論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	Balancing Old and New Classes for Class-Incremental Learning Methods
著者(和文)	JODELETQUENTIN MAGIC
Author(English)	Quentin Magic Jodelet
出典(和文)	学位:博士(学術), 学位授与機関:東京科学大学, 報告番号:甲第32号, 授与年月日:2024年12月31日, 学位の種別:課程博士, 審査員:村田 剛志,岡崎 直観,篠田 浩一,横田 理央,金崎 朝子
Citation(English)	Degree:Doctor (Academic), Conferring organization: Institute of Science Tokyo, Report number:甲第32号, Conferred date:2024/12/31, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Balancing Old and New Classes for Class-Incremental Learning Methods

Author: Quentin Jodelet
Supervisor: Tsuyoshi Murata

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Department of Computer Science
School of Computing
Institute of Science Tokyo
(formerly Tokyo Institute of Technology)

December 2024

Abstract

In the Class-Incremental Learning setting, instead of being trained on the complete dataset at once, machine learning models are trained sequentially on new classes while training data from previously learned classes are no longer available. This training paradigm, closer to the human learning process, is challenging for deep neural network models due to catastrophic forgetting: when learning new classes, the model tends to forget the old classes resulting in a significant deterioration of its performance.

This thesis focuses on Class-Incremental Learning for deep neural networks trained for image classification. In order to mitigate catastrophic forgetting, we explore the important trade-off between old and new classes and propose three distinct approaches meant to be seamlessly combined with state-of-the-art approaches for Class-Incremental Learning. First, we propose to balance the contribution of samples from old and new classes in the classification loss by introducing the Balanced Softmax Cross Entropy for Class-Incremental Learning with a rehearsal memory and then extend it to the more challenging setting of Exemplar-Free Class-Incremental Learning where it is impossible to save any samples of past classes. Secondly, we propose to balance the training dataset itself by compensating for the limited size of the replay memory by using additional synthetic samples of past classes generated using pre-trained text-to-image diffusion models. Finally, we propose to pre-train the feature extractor of models designed for Exemplar-Free Class-Incremental Learning better by leveraging knowledge about possible future classes during the initial training step. To do so, our method relies on pre-trained text-to-image diffusion models to generate synthetic samples of future classes in a fast and economical manner.

Extensive experiments on challenging benchmarks for Class-Incremental Learning highlight that our proposed methods significantly improve the performances of state-of-the-art approaches.

Acknowledgement

Firstly, I would like to thank my thesis advisor, Professor Tsuyoshi Murata, for his support and guidance throughout my entire doctoral studies. Secondly, I would like to thank Dr. Xin Liu for his kind mentoring, support, discussion, and invaluable advice. I thank them both for offering me the opportunity to join the National Institute of Advanced Industrial Science and Technology (AIST) and for providing me with all the necessary resources to pursue my research. I also would like to thank Dr. Phua Yin Jun for his in-depth discussion and advice. This thesis would not have been possible without their guidance. Moreover, I would like to express my gratitude to the committee members of my doctoral thesis, Professor Koichi Shinoda, Professor Naoaki Okazaki, Professor Rio Yokota, and Professor Asako Kanezaki for their dedicated time and constructive feedback. I would like to thank former and current members of the Murata laboratory for their encouragement, discussions, and research collaboration.

Finally, I would like to thank Aurélie Peng for her continuous support and her numerous proofreading, as well as my parents for always supporting me and for giving me the opportunity to pursue a doctoral degree.

Contents

List of Figures

List of Tables

List of Acronyms

- **CIL** Class-Incremental Learning
- **CL** Continual Learning
- **CNN** Convolutional Neural Network
- **CV** Computer Vision
- **DIL** Domain Incremental Learning
- **DL** Deep Learning
- **DNN** Deep Neural Network
- **EBCIL** Exemplar-Based Class-Incremental Learning
- **EFCIL** Exemplar-Free Class-Incremental Learning
- **IL** Incremental Learning
- **LFS** Learning From Scratch
- **LFH** Learning From Half
- **TIL** Task Incremental Learning

Notations

- T the total number of incremental steps.

- $\mathcal{D}$ the training dataset.
- $\mathcal{D}_t$ the training dataset for the incremental step t.
- $\mathcal{M}$ the replay memory containing few exemplars from past classes.
- $\mathcal{S}$ the additional dataset of external data.
- $\mathcal{Y}_t$ the set of new classes at the incremental step t.
- θ_t the complete model at the incremental step t, composed of $\{\Phi_t, \mathcal{W}_t\}$.
- Φ_t the feature extractor of θ_t at the incremental step t.
- $\mathcal{W}_t$ the classifier of θ_t at the incremental step t.
- G the pre-trained generative model.

Chapter 1

Introduction

1.1 Scope

During the past decades, tremendous research on artificial intelligence has lead to the development of machine learning models achieving similar, or even superior performances compared to humans on a large variety of tasks: from computer vision (image classification, object detection, image generation, and so on) [Zhao et al., 2019; Minaee et al., 2021; Ozbulak et al., 2023; Croitoru et al., 2023] to natural language processing (question-answering, machine translation, text summarization, and so on) [Young et al., 2018; Zhao et al., 2023], and graph processing (recommendation systems, proteins folding, and so on) [Wu et al., 2021b] . Most of those models rely on artificial deep neural networks composed of millions of parameters that are learned beforehand using large datasets containing representative examples of the desired task.

However, contrary to humans who are able to continually learn new concepts and accumulate knowledge throughout their life, deep neural networks can not. They have to be trained on the complete dataset at once and can not learn any additional concepts or data afterwards due to Catastrophic Forgetting [McCloskey and Cohen, 1989; Ratcliff, 1990; French, 1999; Parisi et al., 2019]. This phenomenon describes the tendency of deep neural networks trained on new additional tasks or data to forget the previously learned ones, resulting in a significant degradation of its performances. This inability to continuously learn means that the model has to be retrained from scratch each time new data are collected. In addition to the significant time and computational cost that regularly retraining a model from scratch induces, there are also various settings where it is impossible. For instance, all the previous data may not be available as storing this ever-growing dataset may not be possible due to privacy concerns or memory limitation. Moreover, on devices with limited computational capabilities such as embedded devices, retraining from scratch may not be conceivable. Incremental Learning, often referred to as Continual Learning, aims at mit-

igating catastrophic forgetting and developing new models that are able to continuously learn.

In this thesis, we focus our study of Class-Incremental Learning to deep neural networks models trained for the task of image classification. Image classification is a fundamental task of computer vision which consists in assigning a specific label to an entire image based on its content. While the problem of catastrophic forgetting is not limited to deep neural networks trained for image classification or to computer vision in general, it is by a large margin the most active domain of study of Incremental Learning and concentrate most of the literature.

1.2 Contribution

In this thesis, we aim to propose new solutions for developing deep neural networks models that can incrementally learn new classes. To achieve this goal, our work focuses on the balance between old and new classes in the classification loss, and in the training dataset. We propose three distinct and complementary approaches that can be seamlessly combined with numerous state-of-the-art methods for Class-Incremental Learning in order to improve their performances.

For our first approach, Balanced Softmax for Class-Incremental Learning [Jodelet et al., 2021, 2022], we propose to address the bias toward new classes induced by the limited size of the rehearsal memory by balancing the classification loss using the Balanced Softmax as a direct replacement for the standard Softmax in the Cross Entropy loss. Experiments on competitive benchmarks show that our proposed method can improve the performance of state-of-the-art approaches while also decreasing the computational cost of their training procedure in some cases. Furthermore, we extended this method to Exemplar-Free Class-Incremental Learning and experimentally demonstrated that it can be used to adapt exemplar-based methods to this more challenging setting.

For our second approach, Memory Augmented using Diffusion Model for Class-Incremental Learning [Jodelet et al., 2023, 2024a], we propose to directly balance the training dataset itself at each incremental step by leveraging widely available pre-trained text-to-image diffusion models in order to generate synthetic images of past classes. Detailed experiments highlight that our method can be combined with various state-of-the-art approaches for Class-Incremental Learning to further improve their performances. Moreover, while using synthetic samples, our method achieves similar or even superior results compared to approaches relying on real unlabeled images as additional source of data.

Finally, for our last approach, Future-Proofing Class-Incremental Learning [Jodelet et al., 2024b], we study to what extend synthetic images of future classes generated using pre-trained Diffusion Model can be used for pre-training during the first incremental step. Experiments show that pre-training the model using synthetic samples of future classes achieves better performances than traditional pre-training using real samples of different

classes.

1.3 Thesis organization

The thesis is organized into the following structure:

- In Chapter 2, we provide a detailed introduction to Continual Learning and Class-Incremental Learning. We first define the setting of Class-Incremental Learning and then present the different approaches and state-of-the-art methods for this setting.
- In Chapter 3, we present our first contribution, Balanced Softmax for Class-Incremental Learning. We approach the problem of bias toward new classes for Exemplar-based Class-Incremental Learning and proposed a modification of the classification loss to address it. We then extend it to the more difficult setting of Exemplar-Free Class-Incremental Learning.
- In Chapter 4, we present our second contribution, Memory Augmented using Diffusion Model for Class-Incremental Learning. Following recent works that leverage additional real data for Class-Incremental Learning, we propose a new flexible method relying on pre-trained Diffusion models to generate additional synthetic data of past classes.
- In Chapter 5, we present our last contribution, Future-Proof Class-Incremental Learning. We study how feature extractors can be better prepared by using pre-trained Diffusion models for generating synthetic images of future classes.
- In Chapter 6, we discuss and detail the relation between our different contributions and their limitations.
- In Chapter 7, we summarize our work and discuss future potential directions for our works and for the field of Class-Incremental Learning in general.

1.4 List of Publications

We provide a list of publication that this thesis focuses on:

- **Quentin Jodelet**, Xin Liu, and Tsuyoshi Murata. "Balanced softmax cross-entropy for incremental learning". In *Artificial Neural Networks and Machine Learning – ICANN 2021*, pages 385–396, September 2021.
- Vincenzo Lomonaco, Lorenzo Pellegrini, Pau Rodriguez, Massimo Caccia, Qi She, Yu Chen, **Quentin Jodelet**, Ruiping Wang, Zheda Mai, David Vazquez, German I. Parisi, Nikhil Churamani, Marc Pickett, Issam Laradji, and Davide Maltoni. "CVPR 2020 continual learning in computer vision competition: Approaches, results, current challenges and future directions". *Artificial Intelligence*, 303:103635, February 2022.

- **Quentin Jodelet**, Xin Liu, and Tsuyoshi Murata. "Balanced softmax cross-entropy for incremental learning with and without memory". *Computer Vision and Image Understanding*, 225:103582, December 2022.
- **Quentin Jodelet**, Xin Liu, Yin Jun Phua, and Tsuyoshi Murata. "Class-incremental learning using diffusion model for distillation and replay". In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 3425–3433, October 2023.
- **Quentin Jodelet**, Xin Liu, Yin Jun Phua, and Tsuyoshi Murata. "Memory Augmented using Diffusion Model for Class-Incremental Learning". *Submitted for review*, 2024.
- **Quentin Jodelet**, Xin Liu, Yin Jun Phua, and Tsuyoshi Murata. "Future-Proofing Class-Incremental Learning". *Machine Vision and Applications*, 2024.

While not directly related, the following article has been completed over the course of the PhD:

- Zarina Rakhimberdina, **Quentin Jodelet**, Xin Liu, and Tsuyoshi Murata. "Natural image reconstruction from fmri using deep learning: A survey". *Frontiers in neuroscience*, 15:795488, December 2021.

Chapter 2

Incremental Learning and State-of-the-Art Methods

2.1 Incremental Learning

2.1.1 Motivation

The most prevailing paradigm, also known as Offline Training or Joint Training in the Incremental Learning community, for training deep neural networks models using the gradient descent consists in a single training phase. In this unique phase, the model's parameters are learned through multiple iterations, also named epochs, on a fixed and complete dataset. This dataset has been prepared beforehand in the course of a long process including, but not limited to, the collection of the data samples, the cleaning and curation of the collected data, and in the case of supervised learning, the annotation of each data sample. After the convergence of the model, the training procedure is considered finished and the model will never be updated again. This training paradigm allows deep neural networks to achieve state-of-the-art performances and outperform humans on numerous tasks from various fields including computer vision, natural language processing, and graph processing.

Conversely, in the Incremental Learning paradigm, often referred to as Continual Learning (CL) or Lifelong Learning, all the training data are not jointly available during the single training phase. Instead, the training procedure is divided into several successive phases, each consisting in different training data, with the objective of training a model able to perform well on the data from all the experienced phases. The Incremental Learning paradigm aims to replicate the learning process of humans who are able to continually learn new concepts as they encounter them throughout their entire lives. In this context, re-training the model from scratch during each phase using the union of the old and new data is not possible due to restrictions: Incremental Learning supposes that due to either privacy concerns

or storage capacity, only a small amount of the previously encountered data can be stored and re-used for training during the following phases. In the most extreme case, no previous data at all can be stored and the model has to be trained solely on the new data from the current phase. A simple and naive approach to Incremental Learning consists in continually finetuning the model during each phase using only the newly available data. However, such a model would be able to achieve satisfactory performance only on the data from the last training phase and not on the previous ones due to the tendency of deep neural networks to forget previously learned knowledge while being trained on new data. Mitigating this phenomenon, known as catastrophic forgetting [McCloskey and Cohen, 1989; Ratcliff, 1990; French, 1999; Parisi et al., 2019], is at the core of Incremental Learning.

While catastrophic forgetting is not limited to deep neural networks trained for classifying images, this thesis focuses on the study of Incremental Learning for the task of image classification as the Incremental Learning literature has largely been dominated by this particular task of computer vision. However, it can noted that during the past few years, there has been an increase in the interest for Incremental Learning applied to other tasks of computer vision such as object detection [Perez-Rua et al., 2020; Feng et al., 2022], semantic segmentation [Cermelli et al., 2020; Douillard et al., 2021a], unsupervised visual representation learning [Madaan et al., 2021; Fini et al., 2022], video understanding [Park et al., 2021], text recognition [Zheng et al., 2023], text-to-image generation [Smith et al., 2023a], and vision-language model Ni et al. [2023]; Garg et al. [2023], as well as for other domain of machine learning such as graphs learning [Febrinanto et al., 2023] and natural language processing [Biesialska et al., 2020; Wu et al., 2022]. Several of these works highlight that methods initially proposed for incremental learning of image classification models can also be applied or adapted to these other tasks and modalities.

2.1.2 The Stability-Plasticity Dilemma

The Stability-Plasticity dilemma is a fundamental challenge of Incremental Learning. Stability, sometimes referred to as rigidity, refers to the capacity of the model to retain past knowledge. Plasticity refers to the capacity of the model to change and accumulate new knowledge. Heavily favoring plasticity would lead to catastrophic forgetting of the old data by the model, while on the other side, heavily favoring stability would result in the incapacity of the model to adapt to new data. Therefore, Incremental Learning approaches have to balance this trade-off between stability and plasticity in order to build a model capable of learning new concepts without forgetting the previous ones.

2.1.3 Incremental Learning settings

Incremental Learning is composed of a multitude of settings among which the three principal are [Hsu et al., 2018; Van de Ven and Tolias, 2019]:

- **Class-Incremental Learning (CIL):** In this setting, the model is sequentially

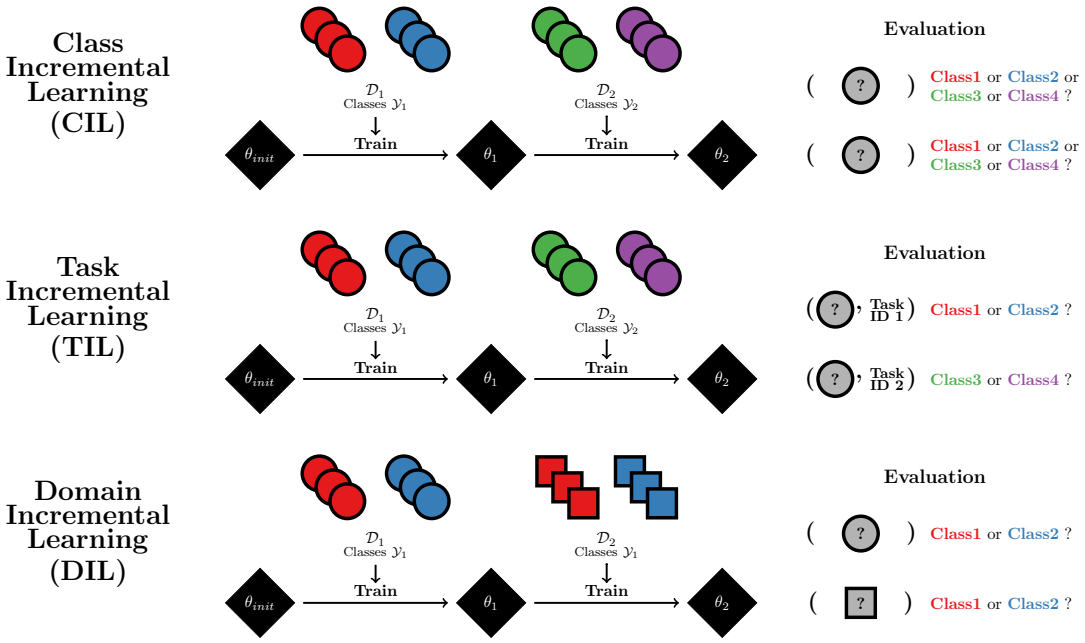


Figure 2.1: Illustration of the training and evaluation procedure for Class-Incremental Learning (CIL), Task-Incremental Learning (TIL), and Domain-Incremental Learning (DIL). Each color represents a class, each shape represents a domain.

trained on incremental steps, also referred to as tasks, each containing classes from disjoint label spaces. During inference, the model has to classify samples from all previously encountered classes without having access to the task identity. *This setting is studied in this thesis.*

- **Task-Incremental Learning (TIL):** Similarly to Class-Incremental Learning, in this setting, the model is sequentially trained on incremental steps, each containing classes from disjoint label spaces. However, during inference, the model has access to the task identity which limits the prediction space to classes of this exact task. While Class-Incremental Learning models have to learn to discriminate among every encountered class, Task-Incremental Learning models only need to learn to discriminate among classes from the same task; this makes the latter setting easier. *This setting is not studied in this thesis.*

- **Domain-Incremental Learning (DIL):** In this setting, the model is continually trained on incremental steps, each containing classes from the same label space but from different input distributions: for example in the case of images classification, the first incremental step would contain photos, the second would contain paintings, the third cliparts ...etc. Similarly to Class-Incremental Learning, the model does not have access to the task identity during inference. *This setting is not studied in this thesis.*

In addition to these three general settings, there are various additional settings such as Online Continual Learning (OCL) [Mai et al., 2022] which only allows a single pass on the training data for Class and Domain Incremental Learning, Class-Incremental with Repetition (CIR) [Hemati et al., 2023] which is Class-Incremental Learning where past classes may be encountered again in the following incremental step, i.e. the label space of each step are not disjoint, or Incremental Implicitly-Refined Classification (IIRC) [Abdelsalam et al., 2021] where each sample has both a coarse label and a fine label, but only one is provided and the model has to determine the other one if it has already encountered it. *These settings are not studied in this thesis.*

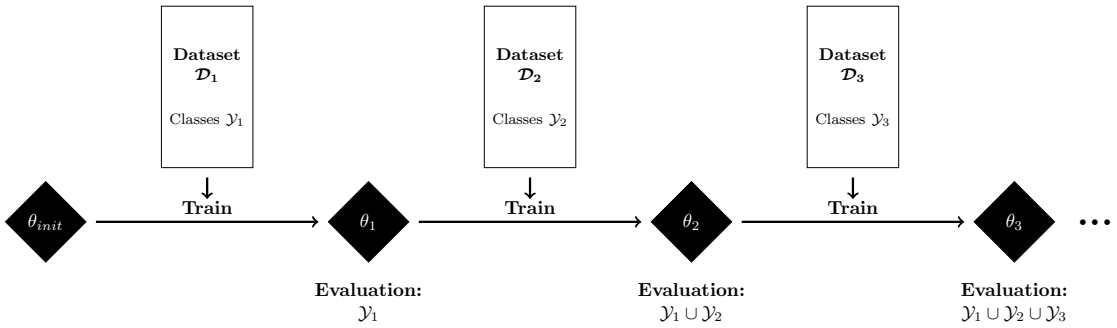
2.2 Class-Incremental Learning

In this thesis, we consider two Class-Incremental Learning [Belouadah et al., 2021; Zhou et al., 2023b] settings: the most actively studied Exemplar-Based Class Incremental Learning (EBCIL), often simply referred to as Class-Incremental Learning, that allows the use of a rehearsal memory for replaying old data, and the more recent and more challenging Exemplar-Free Class Incremental Learning (EFCIL), sometimes referred to Class-Incremental Learning without Memory, which does not allow to save any sample from previously learned classes.

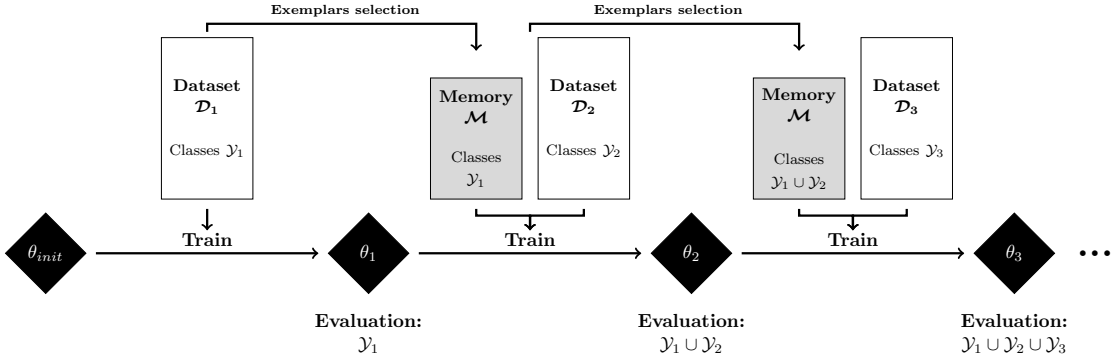
2.2.1 Definition

The Class-Incremental Learning training procedure is divided into T incremental steps, also referred to as tasks, numbered from 1 to T. Each incremental step consists of a training dataset $\mathcal{D}_t$ containing data samples belonging to classes from the set of classes $\mathcal{Y}_t$. Each incremental step only contains classes that were not previously encountered, i.e. there are no classes overlapping between incremental steps, $\forall i, j \in \{1, ..., T\}, \mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset$ for $i \neq j$. The objective is to train a model θ_t which is able to classify samples from all previously learned classes $\bigcup_{i=1}^{t} \mathcal{Y}_i$ without having access to any step or task identity.

In the general case of Exemplar-Based Class Incremental Learning (EBCIL), in addition to the current dataset $\mathcal{D}_t$ containing samples from the new classes $\mathcal{Y}_t$, the model also has access to the rehearsal memory $\mathcal{M}$ containing few samples, also referred to as exemplar, from old classes $\bigcup_{i=1}^{t-1} \mathcal{Y}_i$. The model is then trained on $\mathcal{D}_t \cup \mathcal{M}$. In the specific case of Exemplar-Free Class Incremental Learning (EFCIL), it is not possible to use a rehearsal memory to store samples from past classes and therefore the model is trained solely using $\mathcal{D}_t$. Both training settings are illustrated in Figure 2.2. Additionally, as discussed in Chapter 4, the model may also have access to an additional source of external data denoted $\mathcal{S}$. This additional source of data may or may not change at each incremental step, and may either be an online, and potentially infinite stream of data, that the model can fetch when needed or a fixed set of data accessible in an offline manner.



(a) Exemplar-Free Class-Incremental Learning



(b) Exemplar-Based Class-Incremental Learning

Figure 2.2: Illustration of the training procedure for (a) Exemplar-Free Class-Incremental Learning and (b) Exemplar-Based Class-Incremental Learning.

The model θ_t can be divided into two parts: the feature extractor Φ_t and the classifier $\mathcal{W}_t$. While the use of a feature extractor pre-trained on a large general dataset for initializing Φ is more and more studied in the Incremental Learning community, most works assume that Φ is randomly initialized at the beginning of the first incremental step.

All the notations are shown in the "Notations" section at the beginning of the thesis.

2.2.2 Training Settings

The training settings for Class-Incremental Learning can be characterized by the distribution of the classes into incremental steps, and by the rehearsal memory budget and its management.

Classes Distribution

Two procedures are commonly used by the Class-Incremental Learning research community for distributing the classes into incremental steps: Learning From Scratch (LFS) and

Learning From Half (LFH).

- **Learning From Scratch (LFS):** For this procedure, the classes are evenly divided into the T incremental steps. Supposing C the total number of classes, i.e. $C = \bigcup_{i=1}^{T} \mathcal{Y}_i$, each incremental step consists in learning C/T new classes.
- **Learning From Half (LFH):** This procedure was first introduced by Hou et al. [2019]. For this procedure, the first incremental step, often named the base step, contains half of the total classes and the remaining classes are evenly divided into the following $T-1$ incremental steps. Supposing C the total number of classes, the first incremental step consists in learning $C/2$ classes and is followed by $T-1$ incremental steps containing $C/(2(T-1))$ new classes.

Following [Zhou et al., 2023b] we refer to these two procedures using the unified denomination "Base-i Inc-j", where i is the number of classes in the base step and j is the number of classes in each of the following incremental step. i equals 0 is used to refer to the Training from Scratch procedure.

Rehearsal Memory Budget

For Exemplar-Based Class-Incremental Learning, two rehearsal memory management procedures are commonly used: rehearsal memory with a fixed budget and rehearsal memory with a growing budget. In addition, if no rehearsal memory is allowed, it is referred to as the Exemplar-Free Class-Incremental Learning setting.

- **Fixed Budget Memory:** The total size of the rehearsal memory is fixed. The rehearsal memory stores a constant number of exemplars evenly divided among the classes learned so far at the end of each incremental step. Supposing m the total size of the rehearsal memory, after finishing the t^{th} incremental step, the rehearsal memory contains $\lfloor m/|\bigcup_{i=1}^{t} \mathcal{Y}_i| \rfloor$ exemplars per class, where $\lfloor\ \rfloor$ is the floor function.
- **Growing Budget Memory:** The total size of the rehearsal memory is growing at each incremental step. The rehearsal memory stores exactly n exemplars for each already learned class. Supposing m the current size of the rehearsal memory, after finishing the t^{th} incremental step, we have $m = n|\bigcup_{i=1}^{t} \mathcal{Y}_i|$.

Usually, the Learning From Scratch procedure assumes the use of a rehearsal memory with a fixed budget and the Learning From Half procedure assumes the use of a rehearsal memory with a growing budget.

2.2.3 Evaluation Metrics

In order to evaluate the learned model, it is necessary to define the evaluation metrics. In the case of image classification, the most commonly used metric is the Accuracy. For a

given training dataset $\mathcal{D}$ and its corresponding test dataset $\mathcal{D}_{test}$ containing the same set of classes, the Top-1 Accuracy of the model θ is defined as:

$$\text{acc}(\theta, \mathcal{D}_{test}) = \frac{1}{|\mathcal{D}_{test}|} \sum_{(x,y)\in D_{test}} \mathbb{1}_{f(\theta,x)=y} \tag{2.1}$$

where the test pair (x, y) is composed of the test image x and its corresponding label y, $f(\theta, x)$ is the label predicted by the model θ when passing x as input, $|\ |$ is the cardinal function, and $\mathbb{1}$ is the indicator function that is equal to one if the condition is true and zero otherwise.

In addition to the Top-1 Accuracy, some works also consider the Top-5 Accuracy as an evaluation metric. Subsequently, if not stated otherwise, Accuracy refers to the Top-1 Accuracy.

In the case of general Class-Incremental Learning, the training procedure is divided into T incremental steps, each step containing a training dataset $\mathcal{D}_t$ associated to a test dataset $\mathcal{D}_{test,t}$ with both datasets containing samples belonging to classes from the set of classes $\mathcal{Y}_t$. At the end of each incremental step t, the model θ_t is evaluated on the test set of all the classes learned so far, i.e. $\bigcup_{i=1}^{t} \mathcal{D}_{test,i}$. In this context, the Accuracy of the model at step t is defined as:

$$\text{acc}_t = \text{acc}(\theta_t, \bigcup_{i=1}^{t} \mathcal{D}_{test,i}) \tag{2.2}$$

Two metrics based on this definition of the accuracy at step t are commonly used to evaluate the performance of models for Class-Incremental Learning: the Final Accuracy and the Average Incremental Accuracy.

Final Accuracy

The Final Accuracy is defined as the Accuracy of the model on the complete test dataset after the last incremental step: acc_T, where T is the total number of incremental steps.

Moreover, in order to estimate the trade-off between stability and plasticity, we may be interested in evaluating the Final Accuracy of the model on the base classes $\mathcal{Y}_1$ learned during the first incremental step denoted $\text{acc}_T^{base} = \text{acc}(\theta_T, \mathcal{D}_{test,1})$ and the Final Accuracy on the classes from the following $T-1$ steps denoted $\text{acc}_T^{comp} = \text{acc}(\theta_T, \bigcup_{i=2}^{T} \mathcal{D}_{test,i})$.

Average Incremental Accuracy

The Average Incremental Accuracy has been first proposed by Rebuffi et al. [2017] as a way to evaluate the performance of the model through the course of the complete incremental training procedure. It is defined as the average of the accuracy of the model after each incremental step, including the first one. This is the preferred evaluation metric.

$$\text{acc}_{avg} = \frac{1}{T}\sum_{t=1}^{T}\text{acc}_t = \frac{1}{T}\sum_{t=1}^{T}\text{acc}(\theta_t, \bigcup_{i=1}^{t}\mathcal{D}_{test,i}) \tag{2.3}$$

Other metrics

Additional metrics, such as Forgetting [Chaudhry et al., 2018] or Backward and Forward Transfer [Lopez-Paz and Ranzato, 2017], have been proposed to evaluate some specific aspects of Incremental Learning. Forgetting estimates how much a model has forgotten about a task using the difference between the performance of the current model on that task and the highest performance it has reached in the past. Backward, respectively Forward, Transfer evaluates the impact that learning the task t has on the performance of the old, respectively future, tasks. These metrics were originally proposed for Task-Incremental Learning and are rarely considered for Class-Incremental Learning.

Recent works [Cha et al., 2022; Kim and Han, 2023] have highlighted that state-of-the-art approaches for Class-Incremental Learning tend to highly favor stability resulting in models which feature representation accumulates only little new knowledge through the whole training procedure. They argue that both the Final and Average Incremental Accuracy may not be really indicative of the capability of the model to learn continually and propose to also evaluate the quality of the feature representation when comparing Class-Incremental Learning methods.

2.2.4 Memory and Computational Cost

When comparing approaches, most of the works on Class-Incremental Learning consider the budget for the rehearsal memory as the only memory constraint. The budget for the rehearsal memory, generally expressed as the number of exemplars that can be stored inside, and its management (see Section 2.2.2) are considered as the common denominator which allows a fair comparison of all the considered methods. The budget for storing the weights of the current model itself, and its previous version in the case of distillation-based methods, is not taken into account as most of the works would rely on either the same neural network architecture or architecture with approximately the same number of parameters. With the recent introduction of dynamic-networks-based methods for Class-Incremental Learning, this latter assumption has been questioned. As the size of the deep learning model often grows linearly with the number of incremental steps for dynamic-networks-based methods, Zhou et al. [2023c] have argued that the weights of the models should be taken into account for the total memory budget. They further analyzed in which case it was preferable to store more exemplars of past classes and in which case it was preferable to have larger deep neural network models.

On the other hand, the computational cost of the training procedure is rarely mentioned [Verwimp et al., 2024]. This is most likely due to the emphasis of the research community

on considering the limited size of the rehearsal memory as the central constraint of Class-Incremental learning. Moreover in practice, the difficulty to precisely evaluate the training cost of the different methods further limits its adoption as a metric for comparing Class-Incremental Learning methods.

2.3 Methods for Class-Incremental Learning

Methods designed for continual learning and incremental learning include replay-based methods relying on a rehearsal memory, regularization-based methods which constrain the training of the model by regularizing either the output or the weights of the model, dynamic-networks-based methods which expand the model by allocating new parameters during training, and bias correction methods which mitigate the bias toward new classes.

In practice, most state-of-the-art approaches for Exemplar-based Class-Incremental Learning combine a rehearsal memory with a method to mitigate the bias toward the new classes and an additional mechanism for preserving past knowledge, typically based on either knowledge distillation or network expansion.

2.3.1 Rehearsal Memory

The rehearsal memory [Rebuffi et al., 2017; Chaudhry et al., 2019], also named replay memory, has proven to be a simple yet very effective method for mitigating catastrophic forgetting in Class-Incremental Learning. The rehearsal memory stores few samples, also known as exemplars, of previously learned classes that are used for replay by concatenating it with the current dataset when learning new classes. Following Rebuffi et al. [2017], most of the approaches for Class-Incremental Learning use herding [Welling, 2009] for selecting which exemplars to store in the memory, based on their representation produced by the feature extractor. A recent survey [Belouadah et al., 2021] shows that herding tends to achieve better performance than random selection of exemplars or other algorithms such as K-means or entropy-based selection. Liu et al. [2020b] proposed to parameterize the exemplars from the rehearsal memory and optimize them in an end-to-end manner through a bilevel optimization program.

In order to increase the capacity of the rehearsal memory, it has been proposed to store intermediate representations produced by the feature extractor [Iscen et al., 2020; Hayes et al., 2020] instead of the raw images. Similarly, several authors have studied the use of compression algorithms with the aim of storing more exemplars for a given budget: Zhao et al. [2022] proposed to save low-fidelity exemplars generated using PCA or downsampling, Wang et al. [2022b] proposed to use JPEG compression algorithm, and Luo et al. [2023] proposed to use downsampling on the non-discriminative part of the images. While most approaches for Class-Incremental Learning use a static memory, Reinforced Memory Management (RMM) Liu et al. [2021] learns a two-level policy for allocating the

memory between old and new classes and for allocating the memory to each specific class.

In addition to the previously mentioned methods, which directly store and replay samples from previously encountered classes, some methods proposed to instead generate past samples. Generative Replay, also known as pseudo-rehearsal, methods [Shin et al., 2017; He et al., 2018] rely on a generative model trained alongside the classification model and use it to generate samples of previously learned classes. While those methods can offer more diversity than methods directly storing exemplars, their performances depend on the quality of the generated samples which is challenging for the large scale datasets commonly used for Class-Incremental Learning. Moreover, when trained sequentially, the generative model used for pseudo-rehearsal is also impacted by catastrophic forgetting which makes the problem even more complex.

2.3.2 Distillation-based Methods

Hinton et al. [2015] proposed Knowledge Distillation as a way for transferring knowledge from a large neural network or an ensemble of neural networks, called the teacher, into a smaller neural network called the student. The student model is trained by mimicking the teacher model in order to obtain a student model that achieves competitive, or even superior, performance compared to the teacher.

Distillation-based methods for Class-Incremental Learning use knowledge distillation to build a regularization term to mitigate catastrophic forgetting by retaining knowledge from past classes while learning new ones. At each incremental step t, the current model θ_t is trained on the combination of the classification loss with a distillation loss. This distillation loss constrains the intermediate or the final outputs of the current model θ_t to be the same as those from the previous incremental step model θ_{t-1} which has been frozen. Ideally, knowledge distillation should be performed on the complete dataset used to train the teacher model, which is not possible for Class-Incremental Learning. As a substitute, distillation-based methods use the new training data, the few exemplars from the rehearsal memory, or additional external data [Lee et al., 2019; Zhang et al., 2020; Liu et al., 2024].

Learning Without Forgetting (LWF) [Li and Hoiem, 2017] first used knowledge distillation for Task-Incremental Learning by enforcing the output logits of the old classes for the current model to be the same as the ones of the model from the previous step. This method was subsequently extended to Class-Incremental Learning and combined with a rehearsal memory by Rebuffi et al. [2017]. Logits distillation combined with a memory for replay was further used in numerous works [Castro et al., 2018; Wu et al., 2019]. Instead of using the previous model as the teacher, Buzzega et al. [2020] proposed to store the predicted logits alongside the exemplars using reservoir sampling [Vitter, 1985].

In addition to logits-based knowledge distillation, it is also possible to use feature-based knowledge distillation and relation-based knowledge distillation. Hou et al. [2019] introduced Less-Forget Constraint which is applied to the feature representation. It tackles

catastrophic forgetting by maximizing the cosine similarity between the output features of the current model and those from the previous model. In addition to the final representation from the feature extractor, PODNet [Douillard et al., 2020] proposed a distillation loss that is also applied to the spatially pooled representations from several intermediate stages of the feature extractor. Tao et al. [2020] proposed Topology-Preserving Class-Incremental Learning (TPCIL) which models the topology of the feature space using an elastic Hebian graph and proposed a loss that penalizes change in neighboring relationships of the graph while learning new classes. GeoDL [Simon et al., 2021] proposed to model the feature space of the current and previous step model using two low-dimensional manifolds and apply the distillation loss along the geodesic path connecting them. DDE [Hu et al., 2021] proposed to distill the colliding effect between the new and old data. AFC [Kang et al., 2022] weighted the distillation loss based on the importance of each feature map.

Rather than only using the model from the previous step θ_{t-1}, Zhou et al. [2019] proposed to use all the previous models $\theta_{1:t-1}$ for the distillation loss. To achieve a better trade-off between plasticity and stability, Kim et al. [2023] introduces an auxiliary model that has been specifically trained for the current incremental step and proposes to train the current model θ_t by distilling knowledge from both the previous model θ_{t-1}, for promoting stability, and the auxiliary model, for promoting plasticity, using the same distillation loss.

2.3.3 Weights-Regularization-based Methods

While distillation-based approaches for Class-Incremental Learning aim to mitigate catastrophic forgetting by constraining the changes in the output of the model, weights-regularization-based methods, on the other hand, propose to constrain the changes in the weights of the deep neural networks. To do so, at each incremental step t, the current model θ_t is trained on the combination of the classification loss with a parameters-regularization loss. This regularization loss usually penalizes the change of each of the weights of the current model depending on their importance for carrying out previously learned tasks.

Kirkpatrick et al. [2017] proposed Elastic Weight Consolidation (EWC) which estimates the importance of the parameters at the end of each incremental step using the Fisher Information Matrix. Synaptic Intelligence (SI) [Zenke et al., 2017] estimates the importance of each parameter in an online manner using their contribution to change in the loss through the entire training procedure. RWalk [Chaudhry et al., 2018] combines EWC and SI together in order to estimate the importance of each weight.

2.3.4 Dynamic-Networks-based Methods

As an alternative to accumulating the knowledge from all the incremental steps into a fixed and shared set of weights, dynamic-networks-based methods propose to increase the capacity of the deep neural network to accommodate the evolving stream of training data. At each incremental step, instead of using a distillation loss to protect past knowledge by

regularizing the output of the neural network, dynamic-networks-based methods expand the model by appending new neurons or modules dedicated to the learning of the new classes. This has long been a popular approach for Task-Incremental Learning [Rusu et al., 2016; Yoon et al., 2018; Li et al., 2019] due to the availability of the task identity at test time, allowing the model to process the input through the associated part of the neural network.

Yan et al. [2021] introduced Dynamically Expandable Representation (DER) as the first Dynamic-networks-based method for Class-Incremental Learning. At each incremental step, the model is expanded with an additional feature extractor for learning the new classes, while all the previous feature extractors are kept frozen to preserve past knowledge. The features generated by all the feature extractors are concatenated and fed to the classifier for prediction. To encourage the new feature extractor to learn diverse and discriminative features, the authors emphasize the importance of introducing an auxiliary loss in addition to the classification loss. Concurrently, Li et al. [2021] presented a similar approach but proposed to average the features after adapting them using dedicated linear layers instead of concatenating them and introduced a distillation loss for learning the new feature extractor.

However, by introducing an additional feature extractor at each incremental step, these methods proposed by Yan et al. [2021] and Li et al. [2021] suffer from having models with a linearly growing number of parameters, inducing important memory and computational cost compared to methods relying on a fixed backbone. To address this concern, both methods proposed a different pruning approach to reduce the number of parameters by removing redundant or less important parameters. Inspired by the gradient boosting algorithm, FOSTER [Wang et al., 2022a] proposed a two-stage training procedure. First, it dynamically adds a new feature extractor for learning the new classes and then compress the two feature extractors into a unique one using knowledge distillation; effectively limiting the model to a single feature extractor at the end of each incremental step. Zhou et al. [2023c] analyzed that when adding a new feature extractor at each incremental step, shallow layers of each feature extractor tend to produce highly similar features while only the deeper layers produce diverse features. Consequently, they introduced MEMO which divides the feature extractor into a generalized part and a specialized part. At each incremental step, instead of adding a completely new feature extractor, MEMO keeps a single common generalized part and only adds a new specialized part; resulting in lower memory and computational requirements compared to DER. Based on Vision Transformer [Dosovitskiy et al., 2020], Douillard et al. [2022] proposed DyTox which expands the parameter space of the model by learning one additional task-specific token during each incremental step.

2.3.5 Bias Correction

As previously discussed in Section 2.3.1, the rehearsal memory is a key component of most Class-Incremental Learning methods. However, due to its limited size, that is often restricted to 20 exemplars per class, the actual training dataset $\mathcal{D}_t \cup \mathcal{M}$ resulting of the

concatenation of the current datasets $\mathcal{D}_t$ with the rehearsal memory $\mathcal{M}$ is highly imbalanced. It contains a lot of training samples for the few new classes $\mathcal{Y}_t$ from $\mathcal{D}_t$ and only few training samples for the numerous past classes $\bigcup_{i=1}^{t-1} \mathcal{Y}_i$ from the rehearsal memory $\mathcal{M}$. This imbalance in the training dataset results in a severe bias of the model toward the new classes [Javed and Shafait, 2019; Hou et al., 2019; Wu et al., 2019; Belouadah and Popescu, 2019; Zhao et al., 2020]: the model tends to predict every sample from old classes as new classes. The linear classifier exhibits significantly larger values for the bias and the norm of the weights associated to the new classes compared to old classes. To address this bias in Class-Incremental Learning models, several methods have been proposed.

Rebuffi et al. [2017] first introduced the use of a Nearest-Mean-of-Exemplars (NME) classifier during inference instead of the learned linear classifier in order to mitigate the bias toward new classes. The NME classifier computes a class prototype for each class by averaging the feature representation of its exemplars stored in the rehearsal memory, and at test time, assigns the label of the class whose prototype is the most similar to the feature representation of the input. Castro et al. [2018] proposed to finetune the linear classifier at the end of each incremental step on a small balanced dataset created using the rehearsal memory and containing the same number of samples for both old and new classes. This balanced finetuning is among the most commonly used methods for bias mitigation for Class-Incremental Learning and is part of several approaches Hou et al. [2019]; Douillard et al. [2020]; Yan et al. [2021]. Bias Correction (BiC) [Wu et al., 2019] learn a two-parameter model to rectify the bias in the weight and bias of the linear classifier using a small balanced validation set at the end of each incremental step. Hou et al. [2019] introduced the cosine classifier as a solution to balance the magnitude across all classes. Weight Aligning (WA) [Zhao et al., 2020] corrects the weights of the new classes at the end of each incremental step in such a way that the average norm of the weights for new classes is the same as the one for past classes. Ahn et al. [2021] proposed the Separated-Softmax that splits the output logits into two distinct parts, the new classes part and the old classes part, and compute the Cross Entropy classification loss separately.

2.4 Methods for Exemplar-Free Class-Incremental Learning

Exemplar-Free Class-Incremental Learning has recently gained more and more attention from the continual learning community. Compared to Exemplar-Based Class Incremental Learning, it does not allow the use of a rehearsal memory, resulting in a more challenging setting.

In addition to the general methods presented in the previous section, several approaches have been proposed specifically for this setting. They can be categorized depending on whether the feature extractor is fixed or continually finetuned.

2.4.1 Finetuning-based Methods

Finetuning-based methods for Exemplar-Free Class Incremental Learning sequentially update the feature extractor at each incremental step. Similarly to methods for Exemplar-Based Class-Incremental Learning, they usually rely on a regularization loss to mitigate catastrophic forgetting. Learning Without Forgetting [Li and Hoiem, 2017] adapted to Class-Incremental Learning (LWF-MC) [Rebuffi et al., 2017] and EWC [Kirkpatrick et al., 2017] are two pioneer works for Exemplar-Free Class-Incremental Learning used to regularize the continual finetuning of the feature extractor in order to mitigate catastrophic forgetting.

Yu et al. [2020] showed that feature extractors trained using a metric learning loss suffer less from catastrophic forgetting compared to feature extractors trained using a classification loss. Several authors have highlighted the importance of learning a feature extractor capable of producing more generalizable and transferable features and proposed to use self-supervised learning to do so. Zhu et al. [2021b] proposed to generate novel classes by applying 90°, 180°, and 270° rotations to the current training dataset, transforming the N-class problem into a $(4N)$-class problem. Concurrently, Wu et al. [2021a] also proposed to use rotations and introduced two approaches: SPB-I which incorporates a class-independent learner based on self-supervised rotation prediction task, and SPB-M which incorporates a dedicated classifier per transformed view. Similarly, Zhu et al. [2021a] transformed the N-class problem of the current step into a $(N + M)$-class problem by using Mixup Zhang et al. [2017] for generating M new classes which are used for training the feature extractor before being discarded at the end of the incremental step. Smith et al. [2021] generates synthetic samples of past classes by inverting a frozen copy of the model from the previous step and designs a new importance-weighted feature distillation loss. SSRE [Zhu et al., 2022] combines knowledge distillation with structural expansion and re-parameterization.

Several methods for Exemplar-Free Class-Incremental Learning additionally store one representative prototype for each class, which is usually the class mean in the feature space. However, stored prototypes of previously encountered classes become more and more irrelevant and outdated as the feature extractor is continually finetuned on the data from new classes. To address this change in the feature space, SDC [Yu et al., 2020] compensates the drift of the prototypes from previous incremental steps by using an estimation based on the feature drift experienced by the current step data and uses the prototypes in a Nearest Class Mean (NCM) classifier for inference. To maintain the decision boundary of the previous classes, Zhu et al. [2021b] proposed to augment the prototypes of past classes via Gaussian noise and train the classifier using features from new classes and the augmented prototypes. Zhu et al. [2021a] additionally saves the covariance matrix of each old class in order to generate features of past classes. Zhu et al. [2022] over-samples the prototypes while training the classifier and proposes a prototype selection mechanism to reduce confusion among similar classes.

2.4.2 Fixed-Representation-based methods

Instead of finetuning the feature extractor during each incremental step, several authors have proposed to train the feature extractor during the first incremental step before freezing it and only update the classifier during the following incremental steps. Since only the classifier is updated after the first incremental step, the memory and computational costs of the training procedure for methods relying on a fixed feature extractor are usually significantly lower compared to finetuning-bases methods, resulting in a faster training procedure. On top of the frozen feature extractor, different classifiers can be incrementally trained to make the most of the fixed representation. Using a Nearest Class Mean (NCM) classifier [Rebuffi et al., 2017] on top of the frozen feature extractor is a strong baseline [Janson et al., 2022]. DeeSIL [Belouadah and Popescu, 2018] proposed to use linear SVMs for the classifier and Hayes and Kanan [2020] relies on Streaming Linear Discriminant Analysis [Pang et al., 2005]. FeTrIL [Petit et al., 2023] proposed to generate pseudo-features of the previously encountered classes using one prototype per class and a geometric translation of the new class features; then to train a linear classifier using the pseudo-features for the old classes and the actual features for the new classes. Goswami et al. [2023] proposed FeCAM which relies on the Mahalanobis distance as well as Tukey's transformation, covariance shrinkage, and correlation normalization.

Additionally, there are prompt-based methods, which learn prompt tokens to instruct and adapt a fixed vision transformer to the new tasks. The pioneer method L2P [Wang et al., 2022d] learns a pool of key-prompt pairs and selects the most relevant prompts at inference based on the input. Dual Prompt [Wang et al., 2022c] proposes to learn task-invariant and task-specific prompts. CODA-Prompt [Smith et al., 2023b] uses an attention-based mechanism for selecting the prompts. However, contrary to previously presented methods which train the feature extractor using the data from the first incremental step, most of the prompt-based methods rely on a vision transformer model pre-trained on large external datasets such as ImageNet and are not competitive when solely pre-trained during the first incremental step [Tang et al., 2023].

Chapter 3

Balanced Softmax for Class-Incremental Learning With and Without Memory

3.1 Introduction

In a class incremental learning scenario, the complete training dataset is not available at once. Instead, the training samples are progressively available, few classes at a time. The model has to learn new classes in a sequential manner, similarly to the human learning process. This ability to incrementally learn new knowledge is critical for various real world applications where new samples may appear sequentially and it is not possible to either store them all due to memory constraint or re-train the model from scratch as new data is encountered due to time or computational power limitations. For example, a robot learning to classify new objects while interacting with its environment or a face recognition system should be able to incrementally learn new knowledge during their lifetime without having to be re-trained on previously learned faces or objects each time a new element is encountered. Although deep neural networks achieve state-of-the-art performance across a wide spectrum of applications in computer vision, training them incrementally remains challenging. It is caused by the high propensity of deep neural networks to almost fully forget previously learned classes while learning new ones. This situation is known as catastrophic forgetting [McCloskey and Cohen, 1989; French, 1999; Parisi et al., 2019].

In this context, using a small memory buffer to store samples of previously encountered classes in order to use them for rehearsal while learning new classes has been proven by multitude of recent works to be an effective solution to mitigate catastrophic forgetting. However, as the size of the replay memory is limited in practice, a large imbalance issue

appears, i.e. at a given time, the model will mainly be trained using data from new classes and only a few from previous classes. This leads the model to become biased towards the most recently learned classes and resulting in a deterioration of its overall performance. This phenomenon gets worse as the size of the memory decreases to the point where no memory at all is available. Methods designed to tackle this issue usually rely on additional finetuning steps using a small balanced dataset after each incremental step, oversampling, or specifically designed classifiers.

In this work, we propose a novel approach to mitigate the bias towards new classes in large-scale class incremental learning with and without a replay memory. Our proposed method modifies the training procedure by using the Balanced Softmax activation function [Ren et al., 2020] for the Cross-Entropy loss instead of the commonly used Standard Softmax function. By combining the Balanced Softmax Cross-Entropy loss with state-of-the-art approaches for class incremental learning with rehearsal memory, it is possible to further improve the average incremental accuracy of those methods on competitive benchmarks. Our method also potentially decreases the computational cost of the training procedure as the Balanced Softmax Cross-Entropy loss does not require any additional balanced finetuning step. Moreover, we propose an additional meta-learning algorithm that can further improve the accuracy of the models in some settings. Finally, we propose an extension of the Balanced Softmax Cross-Entropy, named the Relaxed Balanced Softmax Cross-Entropy, specifically designed for class incremental learning without memory. We show that this newly proposed loss can be applied to approaches initially conceived for class incremental learning with a replay memory in order to adapt them to the exemplar-free setting without further modification. They then achieve state-of-the-art results and are competitive with memory-based approaches.

To summarize, our contributions are three-fold:

- We propose the use of the Balanced Softmax Cross-Entropy as a mitigation method for large-scale class incremental learning and we show that it can be seamlessly combined with advanced state-of-the-art methods in order to improve their accuracy while also decreasing the computational complexity of their training procedure.
- We propose a meta-learning algorithm that can improve the accuracy of the Balanced Softmax Cross-Entropy in some settings.
- We introduce an extension of the Balanced Softmax Cross-Entropy specifically designed for class incremental learning without memory, named Relaxed Balanced Softmax Cross-Entropy. We show that it achieves state-of-the-art performances and can efficiently balance the trade-off between stability and plasticity.

In this work, we extend our previously published conference paper [Jodelet et al., 2021] by considering additional settings and experiments for our proposed method. First, we explore the class incremental learning without memory setting. In this setting, no exemplar

of old classes can be stored in the memory, which makes it a challenging problem. It is also not well-addressed by existing methods. We highlight the limitations of the Balanced Softmax Cross-Entropy in this particularly demanding setting and propose an improved version, named Relaxed Balanced Softmax Cross-Entropy, which significantly improves the accuracy of our proposed method and achieves state-of-the-art performances. Secondly, we perform additional experiments on two more challenging scenarios for class incremental learning with replay memory: the 25 incremental steps setting on CIFAR100 and ImageNet-Subset and the 10 incremental steps setting without large base step on ImageNet as initially used by Rebuffi et al. [2017] and Wu et al. [2019]. Finally, we also add supplementary ablation study and figures.

3.2 Related work

In recent years, continual learning and incremental learning for deep neural networks gained more and more attention, following the success of deep learning based methods during the past decade. There exists a large variety of scenarios and settings for continual learning [Hsu et al., 2018; Lomonaco et al., 2022]. In our work, we will only consider the class incremental learning scenario. Most methods designed for this scenario usually combine three distinct components when applied to large scale datasets: constraints to preserve past knowledge, rehearsal memory, and bias mitigation methods.

Hinton et al. [2015] initially proposed the distillation loss for transferring knowledge from a large teacher model into a smaller student model. Li and Hoiem [2017] adapted this method to continual learning by distilling the knowledge of the model learned during the previous step into the next step using the output logits of the models. This method was then applied by several authors [Rebuffi et al., 2017; Castro et al., 2018; Wu et al., 2019]. Recently, several proposals have been made to modify and improve distillation for incremental learning. Hou et al. [2019] proposed a novel distillation loss named Less forget Constraint which is applied on the final class embeddings instead of the output logits. Dhar et al. [2019] proposed the Attention Distillation Loss which penalizes changes in the attention maps of the classifiers. Douillard et al. [2020] proposed a new distillation loss using a pooling function and applied it to several intermediate layers of the neural network in addition to the final class embedding. Tao et al. [2020] proposed to model the class embedding topology using an elastic Hebbian graph and then used a topology-preserving loss to constrain the change of the neighboring relationships of the graph during each incremental step. Similarly, Lei et al. [2020] adopted a feature-graph preservation approach and proposed the weighted-Euclidean regularization to preserve the knowledge. Lastly, Simon et al. [2021] proposed to use the geodesic flow for knowledge distillation in order to preserve past knowledge more efficiently.

Using a small replay memory containing samples from previously encountered classes [Chaudhry et al., 2019] has been shown to be a simple yet effective method to mitigate

catastrophic forgetting. In order to increase the number of stored samples for a fixed size, it is possible to rely on compression [Caccia et al., 2020] or to store intermediate representations [Hayes et al., 2020] instead of the input images. It is also possible to store the output logits for distillation instead of the true labels if the model from the previous step is not available [Buzzega et al., 2020]. Liu et al. [2020b] proposed to parameterize the exemplars of the memory and to learn them in an end-to-end manner through bi-level optimizations.

However, the limited size of the replay memory results in an unbalanced training set mainly composed of samples from the new classes. Several recent works have highlighted this problem and proposed different methods to address it. Rebuffi et al. [2017] introduced iCaRL which relies on a nearest-mean-of-exemplars classifier to tackle the issue. Castro et al. [2018] proposed to use more data augmentation on the training set and to finetune the model on a small balanced dataset after each incremental step. This additional balanced finetuning step was then reused in numerous works [Hou et al., 2019; Douillard et al., 2020]. Wu et al. [2019] and Hou et al. [2019] observed that the classifier part of the neural network is the main reason for the bias of the model towards the new classes. Therefore, Wu et al. [2019] proposed to learn a two-parameter linear model on a small balanced dataset to correct the bias of the last fully connected layer while Hou et al. [2019] proposed to use cosine normalization on the classifier. A recent work [Ahn et al., 2021] also proposed to use oversampling of old classes and to separately compute the softmax probabilities of the new and old classes for the Cross-Entropy loss.

Nevertheless, storing samples from past classes inside the replay memory may not always be possible due to privacy reasons. This situation, referred to as exemplar-free incremental learning or incremental learning without memory, significantly complicates the initial problem. Pioneering works proposed to learn a generative model [Shin et al., 2017; Wu et al., 2018] to generate samples from old classes. Recently, Yu et al. [2020] proposed an approach that does not rely on a generative model but which approximates the semantic drift of the embedding space using new data only while Zhu et al. [2021b] and Wu et al. [2021a] proposed to use self-supervised learning to learn richer representations. Concurrently, Hu et al. [2021] proposed a solution that aims to distill the effects of old data without having to store them in the replay memory.

In our work, we propose to use the Balanced Softmax Cross-Entropy loss to address the problem of unbalanced training set for class incremental learning with and without storing samples from past classes in memory. Compared to previously presented methods, the Balanced Softmax Cross-Entropy loss does not require additional training steps or oversampling while reaching similar or higher accuracy on competitive benchmarks.

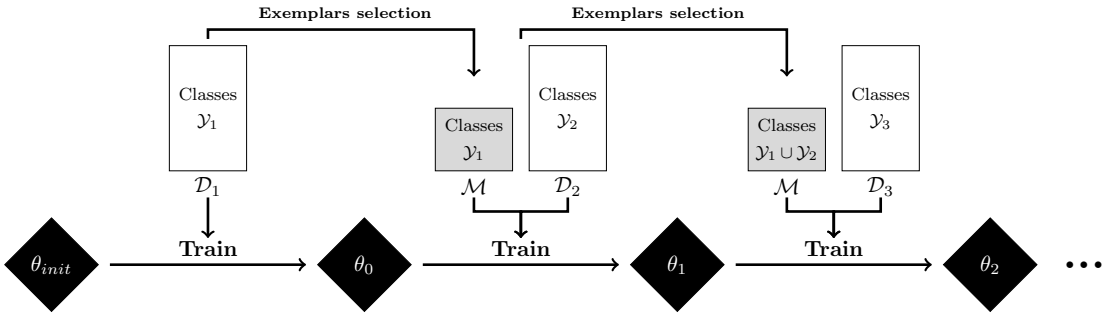


Figure 3.1: Illustration of the class incremental training procedure. At each incremental step t, the model is trained with the new data $\mathcal{D}_t$ containing samples from new classes $\mathcal{Y}_t$ and the memory $\mathcal{M}$ containing few samples from previously encountered classes $\bigcup_{i=1}^{t-1} \mathcal{Y}_t$. The step 1, also called the base step, contains the base classes.

3.3 Proposed method

The objective of class incremental learning is to learn a unified classifier (also denoted single-head classifier) from a sequence of training steps, each containing new and previously unseen classes, as described in Figure 3.1. The first training step called the base step, is followed by several incremental steps numbered from 2 to T, for a total of T training steps, each composed of the training set $\mathcal{D}_t$ containing samples from the set of classes $\mathcal{Y}_t$. Each incremental step contains different classes such that $\forall i, j \in \{1, ..., T\}, \mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset$ for $i \neq j$. During each incremental step, in addition to $\mathcal{D}_t$, the model also has access to the small replay memory $\mathcal{M}$ containing samples from classes encountered during previous incremental steps. The number of classes learned up to the incremental step t included is denoted as N_t. At the end of each training step, the model is evaluated on the test data of all classes seen so far $\bigcup_{i=1}^{t} \mathcal{Y}_i$ where t denotes the current training step.

3.3.1 Incremental learning baseline

To highlight the strengths of our proposed method, we use a simple baseline for incremental learning, denoted IL-baseline, initially proposed by Wu et al. [2019]. This baseline consists of a deep neural network combined with a small replay memory and optimized using two distinct losses: the Standard Softmax Cross-Entropy loss and the distillation loss.

The total loss $\mathcal{L}$ used to train the model, is defined as a weighted sum of the distillation loss $\mathcal{L}_d$ and the Standard Softmax Cross-Entropy loss $\mathcal{L}_c$:

$$\mathcal{L} = \rho \mathcal{L}_d + (1 - \rho)\mathcal{L}_c \tag{3.1}$$

where ρ is defined as $\frac{N_{t-1}}{N_t}$ with N_t the total number of classes at incremental step t and is used to balance the importance of the two losses. The importance of the distillation loss

is modified based on the number of new classes that are learned during the training step compared to the total number of classes that has already been learned.

At the beginning of each incremental step t, the parameters from the previous step θ_{t-1} are first copied to initialize the new parameters θ_t. The old parameters θ_{t-1} are then used to maintain the knowledge of previously learned classes using the distillation loss [Hinton et al., 2015; Li and Hoiem, 2017]. For each training sample $(x, y) \in \mathcal{D}_t \cup \mathcal{M}$, the distillation loss $\mathcal{L}_d$ is defined as:

$$\begin{aligned} \mathcal{L}_d(x) &= \sum_{k=1}^{N_{t-1}} -\hat{p}_k(x) \log(p_k(x))\, T^2 \,, \\ \hat{p}_k(x) = \frac{e^{\hat{z}_k(x)/T}}{\sum_{j=1}^{N_{t-1}} e^{\hat{z}_j(x)/T}}, &\quad p_k(x) = \frac{e^{z_k(x)/T}}{\sum_{j=1}^{N_{t-1}} e^{z_j(x)/T}} \end{aligned} \tag{3.2}$$

where x is the input image, y is the associated ground truth label, $z(x) = [z_1(x), ..., z_{N_t}(x)]$ is the output logits of the current model θ_t, $\hat{z}(x) = [\hat{z}_1(x), ..., \hat{z}_{N_{t-1}}(x)]$ is the output logits of the model at the previous incremental step θ_{t-1} and T is the temperature.

Usually, the replay memory can be constrained by either a global budget or by the number of samples saved per class. Using a global budget means that the total memory size is fixed at the beginning of the training procedure and is equally divided among the already learned classes. As the number of already encountered classes increases, the number of samples stored for each class decreases. On the other hand, using a fixed number of samples stored per class, also known as growing memory, means that the number of samples per class will not change during the training procedure, which means that the total size of the memory increases at each incremental step. The latter approach is usually considered to be the more challenging one. The herding selection [Welling, 2009] is often used to select the samples as it has been shown to be more efficient than the random selection [Belouadah et al., 2021].

3.3.2 Balanced Softmax Cross-Entropy

Due to the limited size of the replay memory, the training set $\mathcal{D}_t \cup \mathcal{M}$ contains only few tens of samples for each of the old classes while containing hundreds or thousands of samples for each of the new classes at each incremental step. The testing set, however, contains the same number of samples for both the new classes and the old classes. This discrepancy induces a bias towards the most recently learned classes [Belouadah and Popescu, 2019; Hou et al., 2019; Wu et al., 2019] as illustrated in Figure 3.2 with the IL-Baseline procedure. As can be seen from the figure, the model tends to predict the classes which had the largest number of samples in the training set during the last incremental steps (the new classes) rather than the old classes. This situation is similar to the Long-Tailed Visual Recognition problem where a model is evaluated on a balanced test dataset after being trained on a dataset composed of a few classes that are over-represented (the head classes) and a large number of classes that are under-represented (the tail classes).

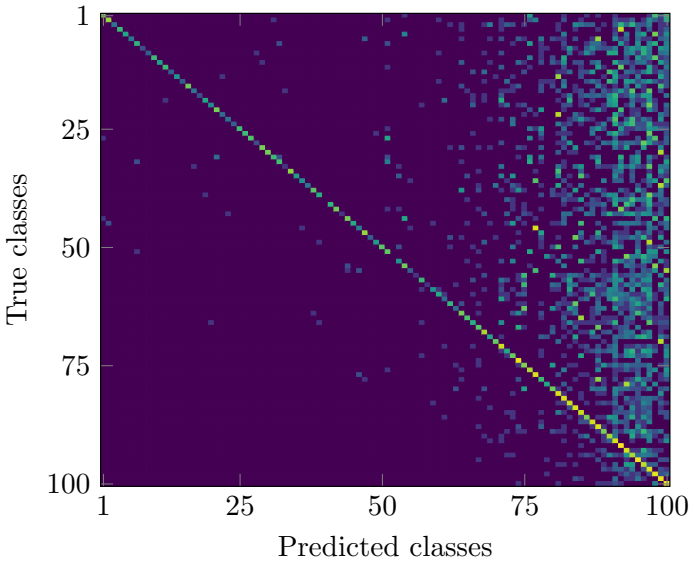


Figure 3.2: Confusion matrix (transformed by log(1 + x) for better visibility) of the IL-Baseline trained on CIFAR100 with a base step containing half of the classes followed by 5 incremental steps, using 20 samples per class stored in the replay memory. Figure best viewed in color.

Based on this observation, we propose to replace the Standard Softmax activation function with the Balanced Softmax for the Cross-Entropy loss during the training procedure. This activation function has been initially introduced by Ren et al. [2020] to address the label distribution shift between the training set and test set in Long-Tailed Visual Recognition.

The Balanced Softmax activation function is defined as:

$$q_k(x) = \frac{\lambda_k e^{z_k(x)}}{\sum_{j=1}^{N_t} \lambda_j e^{z_j(x)}} \text{ with } \lambda_i = n_i \tag{3.3}$$

where x is the input image, $z(x) = [z_1(x), ..., z_{N_t}(x)]$ is the output logits of the current model θ_t, n_i is the number of samples in the training set for the i^{th} class for the current step t and N_t is the number of classes learned up to the current incremental step t included. For detailed analysis of the Balanced Softmax and its relation to the general class imbalance problem, we refer readers to Ren et al. [2020].

The new classification loss $\mathcal{L}_c$ used during the training procedure is then defined as the Cross-Entropy loss using the Balanced Softmax instead of the Standard Softmax:

$$\mathcal{L}_c(x) = \sum_{k=1}^{N_t} - \delta_{k=y} \log(q_k(x)) \tag{3.4}$$

where δ is the indicator function and y is the associated ground truth label to x.

The Balanced Softmax Cross-Entropy loss, denoted BalancedS-CE, can be used as a replacement for the Standard Softmax Cross-Entropy loss in the previously defined IL-Baseline or in any other model for incremental learning. It should be noted that the Balanced Softmax Cross-Entropy does not increase the computational cost of the training procedure in any notable way compared to the Standard Softmax Cross-Entropy.

3.3.3 Meta Balanced Softmax Cross-Entropy

The expression of the Balanced Softmax presented in Equation (3.3) allows for direct control of the importance of each class by selecting a dedicated weighting coefficient λ_i for each of them, which may be different from the number of samples for this class in the training dataset, similarly to Khan et al. [2017]. In the context of large scale incremental learning, the modification of these weighting coefficients offers a new method for controlling the plasticity-rigidity trade-off of the trained model and also for controlling separately the importance of each individual class.

We propose to extend the Balanced Softmax by introducing a new global weighting coefficient α to control the importance of the past classes:

$$q_k(x) = \frac{\lambda_k e^{z_k(x)}}{\sum_{j=1}^{N_t} \lambda_j e^{z_j(x)}} \text{ with } \lambda_i = \begin{cases} \alpha\, n_i, & \text{if } i \in P \\ n_i, & \text{otherwise} \end{cases} \tag{3.5}$$

where P is the set of previously encountered classes and the weighting coefficient α is a real number, usually between 0 and 1.

The expression of the λ_i coefficient for the new classes does not change, but for the old classes, the number of images in the training set n_i is multiplied by the weighting coefficient α. By decreasing α, it is possible to increase the importance of the old classes and reduce the plasticity of the model depending on the specific requirements of the application. This new expression of the Balanced Softmax is equivalent to the one described in Equation (3.3) for α equal to 1.0.

In practice, it appears that 1.0 may not be the optimal value for α when only considering the average incremental accuracy of the model. However, it is difficult to determine beforehand a satisfying value for α without performing several trials with different values. Therefore, to further improve the accuracy of the Balanced Softmax Cross-Entropy for Incremental Learning, we propose a new training procedure, named Meta Balanced Softmax Cross-Entropy (Meta BalancedS-CE), in order to slightly adjust the weighting coefficient α of the Balanced Softmax during the training as described by Algorithm 3.1.

Algorithm 3.1: Meta Balanced Softmax Cross-Entropy training procedure

Input: Data-flow $\{\mathcal{D}_i\}_{i=1}^{T}$
Ouput: Models $\{\theta_i\}_{i=1}^{T}$, and replay memory $\mathcal{M}$.

Initialize parameters θ and function f of the model
Initialize memory $\mathcal{M}$
Train on the base step $\mathcal{D}_1$
for $t \in \{2, 3, ..., T\}$ **do**
    $\alpha \leftarrow 1.0$
    $U_t, B_t \leftarrow \text{split}(\mathcal{D}_t \cup \mathcal{M})$
    for *epoch* $\in \{1, 2, ..., E\}$ **do**
        for $(X, Y) \sim U_t$ **do**
            $\theta^* \leftarrow \theta - \nabla_\theta \text{balancedLoss}(f(\theta, X), Y)$
            $(\bar{X}, \bar{Y}) \sim B_t$
            $\alpha \leftarrow \alpha - \nabla_\alpha \text{softmax\_CE}(f(\theta^*, \bar{X}), \bar{Y})$
            $\theta \leftarrow \theta - \nabla_\theta \text{balancedLoss}(f(\theta, X), Y)$
    $\mathcal{M} \leftarrow \text{updateMemory}(\mathcal{D}_t)$

Instead of using the same fixed weighting coefficient α during the complete training procedure, we propose to jointly learn α during the training of the deep neural network. To achieve this, we propose a meta-learning algorithm that estimates at each optimization step the optimal value of α using a balanced validation set B_t. At the beginning of each incremental step, the unbalanced training set $\mathcal{D}_t \cup \mathcal{M}$ composed of the samples from the new classes and the samples of old classes stored in the memory is split into a training set

U_t and a validation set B_t. Unlike U_t which is a large unbalanced dataset, B_t is a smaller set containing the same number of samples for every class.

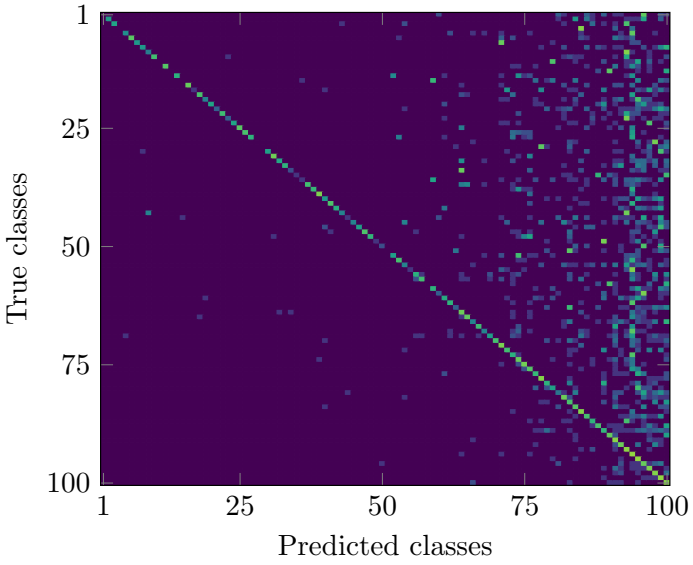
At each optimization step, a temporary model θ^* is created by training the current model θ on the incoming batch of data (X, Y) from U_t using the balanced loss which is the sum of the Balanced Softmax Cross-Entropy loss and secondary losses (such as the distillation loss). By using a batch $(\bar{X}, \bar{Y})$ from the balanced validation set B_t, the value of α is then updated using the gradient of the standard Softmax Cross-Entropy loss of $(f(\theta^*, \bar{X}), \bar{Y})$ with respect to α. Finally, we update the current model θ on the batch (X, Y) previously sampled from U_t using the balanced loss with the newly learned value of α.

Unlike the Balanced Softmax Cross-Entropy which does not change the computational cost of the training procedure compared to the Standard Softmax Cross-Entropy, the Meta Balanced Softmax Cross-Entropy has an impact on the training procedure. The method requires the computation of gradients through the optimization process. One of the main drawbacks is a large increase in the memory requirement which may make it more difficult to combine this method with some existing methods for incremental learning.

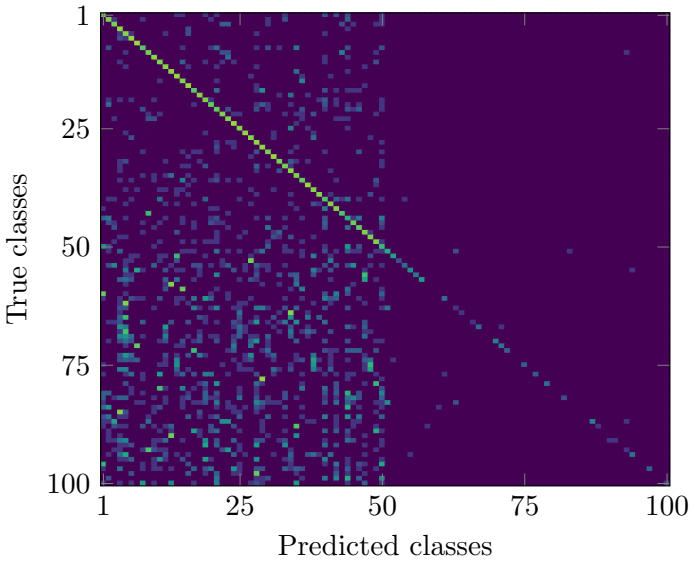
3.3.4 Relaxed Balanced Softmax Cross-Entropy

In the exemplar-free setting, the λ_i coefficient for every old class will be equal to 0 in the Equation (3.3) as it is not possible to save any exemplar from past classes into the memory. Therefore, in this particular setting, the Balanced Softmax Cross-Entropy is equivalent to the standard Softmax Cross-Entropy applied on the new classes only, supposing that the number of images per class in the dataset is the same for all the new classes. This constraint heavily limits the plasticity of the model as shown in Figure 3.3(b). In this experiment, LUCIR [Hou et al., 2019] is trained on ImageNet-Subset with a base step containing half of the classes followed by 5 incremental steps using the Balanced Softmax Cross-Entropy. At the end of the training procedure, while achieving 79.44% accuracy on the 50 base classes, the model only reaches 4.56% on the latest 50 classes. Compared to the 81.16% of accuracy on the base classes at the end of the base step, this means that during the five incremental steps following the base step, the model mostly protected its previous knowledge at the cost of not being able to adapt to the new classes. By comparison, when LUCIR is trained using the Standard Softmax Cross-Entropy, Figure 3.3(a) shows that the opposite happens: the accuracy on the 50 base classes drops from 82.12% at the end of the base step to 29.76% at the end the final step while achieving a similar overall final accuracy. The model trained with the Standard Softmax Cross-Entropy is only able to correctly classify the most recent classes as the method was not designed for class incremental learning without memory. A more suitable solution would therefore be in between those two extreme approaches that are the Balanced Softmax Cross-Entropy and the Standard Softmax Cross-Entropy in the zero memory setting.

To that end, we proposed an extended version of the Balanced Softmax Cross-Entropy



(a) LUCIR
Final overall top-1 accuracy: 38.36%



(b) LUCIR w/ Balanced Softmax Cross-Entropy
Final overall top-1 accuracy: 42.00%

Figure 3.3: Confusion matrix (transformed by log(1 + x) for better visibility) of LUCIR [Hou et al., 2019] trained on ImageNet-Subset with a base step containing half of the classes followed by 5 incremental steps with 0 sample per class stored in the replay memory using (a) the Standard Softmax Cross-Entropy and (b) the Balanced Softmax Cross-Entropy. Figure best viewed in color.

specifically designed for class incremental learning without memory named Relaxed Balanced Softmax Cross-Entropy (Relaxed BalancedS-CE):

$$q_k(x) = \frac{\lambda_k e^{z_k(x)}}{\sum_{j=1}^{N_t} \lambda_j e^{z_j(x)}} \text{ with } \lambda_i = \begin{cases} \epsilon, & \text{if } i \in P \\ n_i, & \text{otherwise} \end{cases} \tag{3.6}$$

where ϵ is usually a small positive number.

While the expression of λ_i for the new classes does not change, instead of using λ_i equals to 0 for the old classes, the Relaxed Balanced Softmax Cross-Entropy uses a small positive value ϵ in order to slightly reduce the constraint on the model during the learning of the new classes. This new hyper-parameter ϵ can be adjusted to tune the balance between stability and plasticity. By using ϵ equal to 0.0, the Relaxed Balanced Softmax Cross-Entropy is equivalent to the Balanced Softmax Cross-Entropy as described in Equation (3.3).

The selection of the value for ϵ remains an open question and is discussed in section 3.4.3. Experimentally, the value of ϵ is fixed at a small percentage of the number of images per class in the training dataset. It should also be noted that the algorithm from Meta-Balanced Softmax Cross-Entropy can not be used in this setting as the balanced validation dataset that is necessary cannot be created in the zero memory setting.

3.4 Experiments

3.4.1 Experimental setups

Datasets

Experiments are conducted on three competitive datasets for large-scale incremental learning: CIFAR100, ImageNet-Subset, and ImageNet. We use the experimental settings defined in [Hou et al., 2019] by initially training the models on the first half of the classes of the dataset (referred to as the base classes) during the base step before learning the remaining classes during the next 5, 10, or 25 incremental steps. This setting known as Learning From Half (LFH) is denoted "Base-i Inc-j" where i is the number of classes in the base step and j is the number of classes in each of the following incremental step. For class-incremental learning with memory, a small growing memory for rehearsal that contains 20 samples per class for the old classes is used. Following [Hou et al., 2019; Rebuffi et al., 2017], the class order is defined by NumPy using the random seed 1993.

- **CIFAR100** [Krizhevsky et al., 2009] is composed of 60,000 32×32 RGB images equally divided among 100 classes with 500 images for training and 100 for testing for each class. There are 50 base classes and the remaining 50 classes are learned in groups of 2, 5, or 10 classes depending on the number of incremental steps.
- **ImageNet** (ILSVRC 2012) [Russakovsky et al., 2015] is composed of about 1.3 million high-resolution RGB images divided among 1,000 classes with around 1,300 images

for training and 50 for testing for each class. There are 500 base classes and the remaining 500 classes are learned in groups of 50 or 100 classes depending on the number of incremental steps.

- **ImageNet-Subset** is a subset of ImageNet only containing the first 100 classes. There are 50 base classes and the 50 remaining classes are learned in groups of 2, 5, or 10 classes depending on the number of incremental steps.

Additionally, we also conduct an experiment on ImageNet without a base step containing half of the dataset. This more challenging setting initially used in [Rebuffi et al., 2017; Wu et al., 2019] consists in learning the 1,000 classes of the ImageNet dataset in 10 training steps, each containing 100 new classes. For this setting, a fixed memory is used with a budget of 20,000 samples in total. This setting known as Learning From Scratch (LFS) is denoted "Base-0 Inc-j" where j is the number of classes in each incremental step.

Baselines

The IL-Baseline which uses the Standard Softmax Cross-Entropy loss is used to highlight the impact of the Balanced Softmax Cross-Entropy loss function for class incremental learning. This baseline is considered as the lower-bound method. Furthermore, the proposed models are compared with several recent approaches for class incremental learning with a replay memory: iCaRL [Rebuffi et al., 2017], BiC [Wu et al., 2019], LUCIR [Hou et al., 2019], Mnemonics [Liu et al., 2020b], PODNet [Douillard et al., 2020], TPCIL [Tao et al., 2020] and DDE [Hu et al., 2021]. For the class incremental learning without memory settings, the Nearest Class Mean (NCM) classifier and the Streaming Linear Discriminant Analysis (SLDA) [Pang et al., 2005; Hayes et al., 2020] are used in addition to two recent approaches: DDE [Hu et al., 2021] and SPB [Wu et al., 2021a] and its derivatives (SPB-I and SPB-M). Contrary to the other considered approaches, both the NCM classifier and SLDA rely on a feature extractor which is only trained during the base step and not further finetuned during the following incremental steps. It results in quick to train methods that achieve competitive results, especially when having a large base step.

To measure the performance of the different models and compare them, the average incremental accuracy is used as in Rebuffi et al. [2017]. It is defined as the average of the Top-1 accuracy of the model on the test data of all the classes seen so far at the end of each training step, including the initial base step.

Implementation details

All compared methods use the 32-layer ResNet [He et al., 2016] for CIFAR100 and the 18-layers ResNet for ImageNet and ImageNet-Subset. In our experiments, the input images are normalized, randomly horizontally flipped, and cropped with no further augmentation applied.

For CIFAR100, the IL-Baseline model is trained for 250 epochs using SGD with momentum optimizer with a batch size of 128 and weight decay set to 0.0002. The learning rate starts at 0.1 and is divided by 10 after the epochs 100, 150, and 200. For ImageNet and ImageNet-Subset, the IL-Baseline model is trained for 100 epochs using SGD with momentum optimizer with a batch size of 256 and weight decay set to 0.0001. The learning rate starts at 0.1 and is divided by 10 after the epochs 30, 60, 80, and 90. For all datasets, the temperature T of the distillation loss for the IL-Baseline is equal to 2. When the Balanced Softmax Cross-Entropy loss or the Relaxed Balanced Softmax Cross-Entropy are combined with other approaches for class incremental learning, the same hyper-parameters and training procedure as those reported in their respective original publications are used. For all datasets, the weighting coefficient α of the Balanced Softmax Cross-Entropy is set to 1.0 as it corresponds to Equation (3.3) and the value of ϵ for the Relaxed Balanced Softmax Cross-Entropy is set to 0.2% of the number of training images per class, which corresponds to $\epsilon = 1$ for CIFAR-100 and $\epsilon = 2.6$ for ImageNet-Subset. For Meta-Balanced Softmax Cross-Entropy, 10% of the memory size is used for the balanced validation set B_t, which represents 2 samples per class for the main experiments. In order to decrease the computational requirement of the method, the α weighting coefficient is only updated every 10 optimization steps instead of every optimization step. PyTorch has been used for the experiments and gradients through the optimization process are computed using Higher library [Grefenstette et al., 2019].

3.4.2 Comparison Results

Class-incremental Learning with memory

The average incremental accuracy (Top-1) on CIFAR100, ImageNet-Subset, and ImageNet of our methods and the different baselines are reported in Table 3.1 using 5, 10, and 25 incremental steps settings. For a fair comparison, each method uses a growing memory containing exactly 20 examples per class.

First, we use the IL-Baseline to support the importance of bias mitigation for large-scale incremental learning. This approach does not use any bias mitigation method and thus highlights the difference between the Standard Softmax Cross-Entropy and the Balanced Softmax Cross-Entropy. On every dataset and in every setting, using the Balanced Softmax Cross-Entropy to train the IL-Baseline instead of the Standard Softmax Cross-Entropy loss results in a huge increase of the average incremental accuracy. Moreover, by using the meta-learning algorithm to learn the weighting coefficient α instead of using the fixed value of 1.0 it is possible to further improve the average incremental accuracy of the IL-Baseline. The largest improvement appears on CIFAR100 with 5 and 10 incremental steps. However, on ImageNet-Subset and ImageNet, there are no significant advantage for the Meta Balanced Softmax Cross-Entropy over the Balanced Softmax Cross-Entropy, and in some cases the Meta Balanced Softmax Cross-Entropy performs worse. On the demand-

Table 3.1: Average incremental accuracy (Top-1) on CIFAR100, ImageNet-Subset, and ImageNet with the Learning From Half procedure: a base step containing half of the classes followed by 5, 10, and 25 incremental steps, using a growing memory of 20 samples per class for all methods. Results for iCaRL and LUCIR are reported from [Hou et al., 2019] ; results for Mnemonics and BiC are reported from [Liu et al., 2020b]; results for PODNet, TPCIL, and DDE are reported from their respective paper. Results marked with "*" correspond to our own experiments. Results on CIFAR-100 averaged over 3 random runs. Results on ImageNet and ImageNet-Subset are reported as a single run. Best result is marked in bold and second best is underlined. Results without memory in Table 3.3.

	CIFAR100			ImageNet-Subset			ImageNet	
	B50-Inc10	B50-Inc5	B50-Inc2	B50-Inc10	B50-Inc5	B50-Inc2	B500-Inc100	B500-Inc50
iCaRL [Rebuffi et al., 2017]	57.17	52.57	-	65.04	59.53	-	51.36	46.72
BiC [Wu et al., 2019]	59.36	54.20	50.00	70.07	64.96	57.73	62.65	58.72
LUCIR [Hou et al., 2019]	63.42	60.18	-	70.47	68.09	-	64.34	61.28
LUCIR w/ Mnemonics [Liu et al., 2020b]	63.34	62.28	<u>60.96</u>	72.58	71.37	**69.74**	64.54	63.01
LUCIR w/ DDE [Hu et al., 2021]	65.27	62.36	-	72.34	70.20	-	67.51	65.77
PODNet [Douillard et al., 2020]	64.83	63.19	60.72	75.54	74.33	68.31	66.95	64.13
PODNet w/ DDE [Hu et al., 2021]	<u>65.42</u>	<u>64.12</u>	-	**76.71**	**75.41**	-	66.42	64.71
TPCIL [Tao et al., 2020]	65.34	63.58	-	<u>76.27</u>	74.81	-	64.89	62.88
IL-Baseline*	43.80	37.00	30.57	51.52	42.22	30.96	43.23	36.70
IL-Baseline w/ BalancedS-CE (ours)	62.22	58.32	52.10	72.57	68.25	61.49	66.45	62.14
IL-Baseline w/ Meta BalancedS-CE (ours)	64.11	60.08	49.63	72.88	69.26	60.70	66.15	61.59
LUCIR [Hou et al., 2019]*	63.37	60.88	57.07	70.25	67.84	63.23	66.69	64.06
LUCIR w/ BalancedS-CE (ours)	64.83	62.36	58.38	71.18	70.66	65.10	<u>67.81</u>	<u>66.47</u>
PODNet [Douillard et al., 2020]*	64.46	62.69	60.63	74.97	71.57	62.67	65.20	62.87
PODNet w/ BalancedS-CE (ours)	**67.77**	**66.57**	**64.64**	76.08	<u>74.93</u>	<u>69.46</u>	**69.67**	**68.65**

ing 25 incremental steps settings, even though the Meta Balanced Softmax Cross-Entropy achieves better performance than the Balanced Softmax Cross-Entropy during the first few incremental steps, its performance deteriorates in the long term and achieves a lower average incremental accuracy at the end. Overall, the IL-Baseline trained with the Balanced Softmax Cross-Entropy achieves superior average incremental accuracy than BiC [Wu et al., 2019] and similar performances to LUCIR [Hou et al., 2019].

To demonstrate the flexibility of the proposed method, the Balanced Softmax Cross-Entropy loss is combined with two state-of-the-art approaches: LUCIR [Hou et al., 2019] and PODNet [Douillard et al., 2020]. For both methods, the classification loss previously used, the Standard Softmax Cross-Entropy loss in the case of LUCIR and the NCA loss [Goldberger et al., 2005; Movshovitz-Attias et al., 2017] in the case of PODNet, is replaced by the Balanced Softmax Cross-entropy and the balanced finetuning step is removed without any other modifications to the method or its hyper-parameters. On every dataset and in every setting, using the Balanced Softmax Cross-Entropy significantly improves the average incremental accuracy of both LUCIR and PODNet. It also results in a decrease of the computational complexity by removing the necessity of performing an additional finetuning step

Table 3.2: Average incremental accuracy (Top-1 and Top-5) on ImageNet with the Learning From Scratch procedure: 10 training steps of 100 classes each using a fixed memory of 20,000 samples. Results for iCaRL, EEIL, and BiC are reported from [Wu et al., 2019] ; results for SS-IL are reported from its respective paper. Results marked with "*" correspond to our own experiments. Results reported as a single run. Best result is marked in bold and second best is underlined.

	ImageNet B0 Inc100	
	Top-1 Avg. Inc. Acc.	Top-5 Avg. Inc. Acc.
iCaRL [Rebuffi et al., 2017]	-	63.7
EEIL [Castro et al., 2018]	-	72.2
BiC [Wu et al., 2019]	-	84.0
SS-IL [Ahn et al., 2021]	**65.2**	**86.7**
IL-Baseline*	46.65	68.74
IL-Baseline w/ BalancedS-CE (ours)	<u>64.15</u>	85.65
IL-Baseline w/ Meta BalancedS-CE (ours)	63.57	85.43
LUCIR [Hou et al., 2019]*	60.40	82.31
LUCIR w/ BalancedS-CE (ours)	60.75	84.31
PODNet [Douillard et al., 2020]*	62.58	83.50
PODNet w/ BalancedS-CE (ours)	64.08	<u>85.71</u>

on a small balanced dataset after each incremental step, as previously done by both methods in some settings. In addition to reducing the training time, removing the finetuning step also means that it is now possible to perform inference at any time during the training procedure. The Balanced Softmax Cross-Entropy loss increases the average incremental accuracy from 0.93 up to 2.81 percentage points (*p.p.*) for LUCIR and from 1.11 up to 6.79 percentage points for PODNet. On ImageNet with 5 incremental steps, PODNet combined with the Balanced Softmax Cross-Entropy outperforms PODNet by 4.47*p.p.* in terms of average incremental accuracy and reaches final overall accuracy of 64.4% which is about 6*p.p.* below the theoretical Top-1 accuracy of a model trained on the whole dataset at once. It is important to note that our proposed method is complementary to other approaches such as Mnemonics [Liu et al., 2020b], DDE [Hu et al., 2021] or GeoDL [Simon et al., 2021], and they could be combined in order to further improve the performance.

Additionally, Table 3.2 contains the average incremental accuracy (Top-1 and Top-5) of our proposed methods and the baselines on ImageNet using 10 training steps each containing 100 new classes as used by Rebuffi et al. [2017] and Wu et al. [2019]. By replacing the standard Softmax Cross-Entropy with the Balanced Softmax Cross-Entropy, the Top-1 average incremental accuracy of the IL-Baseline increased from 46.65% to 64.15% and the Top-5 average incremental accuracy increased from 68.74% to 85.65% without increasing the training time. Comparing BiC [Wu et al., 2019] with the IL-Baseline combined with Balanced Softmax Cross-Entropy highlights the impact of the proposed Balanced Softmax

Cross-entropy for incremental learning, as the only difference between the two models is the bias mitigation method used. Our proposed method achieves a Top-5 average incremental accuracy of 85.65% which represents a notable improvement compared to BiC while not requiring an additional training step to learn the bias correction parameters at the end of each incremental step. The proposed method also achieves competitive results with the recently proposed SS-IL [Ahn et al., 2021], without using neither the ratio-preserving mini-batch nor the improved distillation loss. The meta-learning approach does not improve the performance of the model in this particular setting, though. In this setting also, the proposed Balanced Softmax Cross-Entropy can be combined with LUCIR and PODNet to improve their performance while decreasing their training time and allowing inference at any time by removing the balanced finetuning step. Even though the difference of TOP-1 average incremental accuracy between standard LUCIR and LUCIR combined with our proposed method seems to be limited in this particular setting, it is important to note that LUCIR with the Balanced Softmax achieves a significantly higher final TOP-1 overall accuracy: 51.41% compared to the 48.30% of the standard LUCIR. Moreover, by removing the balanced finetuning step from the standard LUCIR to match the flexibility of our method, its TOP-1 average incremental accuracy drops to 56.41% and its final TOP-1 overall accuracy also drops to 40.44%.

Class-incremental Learning without memory

Next, we compare our method and the different baselines for class-incremental learning without memory using both 5 and 10 incremental steps. The average incremental accuracy (Top-1) on CIFAR100, ImageNet-Subset, and ImageNet are reported in Table 3.3.

Even though in some cases, using the Balanced Softmax Cross-Entropy loss may improve the average incremental accuracy of methods such as LUCIR and PODNet, it is not a viable option for the no memory setting as it tends to over-favor the old classes to the detriment of the new ones. However, by combining LUCIR with the proposed Relaxed Balanced Softmax Cross-Entropy, it is possible to achieve competitive performances compared to DDE[Hu et al., 2021] without increasing the computational complexity of the training procedure compared to the standard LUCIR. Likewise, by combining PODNet with the proposed Relaxed Balanced Softmax Cross-Entropy, it is possible to outperform SPB [Wu et al., 2021a] and achieve similar performance to SPB-M [Wu et al., 2021a] without using multiple perspectives or self-supervised learning. It should be noted that PODNet combined with the Relaxed Balanced Softmax Cross-Entropy without memory achieves comparable average incremental accuracy against standard PODNet with 20 samples per class in memory. Moreover, on ImageNet, the most challenging dataset, LUCIR and PODNet combined with the proposed Relaxed Balanced Softmax Cross-Entropy can outperform the standard LUCIR and PODNet with 20 samples per class in memory by a significant margin.

As detailed in Section 3.4.3, by changing the ϵ parameter of the Relaxed Balanced

Table 3.3: Average incremental accuracy (Top-1) on CIFAR100, ImageNet-Subset, and ImageNet with the Learning From Half procedure: a base step containing half of the classes followed by 5 and 10 incremental steps without using memory for storing samples from past classes. Results for SPB, SPB-I, and SPB-M are reported from [Wu et al., 2021a] ; results for DDE are reported from [Hu et al., 2021]. Results marked with "*" correspond to our own experiments. Results on CIFAR-100 and ImageNet-Subset averaged over 3 random runs, results on ImageNet reported as a single run. Best result is marked in bold and second best is underlined.

	CIFAR100		ImageNet-Subset		ImageNet	
	B50-Inc10	B50-Inc5	B50-Inc10	B50-Inc5	B500-Inc100	B50-Inc50
SPB [Wu et al., 2021a]	60.9	60.4	68.7	67.2	57.8	-
SPB-I [Wu et al., 2021a]	62.6	62.7	70.1	69.8	58.0	-
SPB-M [Wu et al., 2021a]	<u>65.5</u>	**65.2**	<u>71.7</u>	<u>70.6</u>	59.7	-
DDE [Hu et al., 2021]	59.11	55.31	69.22	65.51	-	-
NCM classifier*	59.80	59.52	65.91	65.71	59.92	59.59
SLDA [Hayes and Kanan, 2020]*	60.20	60.13	67.07	66.84	61.69	61.47
LUCIR [Hou et al., 2019]*	43.71	32.08	60.70	47.97	51.19	39.72
LUCIR w/ BalancedS-CE (ours)	54.68	50.87	58.63	57.71	63.29	62.90
LUCIR w/ Relaxed BalancedS-CE (ours)	59.43	54.93	66.27	65.12	<u>67.07</u>	<u>65.07</u>
PODNet [Douillard et al., 2020]*	41.85	35.87	66.61	55.14	47.65	37.22
PODNet w/ BalancedS-CE (ours)	63.31	62.39	63.94	61.98	55.00	54.62
PODNet w/ Relaxed BalancedS-CE (ours)	**65.66**	<u>63.82</u>	**72.56**	**70.99**	**68.50**	**67.38**

Softmax Cross-Entropy, it is possible to slightly increase the average incremental accuracy. For example, PODNet with the Relaxed Balanced Softmax Cross-Entropy can reach up to 74.4% on ImageNet-Subset with 5 incremental steps.

3.4.3 Ablation study

Impact of the memory size

The average incremental accuracy of the IL-Baseline on CIFAR100 with 5 incremental steps is reported in Table 3.4 for various numbers of samples per class stored in the memory depending on the loss function used for the training. The results for the Meta Balanced Softmax Cross-Entropy are only reported for memory size larger than one as this method requires a distinct validation set and training set. The difference between the Standard Softmax Cross-Entropy and the Balanced Softmax Cross-Entropy increases as the size of the memory decreases. With only one sample per class in the memory, the Balanced Softmax Cross-Entropy reaches almost the same average incremental accuracy as the Standard Softmax Cross-Entropy with 50 samples per class. The Meta Balanced Softmax Cross-Entropy further improves the performance and appears to be especially efficient in scenarios with highly restricted memory.

Table 3.4: Average incremental accuracy (Top-1) on the test set of CIFAR100 with the Learning From Half procedure B50-Inc10: a base step containing half of the classes followed by 5 incremental steps of the Incremental Learning Baseline depending on the number of samples stored in memory for each class and the training loss. Results averaged over 3 random runs.

	Memory size				
	1	5	10	20	50
IL-Baseline	24.83	30.59	37.27	43.80	52.99
IL-Baseline w/ BalancedS-CE	**51.55**	56.93	60.11	62.22	64.44
IL-Baseline w/ Meta BalancedS-CE	-	**62.13**	**63.02**	**64.11**	**65.60**

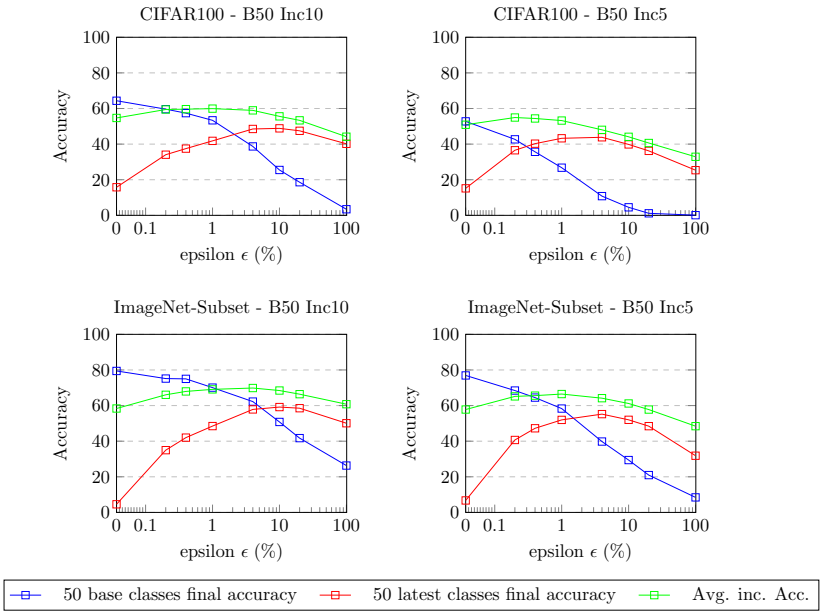
Table 3.5: Accuracy on the test set of CIFAR100 with the Learning From Half procedure: a base step containing half of the classes followed by (a) 5 and (b) 10 incremental steps of the Incremental Learning Baseline (IL-Baseline) depending on the value used for the weighing coefficient α of the Balanced Softmax Cross-Entropy; using a growing memory of 20 samples per class. Results averaged over 3 random runs.

(a) CIFAR100 - B50 Inc10

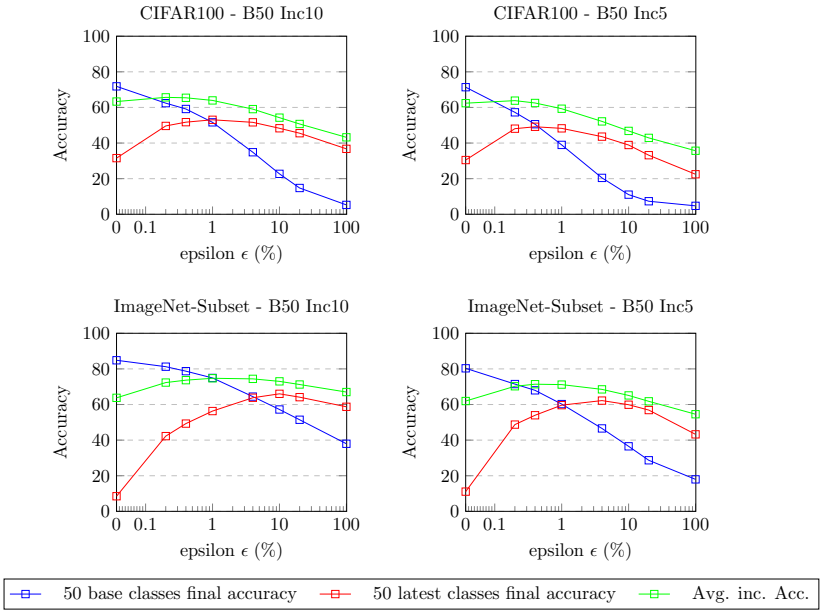
	final base accuracy	final overall accuracy	average inc. accuracy
BalancedS-CE $\alpha = 0.1$	**68.38**	51.54	62.68
BalancedS-CE $\alpha = 0.25$	63.99	54.88	64.09
BalancedS-CE $\alpha = 0.5$	58.78	**55.44**	63.80
BalancedS-CE $\alpha = 1.0$	52.08	54.55	62.22
Meta BalancedS-CE	61.20	55.21	**64.11**

(b) CIFAR100 - B50 Inc5

	final base accuracy	final overall accuracy	average inc. accuracy
BalancedS-CE $\alpha = 0.1$	**63.43**	44.71	57.68
BalancedS-CE $\alpha = 0.25$	60.28	47.86	59.40
BalancedS-CE $\alpha = 0.5$	56.48	49.62	59.52
BalancedS-CE $\alpha = 1.0$	51.01	49.57	58.32
Meta BalancedS-CE	60.39	**49.65**	**60.08**



(a) LUCIR /w Relaxed Balanced Softmax Cross-Entropy



(b) PODNet /w Relaxed Balanced Softmax Cross-Entropy

Figure 3.4: Performance of (a) LUCIR and (b) PODNet combined with the Relaxed Balanced Softmax Cross-Entropy for different values of ϵ on CIFAR100 and ImageNet-Subset without memory, following the Learning From Half procedure. Parameter ϵ is expressed in percentage of the number of samples per class in the training dataset. X-axis represented using linear scale on [0,0.1] and logarithmic scale on [0.1, 100] for better visibility. Results on CIFAR-100 averaged over 3 random runs and results on ImageNet-Subset reported as a single run. Figure best viewed in color.

Impact of the weighting coefficient alpha

The average incremental accuracy of the IL-Baseline on CIFAR100 is reported in Table 3.5 with (a) 5 incremental steps and (b) 10 incremental steps depending on the value of the weighting coefficient α used.

As expected, we can see in both settings that decreasing the value of the weighting coefficient α will reduce the plasticity of the model and thus increase the final accuracy of the 50 base classes. The smaller the weighting coefficient α, the higher the final accuracy of the base classes. Depending on the constraints of the problem, it is possible to use the weighting coefficient α to balance the trade-off between rigidity and plasticity while limiting the impact on the overall final performance of the model: for example on CIFAR-100 with 5 incremental steps by using $\alpha = 0.1$, the accuracy of the 50 base classes increases by $16.30 p.p.$ while the final overall accuracy of the model only decreases by $3.01 p.p.$ compared to the default value of 1.0 for the Balanced Softmax Cross-Entropy. Experimentally, it appears that using the default value of $\alpha = 1.0$ achieves the most balanced performances between the base classes and the latest classes in both the 5 and 10 incremental steps settings. However, it does not achieve the highest average incremental accuracy, and using a smaller value for α may improve it. To this end, using the proposed Meta Balanced Softmax Cross-Entropy is an effective solution as it reaches the highest average incremental accuracy in both settings. This demonstrates the advantage of this method for determining a correct value for the weighting coefficient α without performing several trials. However, the Meta Balanced Softmax Cross-Entropy seems to favor old classes. This may explain why this method does not improve the accuracy compared to the Balanced Softmax cross-entropy in cases where the latter already favors old classes as it is the case when combined with LUCIR for example.

Impact of the hyper-parameter epsilon

Figure 3.4 shows the performance of LUCIR and PODNet in the class incremental learning without memory setting when combined with the Relaxed Balanced Softmax Cross-Entropy depending on the value of ϵ. For ϵ equals to 0% of the number of samples per class in the training dataset, the Relaxed Balanced Softmax Cross-Entropy is equivalent to the Balanced Softmax Cross Entropy and for ϵ equals to 100% of the number of samples per class in the training dataset it is almost equivalent to the Standard Softmax Cross Entropy.

As expected, increasing the value of ϵ for the Relaxed Balanced Softmax Cross-Entropy increases the plasticity of the model, the final accuracy of the 50 base classes decreases while the final accuracy of the 50 latest classes increases. This continues until a point where the final accuracy of the latest classes also decreases as the model is only able to retain the last few classes from the last incremental steps. By replacing the Balanced Softmax Cross-Entropy with the Relaxed Balanced Softmax Cross-Entropy using the default value of ϵ, the final accuracy on the 50 latest classes can be drastically increased while limiting the

impact on the 50 base classes: for example on ImageNet-Subset with 5 incremental steps using LUCIR, the final accuracy on the latest classes is increased by $30.40p.p.$ while the final accuracy on the base classes is only decreased by $4.28p.p.$.

As epsilon is comparable to a number of images, the default value was chosen to be equal to 1 on CIFAR-100, the smallest strictly positive integer. This corresponds to 0.2% of the number of images per class in the training dataset. It should be noted that the value of zero is equivalent to the "non-relaxed" Balanced Softmax Cross-Entropy. Experiments in Table 3.3 empirically show that this default value achieves competitive accuracy and can be generalized to all considered datasets regardless of the number of incremental steps or the approach it is combined with. Although manual tuning is not required to achieve high performance, doing so can slightly increase the average incremental accuracy. For example, by using ϵ equals to 1.0% of the number of training images per class for ImageNet-Subset, the average incremental accuracy can be increased from $0.94p.p.$ to $3.05p.p.$ depending on the method and the number of incremental steps. However, even though Figure 3.4 shows that there is a large range of ϵ values achieving similar competitive performance, determining the optimal value of ϵ beforehand remains an open challenge and will be further investigated in future works. Experimental results show that the value of ϵ achieving either the highest accuracy or the best balance between rigidity and plasticity after the first incremental step is not the value that will achieve the best average incremental accuracy, the best final accuracy, or the best balance at the end of the complete training. This suggests that modifying the value of ϵ during the training procedure instead of using a fixed value may further improve the performance of the proposed approach.

Experimentally, it appears that by using a specific value of ϵ, the Relaxed Balanced Softmax Cross-Entropy can perfectly balance the trade-off between rigidity and plasticity in all settings. This is shown by the Relaxed Balanced Softmax Cross-Entropy achieving the same accuracy for both the 50 base classes and the 50 latest classes learned during the following 5 or 10 incremental steps. Moreover, the highest average incremental accuracy occurs at or in the neighborhood of the value of ϵ achieving the best balance between rigidity and plasticity at the end of the training procedure.

Mitigation of imbalance

In Table 3.6, different bias correction methods are compared with the Balanced Softmax Cross-Entropy and the Meta Balanced Softmax Cross-Entropy on CIFAR100 with 5 incremental steps. Both the IL-Baseline with memory oversampling and the IL-Baseline with class oversampling correspond to the IL-Baseline combined with a form of oversampling on the replay memory. Memory oversampling ensures that each training mini-batch theoretically contains the same number of samples from new and old classes while class oversampling ensures that each class has the same probability of appearing in each training mini-batch. The IL-Baseline with loss rescaling corresponds to the IL-Baseline where the standard Soft-

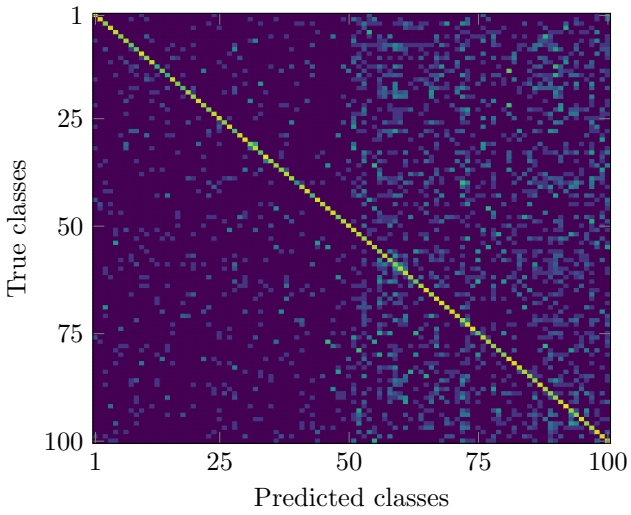
Table 3.6: Average incremental accuracy (Top-1) on the test set of CIFAR100 with the Learning From Half procedure B50-Inc10: a base step containing half of the classes followed by 5 incremental steps of the Incremental Learning Baseline depending on the bias correction method used; using a growing memory of 20 samples per class. Results averaged over 3 random runs.

	average incremental accuracy
IL-Baseline	43.80
IL-Baseline w/ memory oversampling	49.75
IL-Baseline w/ class oversampling	55.95
IL-Baseline w/ loss rescaling	57.01
IL-Baseline w/ balanced finetuning	59.46
IL-Baseline w/ Separated Softmax [Ahn et al., 2021] + oversampling	61.21
IL-Baseline w/ BalancedS-CE (ours)	62.22
IL-Baseline w/ Meta BalancedS-CE (ours)	**64.11**

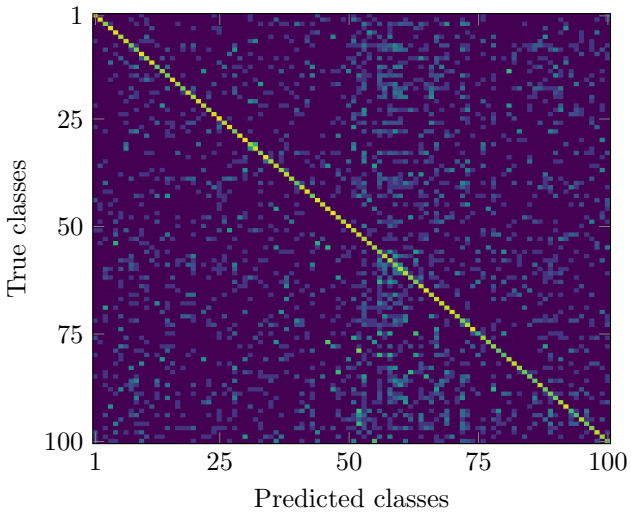
max Cross-Entropy loss for each sample is scaled inversely proportionally to the number of samples for the corresponding label in the training dataset. The IL-Baseline with balanced finetuning corresponds to the IL-Baseline being finetuned after each incremental step on a small balanced training set for few epochs similar to the procedure proposed by Castro et al. [2018] and used by PODNet and LUCIR. IL-Baseline with Separated Softmax relies on oversampling of the replay memory combined with the Separated Softmax layer [Ahn et al., 2021].

In the experiments, it appears that using oversampling for the old classes is a simple yet effective method for mitigating the imbalance occurring in large scale incremental learning scenarios. However, this approach increases the total number of mini-batches resulting in a larger computational complexity of the training procedure and may also be prone to overfitting on the old classes. By rescaling the Softmax Cross-Entropy loss, it is possible to further increase the accuracy of the IL-Baseline without relying on oversampling. Finally, finetuning the model afterward on a small balanced dataset, as it is done by several state-of-the-art methods, further improves the final overall accuracy of the model at the cost of an additional training step after each incremental step. Our two proposed methods achieve higher average incremental accuracy than the other bias correction methods without requiring a two steps training procedure or oversampling.

Figure 3.5 shows the confusion matrices for IL-Baseline trained on CIFAR100 with 5 incremental steps using 20 sample per class stored in the replay memory when trained with (a) the Balanced Softmax Cross-Entropy and (b) the Meta Balanced Softmax Cross-Entropy. Visual comparison with Figure 3.2 depicting the IL-Baseline trained using the



(a) IL-Baseline w/ Balanced Softmax Cross-Entropy
Final overall top-1 accuracy: 54.76%



(b) IL-Baseline w/ Meta-Balanced Softmax Cross-Entropy
Final overall top-1 accuracy: 55.35%

Figure 3.5: Confusion matrix (transformed by log(1 + x) for better visibility) of the IL-Baseline trained using (a) the Balanced Softmax Cross-Entropy and (b) the Meta-Balanced Softmax Cross-Entropy on CIFAR100 with the Learning From Half procedure B50-Inc10: a base step containing half of the classes followed by 5 incremental steps, using 20 sample per class stored in the replay memory. Figure best viewed in color.

Standard Softmax Cross-Entropy in the same setting confirms that our proposed method is efficient in mitigating the imbalance between the new and old classes in the class incremental learning scenario.

3.5 Conclusion

In our work, following recent advances in Long-Tailed Visual Recognition, we proposed to use the Balanced Softmax Cross-Entropy loss function as a straightforward replacement for the Standard Softmax Cross-Entropy loss in order to correct the bias towards new classes in class incremental learning with a replay memory. The expression of this loss offers control on the plasticity-rigidity trade-off of the model but also on the importance of each class individually. We further investigated how it could be adapted for the challenging class incremental learning without memory setting and proposed an extension called the Relaxed Balanced Softmax Cross-Entropy.

Experiments on CIFAR100, ImageNet-Subset, and ImageNet with various settings showed that by seamlessly combining our proposed method with state-of-the-art approaches for incremental learning, it is possible to further increase the accuracy of those methods while also potentially decreasing the computational cost of the training procedure by removing the need for a balanced finetuning step.

In our future work, we will explore how the coefficients of the Balanced Softmax Cross-Entropy can be tuned during the training procedure in order to improve the accuracy of the model and the balance between rigidity and plasticity when combined with memory and when used without memory. The use of the Balanced Softmax Cross-Entropy for general incremental learning on imbalanced datasets is also left for our future work.

Chapter 4

Memory Augmented using Diffusion Model for Class-Incremental Learning

4.1 Introduction

In a class-incremental learning (CIL) scenario, deep neural network models are not trained offline on a fixed pre-collected dataset but sequentially on new incoming data. The model has to be incrementally updated using only a limited number of new classes at a time, with the objective of learning a unified classifier among all seen classes. This paradigm, similar to the human learning process, is challenging as models are suffering from catastrophic forgetting [McCloskey and Cohen, 1989; Ratcliff, 1990; Parisi et al., 2019]: learning new data or classes will result in severe degradation of the performance on the past ones.

Multiple researches have highlighted the importance of rehearsal for continual and incremental learning: by storing samples from each previously encountered class in a memory, models are able to replay past data while learning new ones in order to mitigate catastrophic forgetting. However, the size of this memory is often limited to a few samples per class due to either storage limitations or privacy concerns. Several authors [Lee et al., 2019; Zhang et al., 2020; Lechat et al., 2021; Liu et al., 2024] have proposed to leverage additional external training data as a means to work around this constraint. Those additional training data are sampled from large curated datasets of real images, such as ImageNet, and belong to classes different from the ones encountered by the incremental learner. They are mostly used in an unsupervised manner, for distilling knowledge from the past model to the new one.

In this work, we propose to use pre-trained state-of-the-art generative diffusion models

as a source of complementary data for class-incremental learning. Compared to approaches that rely on additional external datasets made of real images, our method generates synthetic samples belonging to the same classes as the ones previously encountered by the model. This fundamental difference allows us to make use of those samples not only for the knowledge distillation loss but also for replay in supervised losses such as the classification loss whereas concurrent works mostly focus on the distillation loss. The ablation study highlights the improvement resulting from using the additional data for both losses. To the best of our knowledge, this is the first attempt to use large pre-trained text-to-image generative models to improve general class-incremental learning methods on large scale datasets. To summarize, our contribution is three-fold:

- We propose to use pre-trained text-to-image diffusion models to generate synthetic images of past classes and show that they can be used with state-of-the-art approaches to significantly improve their performances.

- We emphasize the advantages of generating synthetic labeled images from past classes which can therefore also be used in supervised losses compared to concurrent works limited to the distillation loss due to the use of real unlabeled data from different classes.

- We perform an in-depth ablation study and highlight important components and hyper-parameters when using pre-trained diffusion models for class incremental learning.

This work is an extension of our previously published workshop paper [Jodelet et al., 2023], that contains additional experiments and considers a more general context of application for our proposed approach, that does not limit it to distillation-based methods for class incremental learning. First, we combined our proposed method with the dynamic-networks-based methods for class-incremental learning DER [Yan et al., 2021] and MEMO [Zhou et al., 2023c] which do not rely on a distillation loss. Secondly, we provide a more detailed analysis of the impact of the distribution gap between the real and synthetic images and discussed the methods to mitigate it. Finally, we conducted additional experiments, including : a wider range of settings such as Learning-From-Scratch class-incremental learning, the use of larger neural networks for the backbone in order to evaluate the scalability of our method, a comparison of different image augmentation procedures and a study of the impact of the image generation process which includes the selection of the guidance scale, a comparison of different pre-trained diffusion models, and a comparison of different prompting mechanisms.

4.2 Related work

4.2.1 Class-incremental Learning (CIL)

Methods for class-incremental learning [Zhou et al., 2023b] usually rely on knowledge distillation and a small replay memory containing past samples combined with a bias mitigation method. Knowledge distillation is used to preserve previous knowledge, it may be logit distillation [Li and Hoiem, 2017; Rebuffi et al., 2017; Wu et al., 2019], feature distillation [Hou et al., 2019; Douillard et al., 2020] or relational distillation [Tao et al., 2020]. Using a memory for replaying previously learned exemplars is a simple yet effective method to recall knowledge of old classes. However, the limited size of the memory induces a bias toward the new classes that strongly degrades the performance of the model if it is not mitigated. To this end, several methods have been proposed such as Nearest-Mean-of-Exemplars (NME) classifier [Rebuffi et al., 2017], cosine classifier [Hou et al., 2019], bias correction layer [Wu et al., 2019], weight alignment [Zhao et al., 2020], finetuning on a balanced subset [Castro et al., 2018] or specific losses [Ahn et al., 2021; Jodelet et al., 2021, 2022].

Instead of storing few exemplars inside the replay memory, some authors have proposed to learn to generate artificial samples from past classes. Those methods, referred to as generative replay or pseudo-rehearsal, can overcome the limitations induced by the limited size of the exemplar memory and can be applied in settings where privacy concerns limit the storage of past samples. Most generative replay methods rely on two models: one main model trained for the classification task and one secondary model for data generation, usually generative adversarial networks [Shin et al., 2017; He et al., 2018; Wang et al., 2021]. Sun et al. [2023] proposed to use variational auto-encoders as one-class classifier for their capacity to both estimate the probability that a sample belong to the associated class and generate synthetic samples to train next classifiers. Model inversion [Yin et al., 2020; Smith et al., 2021] has been shown to be an effective alternative to generative model. However, generative replay methods heavily rely on the quality of the generated data, which poses a challenge for large scale datasets and are also impacted by catastrophic forgetting when the generative model is trained incrementally, limiting their application to smaller datasets. Alternatively, several works [Kemker and Kanan, 2017; Liu et al., 2020a; Petit et al., 2023] proposed the simpler approach of generating feature instead of raw input images of past classes, nonetheless this mostly restricts replay to the classifier and it cannot be used to mitigate catastrophic in the feature extractor.

Recently, dynamic-networks-based methods [Yan et al., 2021; Wang et al., 2022a; Zhou et al., 2023c] have gained popularity and achieved new state-of-the-art performance for class-incremental learning. These methods dynamically expand the network to accommodate new classes and freeze previously learned weights to preserve knowledge from old classes. Most of those methods rely on an additional supervised loss, often named auxiliary loss, to encourage more diverse representations and help learn the new classes.

4.2.2 Additional data for CIL

Various approaches have studied the use of additional external data, either as a direct substitute in case of the absence of memory [Lee et al., 2019; Zhang et al., 2020] or as a complementary resource during training [Lechat et al., 2021; Liu et al., 2024]. Global distillation (GD) [Lee et al., 2019] proposed a confidence-based sampling strategy to sample additional external data and used them for a specifically designed distillation loss. Deep Model Consolidation (DMC) [Zhang et al., 2020] consists in training one new model for each incremental step and then creates a unified model by merging this new model with the previous one using additional external data. Lechat et al. [2021] proposed a pseudo-labeling approach to better utilize the additional external unlabeled data for representation learning. Liu et al. [2024] observed that using samples from new classes in the knowledge distillation loss negatively impacts the training of incremental models and proposed to replace them using placebos of old classes sampled from an external dataset using an online learning method. Those different approaches rely solely on real images, most of the time unlabeled and from classes different to the one encountered by the model, by sampling their additional data from large curated datasets: primarily ImageNet [Russakovsky et al., 2015] or TinyImages dataset [Torralba et al., 2008].

4.2.3 Diffusion models

Following recent advances, diffusion models [Sohl-Dickstein et al., 2015; Ho et al., 2020] have reached new state-of-the-art performances for numerous image generation tasks, achieving both better fidelity and better diversity than GAN-based methods. Diffusion models generate data samples through an iterative denoising process. The diffusion process can be conducted either in the image space [Saharia et al., 2022] or in the latent space [Rombach et al., 2022]. In the case of text-to-image generation, the diffusion process is usually conditioned on a prompt consisting of a textual description of the desired output. Recent approaches [Li et al., 2023b; Zhang et al., 2023] have offered finer control on the image generation process by completing the textual prompt with complementary information such as semantic map, bounding boxes, or sketch drawings.

4.2.4 Learning from synthetic data

In one of the pioneer works on learning from synthetic images, Besnier et al. [2020] trained a classifier for 10 classes of ImageNet using samples generated by a class-conditional GAN trained on ImageNet. Jahanian et al. [2022] used a GAN to generate multiple views for contrastive learning methods. Recently, following the recent breakthrough in text-to-image diffusion models, Sarıyıldız et al. [2023] proposed the use of those general-purpose text-conditioned pre-trained generative models for synthesizing datasets as a direct replacement of real image datasets for the training of large scale image-level classification models. While

these models cannot reach the same accuracy as the models trained using real images when tested on real test datasets, they exhibit other advantages such as strong generalization capability. Concurrently, He et al. [2022] also proposed to use large synthetic datasets to improve pre-trained models on zero-shot and few-shot learning, and transfer learning.

Compared to previously mentioned approaches that leverage unlabeled real images sampled from the web as additional data for class-incremental learning, in this work we propose to study the use of widely available pre-trained generative models such as Stable Diffusion [Rombach et al., 2022] as a source of additional data. This allows for generating labeled samples that belong to the same classes as those previously encountered by the incremental model. Moreover, compared to pseudo-rehearsal approaches that generate samples from past classes by incrementally training a generative model on the target dataset solely, we instead propose to leverage frozen diffusion model pre-trained on large varied natural images datasets and take advantage of their general generative capability, effectively circumventing the challenges inherent to pseudo-rehearsal.

4.3 Proposed method

The Class-Incremental Learning training procedure is divided into T incremental steps. Each incremental step consists of a training dataset $\mathcal{D}_t$ containing images belonging to the set of new classes $\mathcal{Y}_t$; $\forall i, j \in \{1, ..., T\}, \mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset$ for $i \neq j$, i.e. there is no class overlapping between steps. During each incremental step, the model θ_t is trained on $\mathcal{D}_t \cup \mathcal{M}$ where $\mathcal{M}$ is the replay memory containing few samples from previously encountered classes. Additionally, the model also has access during every step to an external source of data $\mathcal{S}$ that may or may not change during the training. This external source of additional data $\mathcal{S}$ can either be an online, and potentially infinite stream of data that the model can fetch when needed or a fixed set of data accessible in an offline manner.

After each incremental step, the model θ_t is evaluated on the test set of all the classes learned so far without having access to any step or task identity.

4.3.1 Synthetic images

Following the recent in-depth study of Sarıyıldız et al. [2023], we propose to use a pre-trained text-to-image Diffusion model as the source of additional data $\mathcal{S}$ for incremental learning. We primarily studied the use of the widely available Stable Diffusion [Rombach et al., 2022] model due to it being open-source, allowing a complete control over the generation process.

Figure 4.1: Samples from $\mathcal{S}$, the set of additional synthetic images generated for class-incremental learning on CIFAR100 by Stable Diffusion 1.4 with a guidance scale of 2.0 . Images are generated at a size of 512×512 and are being resized to 32×32 before being fed to the model. Images are displayed before resizing to 32×32 for better appreciation. From left to right, top to bottom: Apple, Aquarium fish, Baby, Bear, Beaver, Bed, Bee, Bicycle, Bottle, Bridge, Camel, Clock, Cloud, Crab, Crocodile, Fox, Hamster, Lawn-mower, Maple tree, Orchids.

Prompting

To generate a sample for $\mathcal{S}$, the generative diffusion model requires a textual prompt. As proposed in [Sarıyıldız et al., 2023], to generate a synthetic image for the class c, we use the prompt, denoted as default prompt, "c, d_c" where "c" is the textual name of the class and "d_c" its description. Although using only "c" as a prompt may be enough in most cases, this may lead to semantic errors with homographs. For example, in the CIFAR100 dataset using only "apple" as a prompt may generate images of products or shops from the homonym brand instead of the fruit. Adding the definition of the class to the prompt helps solve this issue. In order to produce these prompts automatically without any further engineering, WordNet [Miller, 1995] is used to provide the lemmas of the synset as the class name "c" and the definition of the synset as the description "d_c". For datasets such as ImageNet, the association of each class with a synset of WordNet is included in the dataset itself. For other datasets such as CIFAR100, the association with WordNet can be done semi-automatically at a negligible cost. Figure 4.1 shows synthetic samples for CIFAR100 before being resized to 32×32.

In addition to this default prompting scheme, following Fan et al. [2024a], we also explore the use of two more advanced methods to diversify the textual prompts in the ablation study by generating several textual prompts per class instead of using a single one. The first method, denoted CLIP-prompt, generates prompts using the templates engineered via trial and error for zero-shot classification on ImageNet with CLIP [Radford et al., 2021]. It is composed of 7 templates such as "a photo of the small r_c.", "a bad photo of the r_c.", and "a r_c in a video game." where "r_c" is the manually revised name of the class. The second method, denoted Caption-prompt, generate multiple prompts per class using the template "r_c, a_c" where "r_c" is the manually revised name of the class from CLIP-prompt and "a_c"

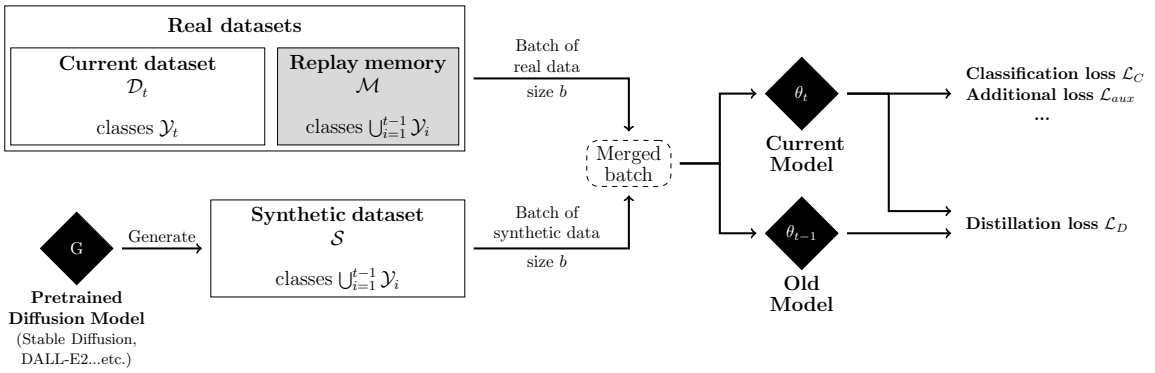


Figure 4.2: Diagram representing our proposed method **M**emory **A**ugmentation using **D**iffusion **M**odel (**MADM**) for leveraging additional external synthetic labeled samples for class-incremental learning. Synthetic data can be used in the distillation loss but also in any supervised losses such as the classification loss or the auxiliary loss of dynamic-networks-based methods. More details in Algorithm 4.1.

are textual captions automatically generated from the real training set of the class using the pre-trained image-to-text model BLIP2 [Li et al., 2023a]. Furthermore, the textual prompts could also be automatically diversified by using large language models, similarly to Yu et al. [2023], Fan et al. [2024b], and Hammoud et al. [2024].

Guidance scale

Several diffusion models, including Stable Diffusion, rely on classifier-free guidance [Ho and Salimans, 2021] which combines the estimate of two diffusion processes: one text-conditioned on the prompt and one unconditioned. The trade-off between the quality and the diversity of the synthetic samples generated by the diffusion models can be controlled to some extent by using a guidance scale used to combine the two diffusion processes.

For Stable Diffusion, following Sarıyıldız et al. [2023], we also used a guidance scale of 2.0, lower than the default value of 7.5, to increase the diversity of our source of additional data at the cost of a slight decrease of the quality of each sample. For DALL-E2 [Ramesh et al., 2022] used through the API from OpenAI, it is not possible to change the guidance scale and we are using the default unknown value.

4.3.2 MADM for Class-incremental learning

Using the previously described generative model, we propose a new method for class-incremental learning: **M**emory **A**ugmentation using **D**iffusion **M**odel (**MADM**), which jointly uses the additional synthetic data $\mathcal{S}$ with the real images from the replay memory $\mathcal{M}$. Compared to the original workshop paper, our proposed method has been renamed from SDDR (Stable Diffusion for Distillation and Replay) to MADM to emphasize the

versatile nature of our approach: it is not limited to Stable Diffusion, can be applied to class-incremental learning methods that don't rely on distillation, and can be used with any losses, supervised or not.

Datasets of real images used by concurrent approaches as an external source of additional data for class-incremental learning do not contain any image belonging to the same classes as the one encountered by the incremental model and are usually considered unlabeled. This leads these methods to mainly use this external source of data in an unsupervised manner for knowledge distillation only. By leveraging the pre-trained text-to-image generative model, our method is able to generate an additional dataset $\mathcal{S}$ containing labeled images belonging to the same classes as the one previously learned by the model without the need for costly manual labeling and curating process. This allows us to use the additional dataset for distillation but also for replay in any supervised loss including the classification loss or the auxiliary loss of dynamic-networks-based methods for example. The ablation study in Section 4.4.3 shows the advantages of our approach.

Our approach is designed as a complementary method that can be seamlessly combined with other standard methods for class-incremental learning. Algorithm 4.1 describes the training procedure of our approach MADM when combined with a class-incremental learning model. Each step of the class-incremental learning procedure is divided into two distinct phases. During the first phase, the model θ_t is trained following the method it is combined with, using a batch of mixed data. As illustrated in Figure 4.2, half of each batch used to train the model is composed of synthetic images sampled from $\mathcal{S}$ while the other half is composed of real data sampled from the memory and the current dataset $\mathcal{D}_t$. Then, in the second phase, before moving to the next incremental step, the synthetic dataset $\mathcal{S}$ is extended by using the Diffusion Model to generate n synthetic samples for each new class encountered in the current dataset $\mathcal{D}_t$, following the method detailed in Section 4.3.1. Therefore, at the end of each incremental step, the additional dataset $\mathcal{S}$ would contain $n \cdot N_t$ samples where $N_t = |\bigcup_{i=1}^{t} \mathcal{Y}_i|$ is the number of classes encountered up to the step t, included.

In our work, we considered this additional dataset of synthetic samples $\mathcal{S}$ as an offline dataset stored on the device itself. Nonetheless, if the storage is a constraint for the incremental learner, the additional dataset $\mathcal{S}$ can be used in an online way as an infinite stream of data, similarly to [Liu et al., 2024], by either using the generative model locally or by relying on a cloud-based model. Compared to concurrent works relying on real datasets as a source of additional data, our method offers more flexibility depending on the constraint of the problem: it can be used with limited storage, limited computational budget, or incapability to communicate with external services.

The quality of the synthetic images, their fidelity to the target classes, and the variety of concepts available in the additional external dataset $\mathcal{S}$ are intrinsically limited by the pre-trained generative text-to-image model used. Using a different and more specialized generative model or finetuning it on the specific target training dataset may further improve

Algorithm 4.1: MADM for Class-Incremental Learning

Input: Data-flow $\{\mathcal{D}_i\}_{i=1}^{T}$ and pre-trained generative model G.
Ouput: Models $\{\theta_i\}_{i=1}^{T}$, replay memory $\mathcal{M}$, and synthetic dataset $\mathcal{S}$.

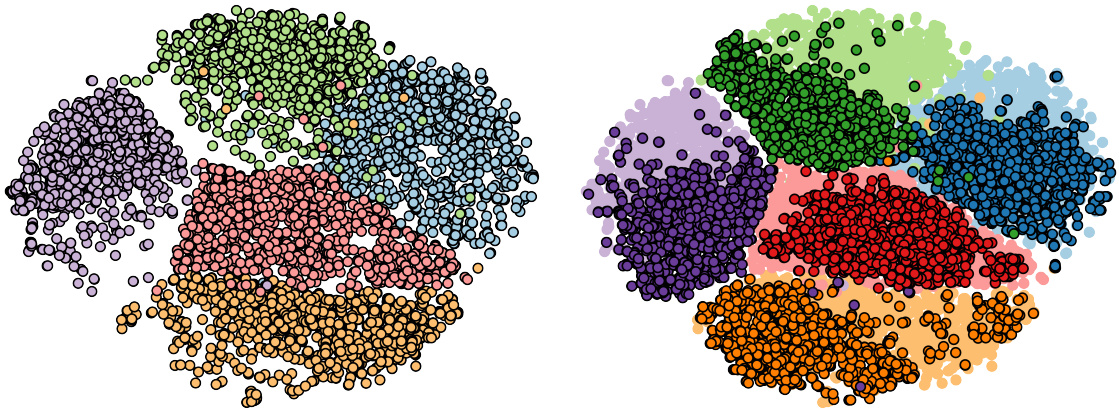
```
for t ∈ {1, 2, ..., T} do
    for epoch ∈ {1, 2, ..., E} do
        while (X, Y) ~ D_t ∪ M do
            if t > 1 then
                // If not base step
                // Sample synthetic data
                (X_S, Y_S) ~ S
                (X, Y) ← (X ∪ X_S, Y ∪ Y_S)
            Update model θ_t using (X, Y)
    M ← UpdateMemory(M, D_t)
    S ← UpdateSynthetic(S, G, Y_t)
```

the performance of the incremental learning model it is combined with.

4.3.3 Distribution gap between real and synthetic images

As highlighted by Sarıyıldız et al. [2023], while the synthetic images generated by Stable Diffusion tend to be highly similar to real ones, there remains a gap that limits the generalization of the model and may negatively impact the performance. In Figure 4.3, the embedding of real and synthetic samples from five classes of the ImageNet dataset are visualised using t-SNE [Van der Maaten and Hinton, 2008]. These embeddings are produced using a ResNet-18 model trained in a non-incremental fashion on the ImageNet dataset solely, and has not been exposed to synthetic data during training. While real and synthetic samples of each classes are correctly clustered together, the distribution of real and synthetic samples are not perfectly overlapping. For example, the real samples of the green class are mostly concentrated in the top-right region of this class's cluster while the synthetic ones are mostly in the bottom-left region, for the red class the distribution of its real samples is spread over a wider region than the one of its synthetic samples which is more concentrated in the middle of the class's cluster. It can also be observed that the importance of the gap between the two distributions may vary for each class.

The effect of this distribution gap can primarily be observed when using the synthetic datasets $\mathcal{S}$ generated by the pre-trained text-to-image diffusion model as a straightforward and complete replacement for the replay memory. While this would allow our proposed method to also be applied to the more challenging exemplar-free class-incremental learning setting, experiments highlighted a significant decrease of accuracy compared to using a few



(a) Real samples **(b) Synthetic samples (w/ real)**

Figure 4.3: Visualisation of the embedding of real and synthetic samples from 5 classes of ImageNet produced by a ResNet-18 model trained in a non-incremental fashion on a large dataset only containing real images (i.e. the ImageNet dataset). Synthetic samples were generated using Stable Diffusion 1.4 with a guidance scale of 2.0 . Light-color points are real samples from the target dataset and deep-color points are synthetic samples generated using our proposed method. To better highlight the distribution gap, (a) real samples only are plotted and (b) synthetic and real samples are plotted together.

real samples as exemplars and it did not achieve competitive results.

More generally, experiments suggest that using synthetic samples in the classification loss induces a shift in the linear classifier due to the distribution gap between the real images and the synthetic ones generated by the diffusion model. This phenomenon is especially important for approaches that apply limited, or no regularization on the classifier. Experiments on CIFAR100 detailed in the ablation study in Section 4.4.3 evaluate the impact of using synthetic data in the classification loss and the auxiliary loss for DER relying on Weight Aligning [Zhao et al., 2020] for the bias mitigation. Simply using the synthetic images in both losses results in a significant decrease of the average incremental accuracy, by up to -5.86 percentage points (*p.p.*). While using the synthetic data only in the auxiliary loss solves the issue and results in up to 1.79*p.p.* improvement of the average incremental accuracy compared to the baseline method, correctly mitigating the shift in the linear classifier afterward results in a higher improvement of the performance: up to 2.30*p.p.*. Two simple and straightforward approaches to mitigate the effect of the distribution gap between real and synthetic samples on the linear classifier is to either use a NME classifier of real exemplars only or finetuning the classifier on a small balanced subset containing only real images from the rehearsal memory. Two methods that are already vastly used to mitigate the bias of the linear classifier toward new classes, inherent to large scale class-incremental learning. Moreover, similarly to Sarıyıldız et al. [2023], the synthetic-to-real gap can further be addressed by using strong augmentation policies and multi-crops: their

experiments on a simple variant of synthetic ImageNet-100 showed a significant 51% relative increase of the accuracy compared to standard limited augmentation.

In addition to mitigating its effect on the linear classifier, the distribution gap between real and synthetic samples itself can also be reduced by improving the generation process. One of the possible improvement is related to the composition of the synthetic images. Advanced prompting mechanisms such as Caption-prompt, which creates prompts by combining textual descriptions of the real training dataset automatically generated by a pre-trained image-to-text model with the revised name of each class, can be used to generate synthetic samples which the composition is more similar to the one of the actual real dataset. As a concrete example, the class n01440764 (tench) from ImageNet has numerous training samples containing fisherman holding the fish in their hands or in a net after catching it while the default prompt mostly results in synthetic images depicting the fish in its natural habitat underwater. Caption-prompt helps to address this issue by generating prompts such as "tench, a man holding a large fish in front of a pond" and "tench, a fish on a net in the grass" resulting in a synthetic dataset more similar to the real one. Detailed experiments in ablation study show that the use of this more advanced prompting mechanism results in an improvement of the performance for our proposed method. Moreover, to further reduce the distribution gap between real and synthetics samples, the generation process could also be improved by using the image embedding of the exemplars from the rehearsal memory to condition the diffusion process: as a replacement for the textual prompt similarly to Zhou et al. [2023d] and general Image Variation [Ramesh et al., 2022], or in addition to the textual prompt similarly to Zhang et al. [2024].

4.4 Experiments

4.4.1 Experimental setups

Datasets

Experiments are conducted on three datasets, CIFAR100 [Krizhevsky et al., 2009], ImageNet-Subset and ImageNet [Russakovsky et al., 2015], defined as:

- **CIFAR100:** The dataset is composed of 60,000 32×32 RGB images from 100 classes with 500 training and 100 testing samples per class.
- **ImageNet (ILSVRC 2012):** The dataset is composed of around 1.28 million high-resolution color images from 1,000 classes with about 1,300 training and 50 testing samples per class.
- **ImageNet-Subset:** The dataset consists in a subset of ImageNet containing the first 100 classes. We also consider an alternative subset of classes, denoted as "ImageNet-Subset Alt.", containing 100 classes randomly sampled from the 1,000 classes of ImageNet.

For all datasets, the class order is defined by NumPy using the random seed 1993 as proposed by Rebuffi et al. [2017]. The classes are then divided into incremental steps following two possible training procedures:

- **Training From Scratch (TFS):** The classes are evenly divided into T incremental steps. In this training procedure, models rely on a fixed replay memory containing a constant number of exemplars evenly divided among the classes learned so far at the end of each incremental step.
- **Training From Half (TFH):** Following the experimental setting firstly proposed by Hou et al. [2019], the first incremental step also denoted base step, containing half of the classes is followed by $T - 1$ incremental step containing the remaining classes evenly divided. In this training procedure, models rely on a growing replay memory containing exactly n exemplars for each already learned class.

Following [Zhou et al., 2023b], we refer to these two training procedures using the denomination "Base-i Inc-j" where i is the number of classes in the base step and j is the number of classes in each of the following incremental step. With i equals 0 referring to the Training from Scratch procedure.

Comparison methods

We combine our proposed method MADM with three methods relying on distillation: the recent state-of-the-art approach FOSTER [Wang et al., 2022a] and two fundamental baselines for class-incremental learning iCaRL [Rebuffi et al., 2017] and LUCIR [Hou et al., 2019]. We also consider two approaches that do not use distillation: the dynamic-networks-based methods DER [Yan et al., 2021] and MEMO [Zhou et al., 2023c]. iCaRL and LUCIR were selected because they are representative of the principal approaches for class-incremental learning: using different distillation losses (logits distillation and feature distillation) and difference bias mitigation methods (Nearest-Mean Exemplar classifier and cosine normalized classifier).

Our proposed method is compared with the recent approach PlaceboCIL [Liu et al., 2024] which leverages external unlabeled datasets of real high-resolution images to improve the distillation loss of class-incremental learning methods.

Implementation details

Training details: Following the standard setting [Rebuffi et al., 2017; Hou et al., 2019], the same network backbone is used for every considered method: a modified 32-layer ResNet [He et al., 2016] for CIFAR100 and a 18-layer ResNet for ImageNet and ImageNet-Subset. Additional experiments are conducted using the standard 18-layer ResNet on CIFAR100 in the ablation study. Every considered method uses a fixed replay memory containing 2,000 exemplars for the Learning From Scratch procedure and a growing memory containing 20

exemplars per class for the Training From Half procedure. When combining our proposed method with an existing baseline for class-incremental learning, the same hyperparameters as the original method are used. The only difference is that the effective batch size during training is doubled by appending synthetic images generated by the generative model to the original batch of real images from the union of the new dataset and the replay memory. Experiments are implemented with Pytorch [Paszke et al., 2019] using officially released implementations of the different baselines methods and PyCIL [Zhou et al., 2023a].

Proposed method details: For iCaRL [Rebuffi et al., 2017] and LUCIR [Hou et al., 2019], synthetic samples are used in both the classification and the distillation losses. For LUCIR, the synthetic data are not used for the margin ranking loss as preliminary experiments suggested that it slightly decreases the performance of the method. For FOSTER [Wang et al., 2022a], the synthetic data are used for the feature enhancement loss and the distillation loss during feature boosting and for the distillation loss during feature compression. We do not use the synthetic data for the logits aligned classification loss as it results in a significant decrease of the accuracy due to a bias of the classifier. For fair comparison with the other baselines, we report the accuracy of FOSTER after feature compression. For comparison of dual branch FOSTER before feature compression, see Appendices. For DER [Yan et al., 2021] and MEMO [Zhou et al., 2023c], synthetic samples are used in both the classification and the auxiliary losses. DER is used without pruning. When combined with our method, MEMO relies on a NME classifier for evaluation due to the distribution gap between real and synthetic samples. For LUCIR and iCaRL, training images are normalized, randomly horizontally flipped, and cropped, and no more augmentation is applied. Additionally, following their respective authors, AutoAugment [Cubuk et al., 2019] is also used for augmentation for FOSTER, and color jitter is also used for augmentation for DER and MEMO on CIFAR100 dataset. For a fair comparison with other dynamic-networks-based methods, experiments for FOSTER without AutoAugment are included in the Ablation Study.

Image generation details: If not stated otherwise, the default generative model we use is Stable Diffusion [Rombach et al., 2022] version 1.4 [1] using the default prompt "c, d_c" [Sarıyıldız et al., 2023], with 50 denoising steps and and the guidance scale g equals to 2.0 . Synthetic images have been generated separately using fixed seeds before the incremental learning experiments. Every method uses the same dataset of synthetic images. For CIFAR100, the synthetic dataset contains 500 32×32 images per class, and for ImageNet and ImageNet-Subset 1300 512×512 images per class. For every dataset, synthetic images are generated with a size of 512×512 and are only resized to 32×32 for CIFAR100 before the incremental training. The same seeds have been used for generating all classes, resulting in closely related classes having some similar images (same background, subject's pose, and position) where only the subject of the image changes. Additional

[1] Available at: https://huggingface.co/CompVis/stable-diffusion-v1-4

experiments have been conducted using the generative model DALL-E2 [Ramesh et al., 2022] through the API of OpenAI. Due to the moderation policy of the OpenAI API, a few definitions "d_c" have been edited to remove words that triggered the moderation filtering system. Moreover, for DALL-E2, the prompt has been changed to "a photo of a c, d_c" as preliminary experiments highlighted that the default prompt could result in over-simplified images for some classes. Regarding the two additional prompting mechanisms, CLIP-prompt uses the revised name of each class and a restricted set of 7 sentences (from the original set of 80 sentences)[2] engineered for zero-shot classification on the complete ImageNet dataset with CLIP [Radford et al., 2021] in order to generate prompts. CLIP-Caption generates prompts for each class by combining the revised name of the class with textual description of each real training sample generated by the pre-trained image-to-text model BLIP2[Li et al., 2023a][3].

Performance Measure

Models are primarily evaluated and compared using the Average Incremental Accuracy defined by Rebuffi et al. [2017] as the average of the Top-1 accuracy of the model on the test data of all the classes seen so far, at the end of each incremental step including the base step. The final overall accuracy of the model after the last incremental step is also reported for the experiments with the Training From Scratch procedure.

4.4.2 Comparison with baselines

The average incremental accuracy (Top-1) of our method and the different baselines are reported for CIFAR100, ImageNet-Subset, and ImageNet with the Learning From Half procedure in Table 4.1 and for the Learning From Scratch procedure in Table 4.2. The final accuracy of the model with the Learning From Half procedure is reported in the Appendices. Due to significant differences in the reproduced accuracy for FOSTER [Wang et al., 2022a] between our work and Liu et al. [2024], we decided to make a more detailed comparison with PlaceboCIL in Table 4.3. The difference in the reproduced accuracy is especially important on ImageNet, however, while our reproduced accuracies for FOSTER on ImageNet is significantly lower than the ones reproduced by Liu et al. [2024], it is similar to the results reproduced by Zhou et al. [2023b] in their recent review. As this difference is superior to 5 percentage points on ImageNet, comparison with PlaceboCIL in this setting may not be relevant, nevertheless we report those results in the Appendices for the sake of completeness.

For all the considered baselines, using our proposed approach MADM leads to a significant improvement of the average incremental accuracy in the Training From Half procedure: up to $2.30p.p.$ for DER, up to $3.06p.p.$ for MEMO, up to $4.54p.p.$ for FOSTER, up to

[2]See: https://github.com/openai/CLIP/tree/main/notebooks
[3]Available at: https://huggingface.co/Salesforce/blip2-opt-2.7b

Table 4.1: Average incremental accuracy (Top-1) on CIFAR100, ImageNet-Subset, and ImageNet with the Learning From Half procedure: a base step containing half of the classes followed by 5, 10, and 25 incremental steps, using a growing memory of 20 exemplars per class. Using ResNet-32 for CIFAR100 and ResNet-18 for ImageNet-Subset and ImageNet. Using 500 synthetic samples per class for CIFAR100 and 1300 per class for ImageNet-Subset and Imagenet. Synthetic samples generated using Stable Diffusion with a guidance scale of 2.0 . Results marked with "*" correspond to our own experiments. Results on CIFAR100 and ImageNet-Subset are averaged over 3 random runs. Results on ImageNet are reported as a single run. Best result is marked in bold and second best is underlined. The final accuracy (Top-1) is reported in the Appendices.

Methods	CIFAR100			ImageNet-Subset			ImageNet	
	B50 Inc10	B50 Inc5	B50 Inc2	B50 Inc10	B50 Inc5	B50 Inc2	B500 Inc100	B500 Inc50
iCaRL* [Rebuffi et al., 2017]	57.68	52.44	48.05	65.34	60.74	54.31	53.58	48.65
iCaRL w/ MADM (ours)	62.01	59.24	56.47	68.64	65.55	62.75	57.15	54.07
LUCIR* [Hou et al., 2019]	63.37	60.88	57.07	70.75	68.48	63.14	66.69	64.06
LUCIR w/ MADM (ours)	65.77	63.84	61.73	72.92	71.48	69.48	67.56	66.44
DER* [Yan et al., 2021]	67.74	65.62	64.82	74.40	73.46	70.16	**69.02**	67.68
DER w/ MADM (ours)	68.73	67.53	67.12	74.82	74.24	72.40	68.61	**68.03**
MEMO* [Zhou et al., 2023c]	67.71	65.68	64.07	75.86	74.42	72.00	66.62	65.87
MEMO NME*	68.11	67.09	65.41	71.65	71.46	71.24	65.95	64.60
MEMO NME w/ MADM (ours)	68.67	67.55	67.13	75.05	74.51	74.05	66.38	65.55
FOSTER* [Wang et al., 2022a]	71.17	68.89	65.07	76.09	75.05	70.96	62.18	60.83
FOSTER w/ MADM (ours)	**72.18**	**70.88**	**68.06**	**77.13**	**76.77**	**75.50**	63.19	62.09

Table 4.2: Average incremental accuracy (Top-1) and Final accuracy (Top-1) on CIFAR100, and ImageNet-Subset with the Learning from Scratch procedure: 10 steps of 10 classes or 20 steps of 5 classes, using a fixed memory of 2,000 exemplars in total. Using ResNet-32 for CIFAR100 and ResNet-18 for ImageNet-Subset. Using 500 synthetic samples per class for CIFAR100 and 1300 per class for ImageNet-Subset. Synthetic samples generated using Stable Diffusion with a guidance scale of 2.0 . Results marked with "*" correspond to our own experiments. "Average" is Average Incremental Accuracy (Top-1) and "Last" is the final overall accuracy of the model after the last incremental step. Results are averaged over 3 random runs. Best result is marked in bold and second best is underlined.

Methods	CIFAR100				ImageNet-Subset			
	B0 Inc10		B0 Inc5		B0 Inc10		B0 Inc5	
	Average	Last	Average	Last	Average	Last	Average	Last
iCaRL* [Rebuffi et al., 2017]	64.79	48.33	63.15	44.47	66.77	51.51	61.15	44.29
iCaRL w/ MADM (ours)	67.44	53.40	67.66	52.73	70.59	57.51	68.52	55.91
DER* [Yan et al., 2021]	70.68	60.95	70.64	59.21	75.47	64.37	73.43	61.69
DER w/ MADM (ours)	71.88	**62.67**	<u>72.32</u>	**62.15**	<u>77.20</u>	<u>66.91</u>	<u>76.23</u>	<u>65.33</u>
MEMO* [Zhou et al., 2023c]	70.02	58.44	68.46	52.96	72.81	60.25	69.06	53.95
MEMO NME*	69.79	57.71	67.46	48.94	71.00	56.36	66.86	50.23
MEMO NME w/ MADM (ours)	71.51	61.51	71.47	<u>59.94</u>	76.23	64.81	73.97	62.27
FOSTER* [Wang et al., 2022a]	<u>73.03</u>	60.89	70.94	56.76	76.44	65.88	72.92	60.96
FOSTER w/ MADM (ours)	**74.02**	<u>62.38</u>	**72.97**	58.23	**79.42**	**70.20**	**77.25**	**66.52**

6.34*p.p.* for LUCIR and up to 8.44*p.p.* for iCaRL, depending on the dataset and the number of incremental steps. This improvement is especially important in more challenging settings with numerous incremental steps or when the replay memory is small as discussed in Section 4.4.3. Our proposed method is also effective in the more challenging Training From Scratch procedure, with improvements of the average incremental accuracy up to 2.80*p.p.* for DER, up to 4.33*p.p.* for FOSTER, up to 4.91*p.p.* for MEMO, and up to 7.37*p.p.* for iCaRL, depending on the dataset and the number of incremental steps.

When looking more closely at the learned model, we can see that our approach improves the capacity of the model to preserve past knowledge without penalizing its plasticity. This is especially true in the most challenging settings with a small replay memory: for example on CIFAR100 B50 Inc5 with 5 exemplars per class in the memory, by combining LUCIR with MADM, the final accuracy on the 50 bases classes increases from 34.15% to 42.06% while the accuracy on the 50 remaining classes also increases from 46.08% to 52.23%. By combining our proposed method MADM with FOSTER, we achieve new state-of-the-art performance on several datasets and settings.

Table 4.3: Average incremental accuracy (Top-1) on CIFAR100, and ImageNet-Subset Alt. with the Learning From Half procedure: a base step containing half of the classes followed by 5, 10, and 25 incremental steps, using a growing memory of 20 exemplars per class. Using 500 synthetic samples per class for CIFAR100 and 1300 per class for ImageNet-Subset Alt. . Synthetic samples generated using Stable Diffusion with a guidance scale of 2.0 . Results marked with "*" correspond to our own experiments and results marked with "†" are reported from [Liu et al., 2024]. Results averaged over 3 random runs.

Methods	CIFAR100			ImageNet-Subset Alt.		
	B50 Inc10	B50 Inc5	B50 Inc2	B50 Inc10	B50 Inc5	B50 Inc2
FOSTER† [Wang et al., 2022a]	70.62	68.43	63.83	80.21	77.63	69.27
FOSTER w/ PlaceboCIL† [Liu et al., 2024]	71.97	70.31	67.02	**82.03**	**79.52**	72.79
Improvement in *p.p.*	**+1.35**	+1.88	**+3.19**	+1.82	+1.89	+3.52
FOSTER* [Wang et al., 2022a]	71.17	68.89	65.07	78.52	76.49	71.24
FOSTER w/ MADM (ours)	**72.18**	**70.88**	**68.06**	80.35	79.21	**77.09**
Improvement in *p.p.*	+1.01	**+1.99**	+2.99	**+1.83**	**+2.72**	**+5.85**

Compared to PlaceboCIL [Liu et al., 2024], our proposed method MADM achieves similar or higher performance while using a significantly smaller dataset of additional external data: 50,000 synthetic images compared to the 1.28 million of real images from ImageNet for the experiments on CIFAR100 and 130,000 synthetic images compared to about 1.1 million of real images sampled without overlapping from ImageNet for the experiments on ImageNet-Subset Alt. dataset. MADM achieves higher average incremental accuracy than PlaceboCIL in 4 out of 6 experiments and outperforms it by up to 4.30*p.p.*. Moreover, by relying on synthetic labeled images from past classes, our proposed method is more versa-

tile and can also be combined with methods that don't rely on a distillation loss such as the dynamic-networks-based methods DER and MEMO whereas PlaceboCIL is limited to distillation-based approaches.

4.4.3 Ablation Study

Components

Table 4.4: Average incremental accuracy (Top-1) on CIFAR100 with the Training From Half procedure, using a growing memory of 20 exemplars per class. Using 500 synthetic samples per class. Synthetic samples generated using Stable Diffusion with a guidance scale of 2.0 . Results marked with "*" correspond to our own experiments. Results averaged over 3 random runs.

Methods	CIFAR100		
	B50 Inc10	B50 Inc5	B50 Inc2
LUCIR* [Hou et al., 2019]	63.37	60.88	57.07
w/ Synth. Distillation	64.62	63.12	**62.15**
w/ Synth. Distillation w/o new	62.66	61.93	61.48
w/ Synth. Replay	64.95	62.44	59.59
w/ MADM	**65.77**	**63.84**	61.73

By leveraging synthetic images supposedly belonging to the same classes as the training dataset, it is possible to use them for both replay and distillation. Table 4.4 compares the performance when using the synthetic images only for distillation (w/ Synth. Distillation), only for replay (w/ Synth. Replay), and for both combined (w/ MADM). Distillation and Replay using synthetic data appear to be complementary, Replay achieving higher performances for a short number of incremental steps T and Distillation for a higher number of incremental steps. Combining both in MADM achieves an overall improvement over each used individually. Moreover, models trained with our proposed method MADM tend to achieve a better balance between stability and plasticity compared to models trained using the synthetic samples only for distillation, the latter favoring more stability.

Size of the synthetic dataset

While synthetic samples could be generated in an online manner at the creation of each mini-batch in order to increase diversity, we instead made the choice of repeatedly sampling synthetic data from the fixed dataset $\mathcal{S}$. The synthetic dataset $\mathcal{S}$ is fixed while training the current model θ_t and is solely extended at the end of each incremental step by generating n synthetic samples for each newly learned class, effectively decrease the computational cost of our proposed method compared to online generation. Table 4.5 shows the impact of the

number of synthetic images n generated for each past class used during training. With only 50 synthetic images per class, our method can improve the average incremental accuracy by up to $3.57p.p.$ compared to the standard LUCIR. This can be further increased to $5.72p.p.$ by expanding the size of the synthetic dataset. However, it appears that beyond a certain limit, further increasing the size of the synthetic dataset does not yield any significant improvement. This supports our approach of not using the diffusion model in an online manner to generate each training batch if the storage is not a constraint of the system.

Table 4.5: Average incremental accuracy (Top-1) on CIFAR100 with the Training From Half procedure depending on n, the number of synthetic samples generated for each class. Synthetic samples generated using Stable Diffusion with a guidance scale of 2.0 . Using a growing memory of 20 exemplars per class. Results marked with "*" correspond to our own experiments. Results averaged over 3 random runs.

Methods	CIFAR100		
	B50 Inc10	B50 Inc5	B50 Inc2
LUCIR* [Hou et al., 2019]	63.37	60.88	57.07
w/ MADM $n = 50$	64.84	62.71	60.64
w/ MADM $n = 500$	65.77	63.84	61.73
w/ MADM $n = 1000$	**66.41**	64.91	62.73
w/ MADM $n = 2000$	66.22	**65.03**	**62.79**

Size of the replay memory

Table 4.6 measures the improvement from our method depending on the number of incremental steps and the size of the replay memory containing exemplars of past classes. While the improvement compared to the baseline method is more significant in the case of a small memory size, our method also significantly improves the average incremental accuracy of both LUCIR and iCaRL when combined with a larger replay memory. Likewise, the improvement is more notable for more difficult settings containing numerous incremental steps. For example, when trained for 25 incremental steps with 5 samples per class in the replay memory, MADM almost doubles the final overall accuracy for iCaRL, increasing it from 20.32% to 40.39% and augments the average incremental accuracy by $18.56p.p.$. Those results highlight that our method can boost the performance of the baseline method it is combined with independently of the setting. Additional results for CIFAR100 with 5 incremental steps can be found in the Appendices.

Table 4.6: Average incremental accuracy (Top-1) on CIFAR100 with the Learning From Half procedure: a base step containing half of the classes followed by 5, 10, and 25 incremental steps depending on the number of exemplars saved in memory for each class. Using 500 synthetic samples per class. Synthetic samples generated using Stable Diffusion with a guidance scale of 2.0 . Results marked with "*" correspond to our own experiments and results marked with "†" are reported from [Liu et al., 2024]. Improvement reported comparatively to the baseline method. Results averaged over 3 random runs.

Methods	CIFAR100											
	5 exemplars/class			10 exemplars/class			20 exemplars/class			50 exemplars/class		
	B50 Inc10	B50 Inc5	B50 Inc2	B50 Inc10	B50 Inc5	B50 Inc2	B50 Inc10	B50 Inc5	B50 Inc2	B50 Inc10	B50 Inc5	B50 Inc2
iCaRL* [Rebuffi et al., 2017]	43.40	39.30	29.59	52.30	46.79	38.78	57.68	52.44	48.05	62.09	58.42	55.36
iCaRL w/ PlaceboCIL† [Liu et al., 2024]	51.55	-	-	59.11	-	-	61.24	-	-	-	-	-
iCaRL w/ MADM (ours)	55.36	50.80	48.15	58.53	55.70	52.51	62.01	59.24	56.47	65.53	63.92	62.18
Improvement in *p.p.*	+11.96	+11.50	+18.56	+6.23	+8.91	+13.73	+4.33	+6.80	+8.42	+3.44	+5.50	+6.82
LUCIR* [Hou et al., 2019]	53.14	52.79	44.67	60.77	57.55	50.77	63.37	60.88	57.07	65.63	62.39	60.80
LUCIR w/ PlaceboCIL† [Liu et al., 2024]	62.74	-	-	64.79	-	-	65.28	-	-	-	-	-
LUCIR w/ MADM (ours)	62.90	59.81	56.32	64.47	62.16	59.44	65.77	63.84	61.73	67.21	65.81	64.48
Improvement in *p.p.*	+9.76	+7.02	+11.65	+3.70	+4.61	+8.67	+2.40	+2.96	+4.66	+1.58	+3.42	+3.68

Diffusion model

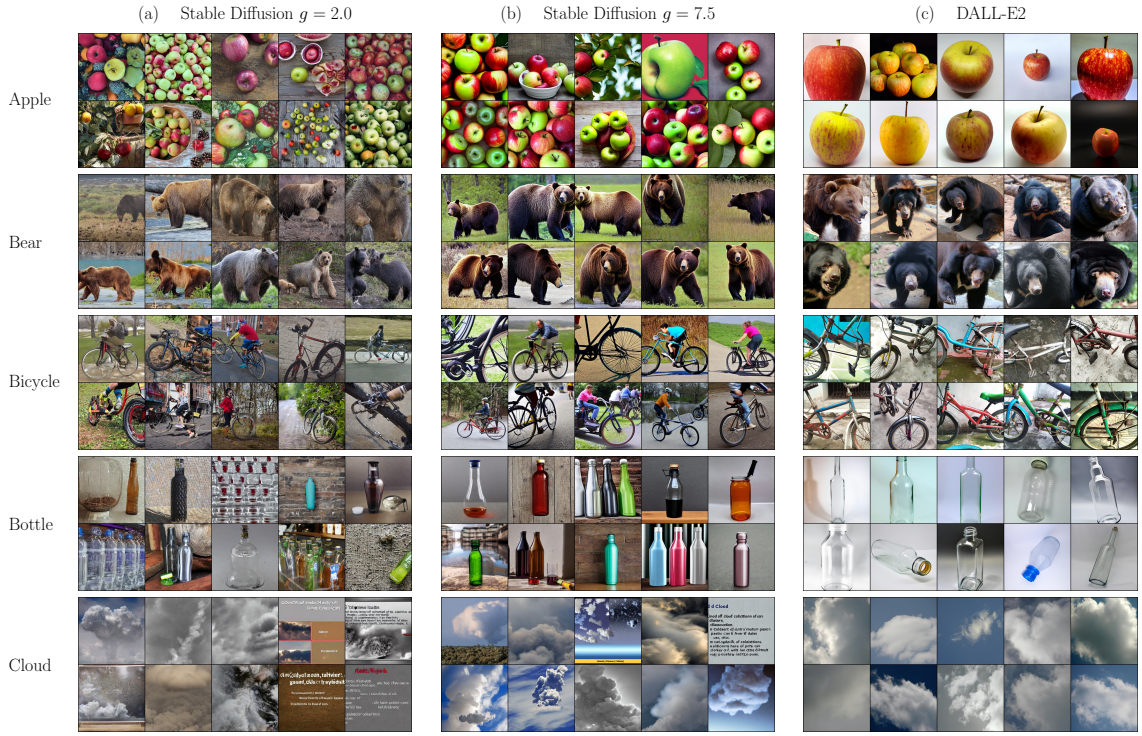


Figure 4.4: Samples from $\mathcal{S}$, the set of additional synthetic images generated for class-incremental learning on CIFAR100, depending on the diffusion model used: (a) Stable Diffusion version 1.4 with a guidance scale of 2.0, (b) Stable Diffusion version 1.4 with a guidance scale of 7.5, and (c) DALL-E2. Images are generated at the size 512×512 and are being resized to 32×32 before being fed to the model. Images are displayed before resizing to 32×32 for better appreciation.

Table 4.7 shows the performance of LUCIR and iCaRL on CIFAR100 with the Learning From Scratch procedure depending on the Diffusion model used for augmenting the replay memory. It compares the default Stable Diffusion version 1.4 with a low value of 2.0 for the guidance scale which favors diversity, the same Stable Diffusion model but with a higher value of 7.5 for the guidance scale which generates more realistic yet less diverse images, and DALL-E2 for which the trade-off between diversity and quality of the generated image cannot be controlled but seems to favors quality in our case, as illustrated in Figure 4.4. Experimentally, it appears that if the number of synthetic samples generated for each class is small, our proposed method achieves higher performance when combined with a Diffusion Model that favors quality over diversity such as DALL-E2 or Stable Diffusion with a high value for guidance scale. Conversely, when it is possible to generate more synthetic images per class, the model that favors diversity reaches a higher accuracy than the two others.

Table 4.7: Average incremental accuracy (Top-1) of (a) iCaRL and (b) LUCIR on CIFAR100 with the Training From Half procedure depending on the Diffusion model used and n, the number of synthetic samples generated for each class. "SD" is refering to Stable Diffusion version 1.4 and g is the guidance scale. Using a growing memory of 20 exemplars per class. Results marked with "*" correspond to our own experiments. Results averaged over 3 random runs.

(a) iCaRL - Using different Diffusion Models

Methods		CIFAR100		
		B50 Inc10	B50 Inc5	B50 Inc2
	iCaRL* [Rebuffi et al., 2017]	57.68	52.44	48.05
$n = 50$	w/ MADM SD $g = 2.0$	59.76	56.32	52.58
	w/ MADM SD $g = 7.5$	**60.84**	**57.39**	**54.50**
	w/ MADM DALL-E2	60.32	57.16	53.11
$n = 500$	w/ MADM SD $g = 2.0$	**62.01**	**59.24**	**56.47**
	w/ MADM SD $g = 7.5$	61.82	58.67	55.59
	w/ MADM DALL-E2	61.42	58.35	55.13

(b) LUCIR - Using different Diffusion Models

Methods		CIFAR100		
		B50 Inc10	B50 Inc5	B50 Inc2
	LUCIR* [Hou et al., 2019]	63.37	60.88	57.07
$n = 50$	w/ MADM SD $g = 2.0$	64.84	62.71	60.64
	w/ MADM SD $g = 7.5$	**65.25**	**63.18**	61.39
	w/ MADM DALL-E2	64.99	63.09	**61.47**
$n = 500$	w/ MADM SD $g = 2.0$	**65.77**	**63.84**	61.73
	w/ MADM SD $g = 7.5$	65.55	63.59	**61.89**
	w/ MADM DALL-E2	65.12	63.41	61.48

Prompting mechanism

Another important component of the generation process used to create synthetic samples of past classes is the textual prompt guiding the diffusion process. In addition to the default prompt mechanism "c, d_c" based on WordNet which create one prompt per class, we also conduct additional experiments by generating synthetic samples using a larger and more diverse set of prompts: using CLIP-prompt which relies on sentences and names prompt-engineered for zero-shot classification on ImageNet and Caption-prompt which relies on textual description of the training dataset generated using a image-to-text captioning pre-trained model. Experiment on the two variants of ImageNet-Subset, reported in Table 4.8, shows that more sophisticated prompting mechanism can improve the performance of our proposed method and are especially effective in the most difficult settings where there are numerous incremental steps. With the training from half procedure, Caption-prompt can

Table 4.8: Average incremental accuracy (Top-1) and Final accuracy (Top-1) on (a) ImageNet-Subset and (b) ImageNet-Subset Alt. with the Learning from Half procedure: a base step containing half of the classes followed by 5, 10, and 25 incremental steps, using a growing memory of 20 exemplars per class. Using ResNet-18 model and 1300 synthetic samples per class. Synthetic samples generated using Stable Diffusion with a guidance scale of 2.0 . Results marked with "*" correspond to our own experiments. "Average" is Average Incremental Accuracy (Top-1) and "Last" is the final overall accuracy of the model after the last incremental step. Results are averaged over 3 random runs.

(a) ImageNet-Subset - Using different prompting mechanisms

Methods	ImageNet-Subset					
	B50 Inc10		B50 Inc5		B50 Inc2	
	Average	Last	Average	Last	Average	Last
iCaRL* [Rebuffi et al., 2017]	65.34	54.48	60.74	49.37	54.31	42.69
w/ MADM - Default prompt	68.64	59.47	65.55	57.17	62.75	54.62
w/ MADM - CLIP-prompt	68.57	59.52	65.73	57.74	63.00	54.91
w/ MADM - Caption-prompt	**68.99**	**60.13**	**67.07**	**59.61**	**64.24**	**57.17**

(b) ImageNet-Subset Alt. - Using different prompting mechanisms

Methods	ImageNet-Subset Alt.					
	B50 Inc10		B50 Inc5		B50 Inc2	
	Average	Last	Average	Last	Average	Last
iCaRL* [Rebuffi et al., 2017]	64.67	53.25	58.92	48.23	54.01	43.63
w/ MADM - Default prompt	69.42	61.07	66.47	59.13	64.98	58.31
w/ MADM - CLIP-prompt	69.84	62.19	67.30	60.53	65.85	59.79
w/ MADM - Caption-prompt	**70.88**	**63.36**	**68.88**	**62.57**	**68.01**	**62.40**

improve the average incremental accuracy by up to $3.03p.p.$ and the final accuracy by up to $4.09p.p.$ compared to the default prompting mechanism. While these two more advanced prompting mechanisms induce additional requirements (lengthy manual annotation and tuning by humans, having access to large-language model or text-to-image model, the storage and computational cost associated to those additional models) compared to the default prompt, they can significantly improve the performances of our proposed method and should be considered if they fit into the constraints of the considered application.

Architecture of the backbone

Following the standard practice, the modified ResNet-32 model with 0.46 million parameters was used for the backbone in our experiments. Additionally, we report in Table 4.9 the performance for iCaRL and FOSTER using a standard ResNet-18 model with 11.2 million parameters. Our proposed method takes advantage of larger backbones and further improves the performance of the baseline methods. When using ResNet-18 instead

Table 4.9: Average incremental accuracy (Top-1) of (a) iCaRL and (b) FOSTER on CIFAR100 with the Learning From Half procedure depending on the backbone used. Using 500 synthetic samples per class. Synthetic samples generated using Stable Diffusion with a guidance scale of 2.0 . Using a growing memory of 20 exemplars per class. "#P" denote the number of parameters in the backbone, in million. Results marked with "*" correspond to our own experiments. Results averaged over 3 random runs.

(a) iCaRL - Using different backbone models

Backbone	Methods	#P	CIFAR100		
			B50 Inc10	B50 Inc5	B50 Inc2
ResNet-32	iCaRL* [Rebuffi et al., 2017]	0.46	57.68	52.44	48.05
ResNet-32	iCaRL w/ MADM	0.46	62.01	59.24	56.47
	Improvement in *p.p.*		+4.33	+6.80	+8.42
ResNet-18	iCaRL* [Rebuffi et al., 2017]	11.2	62.40	55.96	50.79
ResNet-18	iCaRL w/ MADM	11.2	67.18	63.76	60.96
	Improvement in *p.p.*		+4.78	+7.80	+10.17

(b) FOSTER - Using different backbone models

Backbone	Methods	#P	CIFAR100		
			B50 Inc10	B50 Inc5	B50 Inc2
ResNet-32	FOSTER* [Wang et al., 2022a]	0.46	71.17	68.89	65.07
ResNet-32	FOSTER w/ MADM	0.46	72.18	70.88	68.06
	Improvement in *p.p.*		+1.01	+1.99	+2.99
ResNet-18	FOSTER* [Wang et al., 2022a]	11.2	74.64	69.98	64.77
ResNet-18	FOSTER w/ MADM	11.2	77.68	75.60	73.00
	Improvement in *p.p.*		+3.04	+5.62	+8.24

of ResNet-32 for FOSTER, the relative improvement resulting from our proposed method increases from 1.4% to 4.1% for B0 Inc10, from 2.9% to 8.0% for B0 Inc10, and from 4.6% to 12.7% for B0 Inc2.

Augmentation

In our main experiments, following the official implementation, we used AutoAugment [Cubuk et al., 2019] for both FOSTER and FOSTER combined with our method. Additionally, to make a fair comparison with DER [Yan et al., 2021], we report in Table 4.10 the performance of FOSTER on the CIFAR100 dataset with the Training From Half procedure without using AutoAugment.

While the performances of FOSTER are significantly impacted by the choice of the augmentations used during training, our proposed method significantly improves its performance in both cases. When using the same augmentations, FOSTER combined with our proposed method is able to close the gap with DER while using a significantly smaller

Table 4.10: Average Incremental Accuracy (Top-1) of FOSTER and DER on CIFAR100 with the Training From Half procedure depending on the augmentation used during training. Using 500 synthetic samples per class. Synthetic samples generated using Stable Diffusion with a guidance scale of 2.0 . Results averaged over 3 random runs.

	Methods	CIFAR100		
		B50 Inc10	B50 Inc5	B50 Inc2
With AutoAugment	FOSTER* [Wang et al., 2022a]	71.17	68.89	65.07
	FOSTER w/ MADM (our)	72.18	70.88	68.06
	DER* [Yan et al., 2021]	72.01	70.97	69.72
	DER w/ MADM (our)	**72.32**	**71.47**	**70.63**
Without AutoAugment	FOSTER* [Wang et al., 2022a]	64.38	61.11	59.10
	FOSTER w/ MADM (our)	67.56	66.19	64.58
	DER* [Yan et al., 2021]	67.74	65.62	64.82
	DER w/ MADM (our)	**68.73**	**67.53**	**67.12**

model.

Use of data from new classes

In their recent work, Liu et al. [2024] emphasized that using data from new classes for the distillation loss was detrimental and proposed to only use data of past classes from the memory in addition to the unlabeled samples from the additional external dataset. Following their study, we also report in Table 4.4 the results of our method while using the distillation loss only on the synthetic images of past classes and on the exemplar from the memory (w/ Synth. Distillation w/o new). However, it appears that in our case, this significantly decreases the performance of the model. As further discussed below, we suppose that it may be due to the gap between the distribution of synthetic images and the distribution of real images: while real samples from new images are generally harmful to the distillation loss, they may be useful for transferring general knowledge about real data in our case.

4.4.4 From synthetic to real images

As discussed in Section 4.3.3, using synthetic data in the classification loss may negatively impact the performance of the model.

We first observed this phenomenon when using the synthetic dataset generated by Stable Diffusion as a direct replacement for the replay memory in the exemplar-free class-incremental learning setting. As depicted in Table 4.11, when using synthetic images as exemplars for the past classes, LUCIR has difficulty generalizing to the real test dataset and achieves significantly lower accuracy compared to using real exemplars in the memory.

Table 4.11: Performances of standard LUCIR [Hou et al., 2019] on CIFAR100 with the Training From Half procedure depending on the number of exemplars per class saved in the memory and whether the exemplars are real or synthetic images. Synthetic samples generated using Stable Diffusion with a guidance scale of 2.0 . "Average" is Average Incremental Accuracy (Top-1) and "Last" is the final overall accuracy of the model after the last incremental step. Results averaged over 3 random runs.

Memory	CIFAR100			
	B0 Inc10		B0 Inc5	
	Average	Last	Average	Last
Real 20	**63.37**	**53.91**	**60.88**	**51.42**
Synthetic 20	52.38	33.68	42.79	21.83
Synthetic 100	54.54	36.56	48.56	27.93
Synthetic 500	55.77	38.78	52.19	32.58

Table 4.12: Average Incremental Accuracy (Top-1) of DER [Yan et al., 2021] on CIFAR100 with the Training From Half procedure depending on the bias correction method used. Using 500 synthetic samples per class. Synthetic samples generated using Stable Diffusion with a guidance scale of 2.0 . "(NME)" indicates that the model is evaluated using a Nearest-Mean-of-Exemplars classifier instead of the default linear classifier. Results marked with "*" correspond to our own experiments. Results averaged over 3 random runs.

Bias Correction	Methods	CIFAR100		
		B50 Inc10	B50 Inc5	B50 Inc2
Weight Aligning	DER*	67.00	63.64	62.89
	w/ Synth. aux.	67.60	65.43	64.09
	w/ Synth. aux. (NME)	68.23	65.94	64.17
	w/ MADM	61.14	61.07	60.85
	w/ MADM (NME)	**68.75**	67.13	66.30
Balanced Finetuning	DER*	67.74	65.62	64.82
	w/ MADM	68.73	**67.53**	**67.12**

Moreover, while it performs better than exemplar-based methods used without any replay memory, it is not competitive with approaches specifically designed for this setting.

Secondly, in Table 4.12, the performance of DER [Yan et al., 2021] are compared for two different bias mitigation methods: balanced finetuning on real exemplars and Weight Aligning [Zhao et al., 2020] similarly to MEMO [Zhou et al., 2023c]. This experiment shows the impact on the classifier of the synthetic images when used in the classification loss. Without considering the distribution gap between real and synthetic samples, it results in a significant decrease of the performance compared to the baseline. But when correctly mitigated, using either an NME classifier of real exemplars only or finetuning the classifier on a small balanced subset containing only real images, it reaches higher performance compared to naively using the synthetic data for the auxiliary loss only. In comparison, Weight Aligning [Zhao et al., 2020] notably used by MEMO for bias mitigation cannot address the distribution gap between synthetic and real data, which forces us to use an NME classifier for MEMO when combined with our method. This explains some lower results for MEMO as a linear classifier correctly trained is often more accurate than an NME classifier. For example, on ImageNet-Subset B50 Inc10, MEMO combined with our method using an NME classifier achieves 75.05% of average incremental accuracy, slightly lower than the base method MEMO with a linear classifier 75.86% but significantly higher than with a NME classifier 71.65%. Similarly, while being less significant, the negative impact of the distribution gap on methods such as LUCIR, which rely on cosine classifier and fix the class embeddings after learning it using real data only, can still be observed in the most challenging settings: on CIFAR100 with B50 Inc2 setting and 5 exemplars per class in the replay memory, using an NME classifier for LUCIR when combined with our method improves the average incremental accuracy by $0.89 p.p.$, reaching 57.21% while for standard LUCIR it decreases it by $0.01 p.p.$.

4.5 Conclusion

In this work, we showed that pre-trained generative text-to-image model can be seamlessly combined with general methods for class-incremental learning in order to further improve their performances. Our approach leverages text-to-image diffusion models to generate labeled synthetic images belonging to the same classes as the ones previously encountered by the model and use those synthetic data for both the distillation loss and the classification loss. Complete experiments show that our approaches significantly increase the average incremental accuracy of state-of-the-art methods for class-incremental learning, especially in settings with highly restricted memory, and achieve comparable or superior results compared to other competitive approaches relying on additional real data samples while offering more flexibility.

While our work focused on a simple and straightforward integration of the synthetic images into the class-incremental learning setting, it would be interesting in future works

to consider how the different losses could be modified and adapted to make the most of the synthetic images. Moreover, in order to reduce the distribution gap between real and synthetic samples, the generation process should be improved. For example by finetuning the diffusion models on the training dataset beforehand, or by relying on the image embedding of the exemplars from the rehearsal memory in combination with the textual prompts for conditioning the diffusion process. Finally, we will also explore the different possible uses of those pre-trained generative models for class-incremental learning: for example to prepare the model by learning in advance, during the initial step, classes that may be encountered in the future, based on already encountered classes.

Chapter 5

Future-Proofing Class-Incremental Learning

5.1 Introduction

While humans naturally learn and accumulate knowledge continuously during their whole lifespan, deep neural networks struggle to do the same. Continuously training deep neural networks on new data will inevitably induce a severe decrease of the performance on the previously learned data. This phenomenon, named catastrophic forgetting [McCloskey and Cohen, 1989; Ratcliff, 1990], has received more and more attention during the past decade, and mitigating it is the core problem of incremental learning and continual learning. Among the different settings of incremental learning, Class-Incremental Learning (CIL) is among the most challenging ones. In this setting, the model is incrementally trained on a few number of classes at a time, without having access to the previously learned ones. The objective is to train a unified model that performs well on every encountered class.

Rehearsal memory [Rebuffi et al., 2017; Chaudhry et al., 2019; Lomonaco et al., 2022] is one of the most commonly used methods to mitigate catastrophic forgetting and is used by most of the approaches for Class-Incremental Learning. It consists of a small buffer containing few representative exemplars of each previously learned class that are later used for replay while learning new classes. However, there remain situations where using a rehearsal memory is not possible, either due to privacy concerns regarding the training data (especially for tasks such as person re-identification) or due to limitations in the storage space. This led to the emergence of a new setting named Exemplar-Free Class-Incremental Learning (EFCIL) or Class-Incremental Learning Without Memory. Methods for EFCIL often freeze the feature extractor after the first incremental step [Hayes and Kanan, 2020; Petit et al., 2023], achieving competitive performances while allowing for a faster and less computationally expensive learning of new classes compared to methods that have to back-

propagate through the entire deep neural network. However, this may also lead to poor performance when the number of classes available in the initial step to train the feature extractor is too limited.

To address this limitation, several works have proposed to pre-train the feature extractor [Lomonaco et al., 2022; Zhou et al., 2024] using additional data from classes similar yet different from those in the target dataset. However, with the advent of foundational diffusion model for image generation, pre-training the feature extractor on a more targeted dataset, specifically classes that are highly likely to be encountered during the future incremental steps, has the potential to lead to better improvements in this setting. While the aspect of using future classes to pre-train the model beforehand seems counter to the concept of Class-Incremental Learning, we argue that it is indeed a realistic setting. In many real world applications, collecting training data for every possible object in an ever changing world to train an omniscient classifier is impossible. It is much more realistic to train a classifier that is more specific to the needs of the user. Even under such a scenario, collecting every possible object that the user needs is an insurmountable task. Therefore, it is much more desirable to attempt to predict specific objects that are likely to be encountered, pre-train the feature extractor, and then perform Incremental Learning while it is deployed to the user. In our case, using a diffusion model that generates training data makes it possible to pre-train especially when objects can be difficult or impossible to obtain.

In this work, we propose to leverage pre-trained text-to-image diffusion models to better prepare Exemplar-Free Class-Incremental Learning methods relying on a frozen feature extractor for the future. The diffusion models generate synthetic samples of possible future classes that will be jointly used with actual real data during the first incremental step to train a feature extractor that better generalizes to future classes. Experiments using various challenging settings and datasets for EFCIL highlight how our method can be combined with state-of-the-art approaches and results in significant improvement of the average incremental accuracy. Moreover, it appears that using synthetic images of future classes achieves higher performance than using real images from different yet similar classes.

Our contributions can be summarized into three points:

- We propose a framework relying on pre-trained diffusion models to better prepare Exemplar-Free Class-Incremental Learning methods for the future, reaching new state-of-the-art performances.

- While previous approaches for Incremental Learning used pre-trained diffusion models for generating samples from past classes, our proposed method aims at generating samples from future classes instead.

- We experimentally show that using synthetic samples of future classes for training the feature extractor achieves higher performance than using real images of classes different from those in the target dataset.

5.2 Related Work

5.2.1 Class-Incremental Learning (CIL) and Exemplar-Free Class-Incremental Learning (EFCIL)

While Class-Incremental Learning may be applied to various tasks and settings such as semantic segmentation [Douillard et al., 2021a; Cong et al., 2024] or federated learning [Dong et al., 2022, 2023b], in this work we only consider the image classification task which is the most extensively studied. Most approaches for Class-Incremental Learning [Belouadah et al., 2021; Zhou et al., 2023b] leverage knowledge distillation [Rebuffi et al., 2017; Hou et al., 2019; Douillard et al., 2020; Dong et al., 2023a], a rehearsal memory and a method to mitigate the bias toward the new classes [Hou et al., 2019; Wu et al., 2019; Jodelet et al., 2021; Ahn et al., 2021] induced by the limited size of the replay memory. Recent works [Yan et al., 2021; Zhou et al., 2023c] proposed to extend the feature extractor at each incremental step to learn the new classes while preserving past knowledge by freezing previously learned weights.

Exemplar-Free Class-Incremental Learning is a more challenging setting in which there is no rehearsal memory, making the replay of samples from the previously encountered classes impossible. Similarly to approaches designed for Class-Incremental Learning, numerous methods for Exemplar-Free Class-Incremental Learning also continuously finetune the feature extractor and the classifier [Wu et al., 2021a; Zhu et al., 2021a; Jodelet et al., 2022; Zhu et al., 2021b, 2022]. Several authors have highlighted the risk of overfitting on the current task and the importance of having general and transferable feature representation for EFCIL. To address this concern, approaches often rely on self-supervised learning [Wu et al., 2021a; Zhu et al., 2021b] or by creating augmented classes using Mixup [Zhang et al., 2017]. Recently, methods relying on a fixed feature extractor have gained more attention [Hayes and Kanan, 2020; Ostapenko et al., 2022]. Those methods either rely on a pre-trained model [Wang et al., 2022c,d; Smith et al., 2023b; Zhou et al., 2024] or by freezing the feature extractor after learning it during the initial step [Hayes and Kanan, 2020; Petit et al., 2023; Goswami et al., 2023]. Petit et al. [2023] proposed FeTrIL which achieves state-of-the-art performances by relying on a frozen feature extractor and a linear classifier trained using actual features of new classes and pseudo-features of past classes generated by a geometric translation of new class features. FeCAM [Goswami et al., 2023] is a recently proposed approach for Exemplar-Free Class-Incremental Learning with a fixed feature extractor that rely on the Mahalanobis distance combined with Tukey's transformation, covariance shrinkage, and correlation normalization. Those methods exhibit numerous advantages including a less computationally expensive training procedure that allows for for a deployment on edge devices. However, they also depend on the initial training strategy used [Janson et al., 2022; Petit et al., 2024] and may achieve poor performance if the quantity of data available for training the feature extractor during the initial step is too

limited.

5.2.2 Training on synthetic images

While few attempts have been made using GAN-based models [Besnier et al., 2020; Jahanian et al., 2022], the use of synthetic images for training deep learning models has recently gained momentum following the advances in diffusion models [Sohl-Dickstein et al., 2015; Ho et al., 2020] and their wide availability. Diffusion models generate synthetic samples using an iterative denoising process conditioned on an input, usually a text prompt. Imagen [Saharia et al., 2022] performs the denoising task in the pixel space while Stable Diffusion [Rombach et al., 2022] and DALL-E2 [Ramesh et al., 2022] perform it in the latent space in order to decrease the computational cost of the generative process. In one of the pioneer works, Sarıyıldız et al. [2023] studied the use of the pre-trained text-to-image diffusion model Stable Diffusion to generate a synthetic replacement for the original training dataset. Their experiments highlighted that while the model trained using the real train dataset reaches a higher accuracy on the real test dataset compared to the one trained on synthetic images, they perform on par for transfer learning tasks. Concurrently, Zhang et al. [2024] proposed to expand real datasets using synthetic samples generated with various diffusion models and achieved significant improvement on various image classification datasets. Several authors proposed to use synthetic images generated by pre-trained diffusion models for different problems of computer vision such as zero-shot and few-shot learning [He et al., 2022], long-tailed image classification [Shin et al., 2023], incremental learning [Jodelet et al., 2023; Duan et al., 2023], out-of-distribution detection [Du et al., 2024], or semantic segmentation [Nguyen et al., 2024]. Tian et al. [2023] showed that the representation learned using contrastive learning on synthetic images generated by Stable Diffusion outperforms state-of-the-art methods trained on real datasets.

5.2.3 Additional data for Incremental Learning

Several authors [Lee et al., 2019; Zhang et al., 2020; Lechat et al., 2021; Bellitto et al., 2022; Liu et al., 2024] have considered the use of additional data for Class-Incremental Learning by leveraging datasets of real images belonging to classes different from the ones that the model has to learn. Most of these approaches assume that the additional dataset is unlabeled, restricting the use of those additional data samples to the distillation loss [Lee et al., 2019; Zhang et al., 2020; Liu et al., 2024]. Jodelet et al. [2023] first proposed to use large pre-trained text-to-image diffusion models for Incremental Learning in order to generate additional synthetic images for the classes previously encountered by the model. Using a textual prompt based on the class name, this method generates labeled samples that can be used for the distillation loss but also for any supervised losses such as the classification loss. More recently, pre-trained diffusion models were also used for Class-Incremental semantic segmentation [Chen et al., 2023] and object detection [Kim et al.,

2024]. However, there remains a significant distribution gap between the real samples and the synthetic ones [Jodelet et al., 2023] whose impact on the classifier makes the use of pre-trained diffusion model for replaying samples from past classes especially difficult for Exemplar-Free Class-Incremental Learning.

5.2.4 Preparing for the future

While most of the methods for Class-Incremental Learning only focus on how to prevent the forgetting of past data, some authors have studied how to better prepare the model for future updates. Douillard et al. [2021b] took inspiration from zero-shot learning and proposed to incorporate into the classifier feature vectors of unseen future classes during training. Those possible future vectors, named ghost features, are produced by a generator using actual features of seen classes and prior information about all the classes. [Pernici et al., 2021] proposed to use a pre-allocated Regular Polytope Classifier which can exploit future unseen classes as negative examples. Zhou et al. [2022] proposed to reserve embedding space for future classes by pre-assigning virtual prototypes and by forecasting future classes using manifold Mixup [Zhang et al., 2017]. Bellitto et al. [2022] proposed to use an auxiliary dataset of real images containing classes unrelated to the main task in order to prepare the model for the future. Those additional data samples are used to learn more general and stable representation that can be leveraged when learning future classes.

Compared to previously mentioned methods, in this work, we aim to leverage pre-trained text-to-image diffusion models not for generating additional past data but for generating synthetic samples for possible future classes. We study how those synthetic images of future classes can be used during the initial incremental step and to what extent they can help to better prepare Exemplar-Free incremental learning methods relying on a frozen feature extractor.

5.3 Proposed method

The objective of Class-Incremental Learning (CIL) is to learn a unified model θ on a gradually increasing set of classes. θ can be decomposed into $\{\Phi, \mathcal{W}\}$ with Φ the feature extractor and $\mathcal{W}$ the classifier. The training procedure is divided into T incremental steps, each step consisting of a dataset $\mathcal{D}_t$ containing samples from the set of new classes $\mathcal{Y}_t$ such that $\forall i, j \in \{1, 2, ..., T\}, \mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset$ for $i \neq j$. Contrary to standard Class-Incremental Learning, in the Exemplar-Free Class-Incremental Learning (EFCIL) setting, the model does not have access to any replay memory: during each incremental step t, the model θ_t is trained solely on the current dataset $\mathcal{D}_t$. After each incremental step, the model is evaluated on the test dataset of all the classes observed so far without having any step descriptor.

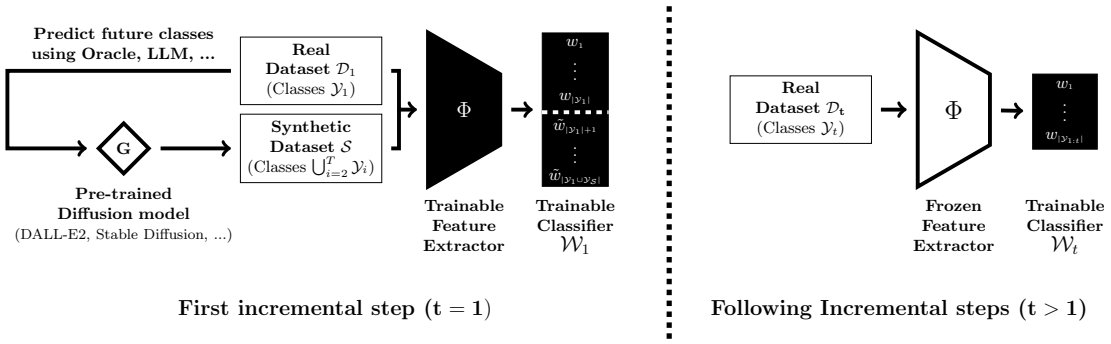


Figure 5.1: Diagram representing our proposed method **F**uture-**P**roof **C**lass **I**ncremental **L**earning (**FPCIL**) which prepares the feature extractor for the future by jointly training it on an auxiliary dataset in addition to the current dataset during the initial step. The auxiliary dataset contains synthetic samples of future classes generated using a pre-trained text-to-image diffusion model. After the first incremental step, the feature extractor ϕ is frozen and the weights of the classifier $\tilde{w}$ corresponding to the future classes are removed. Afterward, in the following incremental step, only the classifier $\mathcal{W}$ is trained using dedicated method (e.g. FeTrIL, FeCAM).

5.3.1 Future-Proofing Class-Incremental Learning

In order to create a future-proof model, as illustrated in Figure 5.1, we introduce **F**uture-**P**roofing **C**lass **I**ncremental **L**earning (**FPCIL**), a straightforward approach which can be combined with any method for Class-Incremental Learning relying on a frozen feature extractor. During the first incremental step, when training the feature extractor, in addition to the initial step dataset $\mathcal{D}_1$, an additional dataset $\mathcal{S}$ containing $|\mathcal{Y}_\mathcal{S}|$ different classes is used. While any labeled datasets could be used for $\mathcal{S}$, we propose a novel approach which consists in using images from classes that will be encountered by the model during the future incremental steps. In detail, the objective of the initial incremental step is extended from solely learning the initial classes $\mathcal{Y}_1$ into learning $\mathcal{Y}_1 \cup \mathcal{Y}_\mathcal{S}$ where $\mathcal{Y}_\mathcal{S} = \bigcup_{i=2}^{T}\mathcal{Y}_i$ are the future classes. The feature extractor is jointly trained with a classifier on $\mathcal{D}_1 \cup \mathcal{S}$. At the end of the initial step, the classifier is restricted to the base classes $\mathcal{Y}_1$ by dropping the weights corresponding to the future classes from the auxiliary dataset $\mathcal{S}$, and the feature extractor is frozen. Then, the training procedure continues according to the baseline method for the remaining incremental steps. Our proposed method is detailed in Algorithm 5.1.

In the initial step, our proposed method FPCIL trains on both the initial training dataset $\mathcal{D}_1$ and an additional dataset $\mathcal{S}$ which contains training data for predicted future classes. As only the data for the current step are available in the Class-Incremental Learning setting, we propose to use synthetic data instead of the real ones. We use a pre-trained text-to-image diffusion model to generate synthetic images from the future classes during the

Algorithm 5.1: Future-Proof Class-Incremental Learning (FPCIL)

Input: Data-flow $\{\mathcal{D}_i\}_{i=1}^{T}$ and pre-trained generative model G.
Ouput: Models $\{\theta_i\}_{i=1}^{T}$ composed of $\{\Phi_i, \mathcal{W}_i\}$.

for $t \in \{1, 2, ..., T\}$ **do**
  if $t = 1$ **then**
    `// Predict future classes based on initial classes using oracle or GPT`
    $\mathcal{Y}_\mathcal{S} \leftarrow$ PredictFuture($\mathcal{Y}_1$)

    `// Generate synthetic samples of future classes`
    $\mathcal{S} \leftarrow$ GenerateSynthetic(G, $\mathcal{Y}_\mathcal{S}$)

    `// Train the model using real data of the first step and synthetic data of future steps`
    Train θ_1 using $\mathcal{D}_1 \cup \mathcal{S}$
    Freeze Φ_1 and remove $\mathcal{Y}_\mathcal{S}$ weights from $\mathcal{W}_1$
  else
    `// Incrementally update the classifier using dedicated method (e.g. FeTrIL, FeCAM)`
    Train $\mathcal{W}_t$ using $\mathcal{D}_t$

initial step. This means that only the labels of the future classes are necessary. Compared to recent approaches for Incremental Learning relying on a pre-trained diffusion model [Jodelet et al., 2023; Chen et al., 2023; Kim et al., 2024] for generating synthetic samples of past classes, our proposed method aims at generating synthetic samples of future classes instead. By primarily using the synthetic data to train the feature extractor, our proposed method circumvents the impact of the distribution gap between real and synthetic samples generated by the diffusion model, making it applicable to Exemplar-Free Class-Incremental Learning.

While the requirement of having access to the label of future classes may seem counter to the concept of Class-Incremental Learning, we argue that it is practical and compatible with most real-world applications of Class-Incremental Learning. In general, during the developing a deep learning model meant for continuously accumulating new knowledge while achieving a given task in an ever-evolving environment, both the task and the environment are known. Therefore, it is reasonable to assume that it is possible, based on this initial knowledge of the environment and task, to predict their possible future evolution. However, acquiring beforehand real data corresponding to these future states of the environment and task may be difficult, expensive, or even impossible in some cases which lead us to use

synthetic samples instead. As a concrete example, we propose to consider the case of an agent that would be deployed on an embedded device for assisting its user inside his home. The agent would have to be able to learn new concepts based on data directly collected from its environment depending on the particular need of its user. It would have to rely on incremental learning as frequent retraining from scratch would not be possible on either a cloud server as uploading user's data would be difficult due to privacy reasons, or in local due to the limited computational and memory capacity of the embedded device. Therefore, using our proposed method it is possible to train an initial model precisely tailored for each user (or group of users) before deployment, by relying on available knowledge about the user and its environment (interests, location, particular requests, etc.) to preemptively learn specific concepts that may appear in the future in addition to generic concepts. Since building a specific dataset of real samples for all possible needs of all possible users would not be tractable due to the significant cost in time and resources of the data collection process, the use of synthetic data appears as a faster and cheaper alternative. As a second example, we consider the case of an agent deployed on an embedded device for performing a task in a hard to access and potentially hostile environment where the agent has to be autonomous due to unreliable remote control possibility (deep sea, space, etc). The nature of the environment intrinsically limits the capacity to collect large training datasets in advance but its properties may be known. Using our proposed method, the agent could be trained before its deployment using synthetic data from generative diffusion models, or more realistically in this scenario using simulation-based tools. The agent would then be able to incrementally learn and adjust itself to the environment using real data collected while performing the task.

It should also be noted that based on our experiments detailed in the ablation study, the classes in the auxiliary dataset $\mathcal{S}$ do not need to be perfectly predicted to have a positive impact on the performance. For clarity, we refer to FPCIL as "FPCIL-Oracle" when the list of future classes has been perfectly predicted and "FPCIL-Partial (k%)" otherwise, with k the percentage of future classes correctly predicted.

Predictions of the future classes can be done through expert knowledge or through automated methods. Here, we introduce a simple approach for automatically predicting the future classes based on recently popular large language models. Our proposed method relies on the GPT model [Brown et al., 2020] "GPT-3.5 Turbo Instruct" used through the OpenAI API. The large language model is queried for a completion task using the prompt "The dataset contains the following 100 classes: c_1, [...], c_k," where c_1 to c_k are the names of the k classes from the initial step, with k equals to $|\mathcal{Y}_1|$. The number "100" is used here to encourage the model to generate a longer list of possible future classes as initial experiments without specifying a number would result in only a few number of guesses. In practice, the output of GPT has to be parsed to extract the predicted future classes. The completion function of GPT often continues the prompt as if it was part of a website or a scientific paper by appending imaginary details such as the number of images per class in the dataset,

Figure 5.2: Synthetic images of future classes for CIFAR100 generated using different diffusion models. Images are displayed before being resized for better appreciation. First row has been generated by Stable Diffusion 1.4 with a guidance scale of 2.0, second row using Stable Diffusion 1.4 with a guidance scale of 7.5, and last row using DALL-E2. From left to right: butterfly, can, house, lion, mountain, orange, otter, pine tree, rose, and streetcar.

nonexistent download url for the dataset, or architecture of the model. Due to the random nature of the output from GPT, the prompt is queried 10 times and the predicted classes are ordered according to the number of times they have been predicted. The auxiliary dataset $\mathcal{S}$ is then restricted to the classes which were the most often predicted. We evaluated three versions with an increasing level of restriction: "FPCIL-GPT", "FPCIL-GPT R1", and "FPCIL-GPT R2". Those approaches based on GPT also serve to imitate a more realistic application of Future-Proof Class-Incremental Learning by predicting various number of future classes with various accuracy.

5.3.2 Generating future data

Leveraging recent advances in diffusion models and the wide availability of generative text-to-image models pre-trained on natural images, we propose to use them for generating the synthetic images dataset of future classes following the procedure initially defined by Sarıyıldız et al. [2023].

The auxiliary dataset $\mathcal{S}$ is generated using a pre-trained diffusion model G conditioned solely on a textual prompt. As proposed in [Sarıyıldız et al., 2023], the default prompt used is "c, d_c" with "c" the name of the class and "d_c" a simple description of the class. Adding the description of the class into the prompt is intended to ensure that the generated images belong to the targeted class and not to any homonyms: for example "crane" in the ImageNet dataset, which may refer to either the bird or the machine used to move heavy objects. The definition "d_c" also tends to restrain the domain of the generated images even though the domain could also be enforced by adding it to the prompt. To automatically create the prompts, we rely on WordNet [Miller, 1995] and use the lemmas of the synset as "c" and the definition of the synset as "d_c". For each class, n synthetic images are generated with

n equals to the number of images per class in the target dataset. Compared to traditional pre-training and concurrent method [Bellitto et al., 2022], our proposed method does not require the costly and fastidious process of collecting, curating, and annotating a dataset of real images. Examples of synthetic images generated for CIFAR100 are displayed in Figure 5.2

The guidance scale is an important hyper-parameter of diffusion models relying on classifier-free guidance [Ho and Salimans, 2021]. To some extent, it can be used to control the trade-off between the diversity and the quality of the generated synthetic images. For Stable Diffusion, the default recommended value for the guidance scale is often 7.0 but following [Jodelet et al., 2023; Sarıyıldız et al., 2023] we use the lower value of 2.0 as a higher diversity is beneficial for pre-training.

While this generation procedure allows our proposed method to achieves significant improvement compared to baselines, there remains an important distribution gap between the real and synthetic data [Jodelet et al., 2023]. Reducing this distribution gap and its impacts on the training of the feature extractor would further improve the performance of our proposed method. As possible future research direction, the diffusion model could be finetuned using the data from the first incremental step, a more sophisticated prompting mechanism could be implemented, or a selection process could be used to discard the synthetic samples too far away from the distribution of the target dataset.

Table 5.1: Definition of the different auxiliary datasets used during the first incremental step for each setting. "Class overlap" is the percentage of classes (actual number of classes in parentheses) from the future incremental steps of the target dataset that are in the auxiliary dataset $\mathcal{S}$, i.e. the number of correctly predicted future classes.

Target dataset	Auxiliary dataset $\mathcal{S}$	Nature of data	Setting	Number of classes $\lvert\mathcal{Y}_\mathcal{S}\rvert$	Samples per class n	Class overlap with future steps of target dataset	Note
CIFAR100 (100 classes)	CIFAR10	Real	All	10	5000	0%	
	FPCIL-Partial (k%)	Synthetic	B50 Inc	50	500	k%	k% of the classes have been randomly selected from the target dataset, the remaining have been randomly selected from the wrong predictions of GPT-3.5. For k=100, we use the denomination "FPCIL-Oracle".
			B0 Inc10	90	500	k%	
			B0 Inc5	95	500	k%	
	FPCIL-GPT	Synthetic	B0 Inc10	150	500	58.9% (53)	Classes the most often predicted by GPT-3.5 based on the classes encountered in the first incremental step.
			B0 Inc5	166	500	53.7% (51)	
	FPCIL-GPT R1	Synthetic	B0 Inc10	90	500	47.8% (43)	Subset of "FPCIL-GPT" containing a more restricted set of classes.
			B0 Inc5	102	500	33.7% (32)	
	FPCIL-GPT R2	Synthetic	B0 Inc10	53	500	33.3% (30)	Subset of "FPCIL-GPT" containing a more restricted set of classes. More restrictive than R1.
			B0 Inc5	58	500	21.0% (20)	
ImageNet-Subset (100 classes)	ImageNet-Compl.	Real	B50 Inc	50	∼1300	0%	Classes have been randomly sampled from the remaining 900 classes of ILSVRC 2012. No class overlap with ImageNet-Subset.
			B0 Inc10	90	∼1300	0%	
			B0 Inc5	95	∼1300	0%	
	ImageNet-Compl. X2	Real	B50 Inc	100	∼1300	0%	Classes have been randomly sampled from the remaining 900 classes of ILSVRC 2012. No class overlap with ImageNet-Subset.
			B0 Inc10	180	∼1300	0%	
			B0 Inc5	190	∼1300	0%	
	ImageNet-Compl. Animals	Real	B0 Inc10	90	∼1300	0%	Classes have been randomly sampled among the "animals" classes from the remaining 900 classes of ILSVRC 2012. No class overlap with ImageNet-Subset.
			B0 Inc5	95	∼1300	0%	
	FPCIL-Oracle	Synthetic	B50 Inc	50	1300	100% (50)	Future classes perfectly predicted.
			B0 Inc10	90	1300	100% (90)	
			B0 Inc5	95	1300	100% (95)	
	FPCIL-Partial (0%)	Synthetic	B50 Inc	50	1300	0%	Same classes as "ImageNet-Compl.".
			B0 Inc10	90	1300	0%	
			B0 Inc5	95	1300	0%	
	FPCIL-Partial Animals (0%)	Synthetic	B0 Inc10	90	1300	0%	Same classes as "ImageNet-Compl. Animals".
			B0 Inc5	95	1300	0%	

5.4 Experiments

5.4.1 Experimental setup

Datasets

The experiments are conducted on two large scale datasets: CIFAR100 [Krizhevsky et al., 2009] and ImageNet-Subset. CIFAR100 is a dataset containing 60,000 32×32 RGB images evenly divided among 100 classes with 500 training images and 100 testing images per class. ImageNet-Subset is subset of the high-resolution images dataset ImageNet (ILSVRC 2012) [Russakovsky et al., 2015]. This subset contains 100 randomly selected classes with about 1,300 training images and 50 testing images per class. Following Rebuffi et al. [2017], the classes are ordered using numpy with the random seed 1993.

Two training procedures are used: Training from Scratch (TFS) and Training From Half (TFH). For the Training from Scratch procedure, the classes are evenly divided into T (10, or 20) incremental steps, each incremental step containing the same number of classes. For the Training From Half procedure, following [Hou et al., 2019], the first step contains half of the classes and the remaining classes are evenly divided into the following $T-1$ incremental steps. Following Zhou et al. [2023b], the two training procedures are referred to using the common naming system "B-i Inc-j" with i the number of classes in the first step and j the number of classes each of the following incremental step; i equals to zero is used to refer to the Learning From Scratch procedure.

Model training

Following common practice for Exemplar-Free Class-Incremental Learning [Petit et al., 2023; Zhu et al., 2021b, 2022], ResNet-18 is used for both CIFAR100 and ImageNet-Subset. The feature extractor is trained solely during the first incremental step and is frozen for the rest of the training procedure. The backbone is trained with SDG using the standard Softmax Cross-Entropy loss and AutoAugment [Cubuk et al., 2019] for 160 epochs with a batch size of 128 and a weight decay of $5e^{-4}$. The initial learning rate is set to 0.1 and decrease to zero using a cosine annealing schedule. For the classifier, we use a fully-connected linear layer trained following FeTrIL [Petit et al., 2023]. The classifier is trained using SGD for 50 epochs at each incremental step, with an initial learning rate of 0.1 and cosine annealing schedule. In the ablation study, we also report results using a Nearest Class Mean (NCM) Classifier and FeCAM [Goswami et al., 2023]. If not stated otherwise, the reported results for the different baselines have been reproduced by our own experiments. Experiments are conducted using Pytorch [Paszke et al., 2019] and PyCIL [Zhou et al., 2023a].

Synthetic generation

Two diffusion models are used for generating the synthetic samples during training: Stable Diffusion [Rombach et al., 2022] and DALL-E2 [Ramesh et al., 2022]. If not specified otherwise, the default diffusion model is Stable Diffusion v1.4 [1] used with a guidance scale of 2.0 for 50 steps. Following [Sarıyıldız et al., 2023], "c, d_c" is used for the prompt. The same datasets of synthetic images are used for all the experiments. Those datasets have been generated prior to incremental learning experiments, at size 512×512 using a fixed seed and contains 500 images per class for CIFAR100 and 1300 images per class for ImageNet-Subset. For CIFAR100, the synthetic images are resized to 32×32 before being fed to the model. For DALL-E2, because the model is used through the official API of OpenAI, it is not possible to control the different hyper-parameters of the model; the default, unknown, values are used for the guidance scale, the random seed and the number of denoising step. A slightly different prompt is used for DALL-E2, "a photo of a c, d_c", as the default one would generate unsatisfactory images for some classes. In order to comply with the moderation policy of the API, few definitions d_c have been changed.

Baselines

As a comparison baseline, a dataset containing real images from classes similar yet different to the ones in the target dataset is used as auxiliary dataset $\mathcal{S}$ during the first incremental step. The purpose of this baseline is to compare our proposed method with the more standard pre-training approach. It is similar to the method proposed by Bellitto et al. [2022] adapted to Exemplar-Free Class-Incremental Learning with a frozen feature extractor. For CIFAR100, we use CIFAR10 for the auxiliary dataset as both datasets were created using the same annotation and curation methodology from the TinyImages Dataset [Torralba et al., 2008]. However, while CIFAR10 contains more samples than the datasets generated using our proposed method FPCIL, it is important to note that CIFAR10 contains only 10 classes. For ImageNet-Subset, the auxiliary dataset "ImageNet-Compl." was created by randomly selecting classes from the complement of ImageNet-Subset in ImageNet. "ImageNet-Compl." contains the same number of classes as the datasets generated by our proposed method FPCIL and approximately the same number of samples. This is a strong baseline as both subsets are sampled from the same dataset and, while they do not contain the exact same classes, classes in both datasets are from the same categories and some are highly similar: for example both datasets contain different breeds of dogs or species of insects. We additionally use "ImageNet-Compl. X2" which contains two times more classes than ImageNet-Compl. We also report the performance of "FPCIL-Partial (k%)" which generates a dataset of the same size as FPCIL-Oracle but with only k% of the classes correctly predicted. For CIFAR100, this auxiliary dataset is built by randomly sampling k% of the classes from FPCIL-Oracle and the remaining ones from the wrong prediction of

[1] https://huggingface.co/CompVis/stable-diffusion-v1-4

GPT-3.5. For ImageNet-Subset, "FPCIL-Partial (0%)" contains synthetic images belonging to the same classes as the ones in ImageNet-Compl. The definition of all the different auxiliary datasets used are summarized in Table 5.1.

Evaluation

Models are evaluated and compared using both the "Final Accuracy" and the "Average Incremental Accuracy" [Rebuffi et al., 2017]. The Average Incremental Accuracy is defined as the average of the Top-1 accuracy of the model on the test dataset of all the classes learned so far, at the end of each incremental step including the first one.

5.4.2 Results

Average incremental accuracy and final accuracy of our proposed method FPCIL are reported for CIFAR100 and ImageNet-Subset in Table 5.2 and Table 5.3 for both the Learning From Scratch and Learning From Half procedures.

Our proposed approach FPCIL considerably improves the average incremental accuracy of the state-of-the-art method FeTrIL [Petit et al., 2023], from 5.39 percentage points ($p.p.$) up to $22.12p.p.$ on CIFAR10 and from $4.24p.p.$ up to $33.00p.p.$ on ImageNet-Subset. This improvement is notably higher for the more difficult Learning From Scratch procedure which is particularly challenging for FeTrIL due to the limited quantity of data available during the first incremental step. Compared to using an auxiliary dataset of real images from classes different from those in the target dataset, our method also achieves significantly higher accuracy. This highlights the fundamental advantage of our proposed method: while the standard pre-training procedure improves the generalization capability of the feature extractor by leveraging additional data belonging to classes different from those in the target dataset, our proposed method aims to learn a feature extractor specifically designed for the incoming task by relying on data from future classes instead. Even if those data from future classes are synthetic, it allows to learn a feature extractor more suitable for the future tasks resulting in a significantly higher accuracy compared standard pre-training on real data. Moreover, on ImageNet-Subset, our proposed method FPCIL even outperforms a two times larger auxiliary dataset of real images, demonstrating that our proposed algorithm is more data-efficient that standard pre-training. Furthermore synthetic datasets can be rapidly generated whereas auxiliary datasets of real data require a long and costly collection, curation, and annotation process. Finally, even in the case where the future classes have not been accurately predicted, the auxiliary dataset of synthetic images performs on par with the standard pre-training approach relying on real datasets: for the Learning From Half procedure, the difference of average incremental accuracy between using ImageNet-Compl or its synthetic counterpart is at most $0.29p.p.$. This makes our proposed method beneficial in every setting.

On CIFAR100, our more practical approach FPCIL-GPT also reaches a higher accuracy

Table 5.2: Performances on CIFAR100 with the Learning from Half and Learning From Scratch procedures. Synthetic samples generated using Stable Diffusion with a guidance scale of 2.0 . Using ResNet-18 as the backbone. "AVG" is Average Incremental Accuracy (Top-1) and "Last" is the final overall accuracy of the model after the last incremental step. Best result is marked in bold and second best is underlined. Results averaged over 3 random runs.

Methods	CIFAR100 - LFH						CIFAR100 - LFS			
	B50 Inc10		B50 Inc5		B50 Inc2		B0 Inc10		B0 Inc5	
	AVG	Last	AVG	Last	AVG	Last	AVG	Last	AVG	Last
FeTrIL [Petit et al., 2023]	68.05	58.69	66.81	58.22	65.46	56.94	50.50	33.86	40.80	24.65
FeTrIL w/ Aux. CIFAR10	<u>70.60</u>	60.82	<u>69.34</u>	60.03	<u>68.23</u>	59.26	58.56	41.72	52.42	35.03
FeTrIL w/ FPCIL-Oracle	**73.44**	**65.71**	**72.81**	**65.20**	**72.40**	**64.76**	**68.09**	**55.43**	**62.92**	**49.67**
FeTrIL w/ FPCIL-Partial (0%)	70.25	<u>61.41</u>	69.27	<u>60.56</u>	68.13	<u>59.67</u>	62.63	47.71	57.46	42.71
FeTrIL w/ FPCIL-GPT	-	-	-	-	-	-	<u>67.18</u>	<u>53.89</u>	<u>62.15</u>	<u>48.67</u>
FeTrIL w/ FPCIL-GPT R1	-	-	-	-	-	-	66.79	53.02	60.65	46.08
FeTrIL w/ FPCIL-GPT R2	-	-	-	-	-	-	65.25	51.01	59.89	44.33

Table 5.3: Performances on ImageNet-Subset with the Learning from Half and Learning From Scratch procedures. Synthetic samples generated using Stable Diffusion with a guidance scale of 2.0 . Using ResNet-18 as the backbone. "AVG" is Average Incremental Accuracy (Top-1) and "Last" is the final overall accuracy of the model after the last incremental step. Best result is marked in bold and second best is underlined. Results averaged over 3 random runs.

Methods	ImageNet-Subset - LFH						ImageNet-Subset - LFS			
	B50 Inc10		B50 Inc5		B50 Inc2		B0 Inc10		B0 Inc5	
	AVG	Last	AVG	Last	AVG	Last	AVG	Last	AVG	Last
FeTrIL [Petit et al., 2023]	72.97	64.49	72.01	63.54	71.02	62.19	51.64	34.53	39.64	22.42
FeTrIL w/ Aux. ImageNet-Compl.	75.19	67.81	74.08	66.96	73.25	65.71	70.62	59.53	68.14	57.45
FeTrIL w/ Aux. ImageNet-Compl. X2	<u>76.33</u>	<u>69.52</u>	<u>75.30</u>	<u>68.53</u>	<u>74.69</u>	<u>67.89</u>	<u>73.96</u>	<u>64.68</u>	**73.01**	<u>62.98</u>
FeTrIL w/ FPCIL-Oracle	**77.21**	**71.80**	**76.81**	**71.51**	**76.84**	**71.17**	**74.04**	**65.65**	<u>72.64</u>	**64.29**
FeTrIL w/ FPCIL-Partial (0%)	74.90	67.27	73.93	66.37	73.18	65.25	68.89	57.21	65.94	55.09

than the baseline method FeTrIL yet slightly lower than the FPCIL-Oracle. On CIFAR100 in the Learning From Scratch procedure with 10 steps of 10 classes (B0 Inc10), the less restricted FPCIL-GPT predicted a total of 150 classes with 53 that actually are among the 90 future classes while the most restricted FPCIL-GPT R2 predicted only 53 classes with 30 among the 90 future classes. All the details regarding future class predictions using GPT are reported in Table 5.1. While the more restricted version FPCIL-GPT R1 and R2 tend to predict a lower number of classes that are not among the future classes, the less restricted version predicts a higher number of future classes, achieving higher performances. There is a trade-off between the precision and recall in the prediction of future classes, while predicting more classes improve the performance it also increase the computational cost of the first incremental step due to the cost of generating synthetic images and training the feature extractor. In the ablation study, we further analyze the impact of the correctness of the prediction of future classes.

5.4.3 Ablation study

Accurately predicting the future

Table 5.4: Performances on CIFAR100 with the Learning From Scratch procedure depending on the ratio of correctly predicted future classes in the complementary dataset used during the initial step. Synthetics samples generated using Stable Diffusion with a guidance scale of 2.0 . Using ResNet-18 as the backbone. "AVG" is Average Incremental Accuracy (Top-1) and "Last" is the final overall accuracy of the model after the last incremental step. Best result is marked in bold. Results averaged over 3 random runs.

Methods	CIFAR100 - B0 Inc10	
	AVG	Last
FeTrIL [Petit et al., 2023]	50.50	33.86
FeTrIL w/ FPCIL-Oracle (100%)	**68.09**	**55.43**
FeTrIL w/ FPCIL-Partial (66%)	66.06	52.21
FeTrIL w/ FPCIL-Partial (50%)	64.79	50.76
FeTrIL w/ FPCIL-Partial (33%)	64.32	49.90
FeTrIL w/ FPCIL-Partial (0%)	62.63	47.71

In Table 5.4, we compare the performance of our proposed methods combined with FeTrIL on CIFAR100 B0 Inc10 depending on the ratio of correctly predicted future classes in the auxiliary dataset used during the initial step. Experimentally, for a fixed dataset size, the average incremental accuracy increase with the correctness of the prediction of the future classes. This confirms the importance of predicting the future classes compared to the standard pre-training procedure which relies on real data of classes different from those in the target dataset that will be encountered by the continual learner in the future.

Moreover, even in the case where no future class has been correctly predicted, our proposed method still improves the performance of the baseline approach as, in most cases, additional data help to learn a feature extractor with better generalization capability.

Table 5.5: Performances on ImageNet-Subset with the Learning From Scratch procedure depending on the complementary dataset used during initial step. Synthetic samples generated using Stable Diffusion with a guidance scale of 2.0 . Using ResNet-18 as the backbone. "AVG" is Average Incremental Accuracy (Top-1) and "Last" is the final overall accuracy of the model after the last incremental step. Best result is marked in bold. Results averaged over 3 random runs.

Methods	ImageNet-Subset - LFS			
	B0 Inc10		B0 Inc5	
	AVG	Last	AVG	Last
FeTrIL [Petit et al., 2023]	51.64	34.53	39.64	22.42
FeTrIL w/ FPCIL-Oracle	**74.04**	**65.65**	**72.64**	**64.29**
FeTrIL w/ Aux. ImageNet-Compl. Animals	68.74	57.87	64.09	54.29
FeTrIL w/ Aux. ImageNet-Compl.	70.62	59.53	68.14	57.45
FeTrIL w/ FPCIL-Partial Animals (0%)	66.08	54.97	61.63	51.11
FeTrIL w/ FPCIL-Partial (0%)	68.89	57.21	65.94	55.09

Additionally, in Table 5.5 we report the accuracy for an alternative version of ImageNet-Compl and FPCIL-Partial named respectively "ImageNet-Compl. Animals" and "FPCIL-Partial Animals". ImageNet-Compl Animals is built by sampling classes from the complement to ImageNet-Subset in ImageNet while restricting the classes to animals; FPCIL-Partial Animals contains the same classes that ImageNet-Compl Animals but the images are generated using a diffusion model. These two more specialized datasets achieve notably lower accuracy than their more general version. This further emphasizes the importance of predicting the future classes and not limiting the scope of the auxiliary dataset.

Quality and diversity of synthetic samples

In Table 5.6, the performance of FPCIL Oracle is reported on CIFAR100 depending on the diffusion model used to generate the synthetic images of future classes. By default in our study, we used Stable Diffusion with a guidance scale of 2.0 [Sarıyıldız et al., 2023] to favor the diversity to the detriment of the quality. We verified this assumption by comparing it against Stable Diffusion with a higher guidance scale and DALL-E2 which both tend to favor more the quality than the diversity. Experiments confirm that when pre-training the feature extractor on future classes, the diversity is more important than the quality of the generated samples: Stable Diffusion with a small guidance scale achieves the highest results, ahead of DALL-E2 and Stable Diffusion with the default guidance scale.

In Table 5.7, the performance of FPCIL Oracle is reported on CIFAR100 for different

Table 5.6: Performances on CIFAR100 with the Learning from Half and Learning From Scratch procedures depending on the Diffusion Model used for generating the synthetic data: Stable Diffusion with a guidance scale of 2.0, Stable Diffusion with a guidance scale of 7.5, and DALL-E2. Using ResNet-18 as the backbone. "AVG" is Average Incremental Accuracy (Top-1) and "Last" is the final overall accuracy of the model after the last incremental step. Best result is marked in bold. Results averaged over 3 random runs.

Methods	CIFAR100					
	B50 Inc5		B0 Inc10		B0 Inc5	
	AVG	Last	AVG	Last	AVG	Last
FeTrIL [Petit et al., 2023]	66.81	58.22	50.50	33.86	40.80	24.65
FeTrIL w/ FPCIL-Oracle SD $g = 2.0$	**72.81**	**65.20**	**68.09**	**55.43**	**62.92**	**49.67**
FeTrIL w/ FPCIL-Oracle SD $g = 7.5$	72.22	64.40	66.48	53.36	61.63	48.50
FeTrIL w/ FPCIL-Oracle DALL-E2	71.08	63.02	63.94	50.39	59.76	46.34

Table 5.7: Performances on CIFAR100 with the Learning from Half and Learning From Scratch procedures for FeTrIL depending on the number of synthetic samples n generated for each future class. Synthetic samples generated using Stable Diffusion with a guidance scale of 2.0 . Using ResNet-18 as the backbone. "AVG" is Average Incremental Accuracy (Top-1) and "Last" is the final overall accuracy of the model after the last incremental step. Results averaged over 3 random runs.

Methods	CIFAR100					
	B50 Inc5		B0 Inc10		B0 Inc5	
	AVG	Last	AVG	Last	AVG	Last
FeTrIL [Petit et al., 2023]	66.81	58.22	50.50	33.86	40.80	24.65
FeTrIL w/ FPCIL-Oracle $n = 50$	68.31	59.67	60.45	45.56	55.52	40.89
FeTrIL w/ FPCIL-Oracle $n = 250$	71.27	63.23	65.53	52.40	61.45	48.38
FeTrIL w/ FPCIL-Oracle $n = 500$	72.81	65.20	68.09	55.43	62.92	49.67
FeTrIL w/ FPCIL-Oracle $n = 1000$	73.95	66.77	70.24	58.37	64.89	52.73

number of synthetic samples n generated for each future classes. While the default value of n equals 500 was used in the other experiments in order to match the size of the target dataset, it appears that just 50 samples per class is enough to improve the performance of the baseline and even outperform the use of eternal dataset of unrelated classes such as CIFAR10 which is more than 10 times bigger. Moreover, it is possible to further improve the performance of our proposed method by generating more synthetic data if the constraints of the system permits it.

In addition to the trade-of between quality and diversity, it is important to emphasize that a significant gap remains between real images and the synthetic samples generated using pre-trained diffusion models. As illustrated in Table 5.5, while Aux. ImageNet-Compl. and FPCIL-Partial (0%) contain the same classes and almost the same number of samples, there is a difference of about 2 percentage points of accuracy between the feature extractor trained on those datasets. The exact same observation can be made for Aux. ImageNet-Compl. Animals and FPCIL-Partial Animals (0%). Similarly, while achieving significantly higher results than the baseline, FPCIL-Oracle still achieves lower performances on both CIFAR100 and ImageNet-Subset compared to a upper-bound model which would have been trained on the complete target dataset at once.

Combining real and synthetic data

Table 5.8: Performances on CIFAR100 with the Learning from Half and Learning From Scratch procedures for FeTrIL depending on the nature (real or synthetic) of the data used to train the feature extractor during the first incremental step. Synthetic samples generated using Stable Diffusion with a guidance scale of 2.0 . Using ResNet-18 as the backbone. "AVG" is Average Incremental Accuracy (Top-1) and "Last" is the final overall accuracy of the model after the last incremental step. Best result is marked in bold. Results averaged over 3 random runs.

Methods	Initial classes	Future classes	CIFAR100					
			B50 Inc5		B0 Inc10		B0 Inc5	
			AVG	Last	AVG	Last	AVG	Last
FeTrIL [Petit et al., 2023]	real	-	66.81	58.22	50.50	33.86	40.80	24.65
FeTrIL w/ FPCIL-Oracle var1	-	synth.	47.00	44.10	64.05	53.30	64.07	52.44
FeTrIL w/ FPCIL-Oracle var2	synth.	synth.	58.04	53.52	65.56	54.42	**65.24**	**53.51**
FeTrIL w/ FPCIL-Oracle	real	synth.	**72.81**	**65.20**	**68.09**	**55.43**	62.92	49.67

During the first incremental step, our proposed method FPCIL combines the real samples of the initial classes with synthetic samples of possible future classes to train the feature extractor. In Table 5.8, different variations of our proposed method are compared on CIFAR100 to estimate the contribution of each component. "FPCIL var1" does not use the real samples from the first classes and trains the feature extractor solely using synthetic sam-

Table 5.9: Performances on CIFAR100 (a) B50 Inc5 and (b) B0 Inc10 depending on the backbone used. Synthetic samples generated using Stable Diffusion with a guidance scale of 2.0 . "AVG" is Average Incremental Accuracy (Top-1) and "Last" is the final overall accuracy of the model after the last incremental step. Best result is marked in bold. Results averaged over 3 random runs.

(a) CIFAR100 - B50 Inc5

Methods	CIFAR100 - B50 Inc5			
	ResNet-32		ResNet-18	
	AVG	Last	AVG	Last
FeTrIL [Petit et al., 2023]	64.27	55.19	66.81	58.22
FeTrIL w/ Aux. CIFAR10	64.48	55.85	69.34	60.03
FeTrIL w/ FPCIL-Oracle	**66.24**	**58.88**	**72.81**	**65.20**
FeTrIL w/ FPCIL-Partial (0%)	63.80	55.67	69.27	60.56

(b) CIFAR100 - B0 Inc10

Methods	CIFAR100 - B0 Inc10			
	ResNet-32		ResNet-18	
	AVG	Last	AVG	Last
FeTrIL [Petit et al., 2023]	46.32	29.80	50.50	33.86
FeTrIL w/ Aux. CIFAR10	56.70	39.42	58.56	41.72
FeTrIL w/ FPCIL-Oracle	**62.73**	**48.96**	**68.09**	**55.43**
FeTrIL w/ FPCIL-Partial (0%)	59.68	45.30	62.63	47.71

ples of future classes. "FPCIL var2" also generates synthetic samples of the initial classes and train the feature extractor using synthetic samples of both initial and future classes. In both cases, the classifier is then retrained at the end of the first incremental step using the real samples of the initial classes. Our proposed approach achieves the best performance overall, confirming the importance of using the available real data and not relying only on synthetic data. Interestingly, when the number of real data in the first incremental step is extremely small, replacing them by synthetic samples improves the overall performance while decreasing the final accuracy on the first classes. This may be due to bias induced in the learned representation by the distribution gap between real and synthetic samples being exacerbated in the most extreme settings.

Architecture of feature extractor

Table 5.9 compares the performance of our method for two different feature extractors: the 11.2 million parameters ResNet-18 and the 0.46 million parameters ResNet-32. Due to the addition of the auxiliary dataset to the often limited dataset of the initial step, our method takes advantage of a larger feature extractor and results in a large improvement with the baseline compared to backbones with less parameters. Nonetheless, FPCIL remains

competitive with smaller deep neural networks.

Pre-trained model and finetuning

Table 5.10: Performances on CIFAR100 with the Learning from Half and Learning From Scratch procedures for FeTrIL using a ResNet-18 feature extractor pre-trained on ImageNet. Synthetic samples generated using Stable Diffusion with a guidance scale of 2.0 . Using ResNet-18 as the backbone. Input data are upscaled before being fed to the model in order to match the pre-training configuration. "AVG" is Average Incremental Accuracy (Top-1) and "Last" is the final overall accuracy of the model after the last incremental step. Best result is marked in bold. Results averaged over 3 random runs.

Methods	CIFAR100					
	B50 Inc5		B0 Inc10		B0 Inc5	
	AVG	Last	AVG	Last	AVG	Last
FeTrIL [Petit et al., 2023]	66.73	57.98	51.05	34.43	42.71	26.19
FeTrIL w/ Pre-training	64.16	59.85	71.05	60.97	70.70	59.81
FeTrIL w/ Pre-training + Finetune	76.68	70.55	66.50	54.73	63.05	51.77
FeTrIL w/ FPCIL-Oracle	73.59	65.83	68.10	55.20	63.38	50.54
FeTrIL w/ Pre-training + FPCIL-Oracle	**79.06**	**73.87**	**76.60**	**67.14**	**74.83**	**64.30**

In addition to the main experiments in which the feature extractor is trained from scratch, we also conducted experiments with a pre-trained feature extractor. Using a feature extractor pre-trained on a large external dataset is an effective approach when the low number of classes in the initial incremental step is not enough to train the feature extractor from scratch.

Compared to experiments where the feature extractor is trained from scratch, when using a pre-trained model the initial learning rate for the feature extractor is set to 0.001 and the number of epochs is reduced to 80, other hyper-parameters are kept unchanged. In Table 5.10, the performances of FeTrIL on CIFAR100 using both a ResNet-18 feature extractor trained from scratch during the first incremental step and a ResNet-18 pre-trained on ImageNet. While it appears that slightly finetuning the feature extractor using the data from the initial incremental step is detrimental if the number of classes in too small, it may also further improve the performance when enough classes are available in the first incremental step, especially if there is a large domain gap between the target dataset and the pre-training dataset. In comparison, FPCIL can be combined with pre-training independently of the number of classes in the first incremental step, increasing the average incremental accuracy by $2.38p.p.$ up to $5.55p.p.$.This highlights the versatility of our proposed method FPCIL which can be used either for training the feature extractor from scratch or for finetuning a feature extractor pre-trained on a large dataset, significantly improving the performance of the baseline in both cases.

Comparison with additional baselines

Table 5.11: Performances on CIFAR100 with the Learning from Half and Learning From Scratch procedures for FeTrIL, FeCAM, and Nearest Class Mean (NCM) Classifier. Synthetic samples generated using Stable Diffusion with a guidance scale of 2.0 . Using ResNet-18 as the backbone. "AVG" is Average Incremental Accuracy (Top-1) and "Last" is the final overall accuracy of the model after the last incremental step. Best result is marked in bold. Results averaged over 3 random runs.

Methods	CIFAR100					
	B50 Inc5		B0 Inc10		B0 Inc5	
	AVG	Last	AVG	Last	AVG	Last
NCM	65.24	55.23	40.48	23.37	30.41	15.00
NCM w/ FPCIL-Oracle	71.61	63.21	60.98	46.39	55.69	41.10
Improvement in *p.p.*	+6.37	+7.98	+20.50	+23.02	+25.28	+26.10
FeTrIL [Petit et al., 2023]	66.81	58.22	50.50	33.86	40.80	24.65
FeTrIL w/ FPCIL-Oracle	72.81	65.20	68.09	55.43	62.92	49.67
Improvement in *p.p.*	+6.00	+6.98	+17.59	+21.57	+22.12	+25.02
FeCAM [Goswami et al., 2023]	70.79	63.04	52.85	38.47	43.30	28.88
FeCAM w/ FPCIL-Oracle	**76.77**	**70.49**	**71.14**	**59.95**	**68.31**	**56.05**
Improvement in *p.p.*	+5.98	+7.45	+18.29	+21.48	+25.01	+27.17

Additionally to FeTrIL [Petit et al., 2023], we report in Table 5.11 the performance of our method combined with a Nearest Class Mean (NCM) Classifier and FeCAM [Goswami et al., 2023]. NCM is a simple yet effective method [Janson et al., 2022; Ostapenko et al., 2022] for Class-Incremental Learning with a fixed feature extractor that does not require memory. FeCAM is a recently proposed approach for Exemplar-Free Class-Incremental Learning achieving new state-of-the-art performances. Similarly to FeTrIL, the performance of both NCM and FeCAM is highly dependent on the quality of the features produced by the backbone. It struggles with the Learning From Scratch procedure where the number of classes and data samples used to train the feature extractor is limited. Our method improves the performance of the NCM classifier and FeCAM in every considered setting and achieves the most significant improvements with the Learning From Scratch procedure on CIFAR100 B0 Inc5, reaching a final accuracy up to two times higher than the baseline approach.

Furthermore, in Table 5.12 we compare our approach with the concurrently released method DiffClass [Meng et al., 2024] which leverages the Stable Diffusion model for generating samples from past classes. While our method takes advantage of a larger backbone and can achieve higher performances as detailed in the ablation study, the comparison on CIFAR100 is done using ResNet-32 for the backbone as the results for DiffClass using ResNet-18 are not available. Experimentally, DiffClass achieves higher performance on

Table 5.12: Performances on CIFAR100 and ImageNet-Subset with the Learning From Scratch procedures. Synthetic samples generated using Stable Diffusion with a guidance scale of 2.0 . Using ResNet-32 as the backbone for CIFAR100 and ResNet-18 for ImageNet-Subset. Results marked with † are reported from [Meng et al., 2024]. "AVG" is Average Incremental Accuracy (Top-1) and "Last" is the final overall accuracy of the model after the last incremental step. Best result is marked in bold. Results averaged over 3 random runs.

Methods	CIFAR100				ImageNet-Subset			
	B0 Inc10		B0 Inc5		B0 Inc10		B0 Inc5	
	AVG	Last	AVG	Last	AVG	Last	AVG	Last
DiffClass† [Meng et al., 2024]	**68.05**	**58.40**	**67.10**	**57.11**	73.87	67.02	72.51	68.68
FeTrIL [Petit et al., 2023]	46.32	29.80	37.75	21.26	51.64	34.53	39.64	22.42
FeTrIL w/ FPCIL-Oracle	62.73	48.96	59.89	46.62	74.04	65.65	72.64	64.29
FeCAM [Goswami et al., 2023]	44.12	27.83	35.57	20.04	55.37	40.03	43.48	29.11
FeCAM w/ FPCIL-Oracle	64.47	52.01	63.25	50.00	**77.78**	**69.85**	**77.41**	**69.17**

CIFAR100 but is outperformed by our approach on ImageNet-Subset. It should be noted that Meng et al. [2024] proposed several new components to mitigate the distribution gap between real and synthetic samples [Jodelet et al., 2023]. Some of those contributions, such as a multi-distribution matching technique to finetune the diffusion model and a selective synthetic image augmentation technique, could be adapted and combined with our proposed approach FPCIL to further improve its performance. Finally, our proposed method FPCIL conducts the computationally expensive task of generating synthetic samples using diffusion model only during the first incremental step, before freezing the feature extractor and only updating the classifier afterward. This allows us to potentially conduct the first incremental step on specialised hardware in a data center before deploying the model on devices with limited hardware where each following incremental step could be conducted in a matter of minutes on a CPU using approaches such as FeTrIL or FeCAM. In comparison, DiffClass finetunes the diffusion model, generates synthetic samples of past classes, and finetunes the whole feature extractor at each incremental step, resulting in much larger computational cost and requiring specialised hardware during whole training procedure. This makes our approach preferable if applicable.

5.5 Conclusion

In this work, we introduced a new method for Exemplar-Free Class-Incremental Learning which leverages large pre-trained diffusion model to generate images from future classes. Experimental results highlight how our method can significantly improve the accuracy of state-of-the-art approaches while only modifying the initial step. We found that our method

requires less additional data than traditional methods relying on real curated datasets. While we focused our present study to feature extractors trained from scratch during the first incremental step, in future work, we will further investigate how synthetic images of future classes can be used to adapt a general pre-trained foundation model. Moreover, we will also explore how synthetic data of future classes can be leveraged during the whole training procedure, not only during the first incremental step.

Chapter 6

Discussion

6.1 Relations

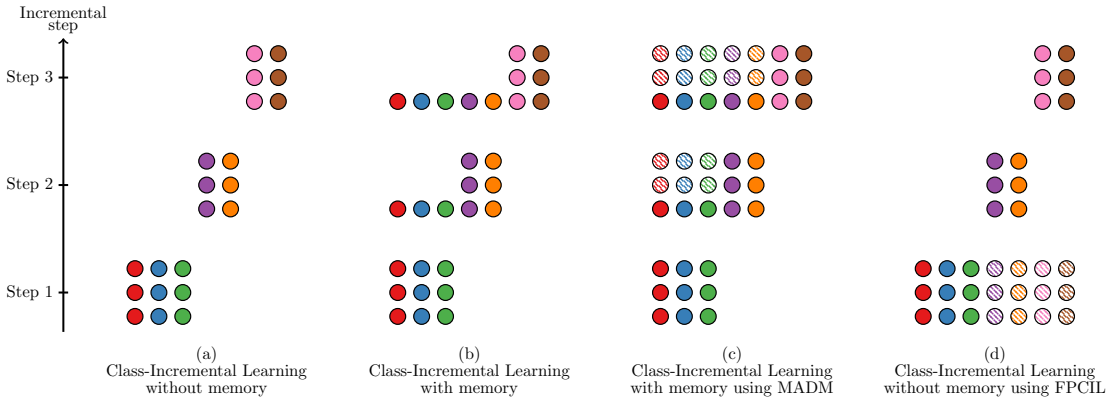


Figure 6.1: Comparison of the different proposed methods from a training data perspective. Circles represent training data, colors represent classes, and plain circles are real data while hatched circles are synthetic data.

The different methods proposed in this thesis are summarized and compared from the perspective of the training data in Figure 6.1:

(a) In the Exemplar-Free Class-Incremental Learning setting, the model is trained solely on the data from the current incremental step $\mathcal{D}_t$ as rehearsal memory can not be used. At each incremental step t, the training dataset only contains real samples of the new classes $\mathcal{Y}_t$. In Chapter 3, we proposed the Relaxed Balanced Softmax Cross Entropy for this challenging setting by extending our previously proposed Balanced Softmax Cross Entropy for Exemplar-Based Class-Incremental Learning.

(b) In the Exemplar-Based Class Incremental Learning setting, the model is trained on the union of data from the current incremental step $\mathcal{D}_t$ and the rehearsal memory $\mathcal{M}$ containing few samples from the previous incremental steps. At each incremental step t, the training dataset contains many real samples of the new classes $\mathcal{Y}_t$ and only a few from the old classes $\bigcup_{i=1}^{t-1} \mathcal{Y}_i$. In Chapter 3, we introduced the Balanced Softmax Cross Entropy to correct the bias toward the new classes induced by the limited size of the rehearsal memory.

(c) In Chapter 4, we proposed to augment the rehearsal memory for Exemplar-Based Class-Incremental Learning using synthetic samples of previously encountered classes generated using pre-trained text-to-image diffusion model. At each incremental step t, the training dataset contains many real samples of the new classes $\mathcal{Y}_t$ in addition to few real samples and many synthetic samples of the old classes $\bigcup_{i=1}^{t-1} \mathcal{Y}_i$.

(d) In Chapter 5, we proposed to prepare the feature extractor better for Exemplar-Free Class-Incremental Learning by using synthetic images of possible future classes generated by a pre-trained text-to-image diffusion model solely during the first incremental step. This allows the feature extractor to produce a representation adapted to discriminate not only the current classes but also the future ones. During the first incremental step only, the training dataset contains many real samples of the new classes $\mathcal{Y}_1$ as well as many synthetic samples of possible future classes $\bigcup_{i=2}^{t} \mathcal{Y}_i$. In the following incremental step $t > 1$, the training dataset only contains real samples of the new classes $\mathcal{Y}_t$, similarly to (a).

In this thesis, we proposed approaches that can be seamlessly combined with state-of-the-art methods for Class-Incremental Learning in order to address the imbalance between new and old classes in different ways. In the next section, we discuss the extent to which our different approaches can be combined together.

6.1.1 Balanced Softmax and MADM

The Balanced Softmax Cross Entropy for Incremental Learning introduced in Chapter 3 and the MADM introduced in Chapter 4 consist in two distinct and orthogonal approaches for mitigating the impact of the limited size of the rehearsal memory.

In Chapter 3, we took inspiration from general imbalance learning and proposed to modify the classification loss using the Balanced Softmax to correct the bias inherent to Class-Incremental Learning during the training procedure. In Chapter 4, similarly to recent works relying on additional data, we propose to augment the rehearsal memory by generating synthetic samples of past classes using a pre-trained text-to-image diffusion model. While not considered in our experiments, these two methods could be combined together but would require some modifications as both methods are related to the composition of the training dataset at each incremental step.

The Balanced Softmax Cross Entropy mitigates the bias toward the new classes by modifying the classification loss using λ parameters equal to the frequency of each class in the training dataset. Combining the Balanced Softmax Cross Entropy with Memory Augmented using Diffusion Model (MADM) raises the question on how to determine the λ parameters. In order to avoid this issue, the straightforward approach consists in restraining the use of the synthetic data of past classes by only using them in distillation loss and not in the classification loss. Although it should result in an improvement of the performances of the model, it would however partially defeat the purpose of MADM which has the advantage of generating labeled synthetic data of past classes that can be used in both the classification and distillation loss. In the case where the synthetic samples of past classes generated by the diffusion model are also used in the classification loss, the λ parameters would have to be appropriately adapted. Only taking into account the number of real samples in the training dataset or considering both the the number of real and synthetic samples in the training dataset for the calculation of the λ parameters may lead to a severe reduction of either the plasticity or the stability. Moreover, as discussed in Section 4.3.3, the use of synthetic samples in the classification loss tends to decrease the performances of the model if the distribution gap between real and synthetic data is not addressed. The Balanced Softmax Cross Entropy may further aggravate this gap by increasing the contribution of the synthetic samples to the loss. This would result in the necessity of using a complementary method for mitigating the effect of this gap, such as the Balanced Finetuning [Castro et al., 2018], which would defeat the purpose of using the Balanced Softmax. Therefore, a more promising solution to combine the two approaches would be to modify the expression of the Balanced Softmax Cross Entropy to handle the real and synthetic samples differently by using different λ parameters depending on the nature of the data. We leave those research directions for future works.

6.1.2 Future-Proofing and Finetuning

In Chapter 5 we propose to synthesize images of possible future classes during the first incremental step in order to better prepare the feature extractor for the future steps. In the experiments, we combined our method with the recent state-of-the-art approach FeTrIL Petit et al. [2023]. As detailed in Chapter 5, FeTrIL relies on a fixed feature extractor making it highly dependent on the quality of the representation learned during the first incremental step, resulting in poor performance when the first incremental step contains only few classes. By combining it with our proposed Future-Proofing approach, the feature extractor has the opportunity to learn a feature representation that discriminates classes more effectively from the future steps by being trained on synthetic samples approximating those future classes. This results in a feature extractor with better generalization capacity and significantly improves the performances of the model. In contrast, not freezing the feature extractor after the first incremental step and continuously finetuning it, would eventually lead the

backbone to learn to extract these discriminative features when encountering the classes in the following incremental steps. This would result in a lower improvement compared to the setting considered in our experiments.

Our future-proofing method could be directly combined with our proposed methods for both Exemplar-Based Class Incremental Learning (Chapters 3 and 4) and Exemplar-Free Class Incremental Learning with a continuously finetuned feature extractor (Chapter 3). Using synthetic samples of future classes during the first incremental step would improve the feature representation of the backbone and result in an improvement of the performances of the model. However, in those settings where the feature extractor is finetuned at each incremental step, having access to synthetic samples of future classes could be useful in the first incremental step, as discussed in Chapter 5, but also in the following ones. In addition to contributing to the training of a better feature extractor, synthetic samples of future classes could also be used for regularization, for example through a distillation loss after the first incremental steps. This would offer various possibilities to better protect and adapt the knowledge through the whole training procedure for both Incremental Learning with and without memory. Moreover, when combined with MADM, synthetic samples of future classes could be used to better prepare the feature extractor and help mitigate the effect of the distribution gap between real and synthetic samples. We leave those research directions for future works.

6.2 Limitations

The limitations of the current study can be summarized as follows:

6.2.1 Limitations of Chapter 3

Through the numerous experiments of Chapter 3, we highlighted that it was possible to adjust the trade-off between stability and plasticity by tuning the λ parameters in the Balanced Softmax Cross Entropy instead of using the frequency of the classes in the training dataset. However, as discussed in Section 3.5, we were not able to propose a method to determine the optimal λ parameters, that would achieve the highest possible performances for either Exemplar-Based Class-Incremental Learning or Exemplar-Free Class-Incremental Learning.

For Exemplar-Based Class-Incremental Learning, we introduced a multiplicative factor α for the λ parameters of the old classes and proposed a meta-learning method to learn it jointly with the model. The improvements resulting from this proposed method compared to using the default Balanced Softmax Cross Entropy (α equals 1) are limited to some settings. Moreover, the meta-learning approach tends to favor stability over plasticity and induces an important increase of the computational cost of the training procedure as discussed in Section 3.3.3. For Exemplar-Free Class-Incremental Learning, our proposed method is to

fix the λ parameters of the old classes to ϵ a small percentage of the number of images per new class instead of zero. While the experiments demonstrated the effectiveness of our proposed method, they also highlighted that this was not the optimal solution and that the performance could be further improved. Moreover, we limited the expression of λ to the constant function for the sake of simplicity.

However, we think that a more general approach could be designed to address both Exemplar-Based and Exemplar-Free Class-Incremental Learning together instead of separately. For instance, it can be reasonably assumed that better performances could be achieved by introducing a more complex function for the λ parameters that would rely not only on the frequency of the classes but also on other variables such as the number of past incremental steps, or the number of classes per incremental step. Instead of using the limited and cumbersome meta-learning approach, this function could be determined by simulating pseudo-incremental steps during the first step, or by estimating it on other datasets similarly to Feillet et al. [2023]. We leave those research directions for future works.

6.2.2 Limitations of Chapters 4 and 5

In Chapter 4 and Chapter 5, we proposed two methods that rely on a pre-trained text-to-image diffusion model for generating synthetic images of past or future classes. Although our proposed methods achieved a significant improvement of the performances compared to the baselines, they are intrinsically limited by the capacity of the pre-trained generative model to produce samples from a similar distribution to the target dataset that we aim to learn.

In our experiments, we only considered the three datasets most commonly used in the Class-Incremental Learning literature: CIFAR100 [Krizhevsky et al., 2009], ImageNet-Subset, and ImageNet [Russakovsky et al., 2015]. Those three datasets only contain natural images which allows our proposed method MADM to achieve high performance by making the most of DALL-E2 [Ramesh et al., 2022] and Stable Diffusion [Rombach et al., 2022]. As both diffusion models have been trained using large scale datasets of natural images and text pairs, they exhibit powerful capabilities for generating natural images, which is favorable for our application. This naturally questions the applicability of our proposed methods datasets to other domains, such as satellite images or the data-scarce domain of medical images.

While the direct application of our proposed methods is limited to domains for which there exists a suitable generative model, we propose different improvements that may be considered in the case such model is not available. A first research direction would be to explore more sophisticated prompting mechanisms, not only by engineering the textual prompts more meticulously, but also by taking advantage of the target dataset itself by using the image variation capacity of models such as DALL-E2 for example. A second research direction would be to adapt widely available generative models to the target dataset

notably by using finetuning as we suggested in Section 4.5. For instance, while both Stable Diffusion and DALL-E2 may struggle to generate medical images due to the significant domain shifts between natural and medical images, Zhang et al. [2024] recently showed that by finetuning these two diffusion models on the target medical dataset, it was possible to generate better medical images which are more domain-similar to the considered target dataset. Moreover, instead of using a frozen pre-trained generative model, it could also be possible to continually train the text-to-image diffusion model at each incremental step on the new data similarly to pseudo-rehearsal-based methods. This would however be beyond the scope of our research and continual learning of generative text-to-image diffusion model remains a challenging problem as they are prone to catastrophic forgetting [Gao and Liu, 2023; Smith et al., 2023a; Masip et al., 2023].

However, it should be noted that concurrent approaches relying on real additional data may be subject to stronger limitations than our proposed approaches. While our method can take advantage of the generalization capacity of those large pre-trained text-to-image generative models and can be improved following the suggestion in the previous paragraph, concurrent approaches relying on real additional data are by definition limited to the data they can access, usually by querying the web.

Chapter 7

Conclusion

7.1 Key Contributions

In this thesis, we aim to develop deep neural networks models able to incrementally learn new classes without forgetting previously encountered ones. To do so, we focused on the importance of balancing old and new classes while training the model. We proposed different approaches that can be combined seamlessly with state-of-the-art approaches for both Exemplar-Based and Exemplar-Free Class-Incremental Learning, in order to improve their performances.

In Chapter 3 we first proposed to use the Balanced Softmax Cross Entropy for Class-Incremental Learning with a rehearsal memory in order to balance the contribution of old and new classes in the classification loss. We further extended this approach to the more challenging setting of Exemplar-Free Class Incremental Learning by proposing the Relaxed Balanced Softmax Cross Entropy. In Chapter 4, we proposed to use a pre-trained text-to-image diffusion model in order to generate synthetic samples of past classes that can be used jointly with the real dataset during training for Exemplar-Based Class-Incremental Learning. Finally in Chapter 5, we showed that it was possible to pre-train the feature extractor for Exemplar-Free Class-Incremental Learning better by generating synthetic samples of future classes, achieving higher performances than traditional pre-training on real samples from unrelated classes.

7.2 Future Research

In this section, we present some future directions for our work in addition to the ones previously discussed in Chapter 6.

7.2.1 Architecture for Incremental Learning

Most approaches for Class-Incremental Learning on large scale datasets rely on architectures for their deep neural network that were designed for the standard, non-incremental, paradigm consisting in training the model on the complete dataset in a single phase. As discussed in Section 2.3.4, the recent breakthrough of the methods relying on dynamically expanding neural networks may emphasize the importance of designing architectures (the components forming the model as well as the architectural organization of those components) and expansion mechanisms specifically for Incremental Learning.

7.2.2 Incremental Learning for Foundation Models

In this thesis, following the current Incremental Learning literature, we restricted our study of Class-Incremental Learning to the setting where only the memory (the number of exemplars that can be stored during each incremental step) is restricted while the computational cost of the training procedure is not. This constraint on the memory has always been justified by privacy concerns or by the limited amount of memory available on embedded devices. However, as discussed by recent works [Prabhu et al., 2023; Verwimp et al., 2024], these limitations may often appear as unrealistic for real-world applications, where the computational cost and the time required by the training procedure are the limiting factors. One possible application of this computationally budgeted Incremental learning is tied to the growing interest in foundation models [Bommasani et al., 2021] for various fields of machine Learning. Those gigantic models are trained in large data-centers where the cost of storing the training data can be considered negligible compared to the millions of GPU hours required for the training of the model itself. However, as they will inevitably become outdated as previously learned concepts evolve and new ones emerge, it is primordial to develop new methods to continuously update those foundation models in a faster and more economical way than by retraining them from scratch regularly.

7.2.3 Adapting models for Incremental Learning

In this thesis, similarly to most of the approaches for Class-Incremental Learning, we considered that the incremental learning model is randomly initialized at the beginning of the initial incremental step. While more challenging, this setting may also appear unrealistic for most real-world applications, especially considering the rapid democratization of foundation models. Incremental Learning with pre-trained models [Zhou et al., 2024] presents new challenges: incremental models have to take advantage from their significant generalization capacity from the first incremental step and adapt their feature representation to the actual future tasks.

Appendix for MADM (Chapter 4)

A.1 Additional results

A.1.1 Final Accuracy for Learning From Half

Table A.1: Final overall accuracy (Top-1) of the model after the last incremental step on CIFAR100, ImageNet-Subset, and ImageNet with the Learning From Half procedure: a base step containing half of the classes followed by 5, 10, and 25 incremental steps, using a growing memory of 20 exemplars per class. Using ResNet-32 for CIFAR100 and ResNet-18 for ImageNet-Subset and ImageNet. Using 500 synthetic samples per class for CIFAR100 and 1300 per class for ImageNet-Subset and ImageNet. Synthetic samples generated using Stable Diffusion with a guidance scale of 2.0 . Results marked with "*" correspond to our own experiments. Results on CIFAR100 and ImageNet-Subset are averaged over 3 random runs. Results on ImageNet are reported as a single run. Best result is marked in bold and second best is underlined.

Methods	CIFAR100			ImageNet-Subset			ImageNet	
	B50 Inc10	B50 Inc5	B50 Inc2	B50 Inc10	B50 Inc5	B50 Inc2	B500 Inc100	B500 Inc50
iCaRL* [Rebuffi et al., 2017]	47.45	43.50	39.51	54.48	49.37	42.69	45.02	40.57
iCaRL w/ MADM (ours)	52.91	51.68	49.62	59.47	57.17	54.62	49.85	47.33
LUCIR* [Hou et al., 2019]	53.91	51.42	47.79	60.87	58.28	50.85	59.48	56.01
LUCIR w/ MADM (ours)	57.00	54.60	51.81	65.04	62.93	60.49	60.82	59.44
DER* [Yan et al., 2021]	61.95	59.30	58.83	66.17	64.67	60.06	<u>63.54</u>	<u>61.77</u>
DER w/ MADM (ours)	63.52	<u>61.79</u>	**61.37**	68.42	67.49	63.95	**63.66**	**62.65**
MEMO* [Zhou et al., 2023c]	59.97	58.52	56.58	68.73	66.11	62.87	60.88	60.51
MEMO NME*	61.34	60.20	58.21	62.21	61.78	61.56	60.01	58.14
MEMO NME w/ MADM (ours)	62.34	61.04	<u>61.01</u>	68.09	67.15	<u>66.14</u>	61.11	59.91
FOSTER* [Wang et al., 2022a]	<u>63.97</u>	60.52	56.44	<u>70.39</u>	<u>68.16</u>	61.96	52.05	49.38
FOSTER w/ MADM (ours)	**64.78**	**63.47**	58.91	**72.49**	**71.07**	**67.81**	53.79	51.31

As a complement to Table 4.1, we report in Table A.1 the final overall accuracy (Top-1) of the model after the last incremental step on CIFAR100, ImageNet-Subset, and ImageNet with the Learning From Half procedure.

This confirms that for all the considered baselines, using our proposed approach MADM leads to a significant improvement of the performances in the Training From Half procedure. It increase the final accuracy of the model by up to $3.89p.p.$ for DER, up to $4.43p.p.$ for MEMO, up to $5.85p.p.$ for FOSTER, up to $9.64p.p.$ for LUCIR and up to $11.93p.p.$ for iCaRL, depending on the dataset and the number of incremental steps. The improvement of the final accuracy of the model is more significant for the most difficult settings, with a numerous incremental steps.

A.1.2 Dual branch FOSTER

Table A.2: Average incremental accuracy (Top-1) on CIFAR100, and ImageNet-Subset with the Learning From Half procedure, using a growing memory of 20 exemplars per class. Results for DER are reported from Yan et al. [2021]. Using 500 synthetic samples per class for CIFAR100 and 1300 per class for ImageNet-Subset. Synthetic samples generated using Stable Diffusion with a guidance scale of 2.0 . Results marked with "*" correspond to our own experiments. Results on CIFAR100 and ImageNet-Subset are averaged over 3 random runs. Best result is marked in bold and second best is underlined.

Methods	CIFAR100			ImageNet-Subset		
	B50 Inc10	B50 Inc5	B50 Inc2	B50 Inc10	B50 Inc5	B50 Inc2
DER [Yan et al., 2021]	68.52	67.09	-	-	**78.20**	-
FOSTER* [Wang et al., 2022a]	71.17	68.89	65.07	76.09	75.05	70.96
FOSTER w/ MADM (ours)	<u>72.18</u>	<u>70.88</u>	<u>68.06</u>	77.13	76.77	<u>75.50</u>
FOSTER-B4* [Wang et al., 2022a]	72.08	69.32	65.41	<u>77.48</u>	76.11	71.40
FOSTER-B4 w/ MADM (ours)	**73.28**	**71.43**	**68.36**	**78.48**	<u>77.71</u>	**75.97**

In our main experiments, to fairly compare our proposed method with other methods, we only evaluated the accuracy of FOSTER after performing the feature compression. To better compare with dynamic-networks-based methods such as DER Yan et al. [2021] whose number of parameters is growing at each incremental steps, we report in Table A.2 the accuracy of the dual branch FOSTER model before the compression, denoted FOSTER-B4.

Experiments show that, if the memory and computational budget of the system allow it, using the dual branch FOSTER for inference with our method further improve the accuracy. On CIFAR100, FOSTER combined with MADM achieves higher performances than DER for both single and dual branch evaluation. Furthermore, on ImageNet-Subset with 10 incremental steps, FOSTER-B4 combined with MADM achieves a Top-1 Average

incremental accuracy 0.49*p.p.* lower and a Top-5 Average incremental accuracy 0.50*p.p.* higher than DER while requiring about five times less parameters.

A.1.3 Additional results for CIFAR-100 B50 Inc10

Table A.3: Performances on CIFAR100 with the Learning From Half procedure: a base step containing half of the classes followed by 5 incremental steps depending on the number of exemplars saved in the growing memory for each class. Using 500 synthetic samples per class. Synthetic samples generated using Stable Diffusion with a guidance scale of 2.0 . Results marked with "*" correspond to our own experiments and results marked with "†" are reported from [Liu et al., 2024]. "Average" is Average Incremental Accuracy (Top-1) and "Last" is the final overall accuracy of the model after the last incremental step. Results averaged over 3 random runs.

Methods	CIFAR100 - B50 Inc10							
	5 exemplars/class		10 exemplars/class		20 exemplars/class		50 exemplars/class	
	Average	Last	Average	Last	Average	Last	Average	Last
iCaRL* [Rebuffi et al., 2017]	43.40	31.90	52.30	40.11	57.68	47.45	62.09	54.23
w/ PlaceboCIL† [Liu et al., 2024]	51.55	39.35	59.11	46.42	61.24	51.47	-	-
w/ MADM (ours)	55.36	43.13	58.53	48.33	62.01	52.91	65.53	58.46
LUCIR* [Hou et al., 2019]	53.14	41.52	60.77	49.65	63.37	53.91	65.63	57.58
w/ PlaceboCIL† [Liu et al., 2024]	62.74	53.25	64.79	55.44	65.28	56.23	-	-
w/ MADM (ours)	62.90	50.94	64.47	54.53	65.77	57.00	67.21	59.73
FOSTER* [Wang et al., 2022a]	56.94	43.74	63.09	54.34	71.17	63.97	70.00	64.10
w/ PlaceboCIL† [Liu et al., 2024]	62.78	50.72	65.12	54.81	71.97	64.43	-	-
w/ MADM (ours)	57.76	44.32	63.26	54.32	72.18	64.78	71.66	66.04

Following Liu et al. [2024], we report in Table A.3 additional results for the different baseline methods on CIFAR100 with a base step containing half of the classes followed by 5 incremental steps depending on the number of exemplars saved in memory for each class.

When FOSTER Wang et al. [2022a] is used with a really small replay memory, we observed that the improvement resulting from using our proposed method MADM is limited. We suppose that this is due to the logits alignment loss used by the authors that too strongly limits the plasticity of the model and restricts our proposed method. By tuning the hyperparameters of the logits aligment loss, it should be possible to further improve the performance of FOSTER when combined with our proposed method.

A.1.4 Comparison with PlaceboCIL on ImageNet

In addition to the results on CIFAR100 and ImageNet-Subset var. reported in Table 4.3, we also report in Table A.4 the results for ImageNet with the Learning From Half procedure. However, as the difference in the reproduced accuracy on ImageNet is more than 5 percentage points, it may be difficult to make a direct comparison between the two methods on this

dataset. It should be noted that while being significantly lower than the ones reproduced by Liu et al. [2024], our reproduced accuracies on ImageNet for FOSTER [Wang et al., 2022a] is similar to the results reproduced by Zhou et al. [2023b] in their recent review.

Table A.4: Average incremental accuracy (Top-1) on ImageNet with the Learning From Half procedure: a base step containing half of the classes followed by 5, and 10 incremental steps, using a growing memory of 20 exemplars per class. Using 1300 synthetic samples per class. Synthetic samples generated using Stable Diffusion with a guidance scale of 2.0 . Results marked with "*" correspond to our own experiments and results marked with "†" are reported from [Liu et al., 2024].

Methods	ImageNet	
	B500 Inc100	B500 Inc50
FOSTER† [Wang et al., 2022a]	69.32	66.07
FOSTER w/ PlaceboCIL† [Liu et al., 2024]	71.02	68.82
Improvement in *p.p.*	+1.70	+2.75
FOSTER* [Wang et al., 2022a]	62.18	60.83
FOSTER w/ MADM (ours)	63.19	62.09
Improvement in *p.p.*	+1.01	+1.26

Bibliography

Mohamed Abdelsalam, Mojtaba Faramarzi, Shagun Sodhani, and Sarath Chandar. Iirc: Incremental implicitly-refined classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11038–11047, June 2021.

Hongjoon Ahn, Jihwan Kwak, Subin Lim, Hyeonsu Bang, Hyojun Kim, and Taesup Moon. Ss-il: Separated softmax for incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 844–853, October 2021.

Giovanni Bellitto, Matteo Pennisi, Simone Palazzo, Lorenzo Bonicelli, Matteo Boschini, and Simone Calderara. Effects of auxiliary knowledge on continual learning. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 1357–1363. IEEE, 2022.

Eden Belouadah and Adrian Popescu. Deesil: Deep-shallow incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, September 2018.

Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Eden Belouadah, Adrian Popescu, and Ioannis Kanellos. A comprehensive study of class incremental learning algorithms for visual tasks. *Neural Networks*, 135:38–54, 2021. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2020.12.003. URL `https://www.sciencedirect.com/science/article/pii/S0893608020304202`.

Victor Besnier, Himalaya Jain, Andrei Bursuc, Matthieu Cord, and Patrick Pérez. This dataset does not exist: training models from generated images. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2020.

Magdalena Biesialska, Katarzyna Biesialska, and Marta R. Costa-jussà. Continual lifelong learning in natural language processing: A survey. In Donia Scott, Nuria Bel, and

Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6523–6541, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.574. URL https://aclanthology.org/2020.coling-main.574.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen A. Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, O. Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir P. Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Benjamin Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, J. F. Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert Reich, Hongyu Ren, Frieda Rong, Yusuf H. Roohani, Camilo Ruiz, Jack Ryan, Christopher R'e, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishna Parasuram Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei A. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models. *ArXiv*, 2021. URL https://crfm.stanford.edu/assets/report.pdf.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

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

Lucas Caccia, Eugene Belilovsky, Massimo Caccia, and Joelle Pineau. Online learned con-

tinual compression with adaptive quantization modules. In *International Conference on Machine Learning*, pages 1240–1250. PMLR, 2020.

Francisco M. Castro, Manuel J. Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 241–257, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01258-8.

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/CVF Conference on Computer Vision and Pattern Recognition*, pages 9233–9242, 2020.

Sungmin Cha, Jihwan Kwak, Dongsub Shim, Hyunwoo Kim, Moontae Lee, Honglak Lee, and Taesup Moon. Towards more objective evaluation of class incremental learning: Representation learning perspective. *arXiv preprint arXiv:2206.08101*, 2022.

Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. *Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence*, page 556–572. Springer International Publishing, 2018. ISBN 9783030012526. doi: 10.1007/978-3-030-01252-6_33. URL `http://dx.doi.org/10.1007/978-3-030-01252-6_33`.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K. Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning, 2019.

Jingfan Chen, Yuxi Wang, Pengfei Wang, Xiao Chen, Zhaoxiang Zhang, Zhen Lei, and Qing Li. Diffusepast: Diffusion-based generative replay for class incremental semantic segmentation, 2023.

Wei Cong, Yang Cong, Jiahua Dong, Gan Sun, and Henghui Ding. Gradient-semantic compensation for incremental semantic segmentation. *IEEE Transactions on Multimedia*, 26:5561–5574, 2024. ISSN 1941-0077. doi: 10.1109/tmm.2023.3336243. URL `http://dx.doi.org/10.1109/TMM.2023.3336243`.

Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):10850–10869, 2023. doi: 10.1109/TPAMI.2023.3261988.

Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019. doi: 10.1109/cvpr.2019.00528. URL `http://dx.doi.org/10.1109/CVPR.2019.00528`.

Jiahua Dong, Lixu Wang, Zhen Fang, Gan Sun, Shichao Xu, Xiao Wang, and Qi Zhu. Federated class-incremental learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2022. doi: 10.1109/cvpr52688.2022.00992. URL `http://dx.doi.org/10.1109/CVPR52688.2022.00992`.

Jiahua Dong, Wenqi Liang, Yang Cong, and Gan Sun. Heterogeneous forgetting compensation for class-incremental learning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, October 2023a. doi: 10.1109/iccv51070.2023.01078. URL `http://dx.doi.org/10.1109/ICCV51070.2023.01078`.

Jiahua Dong, Duzhen Zhang, Yang Cong, Wei Cong, Henghui Ding, and Dengxin Dai. Federated incremental semantic segmentation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2023b. doi: 10.1109/cvpr52729.2023.00383. URL `http://dx.doi.org/10.1109/CVPR52729.2023.00383`.

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

Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, part XX 16*, pages 86–102. Springer, 2020.

Arthur Douillard, Yifu Chen, Arnaud Dapogny, and Matthieu Cord. Plop: Learning without forgetting for continual semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4040–4050, June 2021a.

Arthur Douillard, Eduardo Valle, Charles Ollion, Thomas Robert, and Matthieu Cord. Insights from the future for continual learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, June 2021b. doi: 10.1109/cvprw53098.2021.00387. URL `http://dx.doi.org/10.1109/CVPRW53098.2021.00387`.

Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9285–9295, June 2022.

Xuefeng Du, Yiyou Sun, Jerry Zhu, and Yixuan Li. Dream the impossible: Outlier imagination with diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

Ruxiao Duan, Yaoyao Liu, Jieneng Chen, Adam Kortylewski, and Alan Yuille. Prompt-based exemplar super-compression and regeneration for class-incremental learning, 2023.

Lijie Fan, Kaifeng Chen, Dilip Krishnan, Dina Katabi, Phillip Isola, and Yonglong Tian. Scaling laws of synthetic images for model training ... for now. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7382–7392, June 2024a.

Lijie Fan, Dilip Krishnan, Phillip Isola, Dina Katabi, and Yonglong Tian. Improving clip training with language rewrites. *Advances in Neural Information Processing Systems*, 36, 2024b.

Falih Gozi Febrinanto, Feng Xia, Kristen Moore, Chandra Thapa, and Charu Aggarwal. Graph lifelong learning: A survey. *IEEE Computational Intelligence Magazine*, 18(1): 32–51, 2023.

Eva Feillet, Grégoire Petit, Adrian Popescu, Marina Reyboz, and Céline Hudelot. Advisil - a class-incremental learning advisor. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2400–2409, January 2023.

Tao Feng, Mang Wang, and Hangjie Yuan. Overcoming catastrophic forgetting in incremental object detection via elastic response distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9427–9436, June 2022.

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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9621–9630, June 2022.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

Rui Gao and Weiwei Liu. Ddgr: Continual learning with deep diffusion-based generative replay. In *International Conference on Machine Learning*, pages 10744–10763. PMLR, 2023.

Saurabh Garg, Mehrdad Farajtabar, Hadi Pouransari, Raviteja Vemulapalli, Sachin Mehta, Oncel Tuzel, Vaishaal Shankar, and Fartash Faghri. Tic-clip: Continual training of clip models. *arXiv preprint arXiv:2310.16226*, 2023.

Jacob Goldberger, Geoffrey E Hinton, Sam Roweis, and Russ R Salakhutdinov. Neighbourhood components analysis. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2005.

Dipam Goswami, Yuyang Liu, Bartłomiej Twardowski, and Joost van de Weijer. Fecam: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Edward Grefenstette, Brandon Amos, Denis Yarats, Phu Mon Htut, Artem Molchanov, Franziska Meier, Douwe Kiela, Kyunghyun Cho, and Soumith Chintala. Generalized inner loop meta-learning. *arXiv preprint arXiv:1910.01727*, 2019.

Hasan Abed Al Kader Hammoud, Hani Itani, Fabio Pizzati, Philip Torr, Adel Bibi, and Bernard Ghanem. Synthclip: Are we ready for a fully synthetic clip training? *arXiv preprint arXiv:2402.01832*, 2024.

Tyler L Hayes and Christopher Kanan. Lifelong machine learning with deep streaming linear discriminant analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 220–221, 2020.

Tyler L. Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. *REMIND Your Neural Network to Prevent Catastrophic Forgetting*, page 466–483. Springer International Publishing, 2020. ISBN 9783030585983. doi: 10.1007/978-3-030-58598-3_28. URL http://dx.doi.org/10.1007/978-3-030-58598-3_28.

Chen He, Ruiping Wang, Shiguang Shan, and Xilin Chen. Exemplar-supported generative reproduction for class incremental learning. In *BMVC*, page 98, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016. doi: 10.1109/cvpr.2016.90. URL http://dx.doi.org/10.1109/cvpr.2016.90.

Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? *arXiv preprint arXiv:2210.07574*, 2022.

Hamed Hemati, Andrea Cossu, Antonio Carta, Julio Hurtado, Lorenzo Pellegrini, Davide Bacciu, Vincenzo Lomonaco, and Damian Borth. Class-incremental learning with repetition. In *Conference on Lifelong Learning Agents*, pages 437–455. PMLR, 2023.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL https://openreview.net/forum?id=qw8AKxfYbI.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Yen-Chang Hsu, Yen-Cheng Liu, Anita Ramasamy, and Zsolt Kira. Re-evaluating continual learning scenarios: A categorization and case for strong baselines. *arXiv preprint arXiv:1810.12488*, 2018.

Xinting Hu, Kaihua Tang, Chunyan Miao, Xian-Sheng Hua, and Hanwang Zhang. Distilling causal effect of data in class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3957–3966, June 2021.

Ahmet Iscen, Jeffrey Zhang, Svetlana Lazebnik, and Cordelia Schmid. *Memory-Efficient Incremental Learning Through Feature Adaptation*, page 699–715. Springer International Publishing, 2020. ISBN 9783030585174. doi: 10.1007/978-3-030-58517-4_41. URL http://dx.doi.org/10.1007/978-3-030-58517-4_41.

Ali Jahanian, Xavier Puig, Yonglong Tian, and Phillip Isola. Generative models as a data source for multiview representation learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=qhAeZjs7dCL.

Paul Janson, Wenxuan Zhang, Rahaf Aljundi, and Mohamed Elhoseiny. A simple baseline that questions the use of pretrained-models in continual learning. In *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2022. URL https://openreview.net/forum?id=dnVNYctP3S.

Khurram Javed and Faisal Shafait. Revisiting distillation and incremental classifier learning. In C.V. Jawahar, Hongdong Li, Greg Mori, and Konrad Schindler, editors, *Computer Vision – ACCV 2018*, pages 3–17, Cham, 2019. Springer International Publishing. ISBN 978-3-030-20876-9.

Quentin Jodelet, Xin Liu, and Tsuyoshi Murata. Balanced softmax cross-entropy for incremental learning. In Igor Farkaš, Paolo Masulli, Sebastian Otte, and Stefan Wermter, editors, *Artificial Neural Networks and Machine Learning – ICANN 2021*, pages 385–396, Cham, 2021. Springer International Publishing. ISBN 978-3-030-86340-1.

Quentin Jodelet, Xin Liu, and Tsuyoshi Murata. Balanced softmax cross-entropy for incremental learning with and without memory. *Computer Vision and Image Understanding*, 225:103582, 2022. ISSN 1077-3142. doi: https://doi.org/10.1016/j.cviu.2022.103582. URL `https://www.sciencedirect.com/science/article/pii/S1077314222001606`.

Quentin Jodelet, Xin Liu, Yin Jun Phua, and Tsuyoshi Murata. Class-incremental learning using diffusion model for distillation and replay. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 3425–3433, October 2023.

Quentin Jodelet, Xin Liu, Yin Jun Phua, and Tsuyoshi Murata. Memory augmented using diffusion model for class-incremental learning. *Submitted for review*, 2024a.

Quentin Jodelet, Xin Liu, Yin Jun Phua, and Tsuyoshi Murata. Future-proofing class incremental learning. *arXiv preprint arXiv:2404.03200*, 2024b.

Minsoo Kang, Jaeyoo Park, and Bohyung Han. Class-incremental learning by knowledge distillation with adaptive feature consolidation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16071–16080, 2022.

Ronald Kemker and Christopher Kanan. Fearnet: Brain-inspired model for incremental learning. *arXiv preprint arXiv:1711.10563*, 2017.

Salman H Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A Sohel, and Roberto Togneri. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems*, 29(8):3573–3587, 2017.

Dongwan Kim and Bohyung Han. On the stability-plasticity dilemma of class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20196–20204, 2023.

Junsu Kim, Hoseong Cho, Jihyeon Kim, Yihalem Yimolal Tiruneh, and Seungryul Baek. Sddgr: Stable diffusion-based deep generative replay for class incremental object detection, 2024.

Sanghwan Kim, Lorenzo Noci, Antonio Orvieto, and Thomas Hofmann. Achieving a better stability-plasticity trade-off via auxiliary networks in continual learning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2023. doi: 10.1109/cvpr52729.2023.01148. URL `http://dx.doi.org/10.1109/CVPR52729.2023.01148`.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the Na-*

tional Academy of Sciences, 114(13):3521–3526, March 2017. ISSN 1091-6490. doi: 10.1073/pnas.1611835114. URL `http://dx.doi.org/10.1073/pnas.1611835114`.

Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

Alexis Lechat, Stéphane Herbin, and Frédéric Jurie. Pseudo-labeling for class incremental learning. In *BMVC 2021: The British Machine Vision Conference*, 2021.

Kibok Lee, Kimin Lee, Jinwoo Shin, and Honglak Lee. Overcoming catastrophic forgetting with unlabeled data in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 312–321, 2019.

Cheng-Hsun Lei, Yi-Hsin Chen, Wen-Hsiao Peng, and Wei-Chen Chiu. Class-incremental learning with rectified feature-graph preservation. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023a.

Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *International conference on machine learning*, pages 3925–3934. PMLR, 2019.

Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22511–22521, 2023b.

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

Zhuoyun Li, Changhong Zhong, Sijia Liu, Ruixuan Wang, and Wei-Shi Zheng. Preserving earlier knowledge in continual learning with the help of all previous feature extractors, 2021.

Xialei Liu, Chenshen Wu, Mikel Menta, Luis Herranz, Bogdan Raducanu, Andrew D Bagdanov, Shangling Jui, and Joost van de Weijer. Generative feature replay for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 226–227, 2020a.

Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 12245–12254, 2020b.

Yaoyao Liu, Bernt Schiele, and Qianru Sun. Rmm: Reinforced memory management for class-incremental learning. *Advances in Neural Information Processing Systems*, 34:3478–3490, 2021.

Yaoyao Liu, Yingying Li, Bernt Schiele, and Qianru Sun. Wakening past concepts without past data: Class-incremental learning from online placebos. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, January 2024. doi: 10.1109/wacv57701.2024.00222. URL http://dx.doi.org/10.1109/WACV57701.2024.00222.

Vincenzo Lomonaco, Lorenzo Pellegrini, Pau Rodriguez, Massimo Caccia, Qi She, Yu Chen, Quentin Jodelet, Ruiping Wang, Zheda Mai, David Vazquez, German I. Parisi, Nikhil Churamani, Marc Pickett, Issam Laradji, and Davide Maltoni. Cvpr 2020 continual learning in computer vision competition: Approaches, results, current challenges and future directions. *Artificial Intelligence*, 303:103635, February 2022. ISSN 0004-3702. doi: 10.1016/j.artint.2021.103635. URL http://dx.doi.org/10.1016/j.artint.2021.103635.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Zilin Luo, Yaoyao Liu, Bernt Schiele, and Qianru Sun. Class-incremental exemplar compression for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11371–11380, June 2023.

Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. Representational continuity for unsupervised continual learning. *arXiv preprint arXiv:2110.06976*, 2021.

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

Sergi Masip, Pau Rodriguez, Tinne Tuytelaars, and Gido M. van de Ven. Continual learning of diffusion models with generative distillation, 2023.

Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109 – 165. Academic Press, 1989. doi: https://doi.org/10.1016/S0079-7421(08)60536-8. URL http://www.sciencedirect.com/science/article/pii/S0079742108605368.

Zichong Meng, Jie Zhang, Changdi Yang, Zheng Zhan, Pu Zhao, and Yanzhi WAng. Diffclass: Diffusion-based class incremental learning. *arXiv preprint arXiv:2403.05016*, 2024.

George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, nov 1995. ISSN 0001-0782. doi: 10.1145/219717.219748. URL https://doi.org/10.1145/219717.219748.

Shervin Minaee, Yuri Y. Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–1, 2021. ISSN 1939-3539. doi: 10.1109/tpami.2021.3059968. URL http://dx.doi.org/10.1109/TPAMI.2021.3059968.

Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. doi: 10.1109/iccv.2017.47. URL http://dx.doi.org/10.1109/ICCV.2017.47.

Quang Nguyen, Truong Vu, Anh Tran, and Khoi Nguyen. Dataset diffusion: Diffusion-based synthetic data generation for pixel-level semantic segmentation. *Advances in Neural Information Processing Systems*, 36, 2024.

Zixuan Ni, Longhui Wei, Siliang Tang, Yueting Zhuang, and Qi Tian. Continual vision-language representation learning with off-diagonal information. In *International Conference on Machine Learning*, pages 26129–26149. PMLR, 2023.

Oleksiy Ostapenko, Timothee Lesort, Pau Rodríguez, Md Rifat Arefin, Arthur Douillard, Irina Rish, and Laurent Charlin. Continual learning with foundation models: An empirical study of latent replay, 2022.

Utku Ozbulak, Hyun Jung Lee, Beril Boga, Esla Timothy Anzaku, Ho min Park, Arnout Van Messem, Wesley De Neve, and Joris Vankerschaver. Know your self-supervised learning: A survey on image-based generative and discriminative training. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=Ma25S4ludQ. Survey Certification.

Shaoning Pang, S. Ozawa, and N. Kasabov. Incremental linear discriminant analysis for classification of data streams. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(5):905–914, 2005. doi: 10.1109/TSMCB.2005.847744.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019.

Jaeyoo Park, Minsoo Kang, and Bohyung Han. Class-incremental learning for action recognition in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13698–13707, October 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

Juan-Manuel Perez-Rua, Xiatian Zhu, Timothy M. Hospedales, and Tao Xiang. Incremental few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Federico Pernici, Matteo Bruni, Claudio Baecchi, Francesco Turchini, and Alberto Del Bimbo. Class-incremental learning with pre-allocated fixed classifiers. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6259–6266. IEEE, 2021.

Grégoire Petit, Adrian Popescu, Hugo Schindler, David Picard, and Bertrand Delezoide. Fetril: Feature translation for exemplar-free class-incremental learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3911–3920, January 2023.

Grégoire Petit, Michaël Soumm, Eva Feillet, Adrian Popescu, Bertrand Delezoide, David Picard, and Céline Hudelot. An analysis of initial training strategies for exemplar-free class-incremental learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1837–1847, January 2024.

Ameya Prabhu, Hasan Abed Al Kader Hammoud, Puneet K Dokania, Philip HS Torr, Ser-Nam Lim, Bernard Ghanem, and Adel Bibi. Computationally budgeted continual learning: What does matter? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3698–3707, 2023.

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

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022.

Roger Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285, 1990.

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 (CVPR)*, July 2017.

Jiawei Ren, Cunjun Yu, shunan sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4175–4186. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/2ba61cc3a8f44143e1f2f13b2b729ab3-Paper.pdf`.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

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.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

Mert Bülent Sarıyıldız, Karteek Alahari, Diane Larlus, and Yannis Kalantidis. Fake it till you make it: Learning transferable representations from synthetic imagenet clones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8011–8021, June 2023.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

Joonghyuk Shin, Minguk Kang, and Jaesik Park. Fill-Up: Balancing Long-Tailed Data with Generative Models. *arXiv preprint arXiv:2306.07200*, 2023.

Christian Simon, Piotr Koniusz, and Mehrtash Harandi. On learning the geodesic path for incremental learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 1591–1600, 2021.

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*, pages 9374–9384, 2021.

James Seale Smith, Yen-Chang Hsu, Lingyu Zhang, Ting Hua, Zsolt Kira, Yilin Shen, and Hongxia Jin. Continual diffusion: Continual customization of text-to-image diffusion with c-lora. *arXiv preprint arXiv:2304.06027*, 2023a.

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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2023b. doi: 10.1109/cvpr52729.2023.01146. URL `http://dx.doi.org/10.1109/CVPR52729.2023.01146`.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015.

Wenju Sun, Qingyong Li, Jing Zhang, Danyu Wang, Wen Wang, and YangLi ao Geng. Exemplar-free class incremental learning via discriminative and comparable parallel one-class classifiers. *Pattern Recognition*, 140:109561, 2023. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2023.109561. URL `https://www.sciencedirect.com/science/article/pii/S0031320323002613`.

Yu-Ming Tang, Yi-Xing Peng, and Wei-Shi Zheng. When prompt-based incremental learning does not meet strong pretraining. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1706–1716, 2023.

Xiaoyu Tao, Xinyuan Chang, Xiaopeng Hong, Xing Wei, and Yihong Gong. Topology-preserving class-incremental learning. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 254–270, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58529-7.

Yonglong Tian, Lijie Fan, Phillip Isola, Huiwen Chang, and Dilip Krishnan. Stablerep: Synthetic images from text-to-image models make strong visual representation learners, 2023.

Antonio Torralba, Rob Fergus, and William T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008. doi: 10.1109/TPAMI.2008.128.

Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

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

Eli Verwimp, Rahaf Aljundi, Shai Ben-David, Matthias Bethge, Andrea Cossu, Alexander Gepperth, Tyler L. Hayes, Eyke Hüllermeier, Christopher Kanan, Dhireesha Kudithipudi, Christoph H. Lampert, Martin Mundt, Razvan Pascanu, Adrian Popescu, Andreas S. Tolias, Joost van de Weijer, Bing Liu, Vincenzo Lomonaco, Tinne Tuytelaars, and Gido M van de Ven. Continual learning: Applications and the road forward. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=axBIMcGZn9.

Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57, 1985.

Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Foster: Feature boosting and compression for class-incremental learning. In *European conference on computer vision*, pages 398–414. Springer, 2022a.

Liyuan Wang, Kuo Yang, Chongxuan Li, Lanqing Hong, Zhenguo Li, and Jun Zhu. Ordisco: Effective and efficient usage of incremental unlabeled data for semi-supervised continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5383–5392, 2021.

Liyuan Wang, Xingxing Zhang, Kuo Yang, Longhui Yu, Chongxuan Li, Lanqing HONG, Shifeng Zhang, Zhenguo Li, Yi Zhong, and Jun Zhu. Memory replay with data compression for continual learning. In *International Conference on Learning Representations*, 2022b. URL https://openreview.net/forum?id=a7H7OucbWaU.

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 *European Conference on Computer Vision*, pages 631–648. Springer, 2022c.

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 *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2022d. doi: 10.1109/cvpr52688.2022.00024. URL http://dx.doi.org/10.1109/CVPR52688.2022.00024.

Max Welling. Herding dynamical weights to learn. In *Proceedings of the 26th annual international conference on machine learning*, pages 1121–1128, 2009.

Chenshen Wu, Luis Herranz, Xialei Liu, yaxing wang, Joost van de Weijer, and Bogdan Raducanu. Memory replay gans: Learning to generate new categories without forgetting. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and

R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/a57e8915461b83adefb011530b711704-Paper.pdf.

Guile Wu, Shaogang Gong, and Pan Li. Striking a balance between stability and plasticity for class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1124–1133, October 2021a.

Tongtong Wu, Massimo Caccia, Zhuang Li, Yuan-Fang Li, Guilin Qi, and Gholamreza Haffari. Pretrained language model in continual learning: A comparative study. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=figzpGMrdD.

Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 374–382, 2019.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, January 2021b. ISSN 2162-2388. doi: 10.1109/tnnls.2020.2978386. URL http://dx.doi.org/10.1109/TNNLS.2020.2978386.

Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3014–3023, 2021.

Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724, 2020.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Sk7KsfW0-.

Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing [review article]. *IEEE Computational Intelligence Magazine*, 13(3):55–75, August 2018. ISSN 1556-6048. doi: 10.1109/mci.2018.2840738. URL http://dx.doi.org/10.1109/MCI.2018.2840738.

Lu Yu, Bartlomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Zhuoran Yu, Chenchen Zhu, Sean Culatana, Raghuraman Krishnamoorthi, Fanyi Xiao, and Yong Jae Lee. Diversify, don't fine-tune: Scaling up visual recognition training with synthetic images. *arXiv preprint arXiv:2312.02253*, 2023.

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

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2017.

Junting Zhang, Jie Zhang, Shalini Ghosh, Dawei Li, Serafettin Tasci, Larry Heck, Heming Zhang, and C-C Jay Kuo. Class-incremental learning via deep model consolidation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1131–1140, 2020.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, October 2023. doi: 10.1109/iccv51070.2023.00355. URL `http://dx.doi.org/10.1109/ICCV51070.2023.00355`.

Yifan Zhang, Daquan Zhou, Bryan Hooi, Kai Wang, and Jiashi Feng. Expanding small-scale datasets with guided imagination. *Advances in Neural Information Processing Systems*, 36, 2024.

Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2020. doi: 10.1109/cvpr42600.2020.01322. URL `http://dx.doi.org/10.1109/CVPR42600.2020.01322`.

Hanbin Zhao, Hui Wang, Yongjian Fu, Fei Wu, and Xi Li. Memory-efficient class-incremental learning for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10):5966–5977, October 2022. ISSN 2162-2388. doi: 10.1109/tnnls.2021.3072041. URL `http://dx.doi.org/10.1109/TNNLS.2021.3072041`.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2023.

Zhong-Qiu Zhao, Peng Zheng, Shou-Tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30 (11):3212–3232, November 2019. ISSN 2162-2388. doi: 10.1109/tnnls.2018.2876865. URL `http://dx.doi.org/10.1109/TNNLS.2018.2876865`.

Tianlun Zheng, Zhineng Chen, Bingchen Huang, Wei Zhang, and Yu-Gang Jiang. Mrn: Multiplexed routing network for incremental multilingual text recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18644–18653, 2023.

Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, Liang Ma, Shiliang Pu, and De-Chuan Zhan. Forward compatible few-shot class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9046–9056, 2022.

Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, and De-Chuan Zhan. Pycil: a python toolbox for class-incremental learning. *SCIENCE CHINA Information Sciences*, 66(9):197101–, 2023a. doi: https://doi.org/10.1007/s11432-022-3600-y.

Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Deep class-incremental learning: A survey, 2023b.

Da-Wei Zhou, Qi-Wei Wang, Han-Jia Ye, and De-Chuan Zhan. A model or 603 exemplars: Towards memory-efficient class-incremental learning. In *ICLR*, 2023c.

Da-Wei Zhou, Hai-Long Sun, Jingyi Ning, Han-Jia Ye, and De-Chuan Zhan. Continual learning with pre-trained models: A survey. *arXiv preprint arXiv:2401.16386*, 2024.

Peng Zhou, Long Mai, Jianming Zhang, Ning Xu, Zuxuan Wu, and Larry S Davis. M2kd: Multi-model and multi-level knowledge distillation for incremental learning. *arXiv preprint arXiv:1904.01769*, 2019.

Yongchao Zhou, Hshmat Sahak, and Jimmy Ba. Training on thin air: Improve image classification with generated data. *arXiv preprint arXiv:2305.15316*, 2023d.

Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Cheng-lin Liu. Class-incremental learning via dual augmentation. *Advances in Neural Information Processing Systems*, 34:14306–14318, 2021a.

Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5871–5880, June 2021b.

Kai Zhu, Wei Zhai, Yang Cao, Jiebo Luo, and Zheng-Jun Zha. Self-sustaining representation expansion for non-exemplar class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9296–9305, 2022.