the end of training, it is also important to measure how the model's performance degrades as the learner encounters an increasing amount of data. To address this, inspiration is drawn from traditional CL evaluation to formulate a MLCL version of the Final Average Adjusted Forgetting measure Section 3.5.2 based on PWJS:

$$\begin{aligned} \text{FAAF}_{\text{ML}} &= \sum_{i=1}^{T-1} \left[\frac{R_i^* - R_i^T}{R_i^*} \right]^+, \\ \text{s.t. } R_i^* &= \max_{l \in \{i,\ldots,T-1\}} R_i^l, \quad \forall i \in \{1,\ldots,T-1\} \text{ and } [\cdot]^+ = max(\cdot, 0). \end{aligned} \tag{6.7}$$

The FAAF_{ML} metric ranges between 0 and 100, where 0 indicates no forgetting and 100 represents complete forgetting of past tasks.

Table 6.3 presents the measurements of FAAF_{ML} for the experiments in Section 6.4. The results indicate that the ACE technique is highly effective in reducing forgetting on the IIRC benchmark. This suggests that computing the loss based on the classes of the current batch helps preserve network parameters that encode information from past tasks. On the Incr. WebVision benchmark, however, the forgetting values for rehearsal techniques are observed to be quite similar. With regard to prompting techniques, the two tested methods exhibit comparable levels of forgetting on the IIRC benchmark, while L2P demonstrates significantly higher forgetting on Incr. WebVision. In general, rehearsal-based methods tend to have lower forgetting values. SCAD achieves the lowest forgetting values on both benchmarks, indicating that the proposed method is effective in preserving knowledge from past tasks.

$FAAF_{ML}$	**IIRC**		**Incr.**	**WebVision**
Joint (UB)	-			-
Finetuning (LB)	98.67			37.39
L2P	97.29			66.41
CODA-Prompt	97.77			44.06
Buffer Size	500	2000	2000	5000
ER	69.98	41.57	34.33	25.64
ER-ACE	45.25	40.06	57.52	32.80
$SCAD_{w/o\ FP}$	25.82	17.71	48.95	31.87
SCAD	24.36	16.40	28.37	24.92

Table 6.3: Multi-Label Adjusted Forgetting $FAAF_{ML}$ (the lower the better) of the benchmarks evaluated.

6.6 Conclusions

This chapter introduced MLCL as a challenging scenario where multiple labels can be assigned to each sample, highlighting how established prompting-based methods—highly effective in single-label settings—struggle to maintain performance. In contrast, rehearsal-based approaches showed greater resilience, although they still face considerable difficulties under noisy real-world conditions.

A new framework, **SCAD**, was proposed to address these challenges. By leveraging a frozen, pretrained ViT as a teacher network and selectively distilling attention-based representations into the student model, SCAD constrains the drift from its pretrained initialization while preserving flexibility to learn new tasks. Extensive experiments on the IIRC and Incr. WebVision benchmarks confirmed that SCAD significantly outperforms prompting methods and other rehearsal-based baselines, achieving higher final accuracy and reduced forgetting. These findings suggest that selective distillation from a pretrained latent space is a promising direction for improving multi-label continual learning.

Part IV

Multi-modal Latent Space

Chapter 7

Semantic Residual Prompting

7.1 Motivation

The emergence of large-scale pre-trained Transformers [130] and foundation models [12] has recently changed the adaptation paradigm, usually moving towards *parameter-efficient* plasticity, in which only a few parameters are optimized to solve a new task while keeping most of the model frozen. A successful example is *prompt tuning*, in which a few learnable vectors ("soft prompts") are appended to the input image tokens. For instance, Context Optimization (CoOp) [152] learns a context prompt, which is concatenated with the textual name of the class label and fed to the textual encoder of CLIP (Contrastive Language-Image Pre-training) [108]. The latter generates a *class prototype* in feature space, which is used to classify the images of the task at hand. Similarly, Visual Prompt Tuning (VPT) [72] learns task-specific visual prompts for each layer of an ImageNet pre-trained ViT [44].

Prompt tuning has recently been adopted in different CL proposals [138, 137, 124, 135, 27] with good results. These methods, inspired by CoOp and VPT, usually devise a shared pool of key-value pairs to manage the prompts learned during each round. As shown in Fig. 7.1 (left), the input image is used as a query to retrieve a subset of *relevant* prompts from the pool. To make the selection strategy effective, the keys are learned along with the values (*i.e.*, the prompts). The learning criteria for these keys usually involve a form of weak supervision that pulls selected keys closer to corresponding query features, with further orthogonality constraints to encourage diversification [124].

Despite the extensive application in incremental scenarios, prompting approaches fall into a pitfall, which is the main focus of this chapter. As the key space is continuously updated with no further access to past queries/tasks, the selection mechanism is itself subject to *catastrophic forgetting* and possible

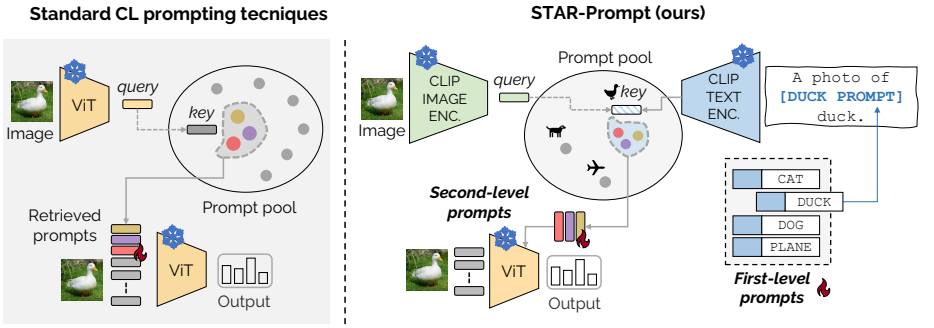


Figure 7.1: Comparison between current approaches (left) and STAR-Prompt (right) with respect to their prompting selection strategies. To enhance the stability of the pool selection, the multi-modal CLIP embedding space is utilized to compute the similarity between the image and prompt-learned class-prototype keys. Subsequently, the retrieved values are used as prompts for another backbone, specifically an ImageNet pre-trained ViT.

misalignments between queries and keys. As outlined in Section 7.2.4, this causes interference in the prompt selection (*i.e.*, prompts of a certain task are mapped to unrelated tasks), thus leading to sub-optimal performance.

The method proposed in this chapter employs a foundation model like CLIP to realize the query-key matching mechanism, yielding a more stable selection strategy. In fact, it has been shown that the fine-tuning regime of CLIP models favors stability over plasticity [27], as distinct parameters can be naturally devoted to distinct concepts, mitigating their interference. However, to enhance plasticity, the prompts retrieved through CLIP serve to condition another backbone, specifically an ImageNet pre-trained ViT. Notably, this leads to a **two-level prompting pipeline** (Fig. 7.1 right), which is the main technical contribution of this work. Briefly: as in CoOp and AttriCLIP [135], class-specific prompts (**first-level prompt pool**) are fed to the CLIP textual encoder, and trained for the current task. The resulting embeddings are not used for computing classification scores, as in other methods, but as keys of a **second** prompt pool. The retrieved **second-level prompts** are used to adapt the ImageNet pre-trained ViT, which provides the final output prediction.

The secondary contribution of this chapter involves a novel mechanism to prompt the pre-trained ViT, utilizing an *additive residual* technique to adapt it. While most methods [137, 124] *concatenate* the learned prompts to the arguments of the Multi-head Self-Attention (MSA) layer, the second-level prompts are used as *semantic residuals* which are *added* before the MLP layer input. While there is no guarantee that a pre-trained and frozen MSA layer will take into account all the prompt tokens, the proposed additive mechanism forces the ViT embeddings to include CLIP-derived semantics.

Inspired by SLCA [149], where first- and second-order feature statistics are

used to generate synthetic samples of past tasks, the approach proposed adopts a generative replay method in both stages. However, unlike SLCA, a Mixture of Gaussians (MoG) is employed to model the *multimodal distribution* of each class, resulting in a more effective generation process.

To evaluate this approach, referred to as **Semantic Two-Tier Additive Residual Prompting (STAR-Prompt)**, experiments are conducted on **nine** image classification datasets. These include well-established CL benchmarks, such as S-Imagenet-R and S-CIF100 [137, 124, 149, 27], as well as datasets from satellite and medical domains, thereby covering scenarios that **differ significantly** [100, 36, 108] from the pre-training distributions of both ImageNet and CLIP. The results highlight that prompting-based CL approaches often struggle with such large domain gaps, frequently being outperformed by older methods that rely on traditional rehearsal schemas. The empirical findings demonstrate a notable +5.96 average improvement in final accuracy compared to the state of the art, with peaks of +3 on S-Imagenet-R and +11 on S-Cars. Importantly, the framework exhibits resilience to severe domain shifts. For instance, when the zero-shot CLIP model performs poorly (*e.g.*, achieving only 54.50% accuracy on S-EuroSAT), the proposed two-level approach achieves significantly higher performance (*i.e.*, 93.70%), highlighting the advantages of stable and continuous adaptation over the direct reuse of frozen large pre-trained models. The key contributions of this chapter are summarized as follows:

- The stability issue of prompt selection is addressed by proposing a two-level prompting strategy that leverages the robustness of foundation models, exploiting their rich latent space.
- A novel prompting approach based on semantic residuals is introduced.
- The SLCA feature generation method is extended to handle multi-modal distributions.
- Insights into incremental adaptation are provided, emphasizing that large pre-trained models still require adaptability.

7.2 Semantic Two-Tier Additive Residual Prompting

7.2.1 Overview

The method proposed, shown in Fig. 7.2, relies on two pre-trained architectures:

- the CLIP model [108], *i.e.* the image $E_{vis}(\cdot)$ and the text $E_{txt}(\cdot)$ encoders.
- an ImageNet pre-trained Vision Transformer [44] $f(\cdot)$.

In summary, all parameters of these networks remain frozen during training. To enable parameter-efficient adaptation while minimizing interference, a **two-level**

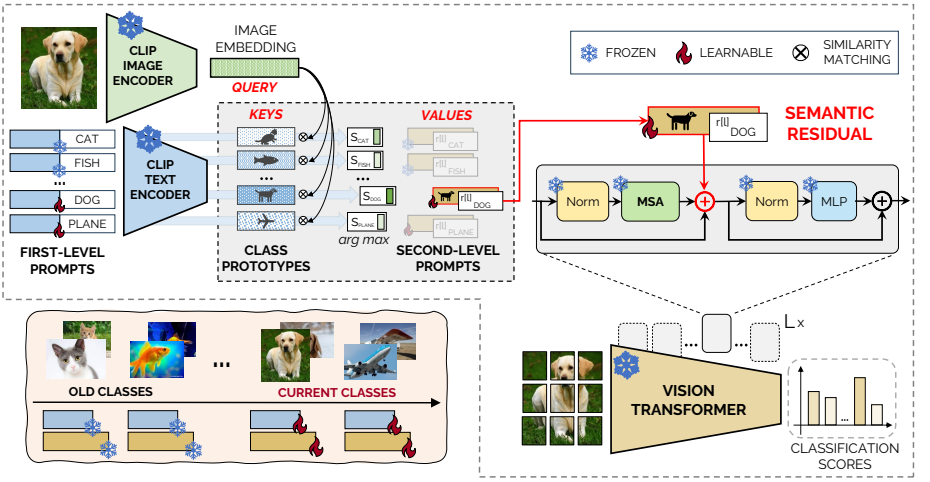


Figure 7.2: The architecture of **STAR-Prompt**. The bottom left box illustrates the CL scenario, in which first- and second-level prompts of the old tasks are frozen.

codebook strategy is introduced. The first level provides learnable context passed as input to the CLIP text encoder, producing *stable* class prototypes. The second level utilizes these text-level prototypes as keys for a secondary codebook of prompts, facilitating a more *plastic* behavior in the second model $f(\cdot)$.

Specifically, the first-level prompting stage is similar to CoOp and conditions the text encoder $E_{txt}(\cdot)$ by concatenating a class-specific learnable vector to the input sequence. However, unlike CoOp, the learned textual class prototypes are not directly used for classification. Instead, they are employed as *keys* for comparison against the visual *query*, which is derived by applying the CLIP visual encoder $E_{vis}(\cdot)$ to the input image. Based on the computed similarities, a second-level learnable prompt R, referred to as the **semantic residual vector**, is retrieved to transfer class-specific semantics to the ViT $f(\cdot)$.

A specific residual vector is designed for each layer of $f(\cdot)$. Formally, considering $\boldsymbol{e}^l$ as the output of the l-th block of $f(\cdot)$, which contains MSA, MLP, and Layer Normalisation (LN) layers [44], the computation is as follows:

$$\boldsymbol{e}' = \mathrm{MSA}(\mathrm{LN}(\boldsymbol{e}_l)) + \boldsymbol{e}_l \tag{7.1}$$

$$\boldsymbol{e}_{l+1} = \mathrm{MLP}(\mathrm{LN}(\boldsymbol{e}')) + \boldsymbol{e}'. \tag{7.2}$$

The first residual connection is leveraged to inject the semantic residual vector R:

$$\boldsymbol{e}' = \mathrm{MSA}(\mathrm{LN}(\boldsymbol{e}_l)) + \boldsymbol{e}_l + R. \tag{7.3}$$

Finally, classification scores are derived from the output of the final classification layer of $f(\cdot)$. Additional details are provided in the subsequent subsections.

7.2.2 STAR-Prompt: Two-Level Prompt Tuning

In the first prompting stage, a learnable prompt p_c is associated with each class $y_c \in \mathcal{Y}_t$. The prompt p_c consists of a single learnable token concatenated with the initial word embedding of the class y_c, denoted as [CLS-NAME]:

$$\boldsymbol{w}_c = E_{txt}([p_c; [\text{CLS-NAME}]]). \tag{7.4}$$

For example, if y_c represents the class '*dog*', then [CLS-NAME] corresponds to the embedding of the word '*dog*'. These two embeddings are fed to $E_{txt}(\cdot)$ to produce a final textual representation $\boldsymbol{w}_c \in \mathbb{R}^d$.

While CoOp directly uses $\boldsymbol{w}_c$ as a class prototype for image classification, STAR-Prompt treats $\boldsymbol{w}_c$ as a **class-specific key**. These class-specific keys are used to retrieve a second set of learnable prompts $Q_c \in \mathbb{R}^{L \times d'}$, designed to adapt each layer of the ImageNet pre-trained Vision Transformer $f(\cdot)$. Let $l < L$ denote the l-th layer of $f(\cdot)$ and d' its embedding dimension. The second-level prompts $Q_c[l] \in \mathbb{R}^{d'}$ adapt the l-th layer of $f(\cdot)$.

The second-level prompting stage leverages the CLIP multi-modal embedding space. Given the visual embedding of an input image $\boldsymbol{z} = E_{vis}(x)$ (query), its cosine similarity $\text{sim}_c = \langle \boldsymbol{z}, \boldsymbol{w}_c \rangle$ is computed with each class prototype key $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{Nt}$. These keys, derived from the first-level prompting stage, correspond to the Nt classes observed so far.

The **semantic residual** $\mathbf{R} \in \mathbb{R}^{L \times d'}$ is then built using the prompt Q_{c_k} of the class with the **highest** cosine similarity:

$$\mathbf{R} = \text{sim}_{c_k} \cdot Q_{c_k}, \tag{7.5}$$

$$\text{where} \quad c_k = \underset{c=1,\ldots,Nt}{\arg\max}\, \text{sim}_c\,. \tag{7.6}$$

In Eq. (7.5), the term sim_{c_k} modulates the residuals based on the confidence CLIP assigns to the selected key $\boldsymbol{w}_{c_k}$. As demonstrated in Section 7.4.1, this modulation yields beneficial empirical effects.

The residuals $\mathbf{R}$ transfer CLIP-based class-specific cues to each MLP layer of $f(x)$. Equation Eq. (7.1) is updated as follows:

$$\boldsymbol{e}' = \text{MSA}(\text{LN}(\boldsymbol{e}_l)) + \boldsymbol{e}_l + \mathbf{R}[l]. \tag{7.7}$$

Note that $\mathbf{R}[l]$ is a single d'-dimensional vector applied uniformly across the entire sequence of visual tokens. For dimensional compatibility during summation, $\mathbf{R}[l]$ is repeated for each token in the sequence.

Main training objective. For each task D_t ($t \in \{1, \ldots, T\}$), the two codebooks are extended by adding the class-specific prompts for the current task. The first-level prompts are updated as: $\mathcal{P}_t := \mathcal{P}_{t-1} \cup \{p_c \mid c \in \mathcal{Y}_t\}$, while the second-level codebook is updated as: $\mathcal{Q}_t := \mathcal{Q}_{t-1} \cup \{Q_c \mid c \in \mathcal{Y}_t\}$. To prevent forgetting, the prompts for previous tasks ($\mathcal{P}_{t-1}$ and $\mathcal{Q}_{t-1}$) **are frozen**, and only the prompts corresponding to the current task are trained. Although both levels can be

jointly optimized end-to-end, the training process is structured as a two-stage procedure for simplicity. For both codebooks, cross-entropy is used as the main loss function:

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{|D_t|} \log p(y_i = c|x_i), \tag{7.8}$$

where, in the first stage, posterior probabilities are computed as:

$$p(y_i = c|x_i) = \frac{\exp(\langle \boldsymbol{z}_i, \boldsymbol{w}_c \rangle / \tau)}{\sum_{c' \in \mathcal{Y}_t} \exp(\langle \boldsymbol{z}_i, \boldsymbol{w}_{c'} \rangle / \tau)}, \tag{7.9}$$

with τ indicating the temperature parameter of CLIP. Conversely, in the second stage, posteriors in Eq. (7.8) use the CLS token $\boldsymbol{e}_L^{\text{CLS}} = f(x_i; \mathbf{R})$ from the last layer L of $f(\cdot)$ [44]:

$$p(y_i = c|x_i) = g_{\theta_t}(\boldsymbol{e}_L^{\text{CLS}}), \tag{7.10}$$

where $g_{\theta_t}(\cdot)$ denotes the classification head for task t. This module consists of a linear projection followed by the softmax layer. The parameters θ_t of the projection layer are initialized at the beginning of task t, and trained while keeping the classification heads of the old tasks frozen.

7.2.3 Additional Details

Query weighting. Inspired by CODA-Prompt [124], class-specific weight vectors $A_c \in \mathbb{R}^d$ are trained to weigh the importance of visual features relative to the c-th class. Specifically, each visual embedding $\boldsymbol{z} = E_{vis}(x)$ is element-wise multiplied with A_c. In Eq. (7.5), $\text{sim}_c = \langle \boldsymbol{z}, \boldsymbol{w}_c \rangle$ is replaced with $\langle \boldsymbol{z} \odot A_c, \boldsymbol{w}_c \rangle$.

Orthogonality. Possible correlations between old and new prompts are discouraged by minimizing the following loss functions for the two codebooks, as proposed in CODA-Prompt:

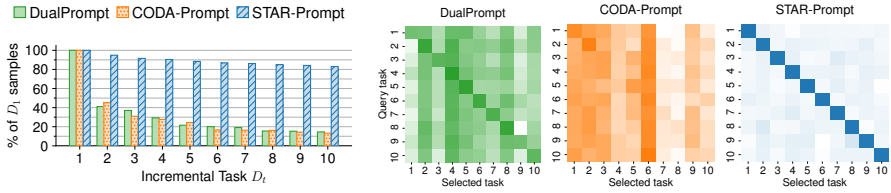
$$\mathcal{L}_{\text{OP}} = \sum_{c \in \mathcal{Y}_t, c' \in \text{past}(t)} \langle \boldsymbol{p}_{c'}, \boldsymbol{p}_c \rangle, \tag{7.11}$$

$$\mathcal{L}_{\text{OQ}} = \sum_{l=1}^{L} \sum_{c \in \mathcal{Y}_t, c' \in \text{past}(t)} \langle Q_{c'}[l], Q_c[l] \rangle. \tag{7.12}$$

where $\text{past}(t) = \{c' | c' \in \mathcal{Y}_{t'}, t' < t\}$ indicates the set of past classes.

7.2.4 Discussion and comparisons to related works

The primary contribution of this chapter relies on the use of a foundation model to improve the **stability** of prompt selection. A prompting strategy is regarded as *stable* if, for a given task, the query-key selection mechanism retrieves prompts that are **relevant** to that task, rather than those introduced for subsequent tasks. The following discussion highlights why existing approaches lack stability and explains how CLIP is utilized to address this issue.



(a) (b)

Figure 7.3: Prompt retrieval on S-Imagenet-R. *Left:* for the test set of the first task, the portion of examples using prompts of the correct task D_1 during the incremental training. *Right:* analysis of all the tasks, computed at the end of training. In each confusion matrix, the y axis represents the task of the query sample, while the x axis shows the task of the corresponding selected key.

On the Stability of Prompting-based CL Methods. Existing approaches such as L2P, DualPrompt, and CODA-Prompt use a query-key matching mechanism to fetch prompts for adapting the pre-trained ViT. These methods learn new keys **from scratch** when a new task is introduced, and employ a customized loss term to align the learned keys with visual queries of the current task (extracted using the pre-trained ViT). However, these learned keys often **lack** clear semantic interpretation and may represent distant concepts.

Concerning this limitation, learning keys in this manner is prone to interference between tasks. Since keys are continuously updated to align with current queries, they may drift and no longer match with queries of earlier tasks. Moreover, introducing novel keys can interfere with those already present in the pool, resulting in the misclassification of examples from previous tasks. To validate this hypothesis, prompts retrieved by different methods—DualPrompt, CODA-Prompt, and STAR-Prompt—are analyzed on the S-Imagenet-R dataset. Results in Fig. 7.3a show the proportion of testing samples from the first task D_1 retrieving the *correct* prompts, specifically those learned during D_1. The precision of selection strategies in DualPrompt and CODA-Prompt **declines** as tasks progress, indicating that examples from D_1 increasingly use prompts from later tasks. In contrast, STAR-Prompt demonstrates greater stability, as evidenced in Fig. 7.3b, which depicts confusion matrices after the final task. The near-diagonal structure of STAR-Prompt's matrix contrasts with the significant task confusion observed in DualPrompt and CODA-Prompt. These results were further verified on datasets from other domains, including aerial and medical datasets (see Figs. 7.4 and 7.5). It is evident that STAR-Prompt exhibits the highest precision in selecting the appropriate prompts.

The stability of STAR-Prompt is attributed to two factors. First, keys are not learned from scratch; instead, the CLIP text encoder is adapted, enabling *distinct learnable parameters* to be assigned to *distinct classes*. This promotes **modularity**, which has also been identified as more stable in PromptFusion [27].

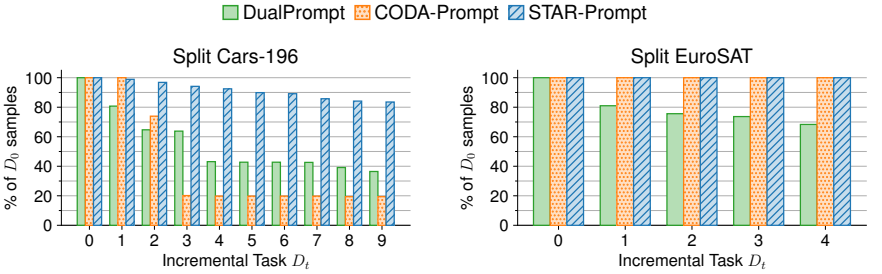


Figure 7.4: Prompt retrieval on the first task of different datasets.

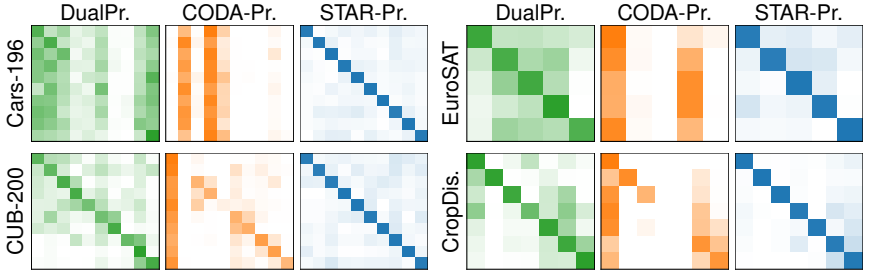


Figure 7.5: Analysis of prompt selection stability for S-Cars, S-EuroSAT, S-CUB, and S-Crop. The y axis indicates the task of the query sample, while the x axis shows the task of the corresponding selected key.

Unlike PromptFusion, however, this approach exploits the stability of the text encoder to fetch a secondary codebook of prompts rather than for direct prediction. Second, **explicit supervision** (Eq. (7.8)) is applied to enforce class-driven **separation** in the key space. This contrasts with other CL prompting approaches, which rely on *weaker* constraints (*e.g.*, orthogonality or aligning the query with the closest key in the pool). As a result, the keys $\boldsymbol{w}_c$ remain both stable and well-separated, enabling the second-level prompts Q_c to capture class-specific information and *reduce interference among tasks and classes*.

Prompting Mechanism. In most methods, prompts are **prepended** to the input sequence **before** each MSA layer. Another approach, **prefix tuning**, operates only on the keys and values within the MSA layer. Since the frozen MSA layer determines the importance of each prompt, there is no guarantee that all tokens will be utilized effectively. In contrast, STAR-Prompt introduces semantic residuals that are **added** after the MSA layer (see Eq. (7.7)). This mechanism resembles the shifting parameters used in FiLM layers [103] to adapt Batch Normalisation [69] in CNNs. However, unlike FiLM, this approach does not use a parameter "generator" network [103]; instead, residuals for the Vision Transformer MLPs are computed using CLIP-guided shifting vectors.

Computational Demands. STAR-Prompt employs a serial two-stage training process. Initially, prompts are learned for the CLIP text encoder, which are then frozen. Subsequently, prompts are learned for the ImageNet-pretrained ViT. Since each phase operates independently, the total computational cost is derived from the requirements of each respective stage. In the first stage, similar to [152], a batch of images undergoes one forward pass through the frozen CLIP image encoder and another through the text encoder to compute class-level textual embeddings. Due to the limited number of trainable parameters, this stage converges within a few epochs and does not require gradient computation for the image encoder. In the second stage, the CLIP text encoder prompts are frozen, and training shifts focus to the ViT prompts. The class prototypes from the text encoder are cached, eliminating the need for additional forward passes through the text encoder. Each batch again involves two forward passes: one through the CLIP image encoder and another through the prompted ViT. The computational complexity of STAR-Prompt aligns with that of L2P, DualPrompt, and CODA-Prompt, which also require two forward passes but operate with the same backbone. In contrast, this approach utilizes two distinct backbones, necessitating additional GPU memory for the CLIP visual encoder. However, this additional memory requirement is found to be negligible compared to the observed performance improvements.

7.2.5 Multi-Modal Generative Replay

In SLCA, class-specific features are utilized to estimate the statistics of a Gaussian representing the class distribution. Subsequent tasks then use this Gaussian to generate synthetic features and perform generative replay. To better capture multiple peaks within each class distribution, this idea is extended by introducing a Mixture of Gaussians (MoG) representation of the feature distribution in both stages of STAR-Prompt training. In the first stage, for each class c, a MoG $\mathcal{G}_c$ is fitted on the CLIP visual features $\boldsymbol{z}_i = E_{vis}(x_i)\ \forall\ (x_i, y_i = c) \in D_t$:

$$\mathcal{G}_c = \mathcal{G}(\cdot; \boldsymbol{\phi_c}) = \sum_{m=1}^{M} \phi_m^c \mathcal{N}(\cdot; \mu_m^c, \Sigma_m^c). \tag{7.13}$$

The parameter vector $\boldsymbol{\phi_c}$ contains the M means, covariances and corresponding Gaussian component weights, and it is estimated by Expectation-Maximization (EM). $\mathcal{G}_c$ is subsequently employed in later tasks t' ($t' \geq t$) to generate $n = 256$ synthetic features $\tilde{\boldsymbol{z}}$ for class c. Using this generative replay technique, the replay loss is:

$$\mathcal{L}_{\text{GR}}^P = -\sum_{c=1}^{Nt} \sum_{i=1}^{n} \log p(y_i = c|\tilde{\boldsymbol{z}}_i), \tag{7.14}$$

where $p(y_i = c|\tilde{\boldsymbol{z}}_i)$ denotes the posterior probability for a synthetic visual point $\tilde{\boldsymbol{z}}_i \sim \mathcal{G}_c$:

$$p(y_i = c|\tilde{\boldsymbol{z}}_i) = \frac{\exp(\langle \tilde{\boldsymbol{z}}_i, \boldsymbol{w}_c \rangle / \tau)}{\sum_{c'=1}^{Nt} \exp(\langle \tilde{\boldsymbol{z}}_i, \boldsymbol{w}_{c'} \rangle / \tau)}. \tag{7.15}$$

Algorithm 3 Training of STAR-Prompt on the current task

Require: Index $t \in 1, \ldots, T$ of the current task D_t, the orthogonality loss weighting coefficient λ, the number of M Gaussian components, the number of E_1 and E_2 training epochs with real and generated samples respectively, learning rate lr.

First stage → *goal*: learning first-level prompts $p_c|_{c\in\{1,\ldots,N\}}$ for each class $c \in \mathcal{Y}_t$ of the current task t.

for $ep := 1, \ldots, E_1$ **do**
  $\mathcal{L}_{\text{CE}} \leftarrow$ use Eqs. (7.8) and (7.9)
  $\mathcal{L}_{\text{OP}} \leftarrow$ use Eq. (7.11)
  $p_c \leftarrow p_c - lr \cdot \nabla_{p_c}\mathcal{L}_{\text{CE}} + \lambda\mathcal{L}_{\text{OP}}$
end for

 # Generative Replay
for $\forall c \in \mathcal{Y}_t$ **do**
  $\mathcal{G}_c \leftarrow \text{EM}(\{E_{vis}(x)\}_{x,y=c\in D_t})$
end for

for $ep := 1, \ldots, E_2$ **do**
  # sample from $\mathcal{G}_{c'}|_{c'\in\mathcal{Y}_1,\ldots,\mathcal{Y}_t}$
  $\mathcal{L}^P_{\text{GR}} \leftarrow$ use Eq. (7.14)
  $p_c \leftarrow p_c - lr \cdot \nabla_{p_c}\mathcal{L}^P_{\text{GR}}$
end for

Second stage → *goal*: learning second-level prompts Q_c, query weights A_c $\forall c \in \mathcal{Y}_t$, and the classifiers' weights $\theta_{t'}|_{t'\in\{1,\ldots,t\}}$.

$\mathtt{W} \leftarrow \{Q_c, A_c\}_{c\in\mathcal{Y}_t} \cup \theta_t$
for $ep := 1, \ldots, E_1$ **do**
  $\mathcal{L}_{\text{CE}} \leftarrow$ use Eqs. (7.8) and (7.10)
  $\mathcal{L}_{\text{OQ}} \leftarrow$ use Eq. (7.12)
  $\mathtt{W} \leftarrow \mathtt{W} - lr \cdot \nabla_{\mathtt{W}}\ \mathcal{L}_{\text{CE}} + \lambda\mathcal{L}_{\text{OQ}}$
end for

 # Generative Replay
for $\forall c \in \mathcal{Y}_t$ **do**
  $\mathcal{H}_c \leftarrow \text{EM}(\{f(x_i;\mathbf{R})\}_{x,y=c\in D_t})$
end for

for $ep := 1, \ldots, E_2$ **do**
  # sample from $\mathcal{H}_{c'}|_{c'\in\mathcal{Y}_1,\ldots,\mathcal{Y}_t}$
  $\mathcal{L}^Q_{\text{GR}} \leftarrow$ use Eq. (7.16)
  $\theta_{t'} \leftarrow \theta_{t'} - lr \cdot \nabla_{\theta_{t'}}\mathcal{L}^Q_{\text{GR}}$
end for

In the denominator, all Nt classes observed so far are included.

Similarly, in the second training stage, ViT features $\boldsymbol{e}^{\text{CLS}}_L = f(x_i;\mathbf{R})$ from the current task are used to estimate the parameters of a second MoG, denoted as $\mathcal{H}_c$, following Eq. (7.13). During rehearsal, $\boldsymbol{e}^{\text{CLS}}_L$ in Eq. (7.10) is replaced with n synthetic features $\tilde{\boldsymbol{e}}^{\text{CLS}}_{L,i}$ sampled from each $\mathcal{H}_c$. Analogous to Eqs. (7.14) and (7.15), $\mathcal{L}^Q_{\text{GR}}$ is then computed:

$$\mathcal{L}^Q_{\text{GR}} = -\sum_{c=1}^{Nt}\sum_{i=1}^{n}\log p(y_i = c|\tilde{\boldsymbol{e}}^{\text{CLS}}_{L,i}), \tag{7.16}$$

where $p(y_i = c|\tilde{\boldsymbol{e}}^{\text{CLS}}_{L,i})$ is the score relating each synthetic representation:

$$p(y_i = c|\tilde{\boldsymbol{e}}^{\text{CLS}}_{L,i}) = g_{\theta_{t'}|_{t'\in\{1,\ldots,t\}}}(\tilde{\boldsymbol{e}}^{\text{CLS}}_{L,i}), \tag{7.17}$$

where $g_{\theta_{t'}|_{t'\in\{1,\ldots,t\}}}(\cdot)$ indicates the employment of *all* classification heads of $g(\cdot)$ corresponding to the t tasks observed so far.

The complete training algorithm is illustrated in Algorithm 3. For simplicity, the dataset is not split into mini-batches during gradient descent computations. In the first training stage, only the first-level prompts are updated ($p_c, c \in \mathcal{Y}_t$). In contrast, the second stage updates the second-level prompts Q_c, the query weighting parameters A_c, and the linear layer of $g_{\theta_t}(\cdot)$. During the rehearsal phase of the second stage, the heads of all tasks observed so far ($\theta_{t' \leq t}$) are updated.

7.3 Experiments

7.3.1 Benchmarks

Settings. The evaluation focuses on the Class-IL setting, where the model is asked to gradually learn to solve all tasks with no information regarding the task identifier, usually depicted as the most challenging [49, 128, 7].

Datasets. Evaluation is conducted across multiple benchmarks to assess the proposed approach. Following recent literature on pre-trained CL models [137, 124, 149, 27], conventional image datasets are initially used. Subsequently, adaptability is evaluated by transitioning to settings with decreasing domain similarity relative to ImageNet pre-training [100, 36]. The evaluation spans the following domains (details about these benchmarks are available in Section 3.5.3):

- **Natural domain**: *Split CIFAR-100 (S-CIF100)* [83] and *Split Imagenet-R (S-Imagenet-R)* [65], featuring 100 and 200 classes split into 10 tasks, respectively.
- **Specialized domain**: following [149], *Split Cars-196 (S-Cars)* [82] and *Split CUB-200 (S-CUB)* [133] are adopted as fine-grained scenarios. Classes are distributed across 10 tasks.
- **Aerial domain**: datasets containing RGB satellite images for land use and land cover classification. The evaluation uses *Split EuroSAT (S-EuroSAT)* [63, 64] (5 binary tasks) and *Split RESISC45 (S-RESISC)* [32], which divides 45 classes into 9 tasks.
- **Medical domain**: three settings are included, ranging from plant to human diseases. *Split CropDiseases (S-Crop)* [68] regards healthy/infected leaves with 7 tasks of 5 classes each. *Split ISIC (S-ISIC)* [33] contains images of 6 skin diseases split into 3 tasks. *Split ChestX (S-ChestX)* [136] consists of chest X-ray images organized into 2 tasks with 3 classes each.

Metrics. The evaluation reports the **Final Average Accuracy (FAA)**, computed after the completion of the last task, and the **Final Average Forgetting (FAF)** [22]. Details are provided in Section 3.5.2. Each experiment is repeated three times with varying class orders using fixed seeds [149], averaging the results to ensure robustness.

Competiting methods. The evaluation focuses on state-of-the-art prompt tuning methods, including L2P [138], DualPrompt [137], AttriCLIP [135], PromptFusion [27], and CODA-Prompt [124]. Additionally, comparisons include methods that fine-tune the entire network, such as SLCA [149], DER++ [17] (*rehearsal-based*), GDumb [107] (*rehearsal-based*), and LwF [89] (*regularization-based*). More details about these methods are available in Section 3.4.

Training-free baselines and lower/upper bounds. The analysis is supplemented with the zero-shot testing of CLIP [108] (**Zero-shot CLIP**), which employs hand-crafted textual prompts (*e.g.*, "A photo of a [CLS-NAME]") to classify test images based on Eq. (7.9). The comparison between STAR-Prompt and Zero-shot CLIP highlights the algorithmic contributions of the proposed approach and disentangles them from the intrinsic strengths of CLIP. Additionally, a zero-shot baseline built on the ImageNet pre-trained backbone is included. This baseline freezes the network and memorizes the latent features of each training example, employing a ***k*-NN** approach for classification.

For a comprehensive comparison, the baseline of STAR-Prompt when trained on all tasks in a stationary scenario is included—**Joint (*STAR-Prompt*)**—serving as an upper bound. Furthermore, **Joint † (*ViT-B/16*)** and **Finetune † (*ViT-B/16*)** are evaluated, where ViT-B/16 is fine-tuned on all tasks respectively jointly and incrementally, with the latter involving no countermeasures to forgetting.

Implementation details. Following the approaches in [137, 124, 149], the same backbone based on ViT-B/16 is adopted as the primary classification architecture for all methods and settings (denoted as $f(\cdot)$). The network is initialized with a fully supervised pre-training on ImageNet-21K. Similarly, for all methods utilizing CLIP, ViT-L/14 is employed, as described in [135] (denoted as $E_{vis}(\cdot)$ and $E_{txt}(\cdot)$). The evaluation includes methods such as PromptFusion and AttriCLIP that leverage the CLIP model in CL scenarios, ensuring fairness in the comparisons.

STAR-Prompt is trained using Adam [78] with a learning rate of 0.001. The number of training epochs is adjusted based on the size of each dataset. For fairness, the same number of epochs is applied to all competing methods, and their hyperparameters (including learning rate and optimizer) are optimized through grid search to ensure the best performance.

Instance- vs. batch-wise Inference. To maintain consistency with most CL studies, inference is conducted independently for each sample within a batch. This approach, referred to as *instance-wise* inference, is the most prevalent in the literature [89, 112, 7, 149, 124]. In contrast, L2P and DualPrompt originally reported results using a *batch-wise* setup, where a single prompt is selected for all samples in a batch through majority voting. This batch-wise configuration may introduce an unfair advantage, as it leverages the absence of sample shuffling during inference, resulting in mini-batches that often contain examples with identical ground-truth labels. To ensure fairness, the results for L2P and DualPrompt are reported using the instance-wise setup, thereby

Model	S-Imagenet-R	S-CIF100	S-Cars	S-CUB	*Avg.*
Joint (*STAR-Prompt*)	90.03	92.32	89.00	85.64	89.25
Joint [(3)] † (*ViT-B/16*)	79.60	93.22	80.31	88.00	85.28
Finetune † (*ViT-B/16*)	17.21	17.47	9.28	11.36	13.83
k-NN (*ViT-B/16*)	18.93	26.14	11.57	21.44	19.52
Zero-shot CLIP [108]	85.36	73.27	76.46	61.14	74.06
LwF † [89]	19.09	19.68	23.24	16.73	19.69
GDumb * † [107]	44.28	57.92	28.74	61.34	48.07
DER++ * † [17]	56.66	79.77	53.66	74.62	66.18
L2P [(3)] [138]	66.49	82.76	38.18	62.21	62.41
DualPrompt [(3)] [137]	68.50	85.56	40.14	66.00	65.05
CODA-Prompt [124]	75.45 [(1)]	86.25 [(1)]	31.99	67.30	65.25
AttriCLIP [135]	86.25	81.40 [(2)]	70.98	50.07	72.18
PromptFusion [(4)] * [27]	80.70	87.40	–	–	–
SLCA [(3)] † [149]	77.00	**91.53**	67.73	**84.71**	80.24
STAR-Prompt	**89.83**	90.12	**87.62**	84.10	**87.92**

Table 7.1: The Class-IL FAA for **natural** and **specialized** domains. † denotes methods fine-tuning the whole model. * highlights rehearsal approaches. Results are taken [(1)] from [124], [(2)] from [135], [(3)] from [149], [(4)] from [27]. Due to the absence of a public codebase, the results of PromptFusion on all datasets could not be reproduced.

removing any potential advantages over other approaches.

7.3.2 Comparison with the State of the Art

As shown in Tables 7.1 and 7.2, STAR-Prompt outperforms all other approaches on average. The comparison is particularly relevant when considering other CLIP-based methods (*e.g.*, AttriCLIP and PromptFusion). For example, in the two fine-grained scenarios—S-Cars and S-CUB—the approach achieves a margin of +11.16 and +22.96 over Zero-shot CLIP, respectively. This outcome is attributed to the fact that fine-grained class differences (*e.g.*, distinguishing a "red-headed woodpecker" from a "red-bellied woodpecker") are not easily captured by the pre-trained CLIP. Consequently, a mechanism promoting plasticity, such as the second-level prompting stage, proves essential. A similar observation can be made for the aerial and medical domains, where the average performance gap relative to Zero-shot CLIP is +37.75.

Notably, STAR-Prompt achieves an average improvement of +5.96 over the runner-up method (SLCA), which is particularly significant considering that STAR-Prompt uses only a fraction of the learnable parameters (details provided in Section 7.4.2). Furthermore, STAR-Prompt closely approximates the performance of its Joint upper bound, with an average margin of just

Model	S-EuroSAT	S-RESISC	S-Crop	S-ISIC	S-ChestX
Joint (*STAR-Prompt*)	97.21	96.41	99.19	78.25	45.36
Joint † (*ViT-B/16*)	98.19	96.88	99.68	88.31	48.92
Finetune † (*ViT-B/16*)	19.91	14.96	13.24	30.30	30.92
k-NN (*ViT-B/16*)	28.18	24.86	30.74	19.09	11.51
Zero shot CLIP	54.50	63.15	27.58	29.14	26.27
LwF †	25.13	15.37	22.31	33.06	32.82
GDumb * †	90.99	60.07	83.61	61.64	32.33
DER++ * †	93.08	51.84	92.53	65.68	35.52
L2P	46.34	63.27	74.68	47.13	32.46
DualPrompt	71.39	76.21	81.41	49.99	35.70
CODA-Prompt	63.12	70.46	77.09	44.87	38.62
AttriCLIP	57.51	66.64	33.21	26.77	28.94
SLCA †	88.69	85.70	93.80	59.19	39.07
STAR-Prompt	**93.70**	**92.28**	**94.92**	**66.67**	**41.85**

Table 7.2: The Class-IL FAA for the **aerial** and **medical** domains. † denotes methods fine-tuning the whole model. * highlights rehearsal approaches.

−3.59. This result is especially promising, as it demonstrates that the proposed approach yields an *almost negligible drop in performance* compared to a stationary scenario. This finding applies equally to aerial and medical scenarios, which typically demand high adaptability. Additionally, existing prompting approaches encounter difficulties in these settings, where they are outperformed by replay-based approaches that enable deeper fine-tuning of the backbone to address domain shifts. In contrast, STAR-Prompt emerges as the best compromise between stability and plasticity.

The results presented in Table 7.3 further validate the superior performance of STAR-Prompt in terms of mitigating forgetting. The only method surpassing STAR-Prompt in two medical domains is CODA-Prompt. However, the FAF metric reflects performance degradation, implying that models achieving low accuracies on tasks may yield lower FAF values. As shown in Table 7.2, CODA-Prompt achieves lower accuracy in these domains, indicating a prioritization of stability over plasticity. In contrast, STAR-Prompt demonstrates a more effective balance between these two aspects, leading to superior overall performance.

7.4 Model Analysis

7.4.1 Ablation studies

Table 7.4 analyzes the impact of each component of STAR-Prompt.

Two-level prompting. The first two comparisons evaluate the role of the two-level prompting strategy. Specifically, the configuration labeled *"Classify*

Model	**S-Imagenet-R**	**S-CIF100**	**S-Cars**	**S-CUB**	***Avg.***
Finetune †	83.62	89.95	85.55	90.69	87.45
LwF †	79.68	87.23	76.59	83.81	81.83
DER++ * †	38.88	19.28	24.42	19.16	25.44
CODA-Prompt	4.06	5.56	7.36	**5.65**	5.66
AttriCLIP	6.9	–	18.8	33.35	19.68
STAR-Prompt	**3.92**	**4.71**	**6.14**	6.94	**5.43**

Model	**S-EuroSAT**	**S-RESISC**	**S-Crop**	**S-ISIC**	**S-ChestX**
Finetune †	98.11	95.23	93.07	94.31	66.76
LwF †	92.88	94.63	90.57	95.06	69.01
DER++ * †	8.02	53.52	8.57	45.61	61.61
L2P	43.87	25.41	18.85	37.79	42.60
DualPrompt	12.88	14.46	10.98	25.01	31.35
CODA-Prompt	16.75	15.05	10.39	**22.97**	**10.49**
AttriCLIP	39.13	32.16	62.56	74.72	48.57
SLCA †	7.74	10.58	4.90	35.25	32.05
STAR-Prompt	**4.31**	**5.25**	**3.26**	26.36	30.75

Table 7.3: The FAF (*the lower the better*). † denotes methods fine-tuning the whole model. * highlights rehearsal approaches. In standard benchmarks, L2P, DualPrompt, and SLCA are lacking, since [149] does not report forgetting values for their experiments. The same applies to PromptFusion and to the experiments of AttriCLIP on S-CIF100 (taken from [135]).

with first-level keys $\boldsymbol{w}_c$" represents an ablation that excludes the second stage. Similar to CoOp, it classifies using the learned keys $\boldsymbol{w}_c$ as class prototypes, with posteriors computed as described in Eq. (7.9). The results in Table 7.4 indicate that this approach achieves competitive performance, underscoring the stability of the learned keys. However, when greater plasticity is required (*e.g.*, S-EuroSAT and S-ISIC), the gap relative to STAR-Prompt widens. Similarly, STAR-Prompt outperforms the configuration labeled *"w/o first-level prompts"*, which omits learning first-level prompts and instead relies on static, hand-crafted CLIP textual templates [108]. The textual embeddings serve as class-prototype keys, with only second-level prompts Q_c being learned.

Semantic residuals. The semantic residuals defined in Eq. (7.7) are replaced with *Prefix Tuning* for evaluation purposes. Following [137], this configuration trains 5 key and 5 value prompt tokens for each ViT layer, which are concatenated to the keys and values of the corresponding MSA. The results in Table 7.4 highlight the advantages of the additive mechanism employed by the semantic residuals.

Model	S-Imagenet-R	S-Cars	S-EuroSAT	S-ISIC
STAR-Prompt	**89.83**	**87.62**	**93.70**	**66.67**
Ablations on two-level prompting				
Classify with first-level keys $\boldsymbol{w}_c$	88.16	87.57	90.12	58.87
w/o first-level prompts	88.97	79.88	86.25	58.68
Other secondary ablations				
Prefix Tuning (no residuals)	71.34	60.03	90.92	62.46
w/o Generative Replay	88.55	87.17	83.78	50.76
w. Unimodal Generative Replay	89.62	87.28	92.58	60.79
w/o Confidence Modulation	88.73	87.29	93.68	63.53

Model	S-CIF100	S-CUB	S-RESISC	S-Crop	S-ChestX
STAR-Prompt	**90.12**	**84.10**	**92.28**	**94.92**	**41.85**
Ablations on two-level prompting					
Classify with first-level keys $\boldsymbol{w}_c$	83.49	81.31	90.80	88.36	31.70
w/o first-level prompts	87.67	81.34	85.85	89.92	37.27
Other secondary ablations					
Prefix Tuning (no residuals)	86.70	82.67	84.26	95.06	39.28
w/o Generative Replay	88.72	82.21	88.2	88.82	37.66
w. Unimodal Generative Replay	90.07	83.16	92.29	94.21	38.77
w/o Confidence Modulation	89.92	83.74	92.05	93.97	39.28

Table 7.4: Ablative studies on STAR-Prompt (FAA).

Generative Replay. A notable performance drop is observed when the generative replay stage is removed, especially in scenarios where tasks deviate significantly from the pre-training data. Additionally, the configuration labeled *w. Unimodal Generative Replay* replaces the MoG with a single Gaussian. While this approach generally outperforms *w/o Generative Replay*, it remains inferior to the MoG-based method used in the full model.

Confidence modulation of the semantic residuals. Eq. (7.5) integrates the *confidence* of the CLIP encoders into the residuals. The final row of Table 7.4 demonstrates that *Confidence Modulation* provides consistent improvements across datasets.

Number of Gaussians. Table 7.5 reports the FAA of STAR-Prompt on S-Imagenet-R when varying the number of Gaussians (M) for each $\mathcal{H}_c$. The results indicate that performance stabilizes when $M \geq 5$, which is therefore set as the default value in all experiments.

M	FAA
2	88.94
5	89.37
10	89.16
20	89.58

Table 7.5: Impact of the number of Gaussians on S-Imagenet-R.

Model	#params (millions)
Joint, Finetune, LwF	86
GDumb, DER++, SLCA	86
L2P	0.12
DualPrompt	0.41
CODA-Prompt	3.91
AttriCLIP	0.0998
PromptFusion	0.35
STAR-Prompt	3.89

Table 7.6: The number of learnable parameters for each tested method.

7.4.2 Number of Trainable Parameters

The number of trainable parameters varies significantly among the compared methods. Two primary adaptation strategies are identified: **fine-tuning**, which involves optimizing the entire model using training data from the target dataset(s), and **parameter-efficient learning**, which adapts the model by optimizing only a subset of parameters (*e.g.*, prompts). In Table 7.6, S-CIF100 is used as the reference scenario, and the number of trainable parameters—those optimized during incremental learning—is reported. For example, SLCA fine-tunes the entire model, whereas the trainable parameters of STAR-Prompt include: $\boldsymbol{p}_c, Q_c, A_c$ $(c \in \mathcal{Y}_1, ..., \mathcal{Y}_T)$ and θ_t $(t \in \{1, ..., T\})$.

7.5 Conclusions

This chapter introduced **STAR-Prompt**, a two-level prompting framework for continual learning designed to address the instability of existing prompt-based approaches. By exploiting the stability of a large-scale foundation model (CLIP) at the first level, class-specific textual embeddings are learned as keys and subsequently used to retrieve a second-level set of prompts. These second-level *semantic residuals* adapt a pre-trained ViT, enabling effective integration of new domain knowledge with minimal interference. Furthermore, the addition of multi-modal generative replay further mitigated forgetting and strengthened the model's ability to tackle complex incremental tasks.

Experiments on nine image classification benchmarks spanning diverse domains demonstrated the robustness of STAR-Prompt, which outperformed both prompting-based and traditional rehearsal methods. Particularly under large domain shifts, STAR-Prompt showed strong adaptability while preserving previously acquired knowledge. Overall, the findings support the view that leveraging foundation models in a two-level prompting pipeline can strike an effective balance between *stability* and *plasticity*. By isolating semantic cues through a

stable text encoder and then transferring them to a pre-trained visual backbone, STAR-Prompt offers a promising approach for robust continual learning with parameter-efficient updates.

Chapter 8

Generative Latent Replay

8.1 Motivation

Continual Learning has been significantly influenced by the emergence of ViTs [44] and Large Vision-Language Models (VLMs) such as CLIP [108, 71]. In addition to providing rich features, pre-trained VLMs like CLIP offer strong zero-shot capabilities. These properties enable impressive continual learning performance without requiring fine-tuning [127], thereby mitigating forgetting by design. However, for tasks deviating from CLIP's pre-training (*e.g.*, satellite and medical domains), adaptation becomes necessary, as shown in Chapter 7. Incremental fine-tuning of CLIP models presents challenges due to their scale and complexity, like parameter count, image-text alignment, and hyper-parameter tuning. Incremental fine-tuning may also result in performance degradation compared to the frozen zero-shot model due to catastrophic forgetting. Additionally, prior research [151] has demonstrated that fine-tuning CLIP can impair its original zero-shot capabilities. Only a limited number of approaches [151, 135, 145] offer adaptation strategies compatible with open-vocabulary classification tests. In contrast, most methods (*e.g.*, PromptFusion [27] and STAR-Prompt—Section 7.2) rely on new classification heads to model posteriors of novel classes, limiting applicability to closed-vocabulary or zero-shot settings.

This chapter introduces a novel and straightforward approach to address these limitations during the incremental adaptation of CLIP. Inspired by CoOp [152] and STAR-Prompt, both the visual and text encoders are frozen, and class-specific prompts are learned for the text encoder, in a different manner *w.r.t.* the first stage of STAR-Prompt (Section 7.2). The proposed method maintains sufficient plasticity to adapt to new domains while preserving the stability needed to retain original zero-shot capabilities. Additionally, for unseen classes, hand-crafted prompts (*e.g.*, `"a photo of a <CLS>"`) are utilized, resulting in a hybrid prompting strategy.

Since prompt learning alone may not be sufficient to address challenges

in a CL scenario (see Sections 7.3 and 8.3), the approach incorporates joint training through **generative replay**. Unlike standard rehearsal methods [112, 17, 19], which depend on a buffer of real examples, this method employs multiple lightweight generative models to learn the underlying data distribution. This design provides two key advantages: *i)* leveraging an effectively infinite supply of synthetic samples rather than a fixed dataset subset, and *ii)* ensuring data privacy by avoiding reliance on real data samples. Furthermore, the proposed method distinguishes itself from existing generative replay techniques [120, 52], which typically focus on generating images in the input space. Instead, it models the data distribution in the latent space, mitigating the *curse of dimensionality*. Building upon the generative replay strategy of STAR-Prompt, which uses MoGs, the proposed method trains a lightweight Variational Autoencoder (VAE) [79] for each new class to model the distribution of CLIP visual embeddings. Operating in the lower-dimensional latent space significantly reduces the complexity of the generative models, enabling their training to be completed within minutes using standard GPUs. As discussed in Section 8.4, the approach is evaluated by comparing various generative and prompt-learning techniques, providing a comprehensive validation of its design choices.

The proposed methodology, referred to as **Continual Generative training for Incremental prompt-Learning (CGIL)**, is evaluated on several standard *class-incremental* CL benchmarks, demonstrating state-of-the-art performance even in domains where zero-shot CLIP fails. Inspired by [151, 145], an additional benchmark is introduced to evaluate zero-shot capabilities on future tasks. This scenario reflects real-world applications in which models encounter new classes before having enough data to be trained on. Evaluations in this setting include all competitors with zero-shot capabilities (*i.e.*, those utilizing VLMs), highlighting the effectiveness of the proposed prompting strategy. The contributions are summarized as follows:

- A novel approach, CGIL, is proposed for incremental fine-tuning of CLIP models. It combines prompt-learning techniques with latent generative replay.

- A new benchmark is introduced to assess zero-shot performance on future tasks.

- Extensive experiments validate the proposed approach, achieving state-of-the-art performance on widely used Class-IL benchmarks and demonstrating superior zero-shot capabilities on unseen classes.

8.2 Continual Generative training for Incremental prompt-Learning

8.2.1 Preliminaries

Contrastive Language-Image Pre-training. CLIP [108] consists of a visual encoder $E_{vis}(\cdot)$, which can be either a ViT or a CNN, and a text encoder $E_{txt}(\cdot)$, typically a transformer. They are trained with a contrastive objective on image-text pairs to obtain aligned latent embeddings. Once trained, CLIP can be used for **zero-shot classification**. To do so, an image x is fed to the visual encoder to compute the embedding $z_{vis} = E_{vis}(x)$.

In parallel, for each candidate class, a text prompt is created by embedding the class label into a template like `"a photo of a <CLS>"`. The resulting class-level prompts are tokenized and fed to the text encoder, producing a textual representation z^i_{txt} for each class y^i. The posterior probabilities are computed as the cosine similarity (noted as $\langle \cdot, \cdot \rangle$) between visual and class-level textual representations:

$$p(y^i|x) = \frac{exp(\langle z^i_{txt}, z_{vis} \rangle / \tau)}{\sum_{j=1}^{|\mathcal{Y}|} exp(\langle z^j_{txt}, z_{vis} \rangle / \tau)}, \tag{8.1}$$

where τ is a temperature parameter and $\mathcal{Y}$ represents the set of classes.

Prompt-learning VLMs. Prompt learning techniques allow the efficient fine-tuning of large pre-trained models. Among these methods, CoOp [152] stands out as particularly effective. In a nutshell, CoOp does not rely on hand-crafted prompts to generate the input for the CLIP text encoder but rather on learnable context vectors V. These context vectors are concatenated with the label token $[CLS]$ and learned through gradient descent, using the similarity scores from Eq. (8.1) as logits of the cross-entropy loss function.

8.2.2 Generative Replay meets Prompt Learning

Fig. 8.1 illustrates the proposed approach, referred to as **Continual Generative training for Incremental prompt-Learning (CGIL)**. From a high-level perspective, CGIL consists of two main phases:

- **Phase 1 (*generative modeling*).** Using all images from the current task, the corresponding image embeddings are extracted through the CLIP visual encoder. These embeddings are then grouped by class and used to train multiple independent Variational Autoencoders (VAEs), with one VAE assigned to each class.

- **Phase 2 (*prompt alignment*).** Context vectors are learned for the text encoder (similar to CoOp). However, instead of computing image embeddings from real images, synthetic embeddings are sampled from all VAEs, covering both past and current classes.

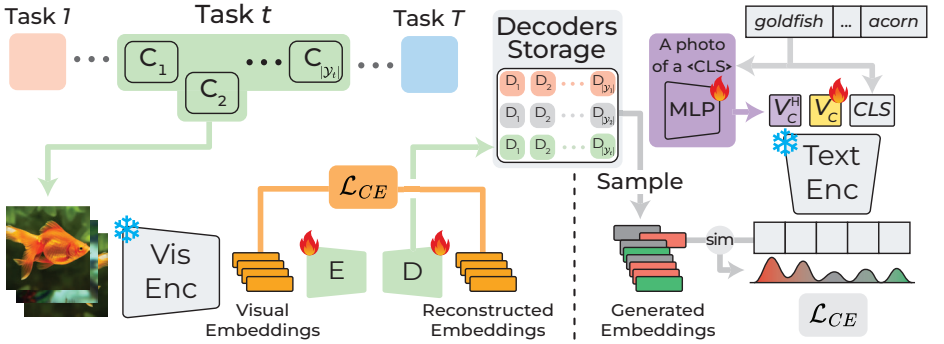


Figure 8.1: Training of the generative models (**left**) and prompt alignment (**right**). For each class C_i of task t, a class-specific generative model (*i.e.*, a VAE) is trained. Afterward, only the decoders are preserved for generative replay, while the encoders are discarded. In the second phase, prompt alignment is performed by matching the features sampled from all stored decoders up to task t with the text features generated using the learnable prompts.

These two phases are repeated for each task to refine previously learned prompts with knowledge from subsequent tasks. A procedural overview is provided in Algorithm 4.

Generative Modeling. The initial objective is to learn, for each class $y \in \mathcal{Y}_t$ of the current task $\mathcal{T}_t$ (with $t \in 1, \ldots, T$), the distribution within the CLIP latent space. This is accomplished by extracting all visual features $z_{vis} = E_{vis}(x)$ for each image $x \in \mathcal{D}_t$ from the current task. Subsequently, the standard Evidence Lower Bound (ELBO) objective is employed to independently train one VAE per class, resulting in $|\mathcal{Y}_t|$ encoders and decoders. Each VAE consists of lightweight components, consisting of three fully connected layers interleaved with LeakyReLU activations. After training, the encoders are discarded, and only the decoders are stored in memory. These decoders are later used to sample new data points during subsequent alignment phases.

Prompt Alignment. In the second phase, a synthetic dataset is constructed by collecting visual embeddings sampled from all stored decoders. Prompt tuning is then performed by aligning the synthetic visual embeddings with corresponding class-level text embeddings generated by the text encoder using learnable prompts. Specifically, the prompt construction involves two distinct tokens: *i)* a learnable class-specific token V_c to capture fine-grained details, and *ii)* a hyper-token V_c^H that models cross-domain knowledge. The hyper-token is generated by a shared, learnable MLP fed with the textual embedding z_{txt} obtained through the standard hand-crafted prompt. The prompt $\mathbf{t}_c$ for class c fed to the text encoder

Algorithm 4 Incremental learning of CLIP with CGIL

Require: datasets $D_t,\ t \in 1, \dots, T$

$\mathcal{M} \leftarrow \{\}$ ▷ initialize a memory buffer for storing the VAE decoders
for each dataset $D_t,\ t \in 1, \dots, T$ **do**
  # Learning class-specific generative models in latent space
  for each class $y \in \mathcal{Y}_t$ **do**
    $\mathcal{D}_t^{(y)} := \{x_i \mid (x^{(n)}, y^{(n)}) \in \mathcal{D}_t,\ y_i = y\}$ ▷ filter examples with label y
    $\mathcal{V}_t^{(y)} := \{E_{vis}(x_i) \mid x_i \in \mathcal{D}_t^{(y)}\}$ ▷ compute CLIP visual embeddings
    TRAIN$(E_y(\cdot), D_y(\cdot), \mathcal{V}_t^{(y)})$ ▷ minimize the ELBO of a new VAE
    $\mathcal{M} \leftarrow \mathcal{M} \cup \{D_y(\cdot)\}$ ▷ store the decoder into the memory buffer
  end for
  # Prompt alignment through generative replay
  $\mathcal{S} = \{(\hat{z}_{vis}, c) \mid \hat{z}_{vis} \sim D_c(\cdot),\ \forall\ c \in \mathcal{Y}_1, \dots, \mathcal{Y}_t\}$ ▷ generate the joint dataset
  for $it := 1, \dots$ **do**
    Construct prompts $\mathbf{t}_c\ \forall\ c \in \mathcal{Y}_1, \dots, \mathcal{Y}_t$ ▷ see Eq. (8.2)
    $\mathcal{L}_{\text{GR}} \leftarrow$ sample a batch of pairs from $\mathcal{S}$ and use Eq. (8.1)
    $\mathbf{t}_c \leftarrow \mathbf{t}_c - lr \cdot \nabla_{\mathbf{t}_c} \mathcal{L}_{\text{GR}}$ ▷ apply Gradient Descent
  end for
end for

E_{txt} is finally defined as:

$$\mathbf{t}_c = [V_c^H]\ [V_c]\ [CLS], \tag{8.2}$$

$$\text{where} \quad V_c^H = \text{MLP}(E_{txt}(\texttt{"a photo of a <CLS>"})). \tag{8.3}$$

The posterior probability of class c is obtained as in Eq. (8.1), using the text embeddings generated with the prompts $z_{txt}^c = E_{txt}(t_c)$. Training of V_c and the MLP is performed via gradient descent, leveraging synthetic data from all previously encountered tasks. Notably, the training process is computationally efficient, as visual embeddings are sampled from the VAEs instead of being computed through the CLIP visual encoder, eliminating the need for costly forward passes. Previously learned contexts are further fine-tuned during training, incorporating knowledge from subsequent tasks without causing forgetting. During inference, images are passed through the visual encoder, and posterior probabilities are computed for each class.

Zero-Shot Inference. A **hybrid** approach is employed to handle both seen and unseen classes. For previously encountered classes (*seen*), the corresponding learned prompts are used. For classes not yet encountered, the text encoder is provided with original hand-crafted prompts (*e.g.*, `"a photo of a <CLS>"`). This approach preserves the zero-shot capabilities of CLIP while enabling adaptation to novel classes introduced incrementally.

Model	S-Imagenet-R	S-Cars	S-CUB	S-EuroSAT	S-ISIC	*Avg.*
Zero-shot CLIP	81.95	64.99	50.52	53.32	26.59	55.47
LwF †	19.09	23.24	16.73	25.13	33.06	23.45
GDumb †	44.28	28.74	61.34	90.99	61.64	57.40
DER++ †	56.66	53.66	74.62	93.08	65.68	68.74
L2P	66.49	38.18	62.21	46.34	47.13	52.07
DualPrompt	68.50	40.14	66.00	71.39	49.99	59.20
CODA-Prompt	75.45	31.99	67.30	63.12	44.87	56.55
AttriCLIP	87.39	75.63	58.28	72.33	28.26	64.38
SLCA †	77.00	67.73	**84.71**	88.69	59.19	75.46
ZSCL	89.14	77.66	62.43	79.13	34.14	68.50
MoE-Adapters	**90.67**	77.76	64.98	80.56	34.52	69.70
CGIL	89.42	**89.27**	83.12	**96.17**	**73.03**	**86.20**

Table 8.1: The FAA on the tested benchmarks. † denotes methods that fine-tune the whole model, while other methods apply PEFT techniques.

8.3 Experiments

8.3.1 benchmarks

Settings. The evaluation is conducted in the more challenging Class-IL setting [128], where task identities are unknown during inference. Additionally, a new benchmark is introduced to evaluate models on future tasks, assessing zero-shot capabilities on unseen classes.

Datasets. CGIL is tested across a diverse set of datasets with varying levels of similarity to the pre-training [36, 100]. Specifically, evaluations are conducted on: *Split Imagenet-R (S-Imagenet-R)* [65], *Split Cars-196 (S-Cars)* [82], *Split CUB-200 (S-CUB)* [133], *Split EuroSAT (S-EuroSAT)* [63, 64], and *Split ISIC (S-ISIC)* [33]. Detailed descriptions of these benchmarks are provided in Section 3.5.3.

Metrics. Performance is assessed using the standard **Final Average Accuracy (FAA)** metric (see Section 3.5.2). To evaluate zero-shot performance on future tasks, a novel metric is introduced, adapting the *Transfer* metric from [151, 145], initially developed for zero-shot evaluation across different domains. Specifically, letting a_i^t represent the accuracy on the i^{th} task after training up to task t, the **Class Incremental Transfer (CI-Transfer)** metric is defined as:

$$\text{CI-Transfer} \triangleq \sum_{t=1}^{T-1} \sum_{i=t+1}^{T} a_i^t. \tag{8.4}$$

Competiting methods. The proposed model is benchmarked against several state-of-the-art prompt-tuning methods, including L2P [138], DualPrompt [137], CODA-Prompt [124], AttriCLIP [135], and ZSCL [151]. Additionally, models that fine-tune the entire architecture are evaluated, including LwF [89], GDumb [107], DER++ [17], and SLCA [149]. Further, MoE-Adapters [145], a parameter-efficient approach aimed at preventing zero-shot accuracy degradation across datasets, is incorporated into the framework. More details about these methods are provided in Section 3.4. A baseline evaluation using zero-shot frozen CLIP is included to assess the effectiveness of prompt-tuning methods. This baseline, along with AttriCLIP, ZSCL, MoE-Adapters, and CGIL, is also evaluated on future tasks to measure how zero-shot capabilities on unseen classes are influenced by incremental training.

Implementation details. Each VAE is trained for 500 epochs using the Adam optimizer [78] with a learning rate of 0.0002. Hidden and latent layer sizes are set to 512 and 256, respectively. Synthetic embeddings, generated approximately $15K$ per class, are shuffled and divided into batches of 128. During the prompt-learning phase, Adam is employed with a learning rate of 0.03. All models utilize ViT-L/14 as their backbone. For CLIP-based methods, the standard CLIP preprocessing is employed, solely involving RGB normalization, while other methods augment the training images with random cropping and horizontal flipping. Results are averaged over three runs with different seeds affecting task composition. All competing models' hyperparameters are tuned for optimal performance to ensure a fair comparison.

8.3.2 Comparison with the State of the Art

Table 8.1 presents the Class-IL performance across all evaluated competitors and benchmarks. The final column reports the average accuracy of each method across all benchmarks. While zero-shot CLIP demonstrates strong results on S-Imagenet-R and S-Cars, it underperforms in other domains, particularly within the medical field. As a result, methods relying on CLIP exhibit a similar performance drop in these domains. In contrast, CGIL effectively addresses the limitations associated with CLIP, delivering state-of-the-art performance in almost all scenarios. On average, the proposed method achieves a significant improvement (**+11**) over the best-performing competitor, SLCA [149], while other PEFT-based approaches lag behind.

Zero-shot performance. Table 8.2 provides the average Class-IL accuracies on unseen tasks, measured using the CI-Transfer metric (Eq. (8.4)). This metric focuses on zero-shot capabilities, and thus only approaches leveraging CLIP or similar VLMs are included in the evaluation. The trends observed are consistent, with CLIP and other competitors performing well on S-Imagenet-R and S-Cars, but facing challenges with other datasets. CGIL demonstrates an exceptional ability to leverage both the zero-shot knowledge of CLIP and information gained

CI-Transfer	S-Imagenet-R	S-Cars	S-CUB	S-EuroSAT	S-ISIC	*Avg.*
Zero-shot CLIP	82.14	66.16	50.86	55.00	22.42	55.32
AttriCLIP	85.75	73.98	54.07	59.69	24.14	59.53
ZSCL	85.30	72.49	62.76	69.74	25.31	63.12
MoE-Adapters	**88.25**	75.82	61.73	55.77	21.06	60.53
CGIL	86.71	**78.80**	**66.34**	**71.52**	**48.18**	**70.31**

Table 8.2: The CI-Transfer on the tested benchmarks. Only methods with zero-shot capabilities (*i.e.*, with CLIP as a backbone) could be tested.

from previously encountered tasks, achieving superior average performance across all benchmarks.

8.4 Model Analysis

To further validate the effectiveness of CGIL and its architectural design, additional experiments are reported in Table 8.3.

Comparison with CoOp. Vanilla CoOp is evaluated under two distinct benchmarks: one trained jointly, *i.e.* without partitioning the dataset into tasks (*Joint*), and the other trained in the conventional Class-IL scenario (*Finetune*). For the latter, two adjustments are made to accommodate the incremental scenario: *i*) a class-specific learnable prompt is allocated to each class, rather than relying on a single global prompt, and *ii*) previously learned prompts are kept frozen during subsequent tasks. This strategy, also employed in [124], helps prevent forgetting. Conversely, training all contexts in subsequent tasks could overwrite previous knowledge by altering learned prompts.

Insights derived from the results of these two approaches – see the first rows of Table 8.3 – highlight the proficiency of CGIL in bridging the gap between fine-tuning and joint training when leveraging prompt learning. The proposed method matches the performance of the *joint* approach. As other ablation studies indicate, this success can be primarily attributed to the effectiveness of the generative replay strategy.

Generative Models. Along with VAEs, various families of generative models are evaluated to determine the most effective complement to the proposed method. The most straightforward approach involves fitting a multivariate Gaussian distribution for each class [149]. As indicated in Table 8.3, this approach alone achieves state-of-the-art results (an average of 76.83, compared to 75.46 for SLCA). However, leveraging more powerful generative models like VAEs considerably enhances the effectiveness of the alignment procedure, suggesting that the quality and variety of generated data are crucial.

	S-Imagenet-R	S-Cars	S-CUB	S-EuroSAT	S-ISIC
CoOp (*Joint*)	89.24	89.89	82.52	96.25	73.57
CoOp (*Finetune*)	84.94	68.61	59.84	79.27	37.08
CGIL	**89.42**	**89.27**	**83.12**	**96.17**	**73.03**
Different generative approaches					
Multinomial Gaussian	83.89	82.59	80.06	85.70	51.89
Mixture of Gaussians	88.54	88.82	82.10	93.04	62.42
Diffusion Models	89.28	90.14	83.48	95.73	68.99
Different techniques to create the context prompt					
Class-specific token	89.09	88.96	83.06	95.59	72.21
MLP-generated token	89.41	88.91	82.82	95.69	70.79
Multiple shared tokens	89.02	88.08	81.50	95.47	70.18

Table 8.3: Ablative studies on CGIL. Results are expressed as FAA.

Additional evaluations are conducted using Mixture of Gaussians (MoG) models, which combine multiple Gaussian components to allow for greater flexibility in representing class variability. Furthermore, Denoising Diffusion Probabilistic Models (DDPMs) [66] are explored, achieving strong performance in generation tasks. These models are trained with the same hyper-parameters as VAEs, described in Section 8.3. Despite their similar capabilities, VAEs are ultimately favored due to faster training times and fewer parameters compared to DDPMs.

Prompting Techniques. The proposed context comprises a class-specific token and a generated token (see Section 8.2). At the bottom of Table 8.3, results for different prompting strategies are presented. Specifically, the evaluation considers: *i)* using a single class-specific context, as in *CoOp (Finetune)*; *ii)* utilizing only the hyper-token generated by the MLP; and *iii)* adopting a method similar to the original CoOp *general context*, where multiple tokens are learned and shared across classes. For the first two variants, increasing the number of contextual tokens during preliminary experiments yielded no improvement, whereas the third variant, which uses a shared context across classes, benefited from this adjustment. The best results are reported.

The outcomes of these alternatives slightly lag behind CGIL, indicating that the primary contributors to performance are the generative rehearsal and alignment phases. This is evident when considering the performance gap between CoOp *(Finetune)* and *Class-specific Context*. Both share the same prompting mechanism, but the latter is enhanced through generative replay.

8.4.1 Discussion and Limitations

Memory and Computational Costs. The VAEs decoders employed are relatively lightweight in terms of memory storage, requiring approximately half a

million parameters each. However, CGIL may encounter scalability limitations as the number of classes increases, potentially constraining its applicability in specific scenarios. When compared with a standard rehearsal approach, the memory requirement for a single decoder is similar to that of a buffer containing 14 RGB images of size 224×224 (approximately ~2MB). It is important to note that these decoders are required only during the training phase, whereas during inference, only the CLIP visual encoder and the embeddings z^c_{txt} of the learned prompts are necessary. Consequently, the computational cost for inference amounts to a single pass through the visual encoder, followed by a matrix multiplication to compute similarity scores. This design offers a significant improvement over other CL prompting methods. For instance, L2P, DualPrompt, and CODA-Prompt perform two forward passes through the ViT, while AttriCLIP calculates both visual and textual embeddings during test time.

Regarding computational costs during training, leveraging generative models in latent space ensures that the VAEs remain highly lightweight, potentially eliminating the need for a GPU. Conversely, the alignment phase exhibits higher complexity, as it involves backpropagating gradients through the CLIP text encoder. Nonetheless, the duration of this phase can be managed by controlling the size of the synthetic dataset generated.

Online Continual Learning. CGIL may also be categorized under the category of online CL methods [7, 91], as training images are processed only once through the visual encoder. Although it requires the temporary storage of visual embeddings to train the VAEs, the associated memory overhead remains minimal due to the compact size of visual features. After training the generative models, these embeddings are then discarded.

Zero-shot Capabilities. Despite the high performance on unseen tasks, the model always performs better when trained on data, since the FAA and CI-Transfer metrics are not directly comparable due to different averaging strategies. Moreover, predictions on unseen classes can be highly biased by the pretraining of the VLM.

8.5 Conclusions

This chapter introduced CGIL, a method that combines prompt learning and latent-space generative replay to incrementally adapt a pre-trained VLM. By training lightweight VAEs to model class-specific distributions in the visual latent space, the approach maintains strong performance on previously encountered tasks without storing original data. The learned generative models supply synthetic embeddings for rehearsal, while newly added textual prompts align these synthetic embeddings with relevant class concepts. Experiments across diverse benchmarks revealed substantial gains over the existing state of the art, especially in domains that diverge significantly from CLIP's pre-training distribution. By introducing a novel scenario to evaluate zero-shot capabilities

on unseen tasks, along with a specific metric to quantify this performance (CI-Transfer), the empirical results underscore the effectiveness of CGIL in leveraging past knowledge to predict future classes. Overall, these findings underscore that carefully designed latent generative replay, paired with parameter-efficient prompting, can effectively address catastrophic forgetting and novel-domain adaptation in incremental learning.

Part V

Conclusions

This thesis investigated the problem of ***catastrophic forgetting*** in Continual Learning (CL) by focusing on the evolving **latent space** of models as they are exposed to sequences of tasks. The overarching goal was to exploit how internal representations change in neural networks during incremental training to mitigate forgetting and improve adaptability. After a technical introduction to the CL problem and its recent literature (Chapter 3), the contributions of this research can be summarized as follows:

- **Spectral Geometry**. In Chapter 4, an initial analysis of latent space in rehearsal-based CL methods revealed that class embeddings become increasingly intertwined as new tasks are encountered. Building on tools from spectral geometry, **Continual Spectral Regularizer for Incremental Learning (CaSpeR-IL)** was introduced to encourage well-separated class-specific clusters by acting on the eigenspectrum of the Laplacian matrix calculated on the latent embeddings. Extensive experiments confirmed that integrating this regularizer into rehearsal-based methods consistently reduces forgetting in various CL scenarios.

- **Equivariant Learning**. Chapter 5 compared the *invariance vs. equivariance* paradigms from self-supervised learning in the context of Online Continual Learning (OCL), as both approaches influence the latent space construction of a model towards different goals. The investigation found that contrastive methods (invariance-based) do not always align well with the rapid adaptation needs of OCL, while leveraging equivariant pretext tasks (*e.g.*, rotation prediction) can learn more robust internal representations, enabling faster adaptation and improved resistance to catastrophic forgetting. This led to the development of **Continual Learning via Equivariant Regularization (CLER)**, which uses equivariant tasks as a lightweight auxiliary loss to strengthen representation quality in online settings.

- **Distillation**. Chapter 6 focused on distilling knowledge from pretrained Vision Transformers (ViTs) to address the challenges of Multi-Label Continual Learning (MLCL). The proposed method, **Selective Class Attention Distillation (SCAD)**, selectively distills an external teacher network into a student model using attention-based filtering on the pretrained latent representations. This approach allows the student to learn only relevant features, achieving notable robustness in incremental multi-label tasks.

- **Semantic Residuals**. Chapter 7 studies how recent *prompting* approaches become unstable when continually updated, leading to increased interference between tasks. This led to the development of **Semantic Two-Tier Additive Residual Prompting (STAR-Prompt)**, a two-level prompting framework that leverages the latent space of a Vision-Language Model (VLM), namely CLIP, to stabilize the key embeddings provided to a separate ViT as additive semantic residuals. The method significantly narrows

the performance gap to offline joint training even under large domain shifts, achieving an optimal balance between *stability* and *plasticity*.

- **Generative Latent Replay**. Chapter 8 further explored the exploitation of CLIP's latent space to enable continual adaptation while preserving zero-shot capabilities. The proposed method, **Continual Generative training for Incremental prompt-Learning (CGIL)**, leverages generative replay over latent embeddings to align different prompts, resulting in a lightweight and efficient approach. CGIL achieves state-of-the-art performance on several benchmarks, demonstrating effective adaptation to new tasks while maintaining strong zero-shot capabilities.

In conclusion, this thesis offers a cohesive perspective on *why* shaping the latent space of Continual Learning models matters, and highlights the critical role played by internal representations in neural networks. Over the course of this research, the field of CL has rapidly expanded, shifting its primary focus from fully training models from scratch to efficiently adapting large pretrained architectures. The gap with offline training has narrowed considerably, and the work presented here may represent a further step toward the eradication of catastrophic forgetting.

Real-World Applications and Future Directions

The ability of CL models to adapt to new tasks without forgetting is essential for various real-world applications. In open-world exploration, robots must continuously adapt to new environments while retaining previously acquired knowledge. In infectious disease detection, models must adapt to emerging virus strains without losing their ability to recognize previously known ones. Similarly, in autonomous driving, systems need to accommodate evolving traffic regulations and road conditions without compromising their ability to drive safely. Furthermore, in the era of large-scale foundation models, frequent retraining is both costly and time-consuming, making CL a crucial component for developing efficient and sustainable AI systems.

The research presented in this thesis contributes to these applications by enhancing our understanding of how to improve model adaptability across diverse CL scenarios. It also paves the way for future research directions. Spectral geometry techniques can be further investigated to analyze latent space dynamics and optimize them for improved adaptability. As self-supervised learning continues to advance, its principles can be leveraged to design more effective auxiliary tasks for CL models. Additionally, novel Parameter-Efficient Fine-Tuning (PEFT) techniques can be explored and compared to assess their resistance to forgetting and their potential for adapting large-scale models to new tasks. Finally, fine-tuning foundational models should prioritize preserving and enhancing their zero-shot capabilities, ensuring that the integration of new knowledge does not come at the cost of forgetting prior knowledge.

Acknowledgments

This thesis is the culmination of my years at the AImageLab group at the University of Modena and Reggio Emilia. I am deeply grateful to everyone who made this journey possible and to the colleagues who made it so enjoyable. Since these people have become friends, I will keep the tone informal and refer to them by their nicknames.

First and foremost, I would like to thank my advisor, Prof. Simone Calderara, for giving me this incredible opportunity, for his guidance, and for bringing together the amazing people of the enviable "Gruppo Simo". I am also grateful to the researchers who mentored and inspired me along the way: thanks to Angel, whose genuine passion for research and patient guidance taught me the ropes, and thanks to Matte Bosc, whose coding wizardry and expertise across all technical things never ceased to amaze me. To all the fantastic members of Gruppo Simo, thank you for sharing both the highs and lows, for the endless laughs, and for putting up with my puns: thanks to Lore, Richi Ben, Gianni, Nello, Monics, Martin, Pit, Moscon, Richi Salam, and Berna. A heartfelt thanks also to all the AImageLab members for sharing the office space, the late-night deadline rushes, and the coffee breaks.

Finally, my deepest gratitude goes to my parents for fueling my work with delicious food and always believing in me.

Appendices

Appendix A

List of Publications

The following list of publications includes all conference papers, journal articles and pre-prints published during my Ph.D.; the contents and experimental results published in these papers have been included in the previous chapters.

[1] Lorenzo Bonicelli, Matteo Boschini, Emanuele Frascaroli, Angelo Porrello, Matteo Pennisi, Giovanni Bellitto, Simone Palazzo, Concetto Spampinato, and Simone Calderara. On the effectiveness of equivariant regularization for robust online continual learning. *arXiv preprint arXiv:2305.03648*, 2023.

[2] Emanuele Frascaroli, Riccardo Benaglia, Matteo Boschini, Luca Moschella, Cosimo Fiorini, Emanuele Rodolà, and Simone Calderara. Latent Spectral Regularization for Continual Learning. *Pattern Recognition Letters*, 2024.

[3] Emanuele Frascaroli, Aniello Panariello, Pietro Buzzega, Lorenzo Bonicelli, Angelo Porrello, and Simone Calderara. Clip with generative latent replay: a strong baseline for incremental learning. In *British Machine Vision Conference*, 2024.

[4] Martin Menabue, Emanuele Frascaroli, Matteo Boschini, Lorenzo Bonicelli, Angelo Porrello, and Simone Calderara. An Attention-based Representation Distillation Baseline for Multi-Label Continual Learning. In *The International Conference on Machine Learning, Optimization, and Data Science*, 2024.

[5] Martin Menabue, Emanuele Frascaroli, Matteo Boschini, Enver Sangineto, Lorenzo Bonicelli, Angelo Porrello, and Simone Calderara. Semantic Residual Prompts for Continual Learning. In *Proceedings of the European Conference on Computer Vision*, 2024.

Appendix B

Activities carried out during Ph.D.

Teaching activities

- Laboratory assistant for the "Machine Learning and Deep Learning" master course at UNIMORE (Master's Degree in Computer Engineering, A.Y. 2022-2023, 2023-2024).
- Teacher for the experimental CS curriculum at Liceo Willigelmo (Scientific High-School, S.Y. 2022-2023, 2023-2024, 2024-2025).
- Laboratory Teacher at the 2nd edition of the "School in AI: Deep Learning, Vision and Language for Industry" organised by UNIMORE (September 2022).
- Lecturer for the Executive Master in "Intelligenza Artificiale, Machine learning e Deep learning", organised by Fondazione Democenter (2023, 2024).
- Lecturer for the Executive Program in "Corso Teorico e Pratico in Machine Learning e Deep learning", organised by the BI-REX Consortium (2022).
- Lecturer for the Executive Program in "Machine Learning e Deep Learning", organised by the BI-REX Consortium (2024).

Thesis supervision

- June 9, 2023. Alessandro Castellucci. Analysis of Transformer Encoders in Knowledge Distillation for Person Re-Identification. Computer Engineering Master's Degree, UNIMORE.

Participation in national and international projects

- "LEGO.AI: LEarning the Geometry of knOwledge in AI systems" – Italian Ministerial grant PRIN 2020 n. 2020TA3K9N.

Reviewing

- Conference on Computer Vision and Pattern Recognition (CVPR)
- European Conference on Computer Vision (ECCV)
- International Conference on Computer Vision (ICCV)
- Conference on Neural Information Processing Systems (NeurIPS)
- International Joint Conference on Artificial Intelligence (IJCAI)
- International Conference on Robotics and Automation (ICRA)
- International Conference on Machine Learning, Optimization, and Data Science (LOD)
- ACM International Conference on Multimedia (ACMMM) Workshop Brave New Ideas

Conferences and schools attended

- VISMAC 23: International Summer School on Machine Vision. Padova. September 4-8, 2023.
- ELLIS Summer School: ELLIS Summer School on Large-Scale AI for Research and Industry. September 18-22, 2023.
- ICML 2024: The 41^{st} International Conference on Machine Learning. Vienna. July 21-27, 2024.
- LOD 2024: The 10^{th} International Conference on Machine Learning, Optimization, and Data Science. Grosseto. September 22-25, 2024.

List of Abbreviations

k-NN k-Nearest Neighbors

ACE Asymmetric Cross-Entropy

AI Artificial Intelligence

ANN Artificial Neural Network

AttriCLIP AttriCLIP

CaSpeR-IL Continual Spectral Regularizer for Incremental Learning

CGIL Continual Generative training for Incremental prompt-Learning

CI-Transfer Class Incremental Transfer

CL Continual Learning

Class-IL Class-Incremental Learning

CLER Continual Learning via Equivariant Regularization

CLIP Contrastive Language-Image Pre-training

CNN Convolutional Neural Network

CODA-Prompt COntinual Decomposed Attention-based Prompting

CoOp Context Optimization

CoPE Continual Prototype Evolution

CSCCT Cross-Space Clustering and Controlled Transfer

CSemiSL Continual Semi-Supervised Learning

CSSL Contrastive Self-Supervised Learning

DDAS Distribution Discriminative Auto-Selector

DDPM Denoising Diffusion Probabilistic Model

DER Dark Experience Replay

DER++ Dark Experience Replay++

DL Deep Learning

DNN Deep Neural Network

Domain-IL Domain-Incremental Learning

DualNet DualNet

DualPrompt DualPrompt

ELBO Evidence Lower Bound

EM Expectation-Maximization

ER Experience Replay

ER-ACE Experience Replay with Asymmetric Cross-Entropy

EWC Elastic Weight Consolidation

FA Final Accuracy

FA-PWJS Final Average PWJS

FAA Final Average Accuracy

FAAF Final Average Adjusted Forgetting

FAF Final Average Forgetting

FAIA Final Average Incremental Accuracy

GDumb Greedy Sampler and Dumb Learner

GPU Graphics Processing Unit

iCaRL Incremental Classifier and Representation Learning

IIRC IIRC CIFAR-100

Incr. WebVision Incremental WebVision

L2P Learning to Prompt

LGG Latent Geometry Graph

LN Layer Normalisation

LwF Learning without Forgetting

LwF.MC MultiClass LwF

ML Machine Learning

MLCL Multi-Label Continual Learning

MLP Multi-Layer Perceptron

MoE-Adapters Mixture-of-Experts Adapters

MoG Mixture of Gaussians

MSA Multi-head Self-Attention

MSP Maximum Softmax Probability

oCIL online Class-Incremental Learning

OCL Online Continual Learning

oEWC online Elastic Weight Consolidation

OnPro Online Prototype Learning

PEFT Parameter-Efficient Fine-Tuning

PNN Progressive Neural Networks

PODNet Pooled Outputs Distillation Network

PromptFusion PromptFusion

PWJS Precision-Weighted Jaccard Similarity

R-MNIST Rotated MNIST

RESISC Remote Sensing Image Scene Classification

RGB Red Green Blue

S-*mini*Img Sequential *mini*ImageNet

S-Cars Split Cars-196

S-ChestX Split ChestX

S-CIF10 Sequential CIFAR-10

S-CIF100 Split CIFAR-100

S-Crop Split CropDiseases

S-CUB Split CUB-200

S-EuroSAT Split EuroSAT

S-Imagenet-R Split Imagenet-R

S-ISIC Split ISIC

S-MNIST Sequential MNIST

S-RESISC Split RESISC45

SCAD Selective Class Attention Distillation

SCR Supervised Contrastive Replay

SGD Stochastic Gradient Descent

SLCA Slow Learner with Classifier Alignment

SOTA state-of-the-art

SSL Self-Supervised Learning

STAR-Prompt Semantic Two-Tier Additive Residual Prompting

Task-IL Task-Incremental Learning

VAE Variational Autoencoder

ViT Vision Transformer

VLM Vision-Language Model

VPT Visual Prompt Tuning

X-DER eXtended Dark Experience Replay

ZSCL Zero-Shot Continual Learning

Bibliography

[1] Mohamed Abdelsalam, Mojtaba Faramarzi, Shagun Sodhani, and Sarath Chandar. Iirc: Incremental implicitly-refined classification. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2021.

[2] Sravanti Addepalli, Kaushal Bhogale, Priyam Dey, and R Venkatesh Babu. Towards Efficient and Effective Self-Supervised Learning of Visual Representations. In *Proceedings of the European Conference on Computer Vision*, 2022.

[3] Hongjoon Ahn, Jihwan Kwak, S. Lim, Hyeonsu Bang, Hyojun Kim, and Taesup Moon. SS-IL: Separated Softmax for Incremental Learning. In *IEEE International Conference on Computer Vision*, 2021.

[4] Deng Ailin, Li Shen, Xiong Miao, Chen Zhirui, and Hooi Bryan. Trust, but Verify: Using Self-Supervised Probing to Improve Trustworthiness. In *Proceedings of the European Conference on Computer Vision*, 2022.

[5] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In *Advances in Neural Information Processing Systems*, 2019.

[6] Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[7] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient Based Sample Selection for Online Continual Learning. In *Advances in Neural Information Processing Systems*, 2019.

[8] Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: on the curvature of deep generative models. In *International Conference on Learning Representations Workshop*, 2018.

[9] Nader Asadi, MohammadReza Davari, Sudhir Mudur, Rahaf Aljundi, and Eugene Belilovsky. Prototype-sample relation distillation: towards

replay-free continual learning. In *International Conference on Learning Representations Workshop*, 2023.

[10] Arjun Ashok, KJ Joseph, and Vineeth N Balasubramanian. Class-incremental learning with cross-space clustering and controlled transfer. In *Proceedings of the European Conference on Computer Vision*, 2022.

[11] Jean M Barnes and Benton J Underwood. "Fate" of first-list associations in transfer theory. *Journal of Experimental Psychology*, 1959.

[12] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[13] Lorenzo Bonicelli, Matteo Boschini, Angelo Porrello, Concetto Spampinato, and Simone Calderara. On the Effectiveness of Lipschitz-Driven Rehearsal in Continual Learning. In *Advances in Neural Information Processing Systems*, 2022.

[14] Matteo Boschini, Lorenzo Bonicelli, Pietro Buzzega, Angelo Porrello, and Simone Calderara. Class-Incremental Continual Learning into the eXtended DER-verse. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[15] Matteo Boschini, Lorenzo Bonicelli, Angelo Porrello, Giovanni Bellitto, Matteo Pennisi, Simone Palazzo, Concetto Spampinato, and Simone Calderara. Transfer without Forgetting. In *Proceedings of the European Conference on Computer Vision*, 2022.

[16] Matteo Boschini, Pietro Buzzega, Lorenzo Bonicelli, Angelo Porrello, and Simone Calderara. Continual semi-supervised learning through contrastive interpolation consistency. *Pattern Recognition Letters*, 2022.

[17] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark Experience for General Continual Learning: a Strong, Simple Baseline. In *Advances in Neural Information Processing Systems*, 2020.

[18] Pietro Buzzega, Matteo Boschini, Angelo Porrello, and Simone Calderara. Rethinking Experience Replay: a Bag of Tricks for Continual Learning. In *International Conference on Pattern Recognition*, 2020.

[19] Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New Insights on Reducing Abrupt Representation Change in Online Continual Learning. In *International Conference on Learning Representations Workshop*, 2022.

[20] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2020.

[21] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In *IEEE International Conference on Computer Vision*, 2021.

[22] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision*, 2018.

[23] Arslan Chaudhry, Albert Gordo, Puneet Dokania, Philip Torr, and David Lopez-Paz. Using hindsight to anchor past knowledge in continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[24] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient Lifelong Learning with A-GEM. In *International Conference on Learning Representations Workshop*, 2019.

[25] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. In *International Conference on Machine Learning Workshop*, 2019.

[26] Jeff Cheeger. A lower bound for the smallest eigenvalue of the Laplacian. In *Problems in analysis*. Princeton University Press, 1969.

[27] Haoran Chen, Zuxuan Wu, Xintong Han, Menglin Jia, and Yu-Gang Jiang. PromptFusion: decoupling stability and plasticity for continual learning. In *Proceedings of the European Conference on Computer Vision*, 2024.

[28] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020.

[29] Xianing Chen, Qiong Cao, Yujie Zhong, Jing Zhang, Shenghua Gao, and Dacheng Tao. Dearkd: data-efficient early knowledge distillation for vision transformers. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[30] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[31] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2021.

[32] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 2017.

[33] Noel CF Codella, David Gutman, M Emre Celebi, Brian Helba, Michael A Marchetti, Stephen W Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, et al. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In *IEEE International Symposium on Biomedical Imaging*, 2018.

[34] Luca Cosmo, Antonio Norelli, Oshri Halimi, Ron Kimmel, and Emanuele Rodolà. LIMP: Learning Latent Shape Representations with Metric Preservation Priors. In *Proceedings of the European Conference on Computer Vision*, 2020.

[35] Luca Cosmo, Mikhail Panine, Arianna Rampini, Maks Ovsjanikov, Michael M. Bronstein, and Emanuele Rodolà. Isospectralization, or How to Hear Shape, Style, and Correspondence. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019.

[36] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018.

[37] Hendrycks Dan and Gimpel Kevin. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In *International Conference on Learning Representations Workshop*, 2017.

[38] Rumen Dangovski, Li Jing, Charlotte Loh, Seungwook Han, Akash Srivastava, Brian Cheung, Pulkit Agrawal, and Marin Soljačić. Equivariant Contrastive Learning. In *International Conference on Learning Representations Workshop*, 2022.

[39] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[40] Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *IEEE International Conference on Computer Vision*, 2021.

[41] Riccardo Del Chiaro, Bartłomiej Twardowski, Andrew Bagdanov, and Joost Van de Weijer. Ratt: Recurrent attention to transient tasks for continual image captioning. In *Advances in Neural Information Processing Systems*, 2020.

[42] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2009.

[43] Mohammad Mahdi Derakhshani, Xiantong Zhen, Ling Shao, and Cees Snoek. Kernel continual learning. In *International Conference on Machine Learning*, 2021.

[44] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations Workshop*, 2021.

[45] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Proceedings of the European Conference on Computer Vision*, 2020.

[46] Sayna Ebrahimi, Franziska Meier, Roberto Calandra, Trevor Darrell, and Marcus Rohrbach. Adversarial continual learning. In *Proceedings of the European Conference on Computer Vision*, 2020.

[47] Sayna Ebrahimi, Suzanne Petryk, Akash Gokul, William Gan, Joseph E Gonzalez, Marcus Rohrbach, and Trevor Darrell. Remembering for the right reasons: Explanations reduce catastrophic forgetting. *Applied AI Letters*, 2021.

[48] Toreini Ehsan, Mhairi Aitken, Kovila P. L. Coopamootoo, Karen Elliott, Carlos Gonzalez Zelaya, and Aad van Moorsel. The relationship between trust in AI and trustworthy machine learning technologies. In *ACM Conference on Fairness, Accountability, and Transparency*, 2020.

[49] Sebastian Farquhar and Yarin Gal. Towards Robust Evaluations of Continual Learning. In *International Conference on Machine Learning Workshop*, 2018.

[50] Enrico Fini, Victor G Turrisi da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-supervised models are continual learners. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[51] Qiankun Gao, Chen Zhao, Bernard Ghanem, and Jian Zhang. R-dfcil: Relation-guided representation learning for data-free class incremental learning. In *Proceedings of the European Conference on Computer Vision*, 2022.

[52] Rui Gao and Weiwei Liu. Ddgr: Continual learning with deep diffusion-based generative replay. In *International Conference on Machine Learning*, 2023.

[53] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Boosting Few-Shot Visual Learning with Self-Supervision. In *IEEE International Conference on Computer Vision*, 2019.

[54] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised Representation Learning by Predicting Image Rotations. In *International Conference on Learning Representations Workshop*, 2018.

[55] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. In *International Conference on Learning Representations Workshop*, 2014.

[56] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, 2020.

[57] Yanan Gu, Xu Yang, Kun Wei, and Cheng Deng. Not just selection, but exploration: Online class-incremental continual learning via dual view consistency. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[58] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.

[59] Yiduo Guo, Bing Liu, and Dongyan Zhao. Online continual learning through mutual information maximization. In *International Conference on Machine Learning*, 2022.

[60] Jeff Z. HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. In *Advances in Neural Information Processing Systems*, 2021.

[61] 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*, 2016.

[62] Yang He, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi Yang. Filter pruning via geometric median for deep convolutional neural networks acceleration. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019.

[63] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In *IEEE International Geoscience and Remote Sensing Symposium*, 2018.

[64] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

[65] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *IEEE International Conference on Computer Vision*, 2021.

[66] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

[67] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019.

[68] David Hughes, Marcel Salathé, et al. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:1511.08060*, 2015.

[69] Sergey Ioffe. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[70] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 1991.

[71] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, 2021.

[72] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Proceedings of the European Conference on Computer Vision*, 2022.

[73] Yu Jin, Andreas Loukas, and Joseph JaJa. Graph Coarsening with Preserved Spectral Properties. In *International Conference on Artificial Intelligence and Statistics*, 2020.

[74] KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2021.

[75] Dahuin Jung, Dongyoon Han, Jihwan Bang, and Hwanjun Song. Generating instance-level prompts for rehearsal-free continual learning. In *IEEE International Conference on Computer Vision*, 2023.

[76] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems*, 2020.

[77] Chris Dongjoo Kim, Jinseo Jeong, and Gunhee Kim. Imbalanced continual learning with partitioning reservoir sampling. In *Proceedings of the European Conference on Computer Vision*, 2020.

[78] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations Workshop*, 2015.

[79] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations Workshop*, 2014.

[80] 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*, 2017.

[81] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019.

[82] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *IEEE International Conference on Computer Vision and Pattern Recognition Workshops*, 2013.

[83] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Toronto, ON, Canada, 2009.

[84] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[85] Carlos Lassance, Vincent Gripon, and Antonio Ortega. Representing deep neural networks latent space geometries with graphs. *MDPI Algorithms*, 2021.

[86] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

[87] James R. Lee, Shayan Oveis Gharan, and Luca Trevisan. Multiway Spectral Partitioning and Higher-Order Cheeger Inequalities. *Journal of the ACM*, 2014.

[88] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017.

[89] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[90] Sihao Lin, Hongwei Xie, Bing Wang, Kaicheng Yu, Xiaojun Chang, Xiaodan Liang, and Gang Wang. Knowledge distillation via the target-aware transformer. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[91] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, 2017.

[92] Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. Rethinking the representational continuity: Towards unsupervised continual learning. In *International Conference on Learning Representations Workshop*, 2022.

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

[94] Zheda Mai, Ruiwen Li, Hyunwoo Kim, and Scott Sanner. Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning. In *IEEE International Conference on Computer Vision and Pattern Recognition Workshops*, 2021.

[95] Riccardo Marin, Arianna Rampini, Umberto Castellani, Emanuele Rodolà, Maks Ovsjanikov, and Simone Melzi. Instant recovery of shape from spectrum via latent space connections. In Vitomir Struc and Francisco Gómez Fernández, editors, *International Conference on 3D Vision*, 2020.

[96] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of learning and motivation*, 1989.

[97] Sanket Vaibhav Mehta, Darshan Patil, Sarath Chandar, and Emma Strubell. An empirical investigation of the role of pre-training in lifelong learning. In *International Conference on Machine Learning*, 2021.

[98] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019.

[99] Luca Moschella, Simone Melzi, Luca Cosmo, Filippo Maggioli, Or Litany, Maks Ovsjanikov, Leonidas J. Guibas, and Emanuele Rodolà. Learning Spectral Unions of Partial Deformable 3D Shapes. *Computer Graphics Forum*, 2022.

[100] Jaehoon Oh, Sungnyun Kim, Namgyu Ho, Jin-Hwa Kim, Hwanjun Song, and Se-Young Yun. Understanding cross-domain few-shot learning based on domain similarity and few-shot difficulty. In *Advances in Neural Information Processing Systems*, 2022.

[101] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (ToG)*, 2012.

[102] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *IEEE International Conference on Computer Vision*, 2019.

[103] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. FiLM: visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[104] Juan-Manuel Perez-Rua, Xiatian Zhu, Timothy M Hospedales, and Tao Xiang. Incremental few-shot object detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2020.

[105] Federico Pernici, Matteo Bruni, Claudio Baecchi, Francesco Turchini, and Alberto Del Bimbo. Class-incremental learning with pre-allocated fixed classifiers. In *International Conference on Pattern Recognition*, 2021.

[106] Quang Pham, Chenghao Liu, and Steven Hoi. DualNet: Continual Learning, Fast and Slow. In *Advances in Neural Information Processing Systems*, 2021.

[107] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. GDumb: A simple approach that questions our progress in continual learning. In *Proceedings of the European Conference on Computer Vision*, 2020.

[108] 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*, 2021.

[109] Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. Effect of scale on catastrophic forgetting in neural networks. In *International Conference on Learning Representations Workshop*, 2022.

[110] Arianna Rampini, Franco Pestarini, Luca Cosmo, Simone Melzi, and Emanuele Rodolà. Universal Spectral Adversarial Attacks for Deformable Shapes. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2021.

[111] Roger Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological Review*, 1990.

[112] 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*, 2017.

[113] Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 1995.

[114] Emanuele Rodolà, Luca Cosmo, Michael M Bronstein, Andrea Torsello, and Daniel Cremers. Partial functional correspondence. In *Computer Graphics Forum*, 2017.

[115] 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.

[116] Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*, 2018.

[117] Hang Shao, Abhishek Kumar, and P. Thomas Fletcher. The Riemannian Geometry of Deep Generative Models. In *IEEE International Conference on Computer Vision and Pattern Recognition Workshops*, 2018.

[118] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[119] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[120] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, 2017.

[121] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 2016.

[122] Alistair Sinclair and Mark Jerrum. Approximate counting, uniform generation and rapidly mixing Markov chains. *Information and Computation*, 1989.

[123] James Smith, Yen-Chang Hsu, Jonathan Balloch, Yilin Shen, Hongxia Jin, and Zsolt Kira. Always be dreaming: A new approach for data-free class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[124] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2023.

[125] Mohammad S Sorower. A literature survey on algorithms for multi-label learning. *Oregon State University, Corvallis*, 2010.

[126] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, 2019.

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

[128] Gido M van de Ven and Andreas S Tolias. Three continual learning scenarios. In *Neural Information Processing Systems Workshops*, 2018.

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

[130] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[131] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 2016.

[132] Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software*, 1985.

[133] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, California Institute of Technology, 2011.

[134] Jiahao Wang, Mingdeng Cao, Shuwei Shi, Baoyuan Wu, and Yujiu Yang. Attention probe: Vision transformer distillation in the wild. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022.

[135] Runqi Wang, Xiaoyue Duan, Guoliang Kang, Jianzhuang Liu, Shaohui Lin, Songcen Xu, Jinhu Lü, and Baochang Zhang. Attriclip: A non-incremental learner for incremental knowledge learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2023.

[136] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017.

[137] 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 *Proceedings of the European Conference on Computer Vision*, 2022.

[138] 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 conference on Computer Vision and Pattern Recognition*, 2022.

[139] Yujie Wei, Jiaxin Ye, Zhizhong Huang, Junping Zhang, and Hongming Shan. Online prototype learning for online continual learning. In *IEEE International Conference on Computer Vision*, 2023.

[140] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019.

[141] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018.

[142] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[143] Guanglei Yang, Enrico Fini, Dan Xu, Paolo Rota, Mingli Ding, Moin Nabi, Xavier Alameda-Pineda, and Elisa Ricci. Uncertainty-aware contrastive distillation for incremental semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[144] Zhendong Yang, Zhe Li, Ailing Zeng, Zexian Li, Chun Yuan, and Yu Li. Vitkd: Practical guidelines for vit feature knowledge distillation. *arXiv preprint arXiv:2209.02432*, 2022.

[145] Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. Boosting continual learning of vision-language models via mixture-of-experts adapters. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2024.

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

[147] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, 2017.

[148] Mengyao Zhai, Lei Chen, Jiawei He, Megha Nawhal, Frederick Tung, and Greg Mori. Piggyback GAN: Efficient Lifelong Learning for Image Conditioned Generation. In *Proceedings of the European Conference on Computer Vision*, 2020.

[149] Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. SLCA: slow learner with classifier alignment for continual learning on a pre-trained model. In *IEEE International Conference on Computer Vision*, 2023.

[150] Yaqian Zhang, Bernhard Pfahringer, Eibe Frank, Albert Bifet, Nick Jin Sean Lim, and Yunzhe Jia. A simple but strong baseline for online continual learning: Repeated Augmented Rehearsal. In *Advances in Neural Information Processing Systems*, 2022.

[151] Zangwei Zheng, Mingyuan Ma, Kai Wang, Ziheng Qin, Xiangyu Yue, and Yang You. Preventing zero-shot transfer degradation in continual learning of vision-language models. In *IEEE International Conference on Computer Vision*, 2023.

[152] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 2022.