Warsaw University of Technology

DISCIPLINE OF SCIENCE INFORMATION AND COMMUNICATIONS TECHNOLOGY

FIELD OF SCIENCE ENGINEERING AND TECHNOLOGY

Ph.D. Thesis

Grzegorz Rypeść

Continual Learning Without Access to Past Data

Supervisor

prof. Tomasz Trzciński, PhD DSc

WARSAW 2026

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Tomasz Trzciński. His guidance throughout this research journey has been invaluable, and I am truly grateful for the numerous life opportunities he has opened for me. His profound insights and extensive research knowledge were fundamental to the completion of this work.

I am also sincerely grateful to my co-authors: Sebastian Cygert, Bartłomiej Twardowski, Daniel Marczak, Bartosz Zieliński and Valeryia Khan. Their collaboration and technical expertise were essential in conducting this research. Finally, I want to thank my family and friends for their unwavering support and encouragement, which kept me motivated every step of the way.

Finally, I acknowledge that this work was supported by the National Science Centre (Poland), PRELUDIUM 23 grant no. 2024/53/N/ST6/03018.

Continual Learning Without Access to Past Data

This doctoral dissertation presents a series of three scientific publications dedicated to the field of Continual Learning in artificial intelligence systems. The primary research challenge addressed is the implementation of the learning process under the constraint of zero access to historical data. This implies that during the model's adaptation to a new task, the use of any data samples available at earlier training stages is prohibited. This represents the most rigorous and demanding variant of incremental learning, in which neural networks exhibit a natural tendency toward so-called catastrophic forgetting. This phenomenon involves the rapid loss of the ability to solve previously mastered tasks due to the overwriting of model weights with new information. The presented publications analyze and generalize the problem of continual learning while proposing innovative methods for its resolution.

The first publication presents a new ensemble learning method in which experts cooperate with each other to solve tasks and autonomously decide which of them will undergo the process of further training for a new task. By applying a heuristic that analyzes the knowledge distributions of the experts, the model selected for adaptation is the one whose degree of forgetting will be the lowest. In the subsequent publication, the scope of continual learning is extended to scenarios in which datasets are only partially labeled. It is demonstrated that combining knowledge distillation with class adaptation is crucial for achieving high model plasticity while simultaneously limiting forgetting. In the final, third publication, it is shown that the common problem of task recency bias results from dimensional collapse in the representation (embedding) space of these models. In response to this phenomenon, a novel loss function is developed that enforces positive eigenvalues of the class covariance matrices, along with an algorithm for their adaptation, which significantly increases the efficiency of continual learning processes in complex deep architectures.

Key words: Artificial Intelligence, Deep Learning, Continual Learning, Catastrophic Forgetting

Uczenie ciągłe bez dostępu do przeszłych danych

W niniejszej rozprawie doktorskiej przedstawiono cykl trzech publikacji naukowych poświęconych tematyce uczenia ciągłego (ang. Continual Learning) w systemach sztucznej inteligencji. Głównym wyzwaniem badawczym podjętym w pracy jest realizacja procesu uczenia w warunkach całkowitego braku dostępu do danych historycznych. Oznacza to, że podczas adaptacji modelu do nowego zadania zabronione jest korzystanie z jakichkolwiek próbek danych dostępnych na wcześniejszych etapach szkolenia. Jest to najbardziej rygorystyczny i wymagający wariant uczenia przyrostowego, w którym sieci neuronowe wykazują naturalną tendencję do tzw. katastroficznego zapominania (ang. catastrophic forgetting). Zjawisko to polega na gwałtownej utracie zdolności do rozwiązywania wcześniej opanowanych zadań na skutek nadpisywania wag modelu nowymi informacjami. Zaprezentowane publikacje analizują i generalizują problem uczenia ciągłego, a także proponują nowatorskie metody jego rozwiązania.

W pierwszej publikacji przedstawiono nową metodę uczenia zespołowego, w której eksperci współpracują ze sobą przy rozwiązywaniu zadań i autonomicznie decydują, który z nich zostanie poddany procesowi douczania do nowego zadania. Dzięki zastosowaniu heurystyki analizującej rozkłady wiedzy ekspertów, do adaptacji wybierany jest ten model, którego stopień zapominania będzie najmniejszy. W kolejnej publikacji problematyka uczenia ciągłego została rozszerzona o scenariusze, w których zbiory danych są tylko częściowo etykietowane. Wykazano, że połączenie destylacji wiedzy z adaptacją klas jest kluczowe dla uzyskania wysokiej plastyczności modelu przy jednoczesnym ograniczeniu zapominania. W ostatniej, trzeciej publikacji pokazano, że powszechny problem faworyzacji ostatnich zadań (ang. task recency bias) wynika z zapaści wymiarowej (ang. dimensional collapse) w przestrzeni reprezentacji (zanurzeń) tych modeli. W odpowiedzi na to zjawisko opracowano nowatorską funkcję straty wymuszającą dodatnie wartości własne macierzy kowariancji klas oraz algorytm ich adaptacji, co znacząco podnosi efektywność procesów uczenia ciągłego w złożonych architekturach głębokich.

Słowa kluczowe: Sztuczna Inteligencja, Uczenie głębokie, Uczenie Ciągłe, Katastroficzne Zapominanie

Abbreviations

NN	Artificial Neural Network
AI	Artificial Intelligence
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
CL	Continual Learning
CIL	Class Incremental Learning
EFCIL	Exemplar Free Class Incremental Learning
SEED	Selection of Experts for Ensemble Diversification
CAMP	Category Adaptation Meets Projected distillation
SGD	Stochastic Gradient Descent
LWF	Learning Without Forgetting
EWC	Elastic Weight Consolidation
FeTrIL	Feature Translation for Exemplar-Free Class-Incremental Learning
FeCAM	Feature Covariance-Aware Metric

Contents

1. Introduction

1.1. Continual Learning

Artificial neural networks (NNs) have become the cornerstone of modern artificial intelligence, demonstrating state-of-the-art performance across a wide range of domains, including computer vision, natural language processing, speech recognition, and decision-making in complex environments. This success has been fueled by advances in deep learning, where hierarchical neural architectures learn rich, multi-level representations directly from raw data [1, 2]. Model architectures such as convolutional neural networks (CNNs) [3, 4], recurrent neural networks (RNNs) [5, 6], and transformers [7, 8] have become standard approaches, achieving state-of-the-art results in benchmark tasks and enabling applications ranging from autonomous driving [9] to large-scale language modeling [10].

The representational power of neural networks stems from their ability to approximate highly nonlinear functions [11, 12], coupled with sample-efficient optimization techniques such as stochastic gradient descent (SGD) [13] and its modern variants like ADAM [14], as well as regularization methods [15, 16] that improve generalization. These models excel in stationary learning settings, where large, representative datasets are available at once and training is conducted offline until convergence. Once trained, a model can be deployed to perform inference, with no expectation of further adaptation.

However, this conventional paradigm - often called offline or batch learning - stands in sharp contrast to the demands of many real-world environments [17, 18]. In practice, data often arrive sequentially, may exhibit distributional shift over time, and can contain novel concepts not present in the original training set. For instance, a medical diagnosis system may encounter new disease variants, an autonomous vehicle may face previously unseen traffic scenarios, and a personal assistant may need to learn about new entities or user preferences. In such settings, retraining from scratch after every data update is computationally expensive, memory-intensive, and often infeasible due to privacy restrictions or real-time constraints. Instead, systems should learn continually - integrating new information as it arrives while retaining previously acquired knowledge.

This requirement motivates the field of continual learning (CL) [19, 20], also referred to as lifelong learning. The overarching goal of CL is to enable a model to acquire, fine-tune, and transfer knowledge throughout its operational lifetime, without catastrophic degradation of performance on earlier tasks. This capability is fundamental for building adaptive, autonomous systems that can operate effectively in non-stationary environments.

The motivation for continual learning is further strengthened by its biological plausibility. Humans and animals learn in a sequential and lifelong manner, acquiring new skills and knowledge every day - across their lifespans. They do so without needing to retrain from scratch and without explicitly storing or rehearsing all prior experiences [21, 22, 23].

1.2. Challenges

Despite decades of machine learning research, achieving robust continual learning in artificial neural networks remains a formidable challenge. The central obstacle is catastrophic forgetting [24, 25], a phenomenon where training on new data leads to a rapid loss of performance on previously learned tasks, as presented in Fig. 1. This occurs because standard gradient-based optimization overwrites weights critical to earlier knowledge when fitting new objectives. Unlike human learning - where old and new knowledge can coexist and even reinforce each other - neural networks lack inherent mechanisms for preserving past information unless explicitly engineered to do so.

The severity of catastrophic forgetting is amplified in *class-incremental learning (CIL)* settings [26, 27], where the model must learn to recognize an expanding set of classes over time, often without access to prior data (EFCIL: *exemplar-free class-incremental learning* setting). This scenario is particularly relevant when legal or ethical considerations preclude the storage of past samples, such as in privacy-sensitive domains (e.g., GDPR-compliant systems) or when operating under strict memory budgets (e.g., embedded devices). In the absence of stored exemplars, the network must rely entirely on *internal mechanisms* - such as parameter regularization [17, 28], knowledge distillation [29, 18], dynamic architectures [30, 31], or generative replay [32] - to preserve prior knowledge. Another challenging scenario is *Generalized Continual Category Discovery (GCCD)* [33] where the data in each task is mostly unlabeled, and part of the classes have no labels at all - these are called novel classes and must be discovered by the method, often by using unsupervised learning strategies.

Moreover, continual learning in neural networks faces a fundamental *stability-plasticity*

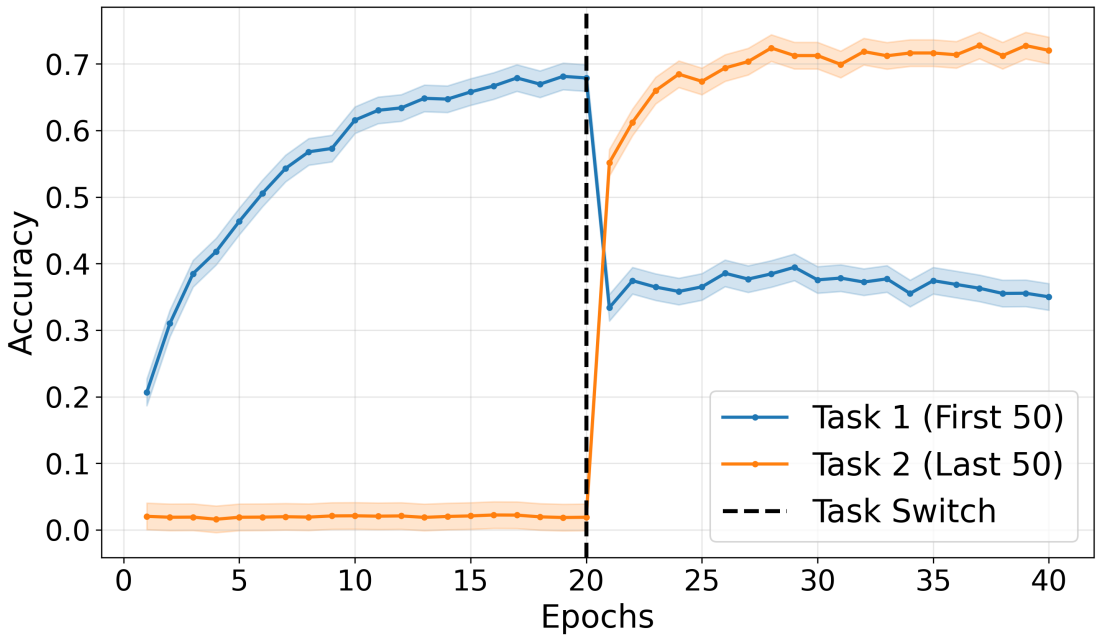

Figure 1. Empirical demonstration of catastrophic forgetting in a multi-head ResNet-32 neural network during sequential task acquisition. Initially, the model achieves peak performance on Task 1 (CIFAR-100, classes 0–49). Upon the introduction of Task 2 (classes 50–99) at epoch 20, a sharp decline in Task 1 accuracy is observed. This performance collapse illustrates the need for Continual Learning methods.

dilemma [34, 35]. *Plasticity* refers to the model's capacity to acquire new information, while *stability* denotes its ability to retain old knowledge. Excessive plasticity leads to forgetting, while excessive stability hinders adaptation to new tasks. Striking the right balance is nontrivial, particularly in scenarios where the sequence of tasks is long, heterogeneous, and potentially adversarial.

Recent years have seen a surge of interest in mitigating forgetting through improved architectural designs [31, 36], refined regularization strategies [28, 18], memory-based approaches [37], and more sophisticated representation learning techniques [26, 27, 38]. While these methods have achieved notable progress, none fully resolves the challenges posed by highly dynamic, open-world scenarios, where models must handle both previously known and newly emerging categories, adapt to evolving data distributions, and operate without access to stored exemplars.

Beyond the general loss of prior knowledge, Continual Learning is frequently plagued by task-recency bias. This phenomenon manifests as a systematic tendency of the model to favor the most recently learned classes during inference, often misclassifying older samples as belonging to the newest task. In the absence of a balanced rehearsal buffer, the weights of the final classification layer and the underlying latent representations become skewed toward the latest data distribution [39]. This bias is

not merely a byproduct of gradient updates but is deeply rooted in the geometric transformation of the latent space, where new tasks may occupy disproportionately large or high-variance regions, effectively "crowding out" the representations of previous tasks. Addressing this imbalance is critical for achieving fairness and high total accuracy across the entire lifetime of the model.

1.3. Overview of the Dissertation

This dissertation presents a series of 3 Continual Learning publications that focus on developing artificial intelligence systems that can learn new information over time without needing to revisit or store data from the past. This approach is essential for real-world applications where keeping old data is not possible due to limited memory or privacy regulations. The primary goal of this work is to create methods that allow these models to stay flexible enough to learn new tasks while remaining stable enough to remember what they have already mastered.

The research is organized into three main areas of contribution:

- **Modular Learning:** We explore how to divide a model into specialized parts that can learn new tasks independently, ensuring that new knowledge does not overwrite previous training.
- **Balancing Stability and Plasticity:** We investigate how to combine different learning strategies so a model can learn and discover new types of data while still retaining its original knowledge.
- **Correcting Model Memory:** We examine how a model's internal representations of classes can change over time and develop ways to statistically correct these shifts to maintain accuracy over the long term.

Through these contributions, this dissertation aims to bridge the gap between theoretical research and the practical need for adaptive, lifelong learning in AI systems.

1.4. Structure of the Dissertation

The dissertation is organized as follows:

- **Chapter 1: Introduction** – Provides a comprehensive overview of the field of Continual Learning. It establishes the necessity for lifelong learning in artificial intelligence and delineates primary challenges, such as catastrophic forgetting.
- **Chapter 2: Background** – Establishes the technical foundations and math-

ematical frameworks essential to this research. This chapter also evaluates existing methodologies developed to address the stability-plasticity dilemma.

- **Chapter 3: Research Synopsis** – Bridges the gap between general theory and the specific contributions of this work. It outlines the research objectives and provides a high-level synthesis of the three primary scientific papers included in this dissertation.
- **Chapter 4: Publications** – Presents the full content of the three scientific papers: SEED [40], CAMP [33], and AdaGauss [41]. These sections contain detailed experimental frameworks, algorithms, mathematical derivations, and empirical results.
- **Chapter 5: Conclusions** – Synthesizes the core findings of the dissertation and discusses potential avenues for future investigation.

2. Background

Continual Learning (CL) is a vast field of machine learning studied under various settings and assumptions. While Chapter 1 highlighted the biological motivation and the general problem of catastrophic forgetting, this chapter provides the formal technical foundations and mathematical frameworks necessary for the methods proposed in this dissertation.

2.1. (Exemplar-Free) Class Incremental Learning

In the Class-Incremental Learning (CIL) scenario, a dataset is considered to be partitioned into T subsets, dubbed tasks, where each task $t \in \{1, \ldots, T\}$ corresponds to a non-overlapping set of classes C_t. These are defined such that $\bigcup_{t=1}^{T} C_t = C$, where $C_t \cap C_s = \emptyset$ for $t \neq s$. In Exemplar-Free CIL (EFCIL), only the current task data $D_t = \{(x, y) \mid y \in C_t\}$ is accessible during training stage t; the storage of any data samples (exemplars) from previous steps is not permitted. The objective is the training of a model that discriminates between all previously encountered ($< t$) and new classes combined. A task-agnostic evaluation [42, 38] is assumed, wherein the task ID is not provided during inference.

2.2. Generalized Continual Category Discovery

Expanding beyond supervised learning, Generalized Continual Category Discovery (GCCD) involves the sequential training of models on partially labeled tasks containing both known and novel categories. Novel categories represent classes for which no label is provided. Formally, a dataset D is partitioned into T disjoint subsets, where each subset represents a task denoted as D_t. Each task D_t is comprised of two subsets: $D_t = D_L^t \cup D_U^t$, where $D_L^t = \{(x, y) \mid x \in X_L^t, y \in Y^t\}$ denotes the labeled set of known classes, and $D_U^t = \{x \mid x \in X_U^t\}$ encompasses unlabeled data samples from both known and novel categories.

Within the GCCD framework, sequential learning is performed from a stream of tasks. At task t, the model is assumed to consist of a feature extractor $F_t : X_t \to Z_t$ and a classifier $C_t : Z_t \to Y_t$, where Z_t signifies the latent space generated by the

feature extractor. The categories of a given task D_t must be recognized while the ability to recognize categories from previous tasks $\bigcup_{i=1}^{t-1} D_i$ is maintained. Similarly to EFCIL, task-agnostic evaluation is assumed, therefore the task ID is not provided at inference. The primary challenge lies in the prevention of catastrophic forgetting on previous tasks while maintaining high plasticity for the learning of new categories on new data.

2.3. Evaluation Metrics

To rigorously assess CL performance, let $a_{t,k}$ denote the accuracy of the model after training on task t, evaluated on the test data of task k (where $k \leq t$).

2.3.1. Average Accuracy

The Average Accuracy (A_t) represents the mean performance across all tasks observed up to the current time step t:

$$A_t = \frac{1}{t} \sum_{k=1}^{t} a_{t,k} \tag{2.1}$$

2.3.2. Average Incremental Accuracy

To evaluate the performance of the model throughout the entire learning process rather than just at the final state, the Average Incremental Accuracy (A_{inc}) is used. It is defined as the mean of the average accuracies A_t recorded after each task t:

$$A_{inc} = \frac{1}{T} \sum_{t=1}^{T} A_t = \frac{1}{T} \sum_{t=1}^{T} \left(\frac{1}{t} \sum_{k=1}^{t} a_{t,k} \right) \tag{2.2}$$

This metric effectively represents the cumulative performance over the learning trajectory, penalizing models that suffer from early catastrophic forgetting.

2.3.3. Forgetting

Forgetting (F) quantifies the drop in performance on previous tasks. It is defined as the average difference between the maximum accuracy obtained for each task k throughout the learning process and its final accuracy at step T:

$$F = \frac{1}{T-1} \sum_{k=1}^{T-1} \left(\max_{s \in \{k,\dots,T-1\}} a_{s,k} - a_{T,k} \right) \tag{2.3}$$

2.3.4. Plasticity

Plasticity (P) measures the model's ability to integrate new information. In this dissertation it is defined as the average of accuracies achieved in the incremental tasks on which the model was trained:

$$P = \frac{\sum_{n=2}^{T} a_{n,n}}{T-1} \tag{2.4}$$

A P value close to 1 indicates that the sequential nature of learning does not hinder the acquisition of new categories.

2.4. The Stability-Plasticity Dilemma

The metrics above allow us to quantify the *Stability-Plasticity Dilemma*. In a perfectly plastic model, P is maximized but F is high (catastrophic forgetting). Conversely, a perfectly stable model has $F = 0$ but P approaches zero. This dissertation focuses on techniques that optimize this trade-off without the use of exemplars.

2.5. Popular EFCIL Methods

In the absence of a memory buffer, EFCIL methods must rely on architectural constraints, weight regularization, or functional distillation to preserve knowledge. The following methods represent the progression of strategies in the field.

2.5.1. Fine-tuning (FT)

Fine-tuning serves as the *lower bound* for continual learning performance. The model parameters θ are updated using only the cross-entropy loss on the current task D_t:

$$L_{FT}(\theta) = \sum_{(x,y)\in D_t} L_{ce}(f(x;\theta), y) \tag{2.5}$$

While this approach maximizes plasticity P, it typically results in a complete collapse of accuracy on tasks $k < t$ because the weights are optimized solely for the new label space C_t.

2.5.2. Joint Training (Oracle)

Joint Training represents the *upper bound* (Oracle) performance. It violates the EFCIL constraint by assuming access to all cumulative data $D_{1:t} = \bigcup_{i=1}^{t} D_i$:

$$L_{Joint}(\theta) = \sum_{(x,y)\in D_{1:t}} L_{ce}(f(x;\theta), y) \tag{2.6}$$

This serves as the performance ceiling, representing the model's capability if it were trained in an i.i.d. (independent and identically distributed) fashion.

2.5.3. Learning Without Forgetting (LwF)

LwF [18] introduces the use of **Knowledge Distillation (KD)** to preserve previous knowledge. Before updating the model on task t, the current task's data $x \in D_t$ is passed through the frozen model from the previous step θ_{t-1} to generate "soft labels." The objective function balances learning new classes and mimicking the old model's behavior:

$$L_{LwF}(\theta) = L_{ce}(f(x;\theta)_t, y) + \lambda L_{dist}(f(x;\theta)_{<t}, f(x;\theta_{t-1})) \tag{2.7}$$

where L_{dist} is typically a temperature-scaled Kullback-Leibler (KL) divergence and λ is a hyperparameter responsible for the plasticity-stability tradeoff.

2.5.4. Elastic Weight Consolidation (EWC)

EWC [17] is a **prior-based regularization** method. It slows down learning on weights that are important for previous tasks. The importance of each parameter is estimated using the diagonal of the **Fisher Information Matrix** (F). The loss function is:

$$L_{EWC}(\theta) = L_{ce}(\theta) + \sum_{k<t} \frac{\Omega}{2} F_k(\theta_i - \theta^*_{k,i})^2 \tag{2.8}$$

where θ^*_k are the optimal parameters after task k and Ω is a hyperparameter governing the stability-plasticity trade-off.

2.5.5. Learning a Unified Classifier (LUCIR)

LUCIR [27] is designed to handle the imbalance between the large number of old classes and the new task data in EFCIL. It replaces the standard Softmax layer with a **Cosine Classifier** and introduces three specific constraints:

1. **Cosine Normalization:** To prevent the magnitude of new class weights from dominating the prediction, the classification logits are computed as the cosine similarity between the feature vector $f(x)$ and the class weight vectors w_j:

$$p_j(x) = \frac{\exp(\eta \cdot \langle \frac{f(x)}{\|f(x)\|}, \frac{w_j}{\|w_j\|} \rangle)}{\sum_k \exp(\eta \cdot \langle \frac{f(x)}{\|f(x)\|}, \frac{w_k}{\|w_k\|} \rangle)} \tag{2.9}$$

   where η is a learnable scalar parameter.
2. **Less-Forgetting Constraint:** Instead of distilling output probabilities, LUCIR preserves the geometry of the feature space. It penalizes the change in the angle between the current feature $f_t(x)$ and the previous feature $f_{t-1}(x)$ for the same input $x \in D_t$:

$$L_{lf} = \sum_{x \in D_t} \left(1 - \frac{\langle f_t(x), f_{t-1}(x) \rangle}{\|f_t(x)\| \|f_{t-1}(x)\|}\right) \tag{2.10}$$

3. **Inter-Class Separation:** To ensure new classes do not overlap with old class regions, a margin-based ranking loss is applied. For a sample x of a new class y, the loss encourages the feature to be closer to its ground truth weight w_y than to the "hardest" old class weights $\hat{w}_{old}$:

$$L_{is} = \sum_{x \in D_t} \max(0, m + \langle \bar{f}_t(x), \bar{w}_{old} \rangle - \langle \bar{f}_t(x), \bar{w}_y \rangle) \tag{2.11}$$

   where m is a pre-defined margin and the bars denote normalized vectors.

The final objective is the weighted sum: $L_{total} = L_{ce} + \lambda_1 L_{lf} + \lambda_2 L_{is}$, where λ_1 and λ_2 are hyperparameters.

2.5.6. Feature Translation for EFCIL (FeTrIL)

FeTrIL [43] is a recent EFCIL method that utilizes a fixed feature extractor (typically pre-trained on the first task) and focuses on the classification head. To combat the lack of old data, FeTrIL stores the mean feature μ_k for each class $k \in C_{1:t-1}$. When learning task t, it generates pseudo-features for previous classes by applying a transformation to the current features $F(x_t)$:

$$\hat{z}_k = F(x_t) - \mu_t + \mu_k \tag{2.12}$$

where μ_t is the mean of the current task features. A linear classifier is then trained on the combination of real features from D_t and these geometric-shifted pseudo-features $\hat{z}_k$. This approach effectively addresses the stability-plasticity dilemma by keeping

the backbone frozen (high stability) while allowing the linear head to adapt to the expanding label space.

2.5.7. Feature Covariance-Aware Metric (FeCAM)

FeCAM [44] addresses the suboptimality of using Euclidean distance for classification in EFCIL. It observes that feature distributions in non-stationary learning are often heterogeneous and anisotropic. To capture this, FeCAM models each class as a multivariate Gaussian distribution $N(\mu_k, \Sigma_k)$, where Σ_k represents the feature covariance.

1. **Covariance Estimation:** For each new class $k \in C_t$, the model calculates the class-specific mean μ_k and the covariance matrix Σ_k:

$$\Sigma_k = \frac{1}{|D_t^k| - 1} \sum_{x \in D_t^k} (F(x) - \mu_k)(F(x) - \mu_k)^T \tag{2.13}$$

   To ensure the matrix is non-singular, especially in low-data regimes, a shrinkage regularization is typically applied: $\hat{\Sigma}_k = (1 - \alpha)\Sigma_k + \alpha I$.
2. **Mahalanobis Distance Classification:** During inference, instead of Euclidean distance, FeCAM employs the **Mahalanobis distance** to account for the correlation between feature dimensions. The distance between a test feature z and a class prototype k is defined as:

$$d_M(z, \mu_k) = \sqrt{(z - \mu_k)^T \hat{\Sigma}_k^{-1} (z - \mu_k)} \tag{2.14}$$

   This metric effectively "normalizes" the feature space according to the spread of each class, allowing for more robust decision boundaries between old and new categories.
3. **Covariance Adaptation:** For previous classes where the raw data is no longer available, FeCAM can adapt the stored covariance Σ_k to account for feature drift caused by updates to the backbone, or it can utilize a shared global covariance if class-specific samples are too sparse.

By exploiting the second-order statistics of the features, FeCAM significantly reduces the overlap between classes in the latent space, which is a primary driver of catastrophic forgetting.

2.6. Experimental Benchmarks and Data Protocols

The evaluation of continual learning methods requires a diverse set of benchmarks to stress-test different aspects of the stability-plasticity trade-off. These range from coarse-grained image classification to fine-grained recognition and cross-domain adaptation, as presented in Fig. 2.6.1.

Table 2.6.1. Summary of dataset characteristics used for image classification experiments.

Dataset	Classes	Resolution	Primary Challenge
CIFAR-100	100	32×32	Standard Benchmark
TinyImageNet	200	64×64	Standard Benchmark
ImageNet-Subset	100	Varied	Standard Benchmark
CUB200	200	Varied	Fine-grained Details
FGVCAircraft	100	Varied	Morphological Similarity
Stanford Cars	196	Varied	Fine-grained Details
DomainNet	345	Varied	Domain/Style Shift

2.6.1. Standard Image Classification Benchmarks

General image classification datasets serve as the baseline for measuring a model's ability to learn distinct semantic concepts sequentially.

- **CIFAR-100** [45]: A cornerstone benchmark consisting of 100 classes with 60,000 images. Its low resolution (32× 32) poses a challenge for maintaining discriminative features as the number of tasks increases.
- **TinyImageNet** [46]: A subset of ImageNet [47] containing 100,000 images across 200 classes at 64×64 resolution. It serves as an intermediate step between CIFAR and full-scale ImageNet.
- **ImageNet-Subset** [48]: Often restricted to 100 classes, this dataset provides high-resolution images that test the scalability of EFCIL methods in more realistic visual environments. The images are of different resolution and are typically clipped to 224×224 pixels.

2.6.2. Fine-Grained Image Classification

Fine-grained classification benchmarks are utilized to evaluate the capacity of models to maintain highly discriminative feature representations at a narrow semantic scale. Unlike standard datasets where categories are distinct (e.g., "dog" vs. "plane"), fine-grained datasets consist of subordinate categories within a single entry-level class. In the context of Continual Learning, this setting is particularly

demanding as it requires highly granular representations. The model must capture and preserve subtle, localized features - such as the specific curvature of a beak or the wingspan patterns of an aircraft - to separate classes that occupy high-density regions of the latent space. Consequently, even minor parameter drift during the acquisition of new tasks can lead to significant inter-class interference, testing the model's ability to maintain a stable yet highly detailed feature space.

- **CUB200** [49]: Contains 11,788 images across 200 bird species.
- **FGVCAircraft** [50]: Consists of 10,200 images across 100 aircraft variants.
- **Stanford Cars** [51]: Features 16,185 images of 196 car types.

In these scenarios, the challenge of catastrophic forgetting is amplified; because the classes are semantically close, the decision boundaries are highly sensitive to the weight drift associated with learning new tasks.

2.6.3. Domain-Shift and Distributional Drift

While standard Class-Incremental Learning primarily focuses on the addition of new semantic categories, **DomainNet** [52] is utilized to simulate shifts in the underlying data distribution. This dataset contains 345 classes across six distinct domains: *clipart, infograph, painting, quickdraw, real,* and *sketch*.

By assigning a different domain to each learning task, the experimental protocol simulates a scenario where the visual style and characteristics of the data change abruptly, even if the semantic categories remain consistent or overlap. This evaluates the model's robustness to *distributional drift* - the ability to adapt to new visual representations of known concepts without suffering from performance degradation on previously encountered styles.

2.6.4. GCCD Data Composition

The Generalized Continual Category Discovery setting introduces a more complex data distribution protocol compared to standard supervised EFCIL. As defined in Section 2.2, each task contains a mixture of known and novel categories.

The experimental protocol generally follows a specific ratio of supervision. For a given dataset D, the classes are partitioned such that the ratio of **known to novel** classes is 4:1. Within the known class subset for any given task, the ratio of **labeled to unlabeled** samples is maintained at 1:1 [33, 53]. This forces the model to not only prevent forgetting but also to perform unsupervised discovery and semi-supervised refinement simultaneously.

2.6.5. Initialization and Task Distribution

The starting state of the model and the relative size of the initial task are critical factors in benchmarking CL performance:

- **Training from Scratch vs. Fine-tuning:** Training a model *from scratch* ensures that the feature extractor must develop its internal representations solely through the incremental stream. This is often contrasted with starting from a *pretrained* backbone (e.g., ImageNet-pretrained ViT [8]). While pretraining provides a more robust starting point, training from scratch is the more rigorous test of a method's ability to build a latent space from zero without leveraging implicit prior knowledge of the visual world.
- **Large First Task vs. Equal Tasks:** Many CIL protocols [43, 44, 54] utilize a "Large First Task" setting (e.g., training on 50 classes of CIFAR-100 first, then 5 tasks of 10 classes). This provides the model with a strong, stable feature extractor before incremental learning begins. However, recent trends - and the protocols followed in this work - favor **equal tasks**, where the model starts with a limited number of classes. This "balanced" protocol is significantly more challenging, as the model must adapt its feature representation early in the process when it has the least amount of information, often leading to more severe drift in subsequent steps.

2.6.6. Task Partitioning and Granularity

The number of tasks (T) defines the temporal resolution of the learning process, typically set to 5, 10, or 20 in standard protocols. The choice of T impacts the evaluation in two primary ways:

- **Interference Frequency:** A higher T increases the number of optimization steps, providing more opportunities for parameter updates to overwrite prior knowledge and exacerbate catastrophic forgetting.
- **Sample Sparsity:** Increasing T reduces the data available per task. This evaluates the model's ability to generalize from limited samples, a critical challenge for methods relying on class prototypes or feature statistics estimation.

3. Research Synopsis

3.1. Research questions

The research presented in this dissertation addresses the fundamental challenge of Catastrophic Forgetting within the context of Exemplar-Free Class Incremental Learning (EFCIL) and Generalized Continual Category Discovery (GCCD). These two settings were prioritized for several key reasons. First, resolving catastrophic forgetting in the most constrained environment (where historical data is entirely inaccessible) establishes a robust framework that likely generalizes to less restrictive scenarios. Second, the inherent complexity of EFCIL and GCCD represents a critical frontier in machine learning, offering significant opportunities for both theoretical and practical advancement. To guide this investigation, the following **research questions** were posed:

RQ 1: **Architectural Isolation:** How can architectural isolation and selective parameter updates be utilized to mitigate catastrophic forgetting without increasing memory overhead?

RQ 2: **Semi-Supervised Stability:** How can the stability-plasticity dilemma be balanced in semi-supervised scenarios where label sparsity is prevalent?

RQ 3: **Evolution of Representations:** How do the representations of previous classes evolve during the acquisition of new tasks, and can these shifts be accurately predicted?

RQ 4: **Covariance Adaptation:** Can statistical corrections be applied to the distributions of former classes to restore their validity and improve decision boundaries?

RQ 5: **Statistical Roots of Bias:** To what extent does dimensionality collapse in latent spaces contribute to task-recency bias, and through what mechanisms can it be avoided?

3.2. Contributions

The investigation of aforementioned research questions within the scope of this dissertation has led to the development of frameworks that outperform current state-of-the-art methods in both EFCIL and GCCD settings. By bridging the gap between theoretical stability-plasticity trade-offs and practical algorithmic implementation, this work offers a cohesive approach to mitigating problems encountered in Continual Learning. The primary technical and theoretical contributions resulting from this doctoral research are summarized as follows:

1. Mitigation of Forgetting through Modular Architectural Isolation

In response to **RQ 1**, this study introduces a modular framework designed to decouple task-specific parameters and prevent catastrophic interference. While monolithic backbones suffer from global weight updates that overwrite previous knowledge, architectural isolation - achieved through multi-expert ensembles - effectively partitions the parameter space without the memory overhead of exemplar storage [40]. Based on this insight, a novel method is proposed that represents classes as multivariate Gaussian distributions within a shared latent space. Furthermore, a specialized algorithm is introduced to determine expert assignment based on the distributional overlap of class representations. This selective training mechanism ensures that new knowledge acquisition occurs with minimal disruption to previously learned parameters, achieving state-of-the-art results on large-scale benchmarks without the memory overhead associated with exemplar storage.

2. Synergistic Interplay between Knowledge Distillation and Category Adaptation

Recent advancements in Continual Learning (CL) frequently employ knowledge distillation mechanisms to enhance model stability [26, 18], or utilize category adaptation to account for distributional shifts in previously learned classes after training on new tasks [55]. Addressing the **RQ 2 & 3**, this study demonstrates that while these mechanisms address distinct challenges when applied in isolation, they are fundamentally complementary. Their integration significantly mitigates catastrophic forgetting and improves overall classification performance. Based on these findings, this work introduces a novel method, CAMP (*Category Adaptation Meets Projected Distillation*), which establishes a new state-of-the-art for both EFCIL and GCCD settings.

3. Discovery of Novel Categories in Non-Stationary Environments

This research extends the scope of stability-plasticity dilemma (**RQ 2**) into the realms of semi-supervised learning and Generalized Category Discovery (GCD) [53, 33]. A primary contribution of this work is the demonstration of how the integration of category adaptation and knowledge distillation enables a model to maintain sufficient plasticity for discovering novel, unlabeled categories while ensuring the stability required to retain supervised knowledge from prior stages. The proposed CAMP method serves as a critical bridge between supervised continual learning and open-world discovery, providing a robust mechanism for autonomous systems to incrementally expand their knowledge bases within non-stationary environments.

4. Statistical Correction of Class Mean and Covariance

A central challenge in Exemplar-Free Class Incremental Learning is overcoming the phenomenon of distributional shift, as posed in **RQ 4**. Many contemporary methods represent classes by storing their distribution statistics within a latent space [43, 44]; however, as the feature extractor is updated during subsequent tasks, these stored statistics lose alignment with the evolving ground truth. This misalignment results in catastrophic forgetting due to the degradation of decision boundaries. To address this, this work introduces AdaGauss [41], a novel method that applies corrective transformations to both the mean and covariance of class distributions. This approach achieves state-of-the-art results across standard EFCIL benchmarks by restoring the statistical validity of previous class representations.

5. Dimensionality Collapse as a Source of Task-Recency Bias and the Anti-Collapse Loss

This research provides a critical diagnostic advancement related to **RQ 5** by identifying that *task-recency bias* (the systematic tendency of models to favor recently learned classes) is a direct symptom of dimensionality collapse within the latent space [41]. Specifically, it is demonstrated that near-singular class covariance matrices and unstable inverse operations lead to skewed decision boundaries. By pinpointing these statistical roots, this work explains why the final classification layer and underlying representations become biased toward the most recent data distribution. To mitigate this effect, a novel **Anti-Collapse loss function** is proposed, which regularizes the latent space to prevent structural degradation, thereby significantly reducing bias and enhancing overall model performance.

3.3. Summaries of Included Publications

This thesis is founded upon a series of peer-reviewed publications that present new developments in the field of continual learning. This section provides a short summary of each one of them, what aforementioned research questions and contributions they address. The publications were accepted and presented on top-tier machine learning conferences **(CORE A*)**.

1. Grzegorz Rypeść, Sebastian Cygert, Valeriya Khan, Tomasz Trzciński, Bartosz Zieliński, Bartłomiej Twardowski. "Divide and not forget: Ensemble of selectively trained experts in Continual Learning" [40]. The Twelfth International Conference on Learning Representations (ICLR), (2024).

To bridge the gap between theoretical Class Incremental Learning (CIL) and practical deployment, this work addresses the limitations of current methodologies that often rely on the storage of exemplars, a practice frequently restricted by privacy regulations or the hardware constraints of memory-limited devices [56]. While many exemplar-free approaches attempt to circumvent catastrophic forgetting [25, 24] by utilizing a frozen feature extractor - either through extensive pre-training or a disproportionately large initial task [27, 57, 43, 58, 59] - these methods demonstrate a significant performance collapse when high-quality representations are not available upfront. Conversely, architecture-based strategies that expand the model for each new task provide necessary plasticity but are prone to parameter explosion, making them unfeasible for long-term learning sequences [60]. Addressing these conflicting challenges, we introduce SEED (Selection of Experts for Ensemble Diversification), a novel framework that maintains a fixed computational budget by utilizing a finite ensemble of experts. Unlike existing ensembling techniques that require task IDs at inference or remain dependent on rehearsal [61, 62], SEED selectively updates only a single expert per task. This selection is driven by multivariate Gaussian distributions of class features, which minimizes representational drift and encourages expert diversification. By leveraging these rich class representations for Bayesian inference, SEED achieves state-of-the-art accuracy in both task-aware and task-agnostic scenarios, effectively balancing stability and plasticity without the need for privileged initial data or memory overhead associated with exemplar buffers. This work addresses **RQ 1** and **Contribution 1** of this dissertation.

Author Contributions: I am the primary author of this work, responsible for the conceptualization of the SEED framework and the development of the selective

expert training strategy. I designed and implemented the method, conducted most of the experiments, and led the manuscript's preparation, specifically authoring Sections I, III, IV, and V.

2. Grzegorz Rypeść, Daniel Marczak, Sebastian Cygert, Tomasz Trzciński, Bartłomiej Twardowski. "Category Adaptation Meets Projected Distillation in Generalized Continual Category Discovery" [33], The European Conference on Computer Vision (ECCV), (2024).

To extend the scope of dynamic learning, Generalized Continual Category Discovery (GCCD) moves beyond traditional closed-world scenarios to address the simultaneous challenges of overcoming catastrophic forgetting [25] and uncovering novel semantic categories in partially labeled data streams [53, 63, 64]. Current methodologies often rely heavily on exemplars to maintain performance, a strategy that is increasingly untenable in privacy-sensitive or resource-constrained environments. While feature distillation (FD) is frequently employed to mitigate forgetting in these exemplar-free settings [65, 66], it often acts as a restrictive "soft" frozen feature extractor, stifling the model's plasticity and its ability to distinguish new categories [67]. Conversely, simply expanding the architecture to accommodate new discovery risks a parameter explosion that limits long-term scalability. To resolve this tension, we propose CAMP (Category Adaptation Meets Projected distillation), which introduces a learnable projector within the distillation framework to grant the model the flexibility to adapt its representations. Recognizing that this newfound plasticity induces a distribution shift in previously learned categories, CAMP utilizes an efficient auxiliary category adaptation network to predict and compensate for the semantic drift of cluster centroids [68, 55]. By synthesizing projected distillation with active centroid recovery, CAMP avoids the rigidity of static extractors and the overhead of unbounded architectural growth, achieving state-of-the-art results across both GCCD and standard Class Incremental Learning benchmarks. This work addresses **RQ 2 & 3** and **Contributions 2 & 3** of this dissertation.

Author Contributions: I served as the lead researcher, formulating the core methodology that integrates projected distillation with category adaptation. I implemented the CAMP method and performed the majority of the experiments. I was responsible for the structural design of the paper and the writing of the introduction, methodology, results and summary sections (Secs. I, III, IV, V).

3. Grzegorz Rypeść, Sebastian Cygert, Tomasz Trzciński, Bartłomiej Twardowski. "Task-recency bias strikes back: Adapting covariances in Exemplar-Free Class Incremental Learning" [41], The Thirty-Eighth Annual Conference on Neural Information Processing Systems (NeurIPS), (2024)

This work addresses significant shortcomings in current EFCIL methods, specifically their susceptibility to task-recency bias and their failure to account for class covariance shifts. While existing approaches often model classes as Gaussian distributions in the latent space [69, 70, 40], they typically assume these distributions remain static. We demonstrate that when a feature extractor is unfrozen to maintain plasticity, the ground truth distributions of previous classes drift, rendering memorized parameters obsolete. Furthermore, we identify that dimensionality collapse [71, 72] particularly in early tasks-leads to low-rank covariance matrices for older classes, which introduces inversion errors during classification and further exacerbates task-recency bias. To solve these issues, we propose AdaGauss (Adapting Gaussians), a novel framework that is the first to adapt both the means and covariances of memorized distributions to align with the evolving feature space. The method introduces a unique anti-collapse loss to preserve the rank of feature representations and utilizes knowledge distillation [29] through a learnable projector network to stabilize the learning process. As presented in Fig. 2 it operates in two stages: training the feature extractor and adapting class distributions. Extensive evaluations show that by explicitly correcting for distribution drift and representation collapse, AdaGauss achieves state-of-the-art performance in both from-scratch and pre-trained EFCIL scenarios. This work addresses **RQ 4 & 5** and **Contributions 4 & 5** of this dissertation.

Author Contributions: I am the lead author of this manuscript. I identified the link between dimensionality collapse and task-recency bias, which led to the development of the AdaGauss algorithm and the Anti-Collapse loss function. I formulated the methodology for adapting class covariance matrices, executed the entirety of the software implementation and experimental benchmarks, and authored the majority of the manuscript (Secs. I–V).

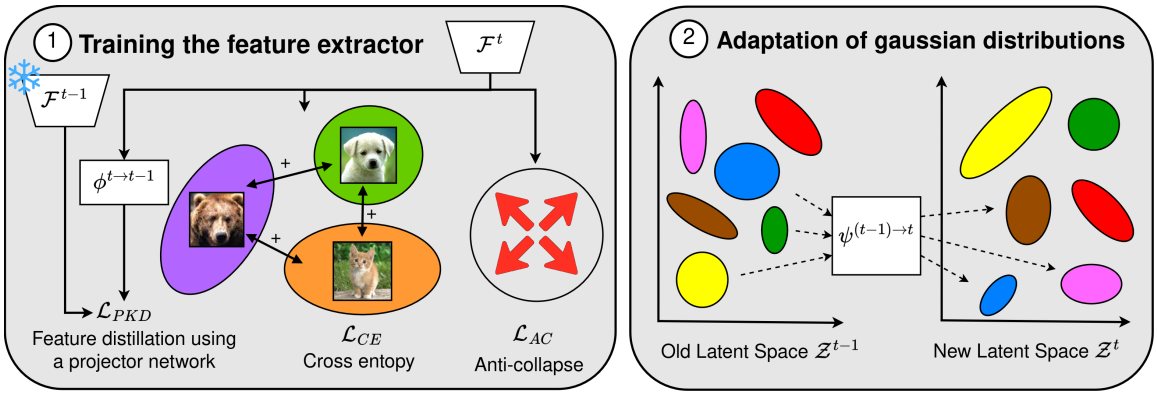

Figure 2. The AdaGauss method operates in two stages for each task. First, the feature extractor is trained using feature distillation through a projector network, alongside cross-entropy and anti-collapse loss functions. Second, the representations of past classes are transformed into the latent space of the updated feature extractor. This adaptation allows the model to maintain a valid decision boundary across all classes.

3.4. Publications not included in the dissertation

1. **Grzegorz Rypeść**, Grzegorz Kurzejamski. "ESC: Evolutionary Stitched Camera Calibration in the Wild", IEEE Congress on Evolutionary Computation (CEC), (2024).
2. Adam Pardyl, **Grzegorz Rypeść**, Grzegorz Kurzejamski, Bartosz Zieliński, Tomasz Trzciński. "Active visual exploration based on attention-map entropy", 32nd International Joint Conference on Artificial Intelligence (IJCAI), (2023).
3. **Grzegorz Rypeść**, Łukasz Lepak, Paweł Wawrzyński. "Reinforcement Learning for on-line Sequence Transformation", 17th Conference on Computer Science and Intelligence Systems (FedCSIS), (2022).
4. **Grzegorz Rypeść**, Grzegorz Kurzejamski, Jacek Komorowski. "Sports Camera Pose Refinement Using an Evolution Strategy", IEEE Congress on Evolutionary Computation (CEC), (2022).

4. Publications

DIVIDE AND NOT FORGET: ENSEMBLE OF SELECTIVELY TRAINED EXPERTS IN CONTINUAL LEARNING

Grzegorz Rypeść[1, 2]*, Sebastian Cygert[1, 3], Valeriya Khan[1, 2], Tomasz Trzciński[1, 2, 4], Bartosz Zieliński[1,5] & Bartłomiej Twardowski[1,6,7]
[1]IDEAS-NCBR, [2]Warsaw University of Technology, [3] Gdańsk University of Technology,
[4] Tooploox, [5] Jagiellonian University, [6] Computer Vision Center, Barcelona
[7] Department of Computer Science, Universitat Autònoma de Barcelona,
{grzegorz.rypesc, sebastian.cygert, valeriya.khan, tomasz.trzcinski, bartosz.zielinski, bartlomiej.twardowski}@ideas-ncbr.pl

ABSTRACT

Class-incremental learning is becoming more popular as it helps models widen their applicability while not forgetting what they already know. A trend in this area is to use a mixture-of-expert technique, where different models work together to solve the task. However, the experts are usually trained all at once using whole task data, which makes them all prone to forgetting and increasing computational burden. To address this limitation, we introduce a novel approach named SEED. SEED selects only one, the most optimal expert for a considered task, and uses data from this task to fine-tune only this expert. For this purpose, each expert represents each class with a Gaussian distribution, and the optimal expert is selected based on the similarity of those distributions. Consequently, SEED increases diversity and heterogeneity within the experts while maintaining the high stability of this ensemble method. The extensive experiments demonstrate that SEED achieves state-of-the-art performance in exemplar-free settings across various scenarios, showing the potential of expert diversification through data in continual learning.

1 INTRODUCTION

In Continual Learning (CL), tasks are presented to the learner sequentially as a stream of non-i.i.d data. The model has only access to the data in the current task. Therefore, it is prone to *catastrophic forgetting* of previously acquired knowledge (French, 1999; McCloskey & Cohen, 1989). This effect has been extensively studied in Class Incremental Learning (CIL), where the goal is to train the classifier incrementally and achieve the best accuracy for all classes seen so far. One of the most straightforward solutions to alleviate forgetting is to store exemplars of each class. However, its application is limited, e.g., due to privacy concerns or in memory-constrained devices (Ravaglia et al., 2021). That is why more challenging, exemplar-free CIL solutions attract a lot of attention.

Many recent CIL methods that do not store exemplars rely on having a strong feature extractor right from the beginning of incremental learning steps. This extractor is trained on the larger first task, which provides a substantial amount of data (i.e., 50% of all available classes) (Hou et al., 2019; Zhu et al., 2022; Petit et al., 2023), or it starts from a large pre-trained model that remains unchanged (Hayes & Kanan, 2020a; Wang et al., 2022c) that eliminates the problem of representational drift (Yu et al., 2020). However, these methods perform poorly when little training data is available upfront. In Fig. 1, we illustrate both CIL setups, with and without the more significant first task. The trend is evident when we have a lot of data in the first task - results steadily improve over time. However, the

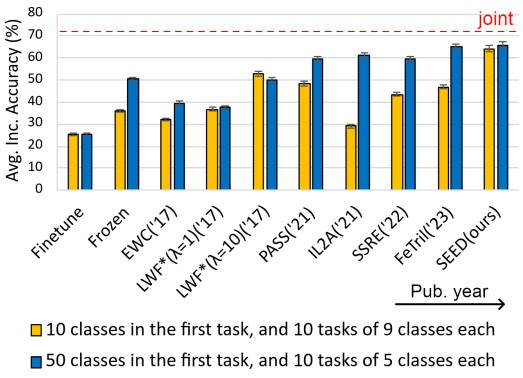

Figure 1: Exemplar-free Class Incremental Learning methods evaluated on CIFAR100 divided into eleven tasks for two different data distributions.

*Code: https://github.com/grypesc/SEED

progress is not evident for the setup with equal splits, where a frozen (or nearly frozen by high regularization) feature extractor does not yield good results. This setup is more challenging as it requires the whole network to continually learn new features (*plasticity*) and face the problem of catastrophic forgetting of already learned ones (*stability*).

One solution for this problem is architecture-based CIL methods, notably by expanding the network structure beyond a single model. Expert Gate (Aljundi et al., 2017) creates a new expert, defined as a neural network, for each task to mitigate forgetting. However, it can potentially result in unlimited growth in the number of parameters. Therefore, more advanced ensembling solutions, like CoSCL (Wang et al., 2022b), limit the computational budget using a fixed number of experts trained in parallel to generate features ensemble. In order to prevent forgetting, regularization is applied to all ensembles during training a new task, limiting their plasticity. Doan et al. (2022) propose ensembling multiple models for continual learning with exemplars for experience-replay. To perform efficient ensembling and control a number of the model's parameters, they enforce the model's connectivity to keep several ensembles fixed. However, exemplars are still necessary, and as in CoSCL, task-id is required during the inference.

As a remedy for the above issues, we introduce a novel ensembling method for exemplar-free CIL called *SEED: **S**election of **E**xperts for **E**nsemble **D**iversification*. Similarly to CoSCL and (Doan et al., 2022), SEED uses a fixed number of experts in the ensemble. However, only a single expert is updated while learning a new task. That, in turn, mitigates forgetting and encourages diversification between the experts. While only one expert is being trained, the others still participate in predictions. In SEED, the training does not require more computation than single-model solutions. The right expert for the update is selected based on the current ensemble state and new task data. The selection aims to limit representation drift for the classifier. The ensemble classifier uses multivariate Gaussian distribution representation associated with each expert (see Fig. 2). At the inference time, Bayes classification from all the experts is used for a final prediction. As a result, SEED achieves state-of-the-art accuracy for task-aware and task-agnostic scenarios while maintaining the high plasticity of the resulting model under different data distribution shifts within tasks.

In conclusion, the main contributions of our paper are as follows:

- We introduce SEED, a new method that leverages an ensemble of experts where a new task is selectively trained with only a single expert, which mitigates forgetting, encourages diversification between experts and causes no computational overhead during the training.
- We introduce a unique method for selecting an expert based on multivariate Gauss distributions of each class in the ensemble that limits representational drift for a selected expert. At the inference time, SEED uses the same rich class representation to perform Bayes classification and make predictions in a task-agnostic way.
- With the series of experiments, we show that existing methods that start CIL from a strong feature extractor later during the training mainly focus on stability. In contrast, SEED also holds high plasticity and outperforms other methods without any assumption of the class distribution during incremental learning sessions.

2 Related Work

Class-Incremental Learning (CIL) represents the most challenging and prevalent scenario in the field of Continual Learning research (Van de Ven & Tolias, 2019; Masana et al., 2022), where during the evaluation task-id is unknown, and the classifier has to predict all classes seen so far. The simplest solution to fight catastrophic forgetting in CIL is to store exemplars, e.g. LUCIR (Hou et al., 2019), BiC (Wu et al., 2019), Foster (Wang et al., 2022a), WA (Zhao et al., 2020). Having exemplars greatly simplifies learning cross-task features. However, storing exemplars can not always be an option due to privacy issues or other limitations. Then, the hardest scenario *exemplar-free* CIL is considered, where number of methods exists: LwF (Li & Hoiem, 2016), SDC (Yu et al., 2020), ABD (Smith et al., 2021), PASS (Zhu et al., 2021b), IL2A (Zhu et al., 2021a), SSRE (Zhu et al., 2022), FeTrIL (Petit et al., 2023). Most of them favor stability and alleviate forgetting through various forms of regularization applied to an already well-performing feature extractor. Some approaches even concentrate solely on the incremental learning of the classifier while keeping the backbone network frozen (Petit et al., 2023). However, freezing the backbone can limit the plasticity and not be sufficient for more complex

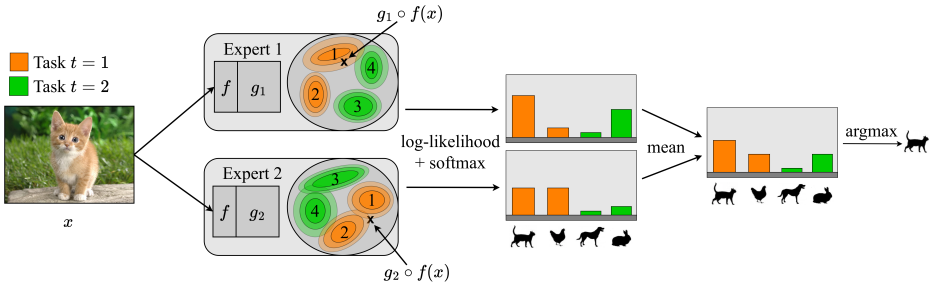

Figure 2: SEED comprises K deep network experts $g_k \circ f$ (here $K = 2$), sharing the initial layers f for higher computational performance. f are frozen after the first task. Each expert contains one Gaussian distribution per class $c \in C$ in his unique latent space. In this example, we consider four classes, classes 1 and 2 from task 1 and classes 3 and 4 from task 2. During inference, we generate latent representations of input x for each expert and calculate its log-likelihoods for distributions of all classes (for each expert separately). Then, we softmax those log-likelihoods and compute their average over all experts. The class with the highest average softmax is considered as the prediction.

settings, e.g., when tasks are unrelated, like in CTrL (Veniat et al., 2020). This work specifically aims at exemplar-free CIL, where the model's plasticity in learning new features for improved classification is still considered an essential factor.

Growing architectures and ensemble. Architecture-based methods for CIL can dynamically adjust some networks' parameters while learning new tasks, i.e. DER (Yan et al., 2021), Progress and Compress (Rusu et al., 2016) or use masking techniques, e.g. HAT (Serrà et al., 2018). In an extreme case, each task can have a dedicated expert network (Aljundi et al., 2017) or a single network per class (van de Ven et al., 2021). That greatly improves plasticity but also requires increasing resources as the number of parameters increases. Additionally, while the issue of forgetting is addressed, transferring knowledge between tasks becomes a new challenge. A recent method, CoSCL (Wang et al., 2022b), addresses this by performing an ensemble of a limited number of experts, which are diversified using a cooperation loss. However, this method is limited to task-aware settings. Doan et al. (2022) diversifies the ensemble by training tasks on different subspaces of models and then merging them. In contrast to our approach, the method requires exemplars to do so.

Gaussian Models in CL. Exemplar-free CIL methods based on cross-entropy classifiers suffer recency bias towards newly trained task (Wu et al., 2019; Masana et al., 2022). Therefore, some methods employ nearest mean classifiers with stored class centroids (Rebuffi et al., 2017; Yu et al., 2020). SLDA (Hayes & Kanan, 2020b) assigns labels to inputs based on the closest Gaussian, computed using the running class means and covariance matrix from the stream of tasks. In the context of continual unsupervised learning (Rao et al., 2019), Gaussian Mixture Models were used to describe new emerging classes during the CL session. Recently, in (Yang et al., 2021), a fixed, pre-trained feature extractor and Gaussian distributions with diagonal covariance matrices were used to solve the CIL problem. However, we argue that such an approach has low plasticity and limited applicability. Therefore, we propose an improved method based on multivariate Gaussian distributions and multiple experts that can learn new knowledge efficiently.

3 METHOD

The core idea of our approach is to directly diversify experts by training them on different tasks and combining their knowledge during the inference. Each expert contains two components: a feature extractor that generates a unique latent space and a set of Gaussian distributions (one per class). The overlap of class distributions varies across different experts due to disparities in expert embeddings. SEED takes advantage of this diversity, considering it both during training and inference.

Architecture. Our approach, presented in Fig. 2, consists of K deep network experts $g_k \circ f$ for $k = 1, \ldots, K$, sharing the initial layers f for improving computational performance. f are frozen after the first task. We consider the number of shared layers a hyperparameter (see Appendix A.3).

Moreover, each expert k contains one Gaussian distribution $G_k^c = (\mu_k^c, \Sigma_k^c)$ per class c for its unique latent space.

Algorithm. During inference, we perform an ensemble of Bayes classifiers. The procedure is presented in Fig. 2. Firstly, we generate representations of input x for each expert k as $r_k = g_k \circ f(x)$. Secondly, we calculate log-likelihoods of r_k for all distributions G_k^c associated with this expert

$$l_k^c(x) = -\frac{1}{2}[\ln(|\Sigma_k^c|) + S\ln(2\pi) + (r_k - \mu_k^c)^T(\Sigma_k^c)^{-1}(r_k - \mu_k^c)], \tag{1}$$

where S is the latent space dimension. Then, we softmax those values $\widehat{l_k^1}, \ldots, \widehat{l_k^{|C|}} = \text{softmax}(l_k^1, \ldots, l_k^{|C|}; \tau)$ per each expert, where C is the set of classes and τ is a temperature. Class c with the highest average value after softmax over all experts (highest $\mathbb{E}_k \widehat{l_k^c}$) is returned as a prediction for task agnostic setup. For task aware inference, we limit this procedure to classes from the considered task.

Our training assumes T tasks, each corresponding to the non-overlapping set of classes $C_1 \cup C_2 \cup \cdots \cup C_T = C$ such that $C_t \cap C_s = \emptyset$ for $t \neq s$. Moreover, task t is a training step with only access to data $D_t = \{(x, y)|y \in C_t\}$, and the objective is to train a model performing well both for classes of a new task and classes of previously learned tasks ($< t$).

The main idea of training SEED, as presented in Fig. 3, is to choose and finetune one expert for each task, where the chosen expert should correspond to latent space where distributions of new classes overlap the least. Intuitively, this strategy causes latent space to change as little as possible, improving stability.

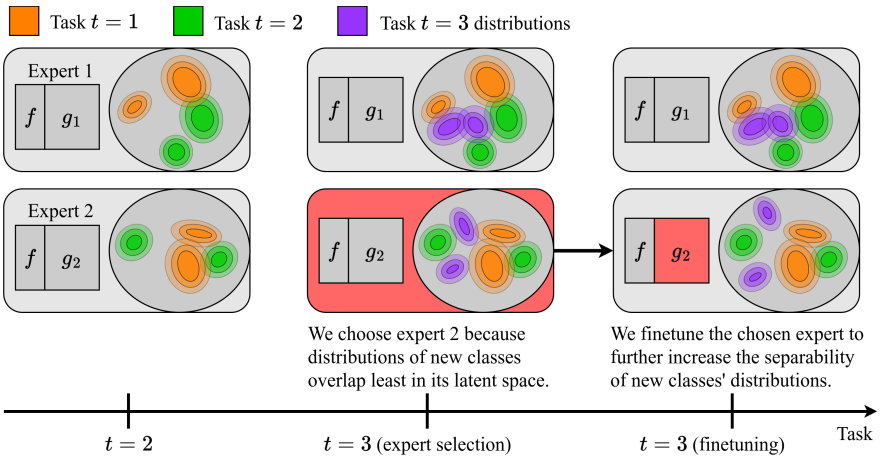

Figure 3: SEED training process for $K = 2$ experts, $T = 3$ tasks, and $|C_t| = 2$ classes per task. When the third task appears with novel classes C_3, we analyze distributions of C_3 classes (here represented as purple distributions) in latent spaces of all experts. We choose the expert where those distributions overlap least (here, expert 2). We finetune this expert to increase the separability of new classes further and move to the next task.

To formally describe our training, let us assume that we are in the moment of training when we have access to data $D_t = \{(x, y)|y \in C_t\}$ of task t for which we want to finetune the model. There are two steps to take, selecting the optimal expert for task t and finetuning this expert.

Expert selection starts with determining the distribution for each class $c \in C_t$ in each expert k. For this purpose, we pass all x from D_t with $y = c$ through deep network $g_k \circ f$. This results in a set of vectors in latent space for which we approximate a multivariate Gaussian distribution $q_{c,k}$. In consequence, each expert is associated with a set $Q_k = \{q_{1,k}, q_{2,k}, ..., q_{|C_t|,k}\}$ of $|C_t|$ distributions. We select expert $\bar{k}$ for which those distributions overlap least using symmetrized Kullback–Leibler divergence d_{KL}:

$$\bar{k} = \underset{k}{\text{argmax}} \sum_{q_{i,k}, q_{j,k} \in Q_k} d_{KL}(q_{i,k}, q_{j,k}), \tag{2}$$

To finetune the selected expert $\bar{k}$, we add the linear head to its deep network and train $g_{\bar{k}}$ using D_t set. As a loss function, we use cross-entropy combined with feature regularization based on knowledge distillation (Li & Hoiem, 2016) weighted with α: $L = (1-\alpha)L_{CE} + \alpha L_{KD}$, where $\mathcal{L}_{KD} = \frac{1}{|B|}\sum_{i \in B} ||g_{\bar{k}} \circ f(x_i) - g^{old}_{\bar{k}} \circ f(x_i)||$, B is a batch and $g^{old}_{\bar{k}}$ is frozen $g_{\bar{k}}$.

While we use CE for its simplicity and effective clustering (Horiguchi et al., 2019), it can be replaced with other training objectives, such as self-supervision. Then, we remove the linear head, update distributions of $Q_{\bar{k}}$, and move to the next task.

Due to the random expert initializations, we skip the selection procedure for K initial tasks and omit L_{KD}. Instead, we select the expert with the same number as the number task ($k = t$) and use $L = L_{CE}$. For the same reason, we calculate distributions of new tasks only for the experts trained so far ($k \leq t$). Finally, we fix f after the first task so that finetuning one expert does not affect others.

4 Experiments

In order to evaluate the performance of SEED and fairly compare it with other models, we utilize three commonly used benchmark datasets in the field of Continual Learning (CL): CIFAR-100 (Krizhevsky, 2009) (100 classes), ImageNet-Subset (Deng et al., 2009) (100 classes) and DomainNet (Peng et al., 2019) (345 classes, from 6 domains). DomainNet contains objects in very different domains, allowing us to measure models' adaptability to new data distributions. We create each task with a subset of classes from a single domain, so the domain changes between tasks (more extensive data drift). We always set $K = 5$ for SEED, so it consists of 5 experts. We evaluate all Continual Learning approaches in three different task distribution scenarios. We train all methods from scratch. Detailed information regarding experiments and the code are in the Appendix. We compare all methods with standard CIL evaluations using the classification accuracies after each task, and *average incremental accuracy*, which is the average of those accuracies (Rebuffi et al., 2017). We train all methods from scratch in all scenarios.

The first scenario is the CIL equal split setting, where each task has the same number of classes. This weakens the feature extractor trained on the first task, as there is little data. Therefore, this scenario better exposes the methods' plasticity. We reproduce results using FACIL(Masana et al., 2022), and PyCIL(Zhou et al., 2021) benchmarks for this setting. We train all methods using random crops, horizontal flips, cutouts, and AugMix (Hendrycks et al., 2019) data augmentations.

The second scenario is similar to the one used in (Hou et al., 2019), where the first task is larger than the subsequent tasks. This equips CIL methods with a more robust feature extractor than the equal split scenario. Precisely, the first task consists of either 50% or 40% of all classes. This setting allows methods that freeze the feature extractor (low plasticity) to achieve good results. We take baseline results for this setting from (Petit et al., 2023).

The third scenario is task incremental on equal split tasks (where the task id is known during inference). Here, the baseline results and numbers of models' parameters are taken from (Wang et al., 2022b). We perform the same data augmentations as in this work.

4.1 Results

Tab. 1 presents the comparison of SEED and state-of-the-art exemplar-free CIL methods for CIFAR-100, DomainNet, and ImageNet-Subset in the equal split scenario. We report average incremental accuracies for various split conditions and domain shift scenarios (DomainNet). We present joint training as an upper bound for the CL training.

SEED outperforms other methods by a large margin in each setting. For CIFAR-100, SEED is better than the second-best method by 14.7, 17.5, and 15.6 percentage points for $T = 10, 20, 50$, respectively. The difference in results increases as there are more tasks in the setting. More precisely, for $T = 10$, SEED has 14.7 percentage points better accuracy than the second-best method (LwF*, which is LwF implementation with PyCIL (Zhou et al., 2021) data augmentations and learning rate schedule). At the same time, for $T = 50$ SEED is better by 15.6%. The results are consistent for other datasets, proving that SEED achieves state-of-the-art results in an equal split scenario. Moreover, based on DomainNet results, we conclude that SEED is also better in scenarios with a significant distributional shift. Detailed results for CIFAR100 T=50 and DomainNet T=36 are presented in

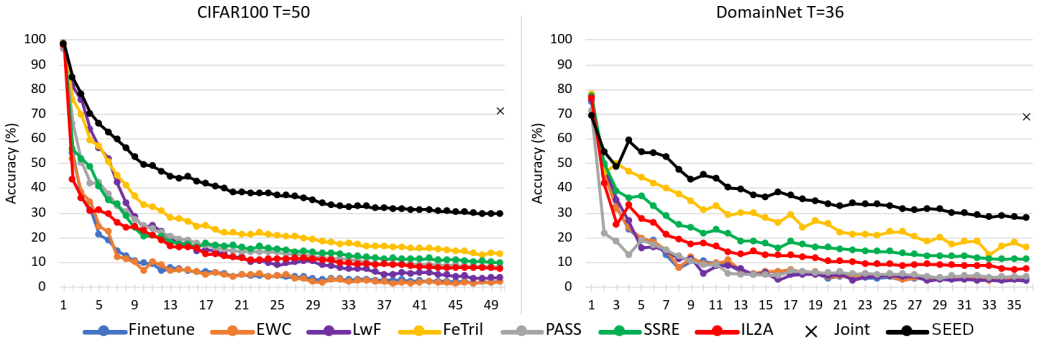

Figure 4: Class incremental accuracy achieved after each task for equal splits on CIFAR100 and DomainNet. SEED significantly outperforms other methods in equal split scenarios for many tasks (left) and more considerable data shifts (right).

Fig. 4. In this extreme setting, where each task consists of just little data, SEED results in significantly higher accuracies for the last tasks than other methods.

Table 1: Task-agnostic avg. inc. accuracy (%) for equally split tasks on CIFAR-100, DomainNet and ImageNet-Subset. The best results are in bold. SEED achieves superior results compared to other methods and outperforms the second best method (FeTrIL) by a large margin.

CIL Method	CIFAR-100 (ResNet32)			DomainNet			ImageNet-Subset
	T=10	T=20	T=50	T=12	T=24	T=36	T=10
Finetune	26.4±0.1	17.1±0.1	9.4±0.1	17.9±0.3	14.8±0.1	10.9±0.2	27.4±0.4
EWC (Kirkpatrick et al., 2017) (PNAS'17)	37.8±0.8	21.0±0.1	9.2±0.5	19.2±0.2	15.7±0.1	11.1±0.3	29.8±0.3
LwF* (Rebuffi et al., 2017) (CVPR'17)	47.0±0.2	38.5±0.2	18.9±1.2	20.9±0.2	15.1±0.6	10.3±0.7	32.3±0.4
PASS (Zhu et al., 2021b) (CVPR'21)	37.8±1.1	24.5±1.0	19.3±1.7	25.9±0.5	23.1±0.5	9.8±0.3	-
IL2A (Zhu et al., 2021a) (NeurIPS'21)	43.5±0.3	28.3±1.7	16.4±0.9	20.7±0.5	18.2±0.4	16.2±0.4	-
SSRE (Zhu et al., 2022) (CVPR'22)	44.2±0.6	32.1±0.9	21.5±1.8	33.2±0.7	24.0±1.0	22.1±0.7	45.0±0.5
FeTrIL (Petit et al., 2023) (WACV'23)	46.3±0.3	38.7±0.3	27.0±1.2	33.5±0.6	33.9±0.5	27.5±0.7	58.7±0.2
SEED	**61.7±0.4**	**56.2±0.3**	**42.6±1.4**	**45.0±0.2**	**44.9±0.2**	**39.2±0.3**	**67.8±0.3**
Joint		71.4±0.3		63.7±0.5	69.3±0.4	69.1±0.1	81.5±0.5

Large first task class incremental scenarios. We present results for this setting in Tab. 2. For CIFAR-100, SEED is better than the best method (FeTrIL) by 4.6, 4.1, and 1.4 percentage points for $T = 6, 11, 21$, respectively. For $T = 6$ on ImageNet-Subset, SEED is better by 3.3 percentage points than the best method. However, with more tasks, $T = 11$ or $T = 21$, FeTrIL with a frozen feature extractor presents better average incremental accuracy.

We can notice that simple regularization-based methods such as EWC and LwF* are far behind more recent ones: FeTrIL, SSRE, and PASS, which achieve high levels of overall average incremental accuracy. However, these methods benefit from a larger initial task, where a robust feature extractor can be trained before incremental steps. In SEED, each expert can still specialize for a different set of tasks and continually learn more diversified features even with using regularization like LwF. The difference between SEED and other methods is noticeably smaller in this scenario than in the equal split scenario. This fact proves that SEED works better in scenarios where a strong feature extractor must be trained from scratch or where there is a domain shift between tasks.

Task incremental with limited number of parameters. We investigate the performance of SEED in task incremental scenarios. We compare it against another state-of-the-art task incremental ensemble method - CoSCL (Wang et al., 2022b) and follow the proposed limited number of models' parameters setup. We compare SEED to: HAT (Serrà et al., 2018), MARK (Hurtado et al., 2021), and BNS (Qin et al., 2021). Tab. 3 presents the results with the number of utilized parameters. Our method requires significantly fewer parameters than other methods and achieves better average incremental accuracy. Despite being designed to solve the exemplar-free CIL problem, SEED outperforms other task-incremental learning methods. Additionally, we check how the number of shared layers (f function) affects SEED's performance. Increasing the number of shared layers decreases required parameters but negatively impacts task-aware accuracy. As such, the number of shared layers in SEED is a

Table 2: Comparison of CIL methods on ResNet18 and CIFAR-100 or ImageNet-Subset under larger first task conditions. We report task-agnostic avg. inc. accuracy from multiple runs. The best result is in bold. The discrepancy in results between SEED and other methods decreases compared to the equal split scenario.

CIL Method	CIFAR-100			ImageNet-Subset		
	T=6 $\lvert C_1\rvert$=50	T=11 $\lvert C_1\rvert$=50	T=21 $\lvert C_1\rvert$=40	T=6 $\lvert C_1\rvert$=50	T=11 $\lvert C_1\rvert$=50	T=21 $\lvert C_1\rvert$=40
EWC* (Kirkpatrick et al., 2017) (PNAS'17)	24.5	21.2	15.9	26.2	20.4	19.3
LwF* (Rebuffi et al., 2017) (CVPR'17)	45.9	27.4	20.1	46.0	31.2	42.9
DeeSIL (Belouadah & Popescu, 2018) (ECCVW'18)	60.0	50.6	38.1	67.9	60.1	50.5
MUC* (Liu et al., 2020) (ECCV'20)	49.4	30.2	21.3	-	35.1	-
SDC* (Yu et al., 2020) (CVPR'20)	56.8	57.0	58.9	-	61.2	-
ABD* (Smith et al., 2021) (ICCV'21)	63.8	62.5	57.4	-	-	-
PASS* (Zhu et al., 2021b) (CVPR'21)	63.5	61.8	58.1	64.4	61.8	51.3
IL2A* (Zhu et al., 2021a) (NeurIPS'21)	66.0	60.3	57.9	-	-	-
SSRE* (Zhu et al., 2022) (CVPR'22)	65.9	65.0	61.7	-	67.7	-
FeTrIL* (Petit et al., 2023) (WACV'23)	66.3	65.2	61.5	72.2	**71.2**	**67.1**
SEED	**70.9±0.3**	**69.3±0.5**	**62.9±0.9**	**75.5±0.4**	70.9±0.5	63.0±0.8
Joint		80.4			81.5	

hyperparameter that allows for a trade-off between achieved results and the number of parameters required for training.

Table 3: Limited parameters setting on CIFAR-100 with random class order. The reported metric is average task aware accuracy (%). Results for SEED are presented for various numbers of shared layers. Although we designed SEED for the task agnostic setting, it achieves superior results to exemplar-free, architecture-based methods using fewer parameters.

Approach	#Params.	20-split	50-split
HAT	6.8M	77.0	80.5
MARK	4.7M	78.3	-
BNS	6.7M	-	82.4
CoSCL(EWC+LWF)	4.6M	79.4±1.0	87.9±1.1
SEED	3.2M	**86.8±0.3**	**91.2±0.4**
SEED(1 shared)	3.2M	86.7±0.6	91.2±0.5
SEED(11 shared)	3.1M	85.6±0.3	89.6±0.2
SEED(21 shared)	**2.7M**	82.4±0.4	88.1±0.5

Table 4: Ablation study of SEED for CIL setting with T=10 on ResNet32 and CIFAR-100. Avg. inc. acc. is reported. Multiple components of SEED were ablated. SEED as-designed presents the best performance.

Approach	Acc.(%)
SEED(5 experts)	**61.7 ±0.4**
standard ensemble	56.9±0.4
weighted ensemble	57.0±0.5
CoSCL ensemble	57.3±0.4
w/o multivariate Gauss.	53.5±0.5
w/o covariance	54.1±0.3
w/o temp. in softmax	59.2±0.5
w/ ReLU	57.8±0.6

4.2 DISCUSSION

Is SEED better than other ensemble methods? We want to verify that the improved performance of our method comes from more than just forming an ensemble of classifiers. Hence, we compare SEED with the vanilla ensemble approach to continual learning, where all experts are initialized with random weights, trained on the first task, and sequentially fine-tuned on incremental tasks. The final decision is obtained by averaging the predictions of ensemble members. We present results in Tab. 4. Using the standard ensemble decreases the accuracy by 4.8%. We also experiment with the approaches where the predictions are weighted during inference by the confidence of the ensemble members (using prediction entropy, as in (Ruan et al., 2023) and where the experts are trained with additional *ensemble cooperation* loss from (Wang et al., 2022b). However, they yielded similar results to uniform weighting.

Diversity of experts Fig. 5 and Fig.11 (Appendix) depict the quality of each expert on various tasks and their respective contributions to the ensemble. It can be observed that experts specialize in tasks on which they were fine-tuned. For each task, there is always an expert who exhibits over 2.5% points better accuracy than the average of all experts. This demonstrates that experts specialize in different tasks. Additionally, the ensemble consistently achieves higher accuracy (ranging from 6% to 10% points) than the average of all experts on all tasks. Furthermore, the ensemble consistently outperforms the best individual expert, indicating that each expert contributes uniquely to the ensemble. See the

details in Fig. 10 (Appendix) for more analysis of overlap strategy from Eq. 2 that also presents how experts are diversified between the tasks.

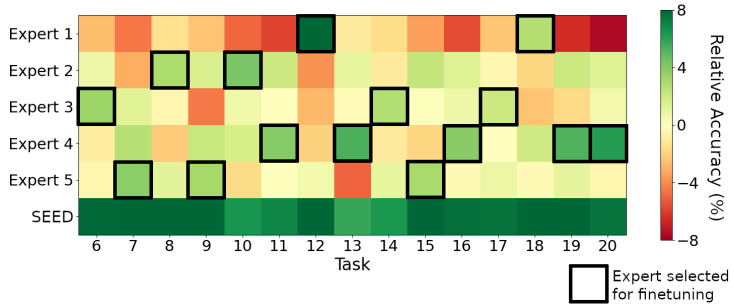

Figure 5: Diversity of experts on CIFAR-100 dataset with $T = 20$ split. The presented metric is relative accuracy (%) calculated by subtracting the accuracy of each expert from the averaged accuracy of all experts. Black squares represent experts selected to be finetuned on a given task. Although we do not impose any cost function associated with experts' diversity, they tend to specialize in different tasks by the design of our method. Moreover, our ensemble (bottom row) always performs better than the best expert, proving that each expert contributes uniquely to the ensemble in SEED.

Expert selection strategy. In order to demonstrate that our minimum overlap selection strategy (KL-max) improves the performance of the SEED architecture, we compare it to three other selection strategies. The first is a random selection strategy, where each expert has an equal probability of being chosen for finetuning. The second is a round-robin selection strategy, where for a task t, an expert with no. $1 + (t - 1 \bmod K)$ is chosen for a finetuning. The third one is the maximum overlap strategy (KL-min), in which we choose the expert for which the overlap between latent distributions of new classes is the highest. We conduct ten runs on CIFAR-100 with a Resnet32 architecture, three experts, and a random class order and report the average incremental accuracy in Fig. 6. Our minimum overlap selection strategy shows a higher mean and median than the other methods.

Ablation study. In Tab. 4, we present the ablation study for SEED. We report task-agnostic inc. avg. accuracy for five experts on CIFAR-100 and ResNet32, where results are averaged over three runs. Firstly, we remove or replace particular SEED components. We start by replacing the multivariate Gaussian distribution with its diagonal form. This reduces accuracy to 53.5%. Then, we remove Gaussian distributions and represent each class as a mean prototype in the latent space and use Nearest Mean Classifier (NMC) to make predictions. This also reduces accuracy, which shows that using multivariate distribution is important for SEED accuracy. Secondly, we check the importance of using temperature in the softmax function during inference. SEED without temperature ($\tau = 1$) achieves worse results than with temperature ($\tau = 3$), allowing more experts to contribute to the ensemble with more fuzzy likelihoods. At last, we analyze various SEED modifications, i.e., adding ReLU activations (like in the original ResNet) at the last layer, which decreased the accuracy by 3.9% points. It is because it is easier for the neural network trained with cross-entropy loss to represent features as Gaussian distribution if nonlinear activation is removed. See Tab. 7 for additional ablation study.

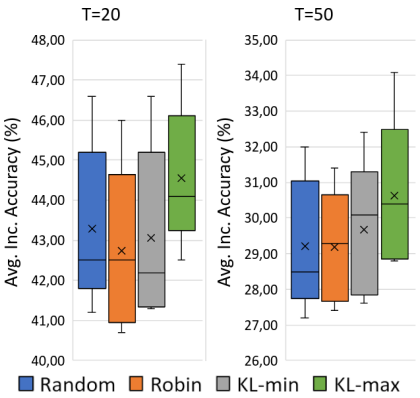

Figure 6: Avg. inc. accuracy of 10 runs with different class orderings for CIFAR-100 and different fine-tuning expert selection strategies for $T = 20, 50$ and three experts. Our KL-max expert selection strategy yields better results than random, round-robin, and KL-min.

Plasticity vs stability trade-off. SEED uses feature distillation in trained expert to alleviate forgetting. To assess the influence of the regularization on overall method performance, we use the forgetting and intransigence measures defined in (Chaudhry et al., 2018). Fig. 7 (left) shows the relationship between forgetting and intransigence for four different regularization-based methods: SEED, EWC as a parameters regularization-based method, LWF as regularization of network's output with distillation,

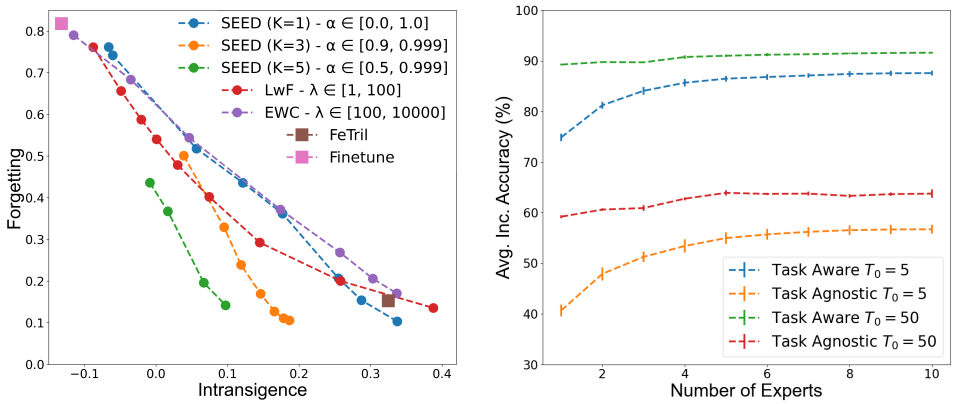

Figure 7: CIFAR-100. (Left) Forgetting and intransigence for different methods when manipulating the stability-plasticity parameters for $T = 10$. SEED with 5 experts achieves the best forgetting-intransigence trade-off. (Right) SEED accuracy as a function of a number of experts for $T = 20$ with 5 or 50 classes in the first task. Bars reflect standard dev. out of three runs.

and a recent FeTrIL method (Petit et al., 2023). For SEED, we adapt plasticity using the α and K parameter, and for both EWC and LWF, we change the λ parameter. FeTrIL method has no such parameter, as it uses a frozen backbone. The trade-off between stability and plasticity is evident. The FeTrIL model is very intransigent, with low plasticity and low forgetting. Plasticity is crucial in the CIL setting with ten or more tasks with an equal number of classes. Thus, EWC and LwF, need to be less rigid and exhibit more forgetting. The SEED model for $K = 3$ and $K = 5$, achieves much better results than FeTrIL while remaining less intransigent and more stable than LwF and EWC. By adjusting the α trade-off parameter of SEED, its stability can be controlled for any number of experts.

Number of experts. In Fig. 7 (right), we analyze how the number of experts influences the avg. incremental accuracy achieved by SEED. Changing the number of experts from 1 to 5 increases task-agnostic and task-aware accuracy by $\approx 15\%$ for $T_0 = 5$. However, for $T_0 = 50$, the increase is less significant (2% and 5% for task aware and task agnostic settings, respectively). These results suggest that scenarios with the significantly bigger first task are simpler than equal split ones. Moreover, going beyond five experts does not improve final CIL performance so much.

5 Conclusions

In this paper, we introduce an exemplar-free CIL method called SEED. It consists of a limited number of trained from scratch experts that all cooperate during inference, but in each task, only one is selected for finetuning. Firstly, this decreases forgetting, as only a single expert model's parameters are updated without changing learned representations of the others. Secondly, it encourages diversified class representations between the experts. The selection is based on the overlap of distributions of classes encountered in a task. That allows us to find a trade-off between model plasticity and stability. Our experimental study shows that the SEED method achieves state-of-the-art performance across several exemplar-free class-incremental learning scenarios, including different task splits, significant shifts in data distribution between tasks, and task-incremental settings. In the ablation study, we proved that each SEED component is necessary to obtain the best results.

Reproducibility and limitations of SEED We enclose the code in the supplementary material, and results can be reproduced by following the readme file. Our method has three limitations. Firstly, SEED may be not feasible for scenarios where tasks are completely unrelated and the number of parameters is limited, as in that case sharing initial parameters between experts may lead to a poor performance. Secondly, SEED requires the maximum number of experts given upfront, which can be found as a limitation of our method for new settings. Thirdly, calculating a distribution for a class may not be possible if the class's covariance matrix is singular. We address the last problem by decreasing latent space size. We elaborate more on this in the Appendix A.2.

ACKNOWLEDGMENTS

This research was supported by National Science Centre, Poland grant no 2020/39/B/ST6/01511, grant no 2022/45/B/ST6/02817, grant no 2022/47/B/ST6/03397, PL-Grid Infrastructure grant nr PLG/2022/016058, and the Excellence Initiative at the Jagiellonian University. This work was partially funded by the European Union under the Horizon Europe grant OMINO (grant number 101086321) and Horizon Europe Program (HORIZON-CL4-2022-HUMAN-02) under the project "ELIAS: European Lighthouse of AI for Sustainability", GA no. 101120237, it was also co-financed with funds from the Polish Ministry of Education and Science under the program entitled International Co-Financed Projects. Bartłomiej Twardowski acknowledges the grant RYC2021-032765-I. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Executive Agency. Neither the European Union nor European Research Executive Agency can be held responsible for them.

REFERENCES

Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *Conference on Computer Vision and Pattern Recognition*, CVPR, 2017.

Eden Belouadah and Adrian Popescu. Deesil: Deep-shallow incremental learning. *TaskCV Workshop @ ECCV 2018.*, 2018.

Arslan Chaudhry, Puneet Kumar Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XI*, Lecture Notes in Computer Science, pp. 556–572, 2018.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pp. 248–255, 2009.

Thang Doan, Seyed Iman Mirzadeh, and Mehrdad Farajtabar. Continual learning beyond a single model. *arXiv preprint arXiv:2202.09826*, 2022.

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

Tyler L. Hayes and Christopher Kanan. Lifelong machine learning with deep streaming linear discriminant analysis. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020a.

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*, pp. 220–221, 2020b.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, CVPR, 2016.

Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. In *International Conference on Learning Representations*, 2019.

Shota Horiguchi, Daiki Ikami, and Kiyoharu Aizawa. Significance of softmax-based features in comparison to distance metric learning-based features. *IEEE transactions on pattern analysis and machine intelligence*, 42(5):1279–1285, 2019.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 831–839, 2019.

Julio Hurtado, Alain Raymond, and Alvaro Soto. Optimizing reusable knowledge for continual learning via metalearning. *Advances in Neural Information Processing Systems*, 34:14150–14162, 2021.

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

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Zhizhong Li and Derek Hoiem. Learning without forgetting. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, volume 9908 of *Lecture Notes in Computer Science*, pp. 614–629, 2016.

Yu Liu, Sarah Parisot, Gregory Slabaugh, Xu Jia, Ales Leonardis, and Tinne Tuytelaars. More classifiers, less forgetting: A generic multi-classifier paradigm for incremental learning. In *European Conference on Computer Vision*, pp. 699–716. Springer, 2020.

Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost van de Weijer. Class-incremental learning: Survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2022.

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

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1406–1415, 2019.

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*, pp. 3911–3920, 2023.

Qi Qin, Wenpeng Hu, Han Peng, Dongyan Zhao, and Bing Liu. Bns: Building network structures dynamically for continual learning. *Advances in Neural Information Processing Systems*, 34: 20608–20620, 2021.

Dushyant Rao, Francesco Visin, Andrei A. Rusu, Razvan Pascanu, Yee Whye Teh, and Raia Hadsell. Continual unsupervised representation learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 7645–7655, 2019.

Leonardo Ravaglia, Manuele Rusci, Davide Nadalini, Alessandro Capotondi, Francesco Conti, and Luca Benini. A tinyml platform for on-device continual learning with quantized latent replays. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 11(4):789–802, 2021.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 5533–5542, 2017.

Yangjun Ruan, Saurabh Singh, Warren R. Morningstar, Alexander A. Alemi, Sergey Ioffe, Ian Fischer, and Joshua V. Dillon. Weighted ensemble self-supervised learning. In *11th International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, Conference Track Proceedings*, 2023.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *CoRR*, abs/1606.04671, 2016.

Joan Serrà, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholm, Sweden, July 10-15, 2018*, pp. 4555–4564, 2018.

James Seale 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 *IEEE/CVF International Conference on Computer Vision, ICCV 2021*, pp. 9354–9364. IEEE, 2021.

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

Gido M. van de Ven, Zhe Li, and Andreas S. Tolias. Class-incremental learning with generative classifiers. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2021, virtual, June 19-25, 2021*, pp. 3611–3620, 2021.

Tom Veniat, Ludovic Denoyer, and Marc'Aurelio Ranzato. Efficient continual learning with modular networks and task-driven priors. *arXiv preprint arXiv:2012.12631*, 2020.

Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. FOSTER: feature boosting and compression for class-incremental learning. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXV*, pp. 398–414, 2022a.

Liyuan Wang, Xingxing Zhang, Qian Li, Jun Zhu, and Yi Zhong. Coscl: Cooperation of small continual learners is stronger than a big one. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXVI*, pp. 254–271, 2022b.

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

Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 374–382, 2019.

Shipeng Yan, Jiangwei Xie, and Xuming He. DER: dynamically expandable representation for class incremental learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 3014–3023, 2021.

Yang Yang, Zhiying Cui, Junjie Xu, Changhong Zhong, Ruixuan Wang, and Wei-Shi Zheng. Continual learning with bayesian model based on a fixed pre-trained feature extractor. In *Medical Image Computing and Computer Assisted Intervention - MICCAI 2021 Strasbourg, France, September, 27 - October 1, 2021, Proceedings, Part V*, pp. 397–406, 2021.

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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 6980–6989. IEEE, 2020.

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 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 13205–13214. IEEE, 2020.

Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, and De-Chuan Zhan. Pycil: A python toolbox for class-incremental learning, 2021.

Fei Zhu, Zhen Cheng, Xu-yao Zhang, and Cheng-lin Liu. Class-incremental learning via dual augmentation. *Advances in Neural Information Processing Systems*, 34, 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*, pp. 5871–5880, 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*, pp. 9296–9305, 2022.

A APPENDICES

A.1 IMPLEMENTATION DETAILS

All experiments are the average over three runs and all methods are trained from scratch as in their original papers. We implemented SEED in FACIL (Masana et al., 2022) framework using Python 3 programming language and PyTorch (Paszke et al., 2019) machine learning library. We utilized a computer equipped with AMD EPYC™ 7742 CPU and NVIDIA A-100™ GPU to perform experiments. On this machine, SEED takes around 1 hour to be trained on CIFAR100 for $T = 10$.

For all experiments, SEED is trained using the Stochastic Gradient Descent (SGD) optimizer for 200 epochs per task, with a momentum of 0.9, weight decay factor equal 0.0005, α set to 0.99, τ set to 3 and an initial learning rate of 0.05. The learning rate decreases ten times after 60, 120, and 160 epochs. As the knowledge distillation loss, we employ the L2 distance calculated for embeddings in the latent space. We set the default number of experts to 5 and class representation dimensionality S to 64. In order to find the best hyperparameters for SEED, we perform a manual hyperparameter search on a validation dataset.

Tab. 1, Fig. 4 and Fig. 9. We perform experiments for all methods using implementations provided in FACIL and PyCIL (Zhou et al., 2021) frameworks. We use ResNet32 as a feature extractor for CIFAR100 and ResNet18 for DomainNet and ImageNet-Subset. For DomainNet $T = 12$, we use 25 classes per task; for $T = 18$, we use 10; for $T = 36$, we use 5.

All methods were trained using the same data augmentations: random crops, horizontal flips, cutouts, and AugMix (Hendrycks et al., 2019). For baseline methods, we set default hyperparameters provided in benchmarks. However, for LwF, we use $\lambda = 10$ as we observed that this significantly improved its performance.

Tab. 2. For baseline results, we provide results reported in (Petit et al., 2023). All CIL methods use the same data augmentations: random resized crops, horizontal flips, cutouts, and AugMix (Hendrycks et al., 2019).

Tab. 3. The setting and baseline results are identical to (Wang et al., 2022b). We train SEED with the same data augmentation methods as other methods: horizontal flips and random crops. Here, we use five experts consisting of the Resnet32 network.

Fig. 6 We calculate relative accuracy by subtracting each expert's accuracy from the average accuracy of all experts as in (Wang et al., 2022b). We perform 10 runs with random seeds.

Fig. 7. Below we report the range of used parameters for plotting the forgetting-intransigence curves (Fig. 7 - left).

- LwF: $\lambda \in \{1, 2, 3, 5, 7, 10, 15, 20, 25, 50, 100\}$
- EWC: $\lambda \in \{100, 500, 1000, 2500, 5000, 7500, 10000\}$
- SEED $K = 1$: $\gamma \in \{0.0, 0.25, 0.5, 0.9, 0.95, 0.97, 0.99, 0.999\}$
- SEED $K = 3$: $\gamma \in \{0.9, 0.95, 0.96, 0.97, 0.98, 0.99, 0.999\}$
- SEED $K = 5$: $\gamma \in \{0.5, 0.9, 0.97, 0.999\}$

A.2 RESNET ARCHITECTURE MODIFICATION

The backbone for the SEED method can be any popular neural network with its head removed. This study focuses on a family of modified ResNet (He et al., 2016) architectures.

ResNet architecture is a typical neural architecture used for the continual learning setting. In this work, we follow this standard. However, there are two minor changes to ResNet in our algorithm.

Due to ReLU activation functions placed at the end of each ResNet block, latent feature vectors of ResNet models consist of non-negative elements. That implies that every continuous random variable representing a class is defined on $[0; \infty)^S$, where S is the size of a latent vector. However, Gaussian distributions are defined for random variables of real values, which, in our case, reduces the ability to represent classes as multivariate Gaussian distributions. In order to alleviate this problem, we remove the last ReLU activation function from every block in the last layer of ResNet architecture.

Secondly, the size S of the latent sample representation should be adjustable. There are two reasons for that. Firstly, if S is too big and a number of class samples is low, Σ can be a singular matrix. This implies that the likelihood function might not be well-defined. Secondly, adjusting S allows us to reduce the number of parameters the algorithm requires. We overcome this issue by adding a 1x1 convolution layer with S kernels after the last block of the architecture. For example, this allows us to represent feature vectors of Resnet18 with 64 elements instead of 512.

A.3 MEMORY FOOTPRINT

SEED requires:

$$|\theta_f| + K|\theta_g| + \sum_{i=1}^{K}\sum_{j=i}^{T}|C_j|(S + \frac{S(S+1)}{2}) \tag{3}$$

parameters to run, where $|\theta_f|$ and $|\theta_g|$ represent number of parameters of f and g functions, respectively. S is dimensionality of embedding space, K is number of experts, T is number of tasks, $|C_j|$ is a number of classes in j-th task.

This total number of parameters used by SEED can be limited in several ways:

- Decreasing S by adding a convolutional 1x1 bottleneck layer at the network's end.
- Pruning parameters.
- Performing weight quantization.
- Using simpler feature extractor.
- Increasing number of shared layers (moving parameters from g into f function).
- Simplifying multivariate Gaussian distributions to diagonal covariance matrix or prototypes.

A.4 NUMBER OF PARAMETERS VS ACCURACY TRADE-OFF

To investigate the trade-off between the number of the SEED's parameters and the overall average incremental accuracy of the method, we conducted several experiments with a different number of experts and shared layers (as in a previous experiment in Tab. 3 we only adjust the number of layers). We see that these two factors indeed control and decrease the number of required parameters, e.g., sharing the first 25 layers in Resnet32 decreases memory footprint by 0.8 million parameters when we use five experts. However, it slightly hurts the performance of SEED, as the overall average incremental accuracy drops by 4.4%. These results, combined with expected forgetting/intransigence, can guide an application of SEED for a particular problem.

We additionally compare SEED to the best competitor - FeTrIL (with Resnet32) in a low parameters regime. FeTrIL stores feature extractor without the linear head and prototypes of each class. For SEED we utilize Resnet20 network with a number of kernels changed from 64 to 48 in the third block. We use 5 experts which share first 17 layers. We present results in Tab. 5. We present the number of network weights and total number of parameters which for SEED also includes multivariate Gaussian distributions. SEED has 13K less parameters than FeTrIL but achieves better accuracy on 3 settings.

Table 5: SEED outperforms competitors in terms of performance and number of parameters on equally split CIFAR100, however it requires decreasing size of the feature extractor network and sharing first 17 layers.

CIL Method	**Network**	**Network weights**	**Total params.**	CIFAR-100		
				T=10	*T*=20	*T*=50
EWC* (Kirkpatrick et al., 2017) (PNAS'17)	ResNet32	473K	473K	24.5	21.2	15.9
LwF* (Rebuffi et al., 2017) (CVPR'17)	Resnet32	473K	473K	45.9	27.4	20.1
FeTrIL (Petit et al., 2023) (WACV'23)	Resnet32	473K	473K	46.3±0.3	38.7±0.3	27.0±1.2
SEED	Resnet20*	339K	460K	54.7±0.2	48.6±0.3	33.1±1.1

A.5 PRETRAINING SEED

We study the impact of using a pretrained network with SEED on its performance. For this purpose we utilize ResNet-18 with weights pretrained on ImageNet-1K as a feature extractor for every expert.

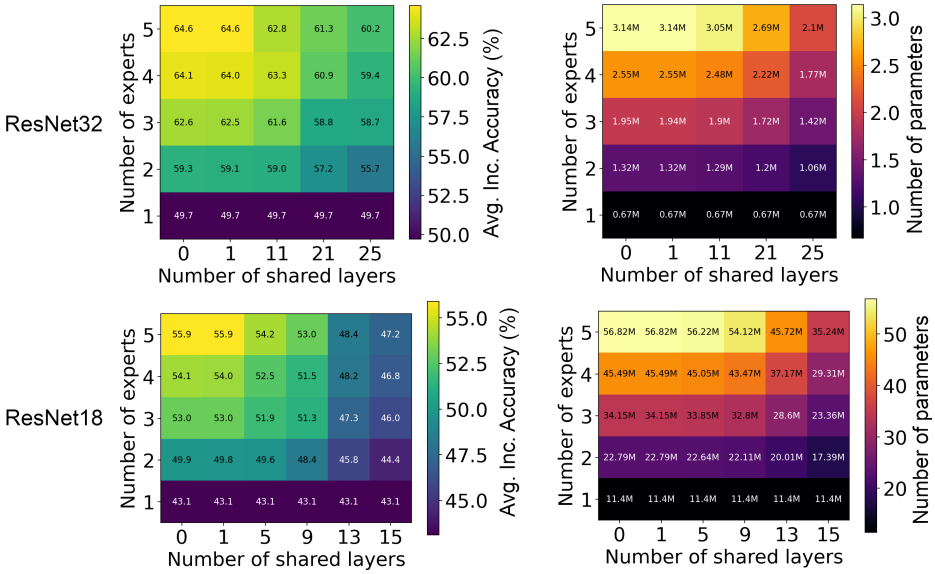

Figure 8: Impact of number of experts and number of shared layers on accuracy and number parameters of SEED. We utilize CIFAR100 with $|T| = 10$. We can observe that accuracy drops when decreasing the number of experts and increasing the number of shared layers.

DomainNet $\|T\|$	Avg. Inc. Accuracy (%)		Forgetting (%)	
	From scratch	Pretained	From scratch	Pretrained
12	45.0	53.1	12.1	12.8
24	44.9	54.2	11.2	12.1
36	39.2	53.6	13.7	15.6

Table 6: Pretraining experts in SEED increases its accuracy while slightly increasing forgetting. We compare ResNet-18 with weights pretrained on ImageNet-1K to a randomly initilized ResNet-18 as feature extractors of each expert.

We compare it to SEED initialized with random weights (training from scratch) in Tab.6. Pretrained SEED achieves better average incremental accuracy by: 8.1%, 9.3%, 14.4% on DomainNet split to 12, 24, 36 tasks respectively.

A.6 Additional results

In this section we provide more experimental results. Fig. 10 presents insight into diversity of experts on CIFAR100 for $T = 50$ and 5 experts. We measure value of overlap per expert in each task given by Eq. 2. Average value of the function differ between tasks, e.g., for task 49. it equals to ≈ 3.5, while for task 33. it equals to ≈ 18.0. This is due to semantic similarity between classes in a given task, classes in task 49. (cups, bowls) are semantically more similar then classes in task 33. (bed, dolphin). However, in each task we can observe significant variance between values for experts. This proves that classes overlap differently in experts, therefore experts are diversified what allows SEED to achieve great results.

In Fig. 9, we present accuracies obtained after equal split tasks for Table 1. We report results for CIFAR100 and DomainNet datasets. We can observe that for DomainNet SEED achieves 15% better accuracy after the last incremental step than FeTril. On CIFAR100 SEED achieves 20% and 16% better accuracy than the best method. This proves that SEED achieves superior results to state-of-the-art methods on equal-split and big domain shift settings.

Table 7 presents an additional ablation study for SEED. We test various ways to approximate class distributions in the latent space on CIFAR100 dataset and $T = 10$. Firstly, we replace multivariate Gaussian distribution with a Gaussian Mixture Model (GMM) consisting of 2 multivariate Gaussian

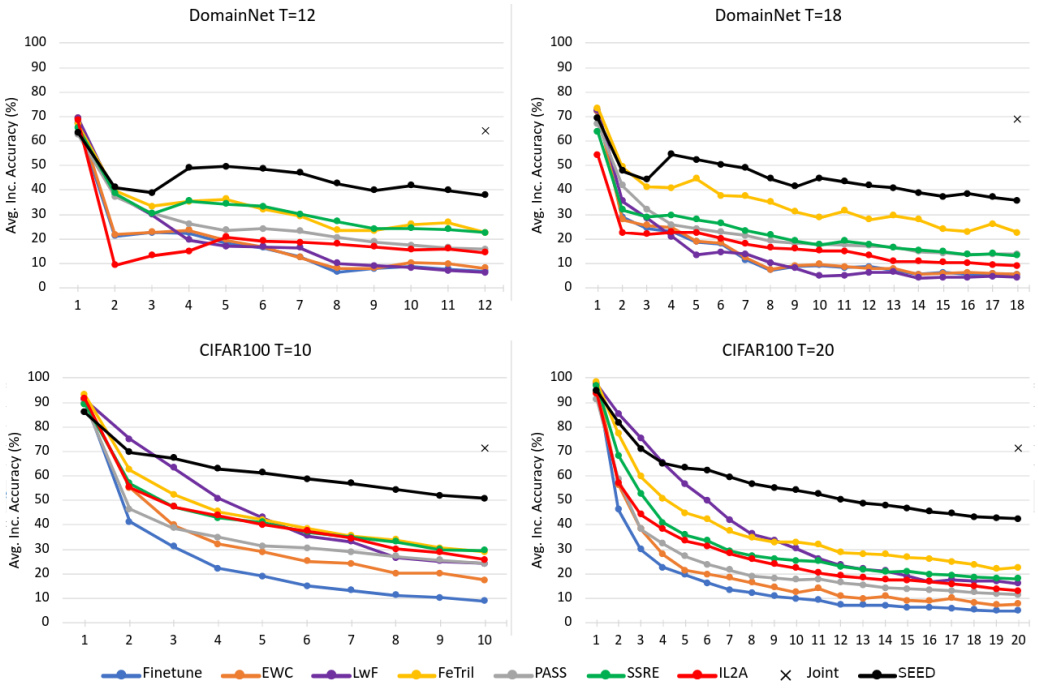

Figure 9: Accuracy after each task for equal splits on CIFAR100 and DomainNet. SEED significantly outperforms other methods in equal split scenarios for many tasks (top) and more considerable data shifts (bottom).

distributions. It slightly reduces the accuracy (by 1.2%). Then, we abandon multivariate Gaussians and approximate classes using 2 and 3 Gaussian distributions with the diagonal covariance matrix. That decreases accuracy by a large margin. We also approximate classes using their prototypes (centroids) in the latent space. This also reduces the performance of SEED.

Table 7: Ablation study of SEED for CIL setting with T=10 on ResNet32 and CIFAR-100. Avg. inc. accuracy is reported. We test different variations of class representation (such as Gaussian Mixture Model, diagonal covariance matrix or prototypes). SEED presents the best performance when used as designed.

Approach	Acc.(%)
SEED(5 experts)	**61.7** ±0.5
w/ 2× multivariate	60.5±0.7
w/ 2× diagonal	53.8 ±0.1
w/ 3× diagonal	53.8±0.3
w/ prototypes	54.1±0.3

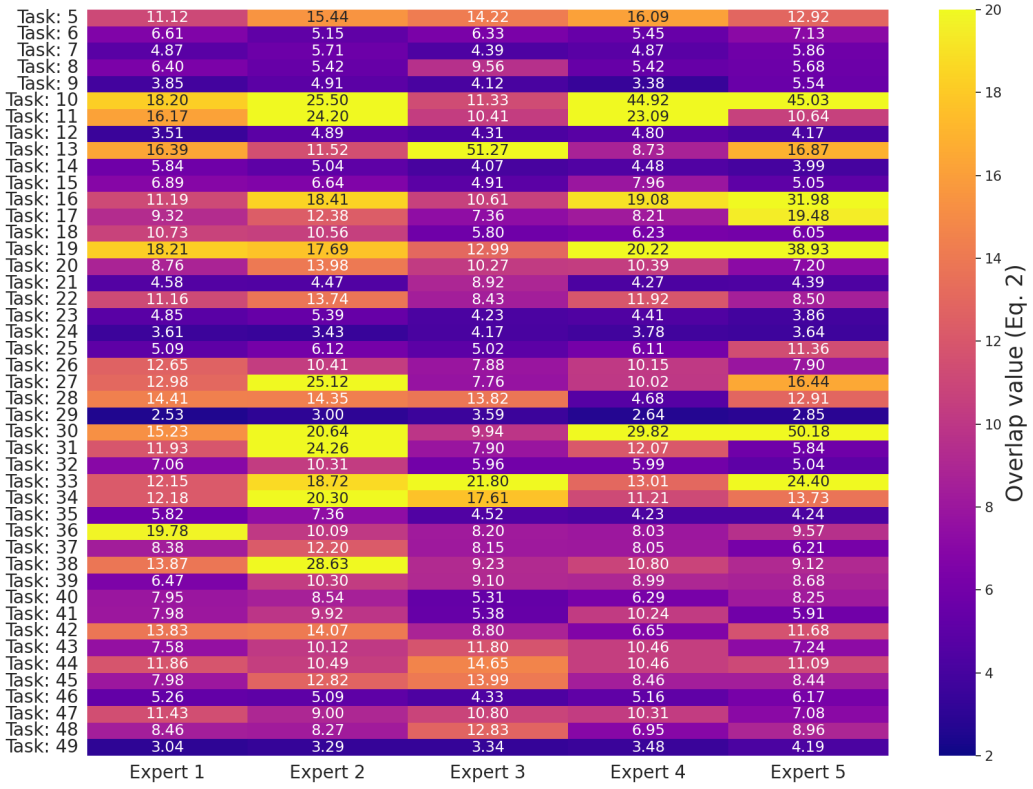

Figure 10: Overlap of class distributions in each task per expert on CIFAR-100 dataset with $T = 50$ split and random class order. Diversification by data yields high variation in overlap values in each task what proves that experts are diversified and learn different features. Average overlap values differ between tasks, as they depend on semantic similarity of classes. Classes in task 49. are semantically similar (cups, bowls) but classes in task 33. are different (beds, dolphins).

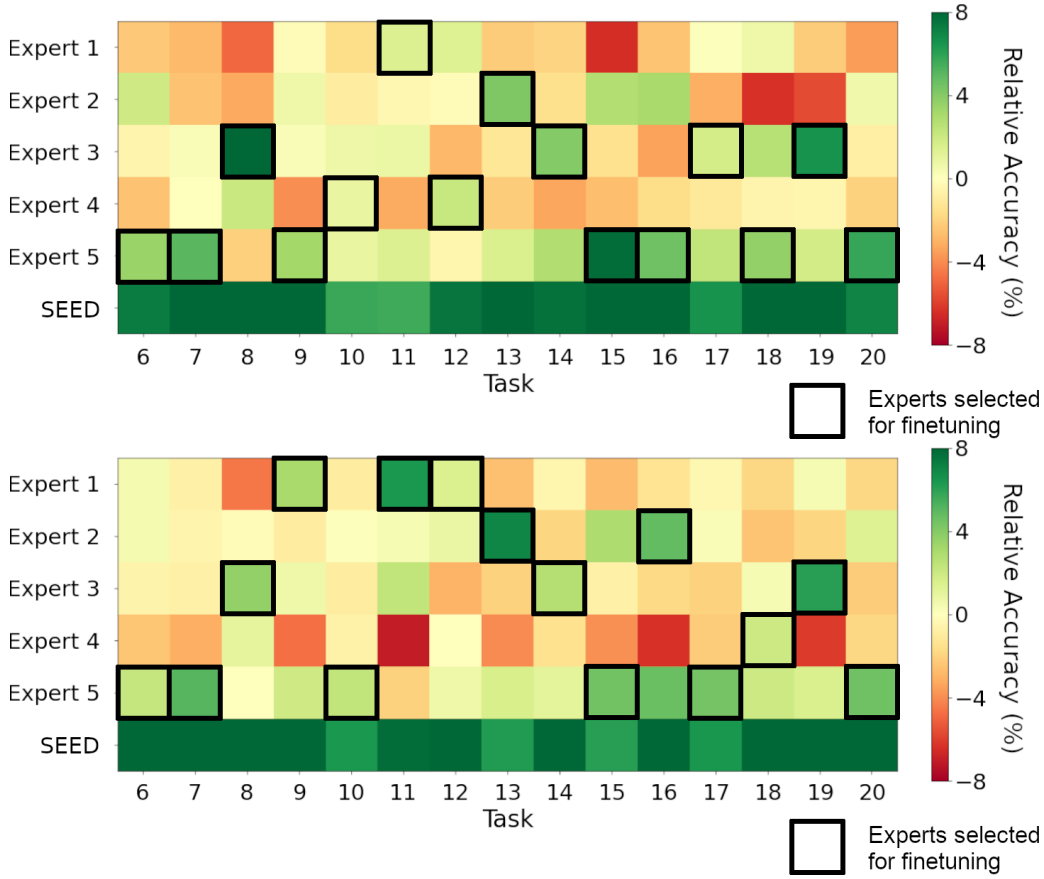

Figure 11: Diversity of experts on CIFAR-100 dataset with $T = 20$ split, different seeds and random class order. The presented metric is relative accuracy (%). Black squares represent experts selected to be finetuned on a given task. We can observe that experts specialize in different tasks.

Category Adaptation Meets Projected Distillation in Generalized Continual Category Discovery

Grzegorz Rypeść*[1,2], Daniel Marczak[1,2], Sebastian Cygert[1,3], Tomasz Trzciński[1,2,4], and Bartłomiej Twardowski[1,5,6]

[1] IDEAS NCBR
[2] Warsaw University of Technology
[3] Gdańsk University of Technology
[4] Tooploox
[5] Autonomous University of Barcelona
[6] Computer Vision Center

Abstract. Generalized Continual Category Discovery (GCCD) tackles learning from sequentially arriving, partially labeled datasets while uncovering new categories. Traditional methods depend on feature distillation to prevent forgetting the old knowledge. However, this strategy restricts the model's ability to adapt and effectively distinguish new categories. To address this, we introduce a novel technique integrating a learnable projector with feature distillation, thus enhancing model adaptability without sacrificing past knowledge. The resulting distribution shift of the previously learned categories is mitigated with the auxiliary category adaptation network. We demonstrate that while each component offers modest benefits individually, their combination – dubbed CAMP (Category Adaptation Meets Projected distillation) – significantly improves the balance between learning new information and retaining old. CAMP exhibits superior performance across several GCCD and Class Incremental Learning scenarios. The code is available on Github.

1 Introduction

Traditional machine learning models provide remarkable performances across various applications. However, they typically operate within the closed-world scenario, which assumes that all the tasks are well-defined and knowledge necessary to solve them is given upfront. Unfortunately, those assumptions are far from real-life applications, where models encounter dynamic streams of ever-changing data that evolves and may contain semantics the models have never encountered before. This variability severely impacts their performance and robustness. To study these problems from a unified perspective, many recent works combine Generalized Category Discovery [36] (GCD) setting with Continual Learning [25, 27] (CL). Resulting combinations [20, 39, 44, 45] (to which we refer as Generalized Continual Category Discovery - GCCD), consider a sequence of tasks, in which the method

* Corresponding author, email: grzegorz.rypesc.dokt@pw.edu.pl

has to continuously handle new data, discover novel categories and maintain good performance on previous tasks by overcoming catastrophic forgetting [12].

A popular trend in the field of discovering categories is to regularize the model with feature distillation [18, 20, 31] (FD) to alleviate forgetting. However, as pointed out by [13], it decreases the plasticity of the model by restricting changes (drift) of the categories' distributions in the feature space. In this work, we show that in order to enable learning robust representations of new data, it is necessary to allow distributions of past categories to drift. Drawing inspiration from recent studies in representation learning for enhanced transferability [3, 34], we incorporate projected representation in a distillation-based regularization for continual learning. Compared to standard FD, this method results in improved plasticity. However, it increases forgetting as it lets old categories drift even more.

To address this issue, we represent categories as centroids in the latent space of the feature extractor and train an auxiliary network to predict the drift of centroids, which we call category adaptation. This enables the model to recover the actual positions of old clusters and improve the overall accuracy of the Nearest Centroid Classifier (NCC). Similarly, works [17, 42] analyze the semantic drift of old categories in closed-world, Continual Learning scenarios. Here, our key observation is that, while centroid adaptation demonstrates overall solid performance, its true effectiveness is realized only through combining it with feature distillation using a projection network, as depicted in Fig. 1. That allows the technique to excel. Building upon these findings, we propose to leverage this combined effect for GCCD by introducing a method called CAMP – Category Adaptation Meets Projected distillation. CAMP presents state-of-the-art performance across GCCD scenarios and exemplar-free Class Incremental Learning (CIL).

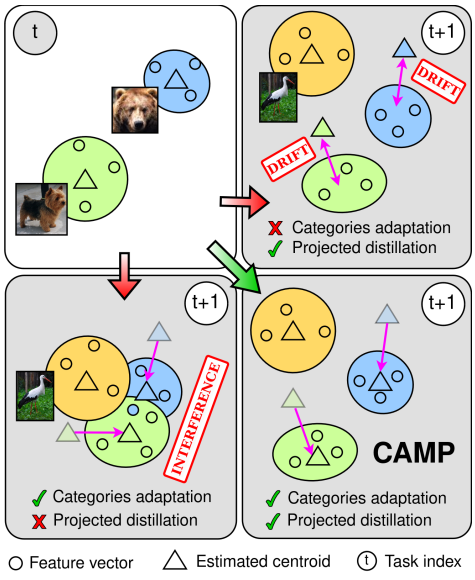

Fig. 1: While knowledge distillation through a projector and category adaption are decent on their own, CAMP combines them to achieve great results without the need of exemplars.

To summarize, the main contributions of our work are as follows. **1:** We study the interplay between knowledge distillation and category adaptation for the problem of Generalized Continual Category Discovery (GCCD). We demonstrate that they address distinct or even opposing challenges when considered separately. However, when combined, they complement each other significantly, reduce

forgetting, and improve overall results. **2:** Based on those insights, we propose a novel method for generalized continual category discovery dubbed CAMP. Built on GCD [36], CAMP improves it for continual learning by adapting categories between tasks and performing knowledge distillation with two separate projectors. **3:** We evaluate our method on numerous GCCD and Class Incremental Learning (CIL) datasets. With a series of experiments, we show that CAMP achieves state-of-the-art results with and without exemplars in different scenarios.

2 Related work

Continual learning (CL) is a setting where an agent learns a sequence of tasks with access to the data from the current task only. The goal is to achieve high performance on all the tasks from the sequence. Regularization-based approaches, such as EWC [22] or MAS [1] penalize changes to important neural network parameters. Alternatively, it is possible to regularize neuron activations as in LwF [25], DER [41] or PODNet [9] by using distillation techniques. However, even with feature distillation, the features from old classes will change, which causes catastrophic forgetting. [42] tries to predict these changes by approximating the drift of old classes based on the drift of current task data. [17] learns a feature adaptation network that translates old features into those learned on new tasks. In our work, we show that such an approach may not necessarily work well when going beyond the closed-world assumption under which CL usually operates.
Novel Class Discovery (NCL) [11, 14–16, 47, 48] aims to discover unknown classes from unlabeled data, thus it relaxes the closed-world assumption. NCL typically assumes that the class space of unlabeled data is completely disjoint with the labeled (training) set. This assumption was removed by Generalized Category Discovery (GCD) [36,38], where a training dataset consists of two parts: known classes where only a fraction of samples are labeled and novel classes which are unlabeled. The goal is to categorize all images from known and novel classes, without any additional information regarding their novelty. Typical solutions in this area include unsupervised clustering [16, 46], contrastive learning [36, 38, 46], pseudo-labeling [38] or pairwise similarities [43].
Continual classes discovery combines the NCL and CL fields. In [32], learning of the novel classes is framed incrementally: that is, in the first stage, the model learns from supervised data, and in the following stages, novel classes are learned without any labels. A similar setting was explored in [18] with the solution that uses pseudo-latent supervision, feature distillation, and a mutual information-based regularizer. However, their setting assumes no class overlap between the initial and incremental training stages, which was later relaxed by various to which we refer as Generalized Continual Category Discovery - GCCD [20, 39, 44, 45]. Grow and Merge (GM) [44] keeps two models: a static one that maintains knowledge of old classes and a dynamic one trained using self-supervision to discover novel classes. Both models are then merged into one when the new task arrives. In IGCD [45] a non-parametric classifier and a small buffer of exemplars from previous tasks to decrease forgetting. PA [20] combines

exemplars and proxy anchors [21] known from deep metric learning. Contrary to previous approaches [20, 44, 45] we focus on the most challenging exemplar-free setting. MetaGCD [39] needs to store the data from the first task, which we avoid, by proposing a purely continual model.

3 Method

3.1 Problem Setting

In Generalized Continual Category Discovery models are trained sequentially on partially labeled tasks containing known and novel categories. Formally, we consider a scenario where a dataset D is partitioned into disjoint subsets, each represented by a task: $D = T^1 \cup T^2 \cup \cdots \cup T^N$, with N denoting the total number of tasks. Each task T^t comprises two subsets: $T^t = T^t_L \cup T^t_U$, wherein $T^t_L = \{(x, y) \mid x \in \mathcal{X}^t_L, y \in \mathcal{Y}^t\}$ denotes the labeled set of known classes and $T^t_U = \{x \mid x \in \mathcal{X}^t_U\}$ encompasses unlabeled data samples from both known and novel classes.

Within the GCCD framework, each method engages in sequential learning from a stream of tasks. We assume that, at task t, each method comprises a feature extractor $\mathcal{F}^t : \mathcal{X}^t \to \mathcal{Z}^t$ and a classifier $C^t : \mathcal{Z}^t \to \mathcal{Y}^t$. Here, $\mathcal{Z}^t$ signifies the latent space generated by the feature extractor trained on the task t. The GCCD model aims to learn to recognize categories of a given task T^t while also maintaining the ability to recognize the categories from previous tasks $T^1 \cup T^2 \cup \cdots \cup T^{t-1}$. The main challenge is to prevent catastrophic forgetting on previous tasks while maintaining high plasticity that allows a method to learn new categories on new data.

3.2 Motivation

In this section, we demonstrate the motivation for our method on a simple experiment performed on ten categories of the CUB200 dataset split into two tasks. We train a neural network with its latent space bottlenecked to two dimensions (see Supplementary Material B for experimental details). In Fig. 2, we present a visualization of the latent space after training on the second task. We observe that naive fine-tuning of GCD (left) causes catastrophic forgetting and results in poor performance on the first task. Training with feature distillation (middle) [18, 20, 31] to some extent alleviates the forgetting and makes the drift of the centroids partially reversible via centroid adaptation (black arrows). However, feature distillation introduces rigid regularization that limits the model's ability to acquire new knowledge and results in low performance on the second task. These results highlight the need for a mechanism that relaxes anti-forgetting regularization constraints while alleviating forgetting. Our proposed CAMP method (right) is able to address these issues by introducing a learnable projector, which makes the regularization less rigid and allows the distribution to drift more, resulting in better accuracy on the second task. At the same time, it causes the

drift to be easily predictable, thus enabling our centroid adaptation to estimate ground truth positions of centroids, leading to great accuracy on the first task. Existing feature adaptation methods [17, 42] do not utilize projected distillation. Therefore, their drift is more challenging to predict (middle). Additionally, our method is more memory efficient than [17] as we store only a single centroid instead of many features per class and we don't require exemplars.

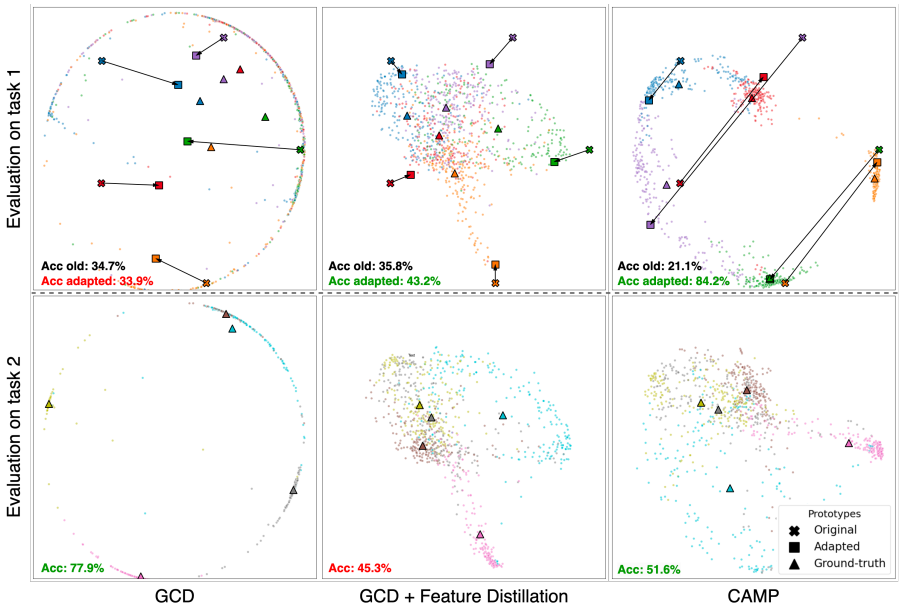

Fig. 2: CAMP utilizes a projected knowledge distillation, resulting in a predictable latent space drift. The drift is revertible via centroid adaptation (black arrows), maintaining high performance on the first task. GCD and GCD with feature distillation fail to prevent forgetting, resulting in a drift that is difficult to predict. This decreases their performance. We report the nearest centroid classification accuracy using stored (Acc old) and adapted centroids (Acc adapted) after training on the second task.

3.3 CAMP

The training procedure for CAMP within each task is divided into three consecutive phases: (1) feature extractor training, (2) clustering on a new task, and (3) adaptation of centroids representing categories of past classes. An overview is depicted in Fig. 3. The first phase focuses on training $\mathcal{F}^t$ to transform images into meaningful features that effectively distinguish between different classes. The second phase is dedicated to identifying novel classes. The third phase, centroid adaptation, is responsible for updating the centroids of previous tasks by esti-

mating the drift caused by updating the feature extractor. Detailed explanations of each phase will follow in the subsequent three subsections.

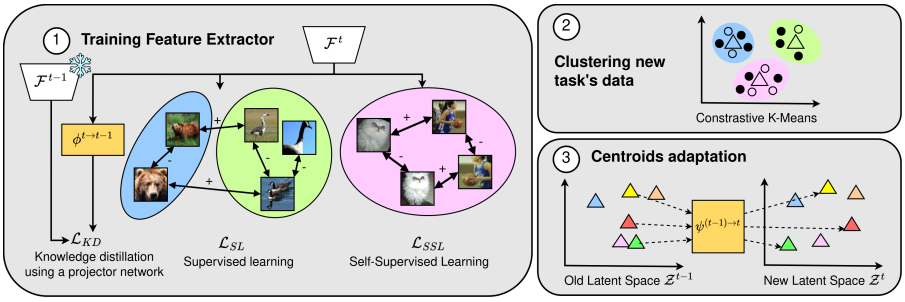

Fig. 3: The training procedure of CAMP consists of three stages: (1) We train the feature extractor in a semi-supervised manner using distillation through a learnable projector. (2) We obtain centroids of known and novel classes in the current task using constrastive K-Means algorithm. (3) We update memorized centroids of old categories to alleviate forgetting.

3.4 Phase 1: Feature extractor training

Our method for training feature extractors consists of three components: self-supervised learning, supervised learning, and knowledge distillation. The first two components allow the use of labeled and unlabeled data for the training feature extractor, while knowledge distillation is used for regularization and preventing forgetting. All are depicted in Fig. 3, left rounded rectangle.

Representation learning To learn the representations of the data from the new task, we follow SimGCD [38] and combine self-supervised and supervised learning. We use self-supervision on all data from known and novel categories T_t^U. We combine SimCLR [6] loss and cross-entropy loss between predictions and pseudo-labels: $\mathcal{L}_{SSL} = \mathcal{L}_{SimCLR} + \mathcal{L}_{pseudo}$. For labeled data T_t^L we use supervised learning loss that combines SupCon [19] loss and cross-entropy loss between predicted and ground-truth labels: $\mathcal{L}_{SL} = \mathcal{L}_{SupCon} + \mathcal{L}_{CE}$. A detailed formal description of each loss component is provided in Supplementary Material A.

Knowledge distillation We use distillation to alleviate forgetting in continual learning scenarios. Following state-of-the-art exemplar-free continual representation learning methods [10, 13], we use a learnable MLP projector $\phi^{t\to t-1} : \mathcal{Z}^t \to \mathcal{Z}^{t-1}$ that maps the features learned on the current task back to the old latent space. In our case, this approach improves the trade-off between plasticity and stability. Additionally, it ensures there is a mapping between old $(t-1)$ and new (t) latent space, which can be later inverted during category adaptation.

Following distillation-based continual learning methods [10, 13, 25], we freeze previously trained feature extractor $\mathcal{F}^{t-1}$ and regularize currently trained feature extractor $\mathcal{F}^t$ with the outputs of $\mathcal{F}^{t-1}$:

$$\mathcal{L}_{KD} = \sum_{i \in B} ||\phi^{t \to t-1}(\mathcal{F}^t(x_i)) - \mathcal{F}^{t-1}(x_i)||^2. \tag{1}$$

where B is a batch of data. This form of distillation does not require labels. Therefore, all the data from T_U^t can be used for regularization.

The final loss function for feature extractor training is defined as follows:

$$\mathcal{L} = (1-\alpha)((1-\beta)\mathcal{L}_{SSL} + \beta\mathcal{L}_{SL}) + \alpha\mathcal{L}_{KD}, \tag{2}$$

where $\alpha, \beta \in [0, 1]$ are hyperparameters defining contribution of regularization and supervision respectively.

3.5 Phase 2: Clustering task's data

The purpose of clustering the new task's data T^t is to find a centroid of each known and novel class in the learned representation space of the feature extractor $\mathcal{F}^t$. Such centroid is the class representation in latent space, later used for classification with the Nearest Centroid Classifier (NCC) during inference time. In order to find class centroids for CAMP, we use a semi-supervised k-means algorithm introduced in [36]. We use labeled data samples of known classes (T_L^t) to initialize known classes' centroids. We obtain the set of remaining centroids (equal to the number of novel classes) from the unlabeled data T_U^t using the k-means++ [2] algorithm, with the constraint that they must be close to the centroids of the labeled ones from T_L^t. To determine the number of classes in a task, we utilize a popular elbow method as in [16, 36]. Thus, for different numbers of K-Means clusters, we measure clustering accuracy on the labeled data and set the number of clusters to the one that yields the highest accuracy.

3.6 Phase 3: Centroid adaptation

Although regularization methods decrease forgetting, they still allow for a drift of ground truth representations of old classes (as discussed in Sec. 3.2). To recover from this issue, we propose to train auxiliary centroid adaptation network $\psi^{(t-1)\to t}$ to predict the drift of data representations from the previous task's latent space $(t-1)$ to the current one (t). To train it, we use all the data available in the current task. Exemplars can also be used for this purpose but are not necessary. We examine the architecture of $\psi^{(t-1)\to t}$ in Sec. 4.3. We train $\psi^{(t-1)\to t}$ by minimizing the MSE loss:

$$\mathcal{L}_{PA} = \sum_{i \in B} ||\mathcal{F}^t(x_i) - \psi^{(t-1)\to t}(\mathcal{F}^{t-1}(x_i))||^2. \tag{3}$$

When $\psi^{(t-1)\to t}$ is trained, we use it to calculate the updated centroids $p_i^t = \psi^{(t-1)\to t}(p_i^{t-1})$ where p_i^t is a centroid of class i calculated after task t. This is presented in Fig. 1, lower right.

4 Experiments

4.1 Experimental setup

We evaluate all methods on five datasets: CIFAR100 [24], Stanford Cars [23], CUB200 [37], FGVCAircraft [26] and DomainNet [28]. We split datasets into five equal tasks, with the exception of Stanford Cars (4 tasks) and DomainNet (6 tasks). Following [33] we set a different domain for each task of DomainNet, to simulate domain shifts. Following [36, 38], for each dataset, we set the ratio of known to novel classes as 4:1. For known classes, we set the ratio of labeled data samples to unlabeled data as 1:1. As the feature extractor $\mathcal{F}$ for all methods we use ViT-small [8] model pretrained with DINO [4] on ImageNet [7]. Following GCD [36] we freeze the first 11 blocks for all methods. We train all methods for 100 epochs in each task using AdamW optimizer with a starting learning rate of 0.1^3 and cosine annealing scheduling. The exception is PA method, for which we set it to 0.1^4 but increase the learning rate for proxies 100 times following the original work. As the distiller ϕ we utilize a 2-layer MLP network consisting of 384 neurons and ReLU activation. As the adapter ψ, we utilize a linear layer consisting of 384 neurons. For CAMP without exemplars, we set α to 0.5; for CAMP with exemplars, we use $\alpha = 0.1$.

Selection of baselines For the simplest baseline, we fine-tune GCD [36] method and couple it with popular CL regularization-based methods: (1) weight regularization method - EWC [22] (GCD+EWC); (2) feature distillation method [13, 25] (GCD+FD). In GCD+FD we distill features $\mathcal{F}(x)$. Moreover, we combine GCD with experience replay [30] (GCD+ER), where for incremental tasks, we add a buffer that stores random images of past known classes. For all the above methods, we perform semi-supervised k-means [36] algorithm to find centroids and utilize the NCC classifier for classification. Additionally, we use SimGCD [38], which improves GCD by utilizing a learnable classifier and adding additional cross-entropy-based losses for training it. We also test MetaGCD [39] that utilizes meta-learning and PA [20] method, which trains the feature extractor using supervised proxy anchor loss, maintains a buffer of exemplars, and uses feature distillation to alleviate forgetting. Lastly, we evaluate IGCD [45] that utilizes a density-based support set selection and replay buffer to mitigate forgetting. We describe all methods in detail in the Supplementary Material C.

Evaluation protocol We assess the performance of each method upon completion of the final task using the test set for each task. Our primary evaluation metrics encompass final task-agnostic (TAg) accuracy [5] (where the model does not know task-id during the inference), average forgetting [5] after last task and plasticity defined as $\frac{1}{N-1}\Sigma_{n=2}^{N} A_n$, where A_n denotes accuracy measured on n-th task after training on n-th task. Following [36, 38], we calculate them separately for known, novel, and all classes to get more insight into the performance of methods. We will release the code upon the acceptance of the manuscript.

Table 1: CAMP achieves state-of-the-art results in most of the scenarios. We report known, novel, and all accuracy (%) calculated over all classes after the last incremental learning step. We assume that task id is not known during the inference.

Method	Exemplars	CIFAR100			Stanford Cars			CUB200			FGVC Aircraft			DomainNet		
	per class	All	Known	Novel	All	Known	Novel	All	Known	Novel	All	Known	Novel	All	Known	Novel
GCD [36]	0	26.7	26.6	27.2	20.6	23.8	5.8	20.6	22.5	13.3	16.8	20.0	4.4	24.7	23.7	28.5
GCD+EWC [22]	0	26.9	28.7	19.7	30.4	35.3	8.7	50.3	53.3	38.1	25.3	28.2	13.8	33.1	37.5	16.8
GCD+FD	0	36.6	40.3	21.6	27.9	31.5	11.8	42.0	45.7	27.2	28.2	30.6	18.6	32.7	34.4	**39.7**
MetaGCD [39]	0	35.8	40.0	19.1	23.6	25.6	14.4	42.0	44.6	31.4	28.6	31.0	19.0	30.8	34.2	37.0
SimGCD [38]	0	31.2	31.9	28.5	21.3	30.1	23.9	35.7	36.6	31.8	22.0	22.3	20.8	36.5	36.4	35.0
CAMP	0	**52.1**	**55.7**	**37.8**	**48.8**	**52.8**	**31.0**	**58.9**	**62.6**	**44.2**	**39.9**	**42.0**	**31.5**	**36.7**	**39.2**	29.7
GCD+ER [30]	5	31.7	32.9	27.0	32.4	38.1	6.5	36.6	42.4	13.7	36.6	42.4	13.7	34.2	38.4	34.2
PA [20]	5	35.9	40.8	16.2	42.8	48.1	18.7	50.1	55.7	27.4	**40.5**	43.8	**27.6**	36.0	43.1	12.4
MetaGCD [39]	5	38.9	43.6	20.0	34.5	38.9	14.5	48.2	52.0	32.9	37.8	41.3	24.0	34.8	38.5	22.1
IGCD [45]	5	40.6	43.6	28.8	37.8	41.3	22.4	50.8	54.8	34.8	34.5	36.4	26.9	**45.9**	35.8	**43.4**
CAMP	5	**52.7**	**56.1**	**39.1**	**50.3**	**55.0**	**29.1**	**61.1**	**65.1**	**45.0**	40.3	**44.6**	23.0	45.1	**48.8**	29.6
GCD+ER [30]	20	42.5	46.3	27.3	46.3	54.0	11.5	52.0	59.9	20.1	48.9	56.5	17.5	46.4	53.7	20.4
PA [20]	20	37.6	43.1	15.7	46.7	53.0	18.7	53.1	58.7	30.4	45.6	50.5	26.0	38.2	42.0	25.3
MetaGCD [39]	20	43.6	47.3	28.6	47.9	51.5	31.8	55.8	56.9	42.2	46.5	49.7	33.9	42.4	47.6	25.8
IGCD [45]	20	51.8	57.3	29.6	57.6	63.9	29.7	64.0	69.0	43.4	49.5	52.6	**37.2**	51.1	58.3	22.0
CAMP	20	**58.0**	**63.6**	**35.7**	**60.5**	**66.5**	**33.6**	**66.1**	**71.6**	**43.9**	**50.8**	**56.7**	27.0	**56.9**	**64.3**	**26.6**

4.2 Comparison to baselines

We compare CAMP to other methods in terms of accuracy in Tab. 1. CAMP outperforms existing methods by a large margin in most scenarios, achieving the best results on CIFAR100, Stanford Cars, and CUB200 with and without exemplars, *e.g.* on CUB200, CAMP with no exemplars achieves 8.6%, 9.3%, 6.1% points better all, known and novel accuracy than the second-best method GCD+EWC. However, results are not that consistent on FGVCAircraft and DomainNet. On DomainNet GCD+FD method achieves 10.0% points better novel accuracy than CAMP, yet still, our method has 4.0% points better all accuracy. We can also study the impact of exemplars. Methods that utilize exemplars outperform exemplar-free competitors at the cost of additional memory complexity. Exemplars drastically improve GCD method, on CUB200 they boost its all accuracy by 16.0% and 31.4% points with 5 and 20 exemplars, respectively.

To better examine how the accuracy changes in incremental steps, we plot averages of all accuracy after each task in Fig. 4 (left). CAMP achieves the highest accuracy from the second task, and the results are consistent with and without exemplars. Next, we analyze the plasticity and forgetting of all methods in Fig. 4 (right). GCD is the most plastic method. However, it has the highest forgetting as it has no form of regularization. On the other hand, IGCD is the most stable as it achieves the lowest forgetting. CAMP has good plasticity and forgetting, allowing it to achieve state-of-the-art results. We can also observe that exemplars prevent methods like GCD, IGCD, PA, and CAMP from forgetting. That is intuitive as the replay buffer allows them to recall the past categories' data. Interestingly, each method forms its own cluster in presented plots, informing that they have different but specific characteristics for continual learning scenarios.

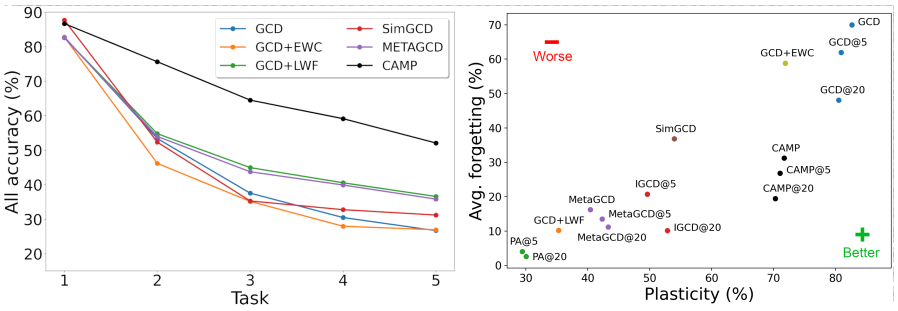

Fig. 4: Results of GCCD methods on CIFAR100 during the continual learning session of five tasks. We report average accuracy after each task (**left**); plasticity and forgetting (**right**). CAMP achieves better accuracy and than competitors.

Class Incremental Learning results Class incremental learning [27] is a special case of GCCD, where each data sample is labeled, and all classes are known. We test CAMP against exemplar-free state-of-the-art methods and train the feature extractor (ResNet-18) from scratch. We train methods for N equal tasks on CIFAR100, Domainnet, and Imagenet-Subset. For training CAMP we remove the self-supervised component from the loss function as all data is labeled. For baselines, we utilize popular exemplar-free CIL methods using their implementations in well-established benchmarks (FACIL [27], PyCIL [49]) and set their hyperparameters to default. We implement CAMP in FACIL and will publish the code upon the acceptance of the manuscript. For details, please refer to Supplementary Material B. We report average incremental accuracy [27], which is the average of task agnostic accuracies measured after each task on tasks seen so far. We provide results in Tab. 2.

CAMP outperforms baselines in terms of average incremental accuracy. On CIFAR100 it is better than the second best method - FeTrIL [29] by 6.5%, 10.4%, 4.3% for 5, 10, 20 tasks respectively. FeTrIL freezes the model after the first task, what decreases its ability to adapt to new data. CAMP does not, thus allowing for better plasticity. Results are consistent on DomainNet and ImageNet-Subset. These results show that CAMP translates well to Class Incremental Learning scenario, therefore combining projected distillation and category adaptation is a promising technique in the field of Continual Learning.

4.3 Method Analysis

How do distillers and adapters affect performance? We verify the influence of different network architectures of adapters and distillers on the accuracy achieved by CAMP. For this purpose, we evaluate three types of adapter networks ψ: linear, two connected layers with batch-norm and ReLU activation after the first layer (MLP) as in [6], and three layers following [35] (T-ReX). We also test a semantic drift compensation [42] (SDC) method to compare adaptation with different, non-learnable methods. It utilizes a vector field to estimate the drift

Table 2: Exemplar-free class incremental results for equal task split. We train all the methods from scratch and report task agnostic average incremental accuracy (%).

CIL Method	CIFAR-100			DomainNet			ImageNet-Subset		
	N=5	N=10	N=20	N=6	N=12	N=24	N=5	N=10	N=20
Fine-tune	40.6	26.4	17.1	27.7	17.9	14.8	41.6	27.4	18.6
GCD [36]	40.1	25.2	16.5	28.2	18.2	15.3	42.0	26.1	18.3
EWC [22]	52.9	37.8	21.0	30.0	19.2	15.7	43.7	29.8	19.1
LwF [25]	49.4	47.0	38.5	30.5	20.9	15.1	54.4	32.3	33.7
PASS [51]	62.2	52.0	41.4	33.6	25.9	14.9	63.6	53.9	36.0
IL2A [50]	-	43.5	28.3	31.3	20.7	18.2	-	-	-
SSRE [52]	57.0	44.2	32.1	34.9	33.2	24.0	56.9	45.0	32.9
FeTrIL [29]	58.5	46.3	38.7	36.8	33.5	27.5	62.9	58.7	43.2
CAMP	**65.0**	**56.7**	**43.0**	**43.6**	**36.6**	**27.9**	**73.1**	**59.9**	**50.2**
Joint		71.4		65.1	63.7	69.3		81.5	

of prototypes in the latent space, and here, we use it to adapt centroids. As a distiller network ϕ, we utilize popular feature distillation [18,20,31] technique and three networks of the same architecture as for adapters (Linear, MLP, T-ReX).

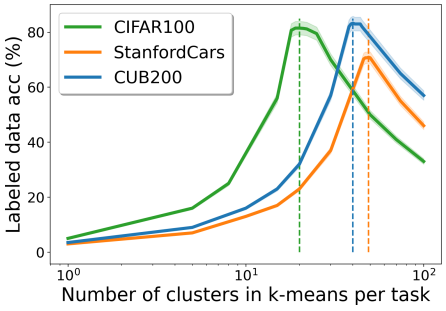

Fig. 5: Results for estimation of number of categories in tasks. We utilize an elbow method following vanilla GCD method. Accuracy calculated on labeled data peaks around ground truth number of clusters.

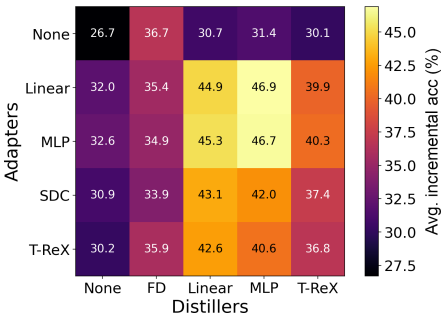

Fig. 6: Performance of CAMP for different adapting (ψ) and distilling (ϕ) networks. It achieves the best results when MLP distiller is combined with Linear or MLP adapter.

We provide results for CIFAR100 in Fig. 6. It is best to combine MLP distiller and linear adapter, which justifies the design choice for CAMP. Utilizing only MLP distiller improves base results by 4.7% points. However, combining MLP and linear adapter improves results by 20.2%, which is a significant gain. We also achieve 12% better accuracy than combination inspired by [17] (MLP adapter and FD distiller). This shows that combining projected distillation with adaptation is crucial for achieving good performance. Interestingly, all adaptation methods

improve no-adaptation results if FD distiller is not used. That suggests that FD changes old representation in a harder-to-predict way than other distillers.

Adaptation with feature distillation or projected distillation? We examine the impact of Feature Distillation (FD) and distillation with an MLP projector on centroid adaptation for α parameter value in range [0.03, 0.99] (plasticity-stability trade-off). We present the results in Fig. 7. We can see that without adaptation, FD imposes more substantial constraints on the drift of old centroids than MLP, as the distance between real and memorized centroids is lower (right). However, after adapting centroids this distance is much lower for MLP, proving that MLP facilitates drift prediction. Moreover, the adaptation method does not work well for FD with $\alpha \geq 0.4$, as the rigid features regularization minimizes the gain from centroid adaptation. The accuracy when using MLP is much better than FD, which proves the design choice of the distilling method.

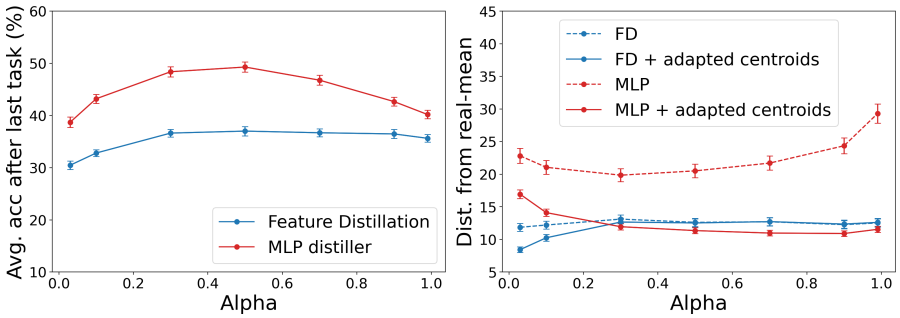

Fig. 7: MLP projector relaxes drifts of class distributions (measured with distance from memorized to ground-truth class centroids) compared to feature distillation. Additionally, it enables better adaptation of centroids, leading to increased accuracy.

Ablation study We perform ablation study on CUB200 and FGVCAircraft datasets. We present results in Tab. 3. We test CAMP without: $\mathcal{L}_{SL}$, $\mathcal{L}_{SSL}$, knowledge distillation through the projector loss ($\mathcal{L}_{KD}$) and without centroid adaptation. We can see that having all of these elements is crucial for obtaining good accuracy on known and novel classes (62.6% and 44.2% respectively on CUB200). Interestingly, without the $\mathcal{L}_{SSL}$ CAMP achieves 2.0% points better accuracy for known classes but 2.8% points lower accuracy for novel classes.

Estimating number of categories in a task. We estimate the number of novel categories in each task using the elbow method. We present results in Fig. 5. It is clear that accuracy on the labeled dataset peaks when the number of clusters for k-means algorithm is equal to the ground truth number of categories.

Does category adaptation improve position of centroids? After each task, we measure the distance between ground truth and estimated centroids with and without our category adaptation method. We present results in Fig. 8. Interestingly, we can see that novel classes tend to be further away from their

Table 3: Ablation study of CAMP.

$\mathcal{L}_{SL}$	$\mathcal{L}_{SSL}$	$\mathcal{L}_{KD}$	Cent. adapt.	CUB200		FGVCAircraft	
				Known	Novel	Known	Novel
✗	✓	✓	✓	29.9±1.6	12.1±1.1	19.0±0.9	18.8±1.9
✓	✗	✓	✓	64.6±2.3	41.4±3.5	43.5±1.8	16.6±2.0
✓	✓	✗	✓	30.9±2.6	32.9±2.6	24.7±1.2	19.8±2.2
✓	✓	✓	✗	36.5±1.9	28.7±1.7	29.6±1.7	24.2±2.8
✓	✓	✓	✓	62.6±2.4	44.2±3.7	42.0±1.9	31.5±3.5

estimated positions than known ones. Our centroid adaptation method consistently improves estimations of real centroids, *e.g.* after the last incremental step it decreases mismatch between positions of known classes by 44.8% and novel classes by 28.8% on CUB200.

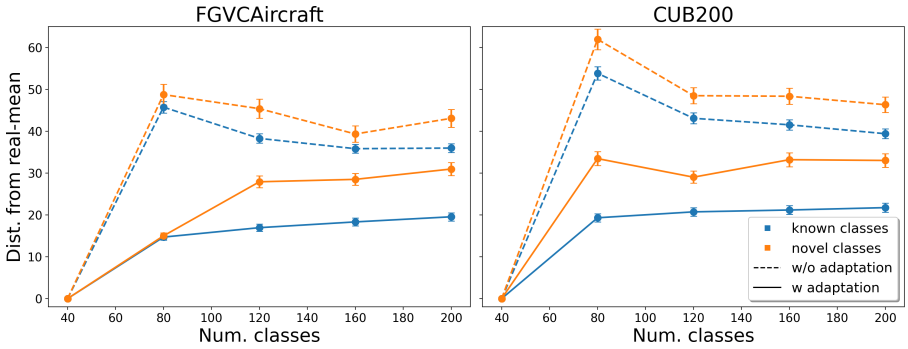

Fig. 8: Distance from centroids estimated by CAMP to ground-truth centroids throughout sequential training. Novel categories tend to be further away from estimated positions than known ones. CAMP centroid adaptation improves these estimations in both cases.

CAMP overcomes task-recency bias. To better understand the performance gap between GCD and CAMP, we check how these methods predict tasks after the last incremental step. We present a confusion matrix calculated for the StanfordCars dataset split into four tasks in Fig. 9. As a fine-tuning method, GCD faces a task-recency bias [40] for known and novel classes, as it mainly classifies test samples into categories of the last task. On the contrary, CAMP does not present such skewed results prediction only towards the last task. Our method of projected distillation, combined with category adaptation, alleviates task-recency bias, which improves CAMP's performance.

CAMP vs. GCD under different setting conditions. We examine the impact of setting parameters (number of novel classes and fraction of labeled data) on results achieved by CAMP and GCD. We set novel to known ratios as

2:8, 4:6, 6:4, 8:2 and fraction of labeled data as 0.25, 0.5, 0.75, 1.0. We provide final all accuracy and average forgetting on StanfordCars in Fig. 10. We can observe that accuracy decreases when increasing the number of novel classes or when decreasing the amount of labeled data. Forgetting behaves similarly, which is understandable - with lower accuracy, there is less to forget. However, in each case, CAMP outperforms GCD, proving that our combination of distillation and category adaptation is robust to different scenarios.

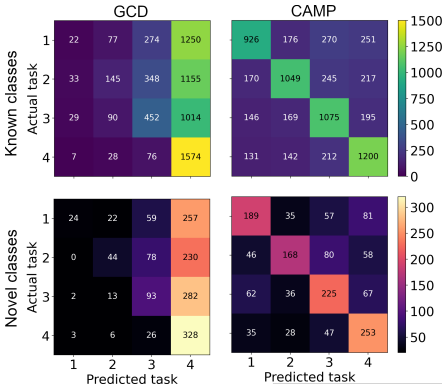

Fig. 9: Vanilla GCD method is prone to task-recency bias as its predictions are skewed towards the last one. However, CAMP overcomes this problem and keeps predictions distributed across all tasks.

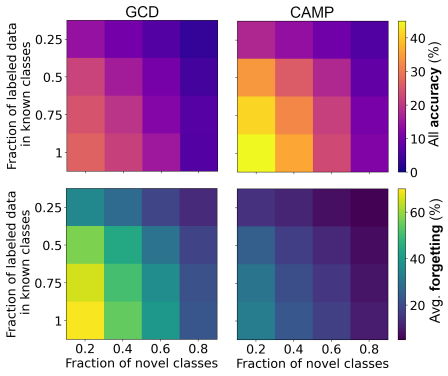

Fig. 10: CAMP improves GCD in different continual learning scenarios in terms of average accuracy and forgetting. We evaluate methods for different numbers of novel classes in each task and different fractions of labeled known data on StanfordCars.

5 Summary

In this work, we study an interplay between knowledge distillation techniques and centroid adaptation in the problem of GCCD and CIL. We find out that combining knowledge distillation through a neural projector followed by centroid adaptation is crucial for achieving high plasticity and low forgetting. Based on this finding, we propose a new method, dubbed CAMP. It utilizes projected knowledge distillation to regularize the network and prevent forgetting. Moreover, CAMP utilizes a centroid adaptation method to estimate better actual positions of centroids representing past categories. CAMP uses the nearest centroid classifier to classify data samples achieving state-of-the-art results in different scenarios.
Limitations. The limitation of our work is that the centroid adaptation mechanism cannot be easily implemented for methods that do not perform centroid-based clustering to represent categories, *e.g.* [38, 45]. Moreover, our method achieves better results for known classes, which can be limiting in use cases where novel classes play the leading role.

Acknowledgments

Daniel Marczak is supported by National Centre of Science (NCN, Poland) Grant No. 2021/43/O/ST6/02482. This research was partially funded by National Science Centre, Poland, grant no: 2020/39/B/ST6/01511, 2022/45/B/ST6/02817, and 2023/51/D/ST6/02846. Bartłomiej Twardowski acknowledges the grant RYC2021-032765-I. This paper has been supported by the Horizon Europe Programme (HORIZON-CL4-2022-HUMAN-02) under the project "ELIAS: European Lighthouse of AI for Sustainability", GA no. 101120237. We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2023/016613.

References

1. Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., Tuytelaars, T.: Memory aware synapses: Learning what (not) to forget. In: ECCV (2018)
2. Arthur, D., Vassilvitskii, S.: K-means++ the advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. pp. 1027–1035 (2007)
3. Bordes, F., Balestriero, R., Garrido, Q., Bardes, A., Vincent, P.: Guillotine regularization: Why removing layers is needed to improve generalization in self-supervised learning. Transactions on Machine Learning Research (2023)
4. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the International Conference on Computer Vision (ICCV) (2021)
5. Chaudhry, A., Dokania, P.K., Ajanthan, T., Torr, P.H.: Riemannian walk for incremental learning: Understanding forgetting and intransigence. In: Proceedings of the European conference on computer vision (ECCV). pp. 532–547 (2018)
6. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations (2020)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. ICLR (2021)
9. Douillard, A., Cord, M., Ollion, C., Robert, T., Valle, E.: Podnet: Pooled outputs distillation for small-tasks incremental learning. In: ECCV (2020)
10. Fini, E., da Costa, V.G.T., Alameda-Pineda, X., Ricci, E., Alahari, K., Mairal, J.: Self-supervised models are continual learners. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022. pp. 9611–9620. IEEE (2022)
11. Fini, E., Sangineto, E., Lathuilière, S., Zhong, Z., Nabi, M., Ricci, E.: A unified objective for novel class discovery. In: ICCV (2021)
12. French, R.M.: Catastrophic forgetting in connectionist networks. Trends in cognitive sciences **3**(4), 128–135 (1999)
13. Gomez-Villa, A., Twardowski, B., Yu, L., Bagdanov, A.D., van de Weijer, J.: Continually learning self-supervised representations with projected functional regularization (2022)
14. Han, K., Rebuffi, S.A., Ehrhardt, S., Vedaldi, A., Zisserman, A.: Automatically discovering and learning new visual categories with ranking statistics (2020)
15. Han, K., Rebuffi, S., Ehrhardt, S., Vedaldi, A., Zisserman, A.: Autonovel: Automatically discovering and learning novel visual categories. IEEE Trans. Pattern Anal. Mach. Intell. **44**(10), 6767–6781 (2022)
16. Han, K., Vedaldi, A., Zisserman, A.: Learning to discover novel visual categories via deep transfer clustering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8401–8409 (2019)
17. Iscen, A., Zhang, J., Lazebnik, S., Schmid, C.: Memory-efficient incremental learning through feature adaptation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16. pp. 699–715. Springer (2020)

18. Joseph, K.J., Paul, S., Aggarwal, G., Biswas, S., Rai, P., Han, K., Balasubramanian, V.N.: Novel class discovery without forgetting. In: Computer Vision - ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXIV. Lecture Notes in Computer Science, vol. 13684, pp. 570–586. Springer (2022)
19. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning (2021)
20. Kim, H., Suh, S., Kim, D., Jeong, D., Cho, H., Kim, J.: Proxy anchor-based unsupervised learning for continuous generalized category discovery. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16688–16697 (2023)
21. Kim, S., Kim, D., Cho, M., Kwak, S.: Proxy anchor loss for deep metric learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3238–3247 (2020)
22. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences **114**(13), 3521–3526 (2017)
23. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: 2013 IEEE International Conference on Computer Vision Workshops. pp. 554–561 (2013)
24. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. . (2009)
25. Li, Z., Hoiem, D.: Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence **40**(12), 2935–2947 (2017)
26. Maji, S., Rahtu, E., Kannala, J., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. arXiv preprint arXiv:1306.5151 (2013)
27. Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A.D., van de Weijer, J.: Class-incremental learning: survey and performance evaluation on image classification. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022)
28. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1406–1415 (2019)
29. Petit, G., Popescu, A., Schindler, H., Picard, D., Delezoide, B.: Fetril: Feature translation for exemplar-free class-incremental learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3911–3920 (2023)
30. Riemer, M., Cases, I., Ajemian, R., Liu, M., Rish, I., Tu, Y., Tesauro, G.: Learning to learn without forgetting by maximizing transfer and minimizing interference. arXiv preprint arXiv:1810.11910 (2018)
31. Roy, S., Liu, M., Zhong, Z., Sebe, N., Ricci, E.: Class-incremental novel class discovery. In: European Conference on Computer Vision. pp. 317–333. Springer (2022)
32. Roy, S., Liu, M., Zhong, Z., Sebe, N., Ricci, E.: Class-incremental novel class discovery. In: Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXIII. Lecture Notes in Computer Science, vol. 13693, pp. 317–333. Springer (2022)
33. Rypeść, G., Cygert, S., Khan, V., Trzciński, T., Zieliński, B., Twardowski, B.: Divide and not forget: Ensemble of selectively trained experts in continual learning. arXiv preprint arXiv:2401.10191 (2024)

34. Sariyildiz, M.B., Kalantidis, Y., Alahari, K., Larlus, D.: No reason for no supervision: Improved generalization in supervised models. In: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net (2023)
35. Sariyildiz, M.B., Kalantidis, Y., Alahari, K., Larlus, D.: No reason for no supervision: Improved generalization in supervised models. In: ICLR 2023-International Conference on Learning Representations. pp. 1–26 (2023)
36. Vaze, S., Han, K., Vedaldi, A., Zisserman, A.: Generalized category discovery. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7492–7501 (2022)
37. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011)
38. Wen, X., Zhao, B., Qi, X.: Parametric classification for generalized category discovery: A baseline study. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16590–16600 (2023)
39. Wu, Y., Chi, Z., Wang, Y., Feng, S.: Metagcd: Learning to continually learn in generalized category discovery. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1655–1665 (2023)
40. Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., Fu, Y.: Large scale incremental learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 374–382 (2019)
41. Yan, S., Xie, J., He, X.: Der: Dynamically expandable representation for class incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3014–3023 (2021)
42. Yu, L., Twardowski, B., Liu, X., Herranz, L., Wang, K., Cheng, Y., Jui, S., Weijer, J.v.d.: Semantic drift compensation for class-incremental learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6982–6991 (2020)
43. Zhang, S., Khan, S., Shen, Z., Naseer, M., Chen, G., Khan, F.S.: Promptcal: Contrastive affinity learning via auxiliary prompts for generalized novel category discovery. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3479–3488 (2023)
44. Zhang, X., Jiang, J., Feng, Y., Wu, Z.F., Zhao, X., Wan, H., Tang, M., Jin, R., Gao, Y.: Grow and merge: A unified framework for continuous categories discovery. Advances in Neural Information Processing Systems **35**, 27455–27468 (2022)
45. Zhao, B., Mac Aodha, O.: Incremental generalized category discovery. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19137–19147 (2023)
46. Zhao, B., Wen, X., Han, K.: Learning semi-supervised gaussian mixture models for generalized category discovery. arXiv preprint arXiv:2305.06144 (2023)
47. Zhong, Z., Fini, E., Roy, S., Luo, Z., Ricci, E., Sebe, N.: Neighborhood contrastive learning for novel class discovery. In: CVPR (2021)
48. Zhong, Z., Zhu, L., Luo, Z., Li, S., Yang, Y., Sebe, N.: Openmix: Reviving known knowledge for discovering novel visual categories in an open world (2020)
49. Zhou, D.W., Wang, F.Y., Ye, H.J., Zhan, D.C.: Pycil: a python toolbox for class-incremental learning. SCIENCE CHINA Information Sciences **66**(9), 197101– (2023)
50. Zhu, F., Cheng, Z., Zhang, X.y., Liu, C.l.: Class-incremental learning via dual augmentation. Advances in Neural Information Processing Systems **34** (2021)
51. Zhu, F., Zhang, X.Y., Wang, C., Yin, F., Liu, C.L.: Prototype augmentation and self-supervision for incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5871–5880 (2021)

52. Zhu, K., Zhai, W., Cao, Y., Luo, J., Zha, Z.J.: Self-sustaining representation expansion for non-exemplar class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9296–9305 (2022)

Supplementary Material: Category Adaptation Meets Projected Distillation in Generalized Continual Category Discovery

Grzegorz Rypeść*[1,2], Daniel Marczak[1,2], Sebastian Cygert[1,3], Tomasz Trzciński[1,2,4], and Bartłomiej Twardowski[1,5,6]

[1] IDEAS NCBR
[2] Warsaw University of Technology
[3] Gdańsk University of Technology
[4] Tooploox
[5] Autonomous University of Barcelona
[6] Computer Vision Center

A CAMP training loss in detail

In this section, we describe the details of CAMP method, more specifically – representation learning during phase 1. Representation learning in CAMP consists of contrastive and entropy-based learning, which we describe below and conclude in a total loss function for CAMP at the end.

Contrastive learning We use unsupervised and supervised contrastive losses, namely SimCLR [5] and SupCon [12].

We calculate SimCLR [5] loss as:

$$\mathcal{L}_{SimCLR} = -\frac{1}{|B|}\sum_{i\in B}\log\frac{\exp\left(\boldsymbol{h}_i^\top \boldsymbol{h}_i'/\tau\right)}{\sum_i^{i\neq n}\exp\left(\boldsymbol{h}_i^\top \boldsymbol{h}_n'/\tau\right)}\,, \tag{1}$$

where x_i and x_i' are two views (random augmentations) of the same image in a mini-batch B, $\mathbf{h}_i = g(\mathcal{F}(x_i))$, and τ is a temperature value. $\mathcal{F}$ is the feature extractor, and g is a multi-layer perceptron (MLP) projection head used in SimCLR method.

For data where labels are available (T_L^t) we can simply use supervised learning. Specifically, we employ SupCon [12] loss:

$$\mathcal{L}_{SupCon} = -\frac{1}{|B_L|}\sum_{i\in B_L}\frac{1}{|\mathcal{N}_i|}\sum_{q\in\mathcal{N}_i}\log\frac{\exp\left(\boldsymbol{h}_i^\top \boldsymbol{h}_q'/\tau\right)}{\sum_i^{i\neq n}\exp\left(\boldsymbol{h}_i^\top \boldsymbol{h}_n'/\tau\right)}\,, \tag{2}$$

where B_L corresponds to the labeled subset of B and $\mathcal{N}_i$ is the set of indices of other images that have the same label as x_i. We use the same projector as in SimCLR to produce representations $\mathbf{h}_i = g(\mathcal{F}(x_i))$ and the same temperature parameter τ.

* Corresponding author, email: grzegorz.rypesc.dokt@pw.edu.pl

Entropy based learning

We also utilize two additional loss functions for training the feature extractor following [26] and applied in the self-distillation [1,4] fashion. We noticed that this improved results achieved by CAMP even though we do not utilize parametrical classifier like [26]. Formally, for a total number of categories K, we randomly initialize a set of prototypes $\mathcal{C} = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K\}$, each representing one category. During training, we calculate the soft label for each augmented view $\boldsymbol{x}_i$ by performing softmax on cosine similarity between the hidden feature $\boldsymbol{z}_i = \mathcal{F}(x_i)$ and the prototypes $\mathcal{C}$ scaled by $1/\tau_s$:

$$\boldsymbol{p}_i^{(k)} = \frac{\exp\left(\frac{1}{\tau_s}(\boldsymbol{z}_i/||\boldsymbol{z}_i||_2)^\top(\boldsymbol{c}_k/||\boldsymbol{c}_k||_2)\right)}{\sum_{k'}\exp\left(\frac{1}{\tau_s}(\boldsymbol{z}_i/||\boldsymbol{z}_i||_2)^\top(\boldsymbol{c}_{k'}/||\boldsymbol{c}_{k'}||_2)\right)}, \tag{3}$$

and the soft pseudo-label $\boldsymbol{q}_i'$ is produced by another view $\boldsymbol{x}_i$ with a sharper temperature τ_t in a similar way. The loss functions are then cross-entropy loss $\ell(\boldsymbol{q}', \boldsymbol{p}) = -\sum_k \boldsymbol{q}'^{(k)} \log \boldsymbol{p}^{(k)}$ between the predictions and pseudo-labels:

$$\mathcal{L}_{pseudo} = \frac{1}{|B|}\sum_{i\in B}\ell(\boldsymbol{q}_i', \boldsymbol{p}_i) - \varepsilon H(\overline{\boldsymbol{p}}), \tag{4}$$

or known labels:

$$\mathcal{L}_{CE} = \frac{1}{|B_L|}\sum_{i\in B_L}\ell(\boldsymbol{y}_i, \boldsymbol{p}_i), \tag{5}$$

where $\boldsymbol{y}_i$ denotes the one-hot label of $\boldsymbol{x}_i$. For the unsupervised objective, we additionally adopt a mean-entropy maximisation regulariser [1]. Here, $\overline{\boldsymbol{p}} = \frac{1}{2|B|}\sum_{i\in B}(\boldsymbol{p}_i + \boldsymbol{p}_i')$ denotes the mean prediction of a batch, and the entropy $H(\overline{\boldsymbol{p}}) = -\sum_k \overline{\boldsymbol{p}}^{(k)} \log \overline{\boldsymbol{p}}^{(k)}$.

Total loss for CAMP On all data provided in a given task: $\mathcal{X}^t = \mathcal{X}_U^t + \mathcal{X}_L^t$, we calculate the following loss as $\mathcal{L}_{SSL} = \mathcal{L}_{SimCLR} + \mathcal{L}_{pseudo}$

Loss that we calculate only on the labeled data $\mathcal{X}_L^t$ is equal to $\mathcal{L}_{SL} = \mathcal{L}_{SupCon} + \mathcal{L}_{CE}$ Finally, the total loss function for CAMP is then equal to:

$$\mathcal{L}_{CAMP} = (1-\alpha)((1-\beta)\mathcal{L}_{SSL} + \beta\mathcal{L}_{SL}) + \alpha\mathcal{L}_{KD}, \tag{6}$$

where $\alpha, \beta \in [0, 1]$ are hyperparameters defining contribution of regularization and supervision respectively and $\mathcal{L}_{KD}$ is knowledge distillation loss calculated using a distiller and defined in Section 3 (main paper). In experimental section we set α to 0.5 or 0.1 (when exemplars are present) and β to 0.35. In case of Class Incremental Learning scenario we set α to 0.9.

B Experimental setup - details

B.1 Generalized Continual Class Discovery

In order to fairly compare existing methods, which often were created for a very specific continual scenario, we evaluate them in a Generalized Continual

Category Discovery framework (GCCD). GCCD consists of an arbitrary number of disjoint tasks and can include exemplars. Each task consists of labeled and unlabeled data from known and novel classes. We extend well-established category incremental setting [3, 21] by adding unlabeled data and novel classes to each task. We formally define it in Section 3 of the main paper. We present differences between GCCD and other setting in Tab.1.

Setting \ Characteristics	Methods	Partially labeled classes	Novel classes	Sequence of disjoint tasks	Arbitrary number of tasks
Class incremental learning	LwF [16], EWC [14], iCaRL [21], DER [3]	×	×	✓	✓
Self-supervised continual learning	CaSSLe [7], PFR [9], LUMP [17], POCON [8]	×	✓	✓	✓
Semi-supervised continual learning	NNCSL [11], CCIC [2], ORDisCo [25]	✓	×	✓	✓
Generalized category discovery	GCD [24], SimGCD [26]	✓	✓	×	×
Incremental generalized category discovery	IGCD [29]	✓	✓	×	✓
Continual generalized category discovery	PA [13]	✓	✓	✓	×
GCCD (ours)		✓	✓	✓	✓

Table 1: Generalized Continual Category Discovery is the most general setting. It includes partial labels, contrary to supervised and self-supervised learning, and requires discovering novel (unlabeled) classes, contrary to supervised and semi-supervised learning. Moreover, it works on a sequence of disjoint tasks, contrary to Incremental Generalized Category Discovery, which assumes that unlabeled data samples become labeled in the next task. Finally, unlike Continual Generalized Category Discovery, GCCD is not limited to only a single category incremental step.

Overall, the proposed setting in this paper is a generalization of the previous ones as we learn both known and novel classes in each of the learning stages and allow for partially-labeled classes, thus we make no distinction between the initial and subsequent learning stages. This scenario holds in many real-life applications, mostly when data comes sequentially and there are insufficient resources to label all images. Additionally, our experiments focus on equal split tasks (so without a sizeable first task) on many incremental steps and investigate the effect of different setting parameters, such as the fraction of novel classes or labeled data proportion, on models performance. For most GCCD experiments (Tab. 1, Fig.6, Fig.9 of the main body) we assume that methods know the number of novel classes due to simplicity purposes. We will publish the code of the framework and our method upon the acceptance of the manuscript. We present data splits which we used for evaluation of GCCD protocol in 2.

Exemplars In our experimental setting we utilize exemplars only for known classes as only they are given labels. We randomly choose which exemplars to store in the buffer. For each past, known class we store the same amount of exemplars.

B.2 Class Incremental Learning

Here, we provide setup details for results presented in Tab.2 of the main body of the paper. Class Incremental Learning [16, 18] is a special case of GCCD, where

Dataset	N	Known	Novel	Lab. known (%)
CIFAR100	5	16	4	50%
Stanford Cars	4	40	9	50%
CUB200	5	32	8	50%
Aircrafts	5	16	4	50%
DomainNet	6	8	2	50%

Table 2: Datasets we utilized in experiments, their splits and characteristics.

each data sample is labeled and there are no novel classes. To combat forgetting, typical approaches utilize exemplars [10, 21], expendable architectures [22, 27] or regularization [14, 16, 28] techniques. We compare CAMP to regularization baselines that consider a single neural network with constant number of parameters and no exemplars.

For the simplest baselines we utilize fine-tuning (no technique to combat forgetting) and joint that trains the network on the whole dataset. Then, we utilize LwF [16] which utilizes knowledge distillation, EWC [14] that regularizes weights of the neural network and PASS [32] as a prototype augmentation technique. Additionally, we use IL2A [31] and [20]. For all this methods we run their implementations in FACIL [18] and if the implementation is not available, we use PyCIL [30]. For all methods we use default hyperparameters and the same augmentations.

To compare CAMP to baselines, we utilize three commonly used benchmark datasets in the field of Continual Learning (CL): CIFAR-100 [15] (100 classes), ImageNet-Subset [6] (100 classes) and DomainNet [19] (345 classes, from 6 domains). DomainNet contains categories of different domains, allowing us to measure models' adaptability to new data distributions. We create each task with a subset of classes from a single domain, so the domain changes between tasks. We split datasets to N equal tasks.

We train CAMP and CIL baseline methods from scratch. We compare all approaches with standard CIL evaluations using the classification accuracies after each task, and *average incremental accuracy*, which is the average of those accuracies [21].

C Baseline methods details

In the following, we describe details of training feature extractors $\mathcal{F}$ of baseline methods.

GCD [24] combines contrastive unsupervised loss from Eq. 1 and supervised loss from Eq. 2 to train $\mathcal{F}^t$ on task t. Combined, the total $\mathcal{L}_{GCD}$ loss equals:

$$\mathcal{L}_{GCD} = (1-\beta)\mathcal{L}_{SimCLR} + \beta\mathcal{L}_{SupCon}, \tag{7}$$

where $\beta \in (0, 1)$ is a weighing parameter and is equal to 0.35 in our experiments following the original work [24].

GCD+FD GCD method was not designed to tackle continual scenarios as it suffers from catastrophic forgetting. In order to adapt it for the GCCD we follow most distillation-based continual learning methods [7, 9, 16]. We freeze previously trained feature extractor $\mathcal{F}^{t-1}$ and regularize currently trained feature extractor $\mathcal{F}^t$ with the outputs of $\mathcal{F}^{t-1}$:

$$\mathcal{L}_{FD} = \frac{1}{|B|} \sum_{i \in B} ||\mathcal{F}^t(x_i) - \mathcal{F}^{t-1}(x_i)||^2, \tag{8}$$

This form of distillation does not require labels and all the data from the current task can be used for a regularization.

The final loss function for feature extractor training is defined as follows:

$$\mathcal{L}_{GCD+FD} = (1-\alpha)\mathcal{L}_{GCD} + \alpha\mathcal{L}_{FD}, \tag{9}$$

where $\alpha \in [0, 1]$ is a hyperparameter defining the contribution of regularization.

GCD+EWC In this baseline method we improve GCD by enforcing Elastic Weight Consolidation regularization on $\mathcal{F}$ parameters using λ parameter as described in [14]. We additionally add a linear head for training and change $\mathcal{L}_{SupCon}$ to cross entropy loss. In our experiments we set λ to 5000 following the original work.

SimGCD [26] improves training $\mathcal{F}^t$ over GCD by adding two additional loss functions: popular cross entropy loss for labeled data which improves clustering capabilities and adapted mean-entropy maximisation regularisation [1] applied to all data present in the task. The loss is equal to

$$\mathcal{L}_{SimGCD} = ((1-\beta)\mathcal{L}_{SSL} + \beta\mathcal{L}_{SL}) \tag{10}$$

IGCD [29] uses the same loss function as SimGCD to train the feature extractor. However, it adapts SimGCD for continual settings by adding a replay buffer that helps to mitigate forgetting and provides for support sample selection. In each incremental step a random subset of data samples of each class in the task is added to the buffer. For classification IGCD utilizes Soft-Nearest-Neighbor classifier.

PA [13] utilizes proxy anchors to train $\mathcal{F}$ and a replay buffer to fight the forgetting. Proxy anchor loss is defined as:

$$\begin{aligned} \mathcal{L}_{SupPA} = &\frac{1}{|P^{0+}|} \sum_{p \in P^{0+}} \log\left(1 + \sum_{z \in Z_p^{0+}} e^{-\mu(s(z,p)-\delta)}\right) \\ &+ \frac{1}{|P^{0}|} \sum_{p \in P^{0}} \log\left(1 + \sum_{z \in Z_p^{0-}} e^{\mu(s(z,p)+\delta)}\right) \end{aligned} \tag{11}$$

where $\mu > 0$ is a scaling factor and $\delta > 0$ is a margin which we set to 32 and 0.1 respectively, following the original work. Here, the function $s(\cdot,\cdot)$ denotes the cosine similarity score. P^{0^+} represents same class PAs(*e.g.*negative) in the batch.

Each proxy p divides the set of embedding vector Z^0 as $Z_p^{0^+}$ and $Z_p^{0^-} = Z^0 - Z_p^{0^+}$. $Z_p^{0^+}$ denotes the same class embedding points with the proxy anchor p. The goal of the first term in the equation is to pull p and its dissimilar but hard positive data together, while the last term is to push p and its similar but hard negatives apart. The proxies are incrementally added in each task.

Following the original work [13], the total loss for PA method for training $\mathcal{F}$ is equal to:

$$\mathcal{L}_{PA} = \mathcal{L}_{SupPA} + \mathcal{L}_{KD} \tag{12}$$

D Additional experiments

Comparison to baselines

We provide additional average accuracy after each task plots in Fig. 1. They represent results achieved by exemplar and exemplar-free methods on StanfordCars, CUB200, and FGVC Aircraft. CAMP achieves the best final accuracy, and results are consistent on all datasets. CAMP also achieves the best average accuracy after most of the tasks. However, on FGVCAircraft CAMP with 20 exemplars, is worse than GCD, with 20 exemplars after the second, third, and fourth tasks.

Impact of β hyperparameter We verify the impact of *beta* hyperparameter (trade-off between SL and SSL losses) on CUB200. We measure average accuracy for all classes for β equal to $0, 0.2, 0.4, 0.6, 0.8$ and 1.0 and present results in 2. CAMP achieves very low results for $\beta = 0$ as SL loss is not utilized in this case. Interestingly, accuracy for novel categories drops for $\beta > 0.6$ showcasing that the SSL part is crucial in obtaining good results for novel categories.

Impact of number of exemplars on centroid adaptation We verify how the number of exemplars (0, 5, 20 per each category) influences distance of memorized centroid to the real category centroid (denoted as distance to real-mean). We plot the results for CUB200 dataset split into 5 tasks in Fig. 3. We measure the distance to the real-mean before and after performing the centroid adaptation. Intuitively, the more exemplars are available, the better is the centroid estimation. This results are consistent for known and novel categories.

Distillers vs adapters for known and novel classes

We verify the impact of using different distillation functions and different adapters for CAMP on StanfordCars dataset split into four tasks. We utilize the same setup as in 4.4 (main body). We provide final all, known, novel accuracies in Fig. 3. The combination of MLP distiller and Linear adapter achieves 38.1%, 43.2%, and 15.0% on all, known and novel categories, respectively, which is the best result. This shows, that such combination improves results for known and novel classes. The results are consistent with Fig.5 (main paper). That proves the design choice of our architecture.

Extended analysis of latent spaces in CAMP

In Fig. 4, we present an analysis of latent spaces in different GCCD methods. We observe that using no distillation (GCD) leads to the best performance on the second task but the worst performance on the first task, as the model is

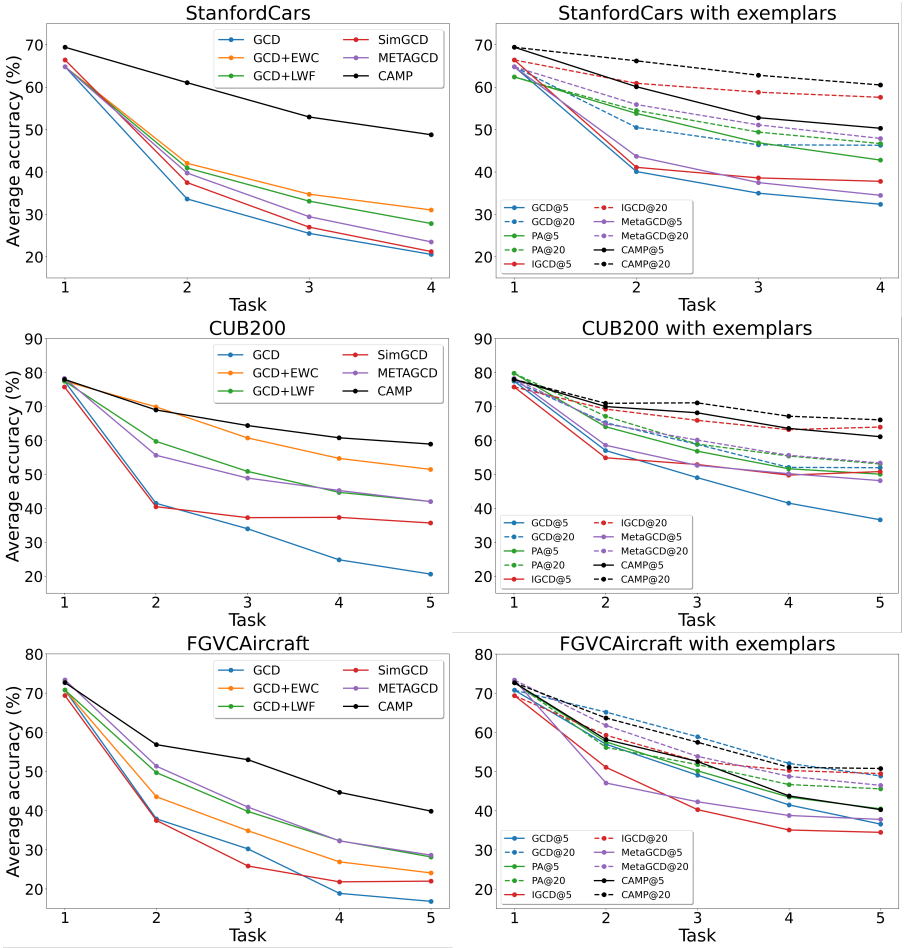

Fig. 1: Average accuracy after each task on three datasets. CAMP achieves the best accuracy after most of tasks.

optimized to fit the new data without regard for the past task. A rigid distillation (GCD + Feature Distillation) helps prevent forgetting but hinders learning new tasks. Moreover, we can see that feature distillation leads to the overlap between tasks. Our CAMP method uses projected knowledge distillation that enhances the ability to learn new tasks and learn representations that overlap less with those of old categories.

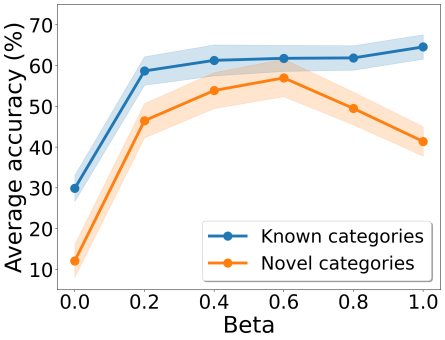

Fig. 2: Impact of β hyperparametrer on known and novel accuracy achieved on CUB200.

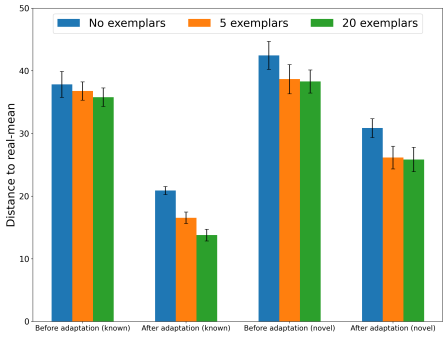

Fig. 3: Distance to real-mean before and after adaptation for 0, 5, 20 exemplars on CUB200.

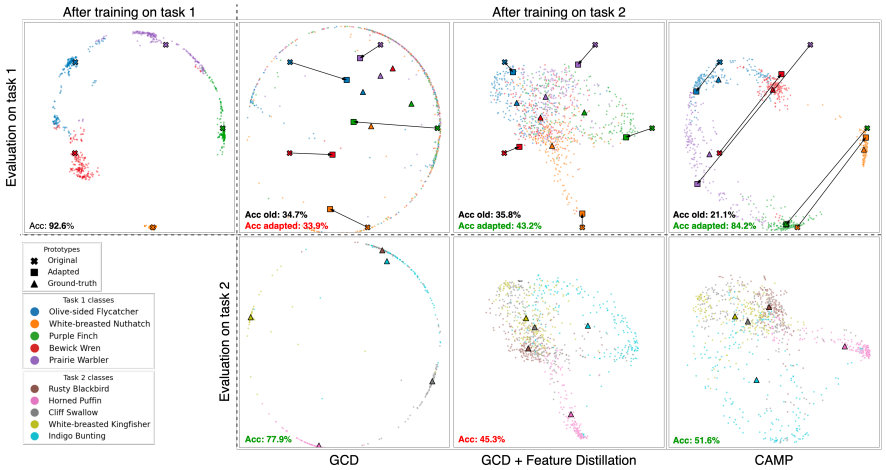

Fig. 4: CAMP utilizes a projected knowledge distillation that results in: (1) predictable drift in latent space that is revertible via centroids adaptation and leads to high performance on the first task and (2) high plasticity of the model and its ability to learn new tasks leading to high performance on the second task. Vanilla GCD fails to prevent forgetting. However, GCD with feature distillation reduces forgetting but diminishes the ability to learn new tasks. We report the nearest centroid classification accuracy. On the first task, we report accuracy using stored prototypes (Acc old) and adapted prototypes (Acc adapted).

Adapter \ Distiller	None	FD	Linear	MLP	T-ReX
None	20.6	28.3	25.2	22.5	21.3
Linear	25.6	28.5	<u>35.5</u>	**38.1**	26.5
MLP	26.6	26.0	34.4	33.7	23.1
T-ReX [23]	24.3	23.9	33.4	28.9	21.8
SDC [28]	23.8	28.1	29.4	31.0	23.8

Adapter \ Distiller	None	FD	Linear	MLP	T-ReX
None	23.9	32.1	28.9	25.8	24.5
Linear	29.4	32.5	<u>40.2</u>	**43.2**	30.1
MLP	30.3	30.1	39.5	38.8	26.5
T-ReX [23]	28.5	27.8	38.5	32.9	24.9
SDC [28]	28.0	32.4	33.3	35.5	27.0

Adapter \ Distiller	None	FD	Linear	MLP	T-ReX
None	5.8	11.3	8.2	7.7	7.0
Linear	8.6	10.2	<u>14.3</u>	**15.0**	10.5
MLP	9.7	7.4	11.4	11.1	8.2
T-ReX [23]	7.2	6.3	10.5	10.7	8.0
SDC [28]	6.9	10.3	9.8	11.0	9.6

Table 3: Impact of different adapters and distiller on our method on StanfordCars. We report all (top), known (middle) and novel (bottom) accuracy after the last task. Combination of MLP distiller and Linear adapter achieves the best results on all types of categories proving the design choice of CAMP.

References

1. Assran, M., Caron, M., Misra, I., Bojanowski, P., Bordes, F., Vincent, P., Joulin, A., Rabbat, M., Ballas, N.: Masked siamese networks for label-efficient learning. In: European Conference on Computer Vision. pp. 456–473. Springer (2022)
2. Boschini, M., Buzzega, P., Bonicelli, L., Porrello, A., Calderara, S.: Continual semi-supervised learning through contrastive interpolation consistency. Pattern Recognition Letters (2021)
3. Buzzega, P., Boschini, M., Porrello, A., Abati, D., Calderara, S.: Dark experience for general continual learning: a strong, simple baseline. Advances in neural information processing systems **33**, 15920–15930 (2020)
4. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the International Conference on Computer Vision (ICCV) (2021)
5. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations (2020)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
7. Fini, E., da Costa, V.G.T., Alameda-Pineda, X., Ricci, E., Alahari, K., Mairal, J.: Self-supervised models are continual learners. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022. pp. 9611–9620. IEEE (2022)
8. Gomez-Villa, A., Twardowski, B., Wang, K., van de Weijer, J.: Plasticity-optimized complementary networks for unsupervised continual learning. In: IEEE/CVF Winter Conference on Applications of Computer Vision (2024)
9. Gomez-Villa, A., Twardowski, B., Yu, L., Bagdanov, A.D., van de Weijer, J.: Continually learning self-supervised representations with projected functional regularization (2022)
10. Iscen, A., Zhang, J., Lazebnik, S., Schmid, C.: Memory-efficient incremental learning through feature adaptation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16. pp. 699–715. Springer (2020)
11. Kang, Z., Fini, E., Nabi, M., Ricci, E., Alahari, K.: A soft nearest-neighbor framework for continual semi-supervised learning. ICCV (2023)
12. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning (2021)
13. Kim, H., Suh, S., Kim, D., Jeong, D., Cho, H., Kim, J.: Proxy anchor-based unsupervised learning for continuous generalized category discovery. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16688–16697 (2023)
14. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences **114**(13), 3521–3526 (2017)
15. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. . (2009)
16. Li, Z., Hoiem, D.: Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence **40**(12), 2935–2947 (2017)

17. Madaan, D., Yoon, J., Li, Y., Liu, Y., Hwang, S.J.: Representational continuity for unsupervised continual learning. In: International Conference on Learning Representations (ICLR) (2022)
18. Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A.D., van de Weijer, J.: Class-incremental learning: survey and performance evaluation on image classification. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022)
19. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1406–1415 (2019)
20. Petit, G., Popescu, A., Schindler, H., Picard, D., Delezoide, B.: Fetril: Feature translation for exemplar-free class-incremental learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3911–3920 (2023)
21. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 2001–2010 (2017)
22. Rypeść, G., Cygert, S., Khan, V., Trzciński, T., Zieliński, B., Twardowski, B.: Divide and not forget: Ensemble of selectively trained experts in continual learning. arXiv preprint arXiv:2401.10191 (2024)
23. Sariyildiz, M.B., Kalantidis, Y., Alahari, K., Larlus, D.: No reason for no supervision: Improved generalization in supervised models. In: ICLR 2023-International Conference on Learning Representations. pp. 1–26 (2023)
24. Vaze, S., Han, K., Vedaldi, A., Zisserman, A.: Generalized category discovery. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7492–7501 (2022)
25. Wang, L., Yang, K., Li, C., Hong, L., Li, Z., Zhu, J.: 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. pp. 5383–5392 (2021)
26. Wen, X., Zhao, B., Qi, X.: Parametric classification for generalized category discovery: A baseline study. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16590–16600 (2023)
27. Yan, S., Xie, J., He, X.: Der: Dynamically expandable representation for class incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3014–3023 (2021)
28. Yu, L., Twardowski, B., Liu, X., Herranz, L., Wang, K., Cheng, Y., Jui, S., Weijer, J.v.d.: Semantic drift compensation for class-incremental learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6982–6991 (2020)
29. Zhao, B., Mac Aodha, O.: Incremental generalized category discovery. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19137–19147 (2023)
30. Zhou, D.W., Wang, F.Y., Ye, H.J., Zhan, D.C.: Pycil: a python toolbox for class-incremental learning. SCIENCE CHINA Information Sciences **66**(9), 197101– (2023)
31. Zhu, F., Cheng, Z., Zhang, X.y., Liu, C.l.: Class-incremental learning via dual augmentation. Advances in Neural Information Processing Systems **34** (2021)
32. Zhu, F., Zhang, X.Y., Wang, C., Yin, F., Liu, C.L.: Prototype augmentation and self-supervision for incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5871–5880 (2021)

Task-recency bias strikes back: Adapting covariances in Exemplar-Free Class Incremental Learning

Grzegorz Rypeść*
IDEAS NCBR
Warsaw University of Technology
grzegorz.rypesc@ideas-ncbr.pl

Sebastian Cygert
IDEAS NCBR
Gdańsk University of Technology
sebastian.cygert@ideas-ncbr.pl

Tomasz Trzciński
IDEAS NCBR
Warsaw University of Technology
Tooploox

Bartłomiej Twardowski
IDEAS NCBR
Autonomous University of Barcelona
Computer Vision Center

Abstract

Exemplar-Free Class Incremental Learning (EFCIL) tackles the problem of training a model on a sequence of tasks without access to past data. Existing state-of-the-art methods represent classes as Gaussian distributions in the feature extractor's latent space, enabling Bayes classification or training the classifier by replaying pseudo features. However, we identify two critical issues that compromise their efficacy when the feature extractor is updated on incremental tasks. First, they do not consider that classes' covariance matrices change and must be adapted after each task. Second, they are susceptible to a task-recency bias caused by dimensionality collapse occurring during training. In this work, we propose *AdaGauss* – a novel method that adapts covariance matrices from task to task and mitigates the task-recency bias owing to the additional anti-collapse loss function. *AdaGauss* yields state-of-the-art results on popular EFCIL benchmarks and datasets when training from scratch or starting from a pre-trained backbone.

1 Introduction

Continual learning (CL), an essential area of machine learning, focuses on developing algorithms that can learn progressively from a continuous stream of data and adapt to new tasks while retaining previously acquired knowledge. This paradigm is paramount for creating systems capable of lifelong learning, much like humans, and robust in dynamic environments where data distribution evolves over time. A significant challenge within CL is exemplar-free class incremental learning (EFCIL) [41, 26], which requires the model to incorporate new classes without storing previous data samples (exemplars). This approach is especially relevant in scenarios with privacy constraints or limited storage capacity, as it compels the model to retain knowledge and prevent catastrophic forgetting [9, 27] solely through internal mechanisms, such as knowledge distillation [12, 21, 45, 51, 24], parameter regularization [17, 2, 6], expanding neural architecture [44, 53, 32, 3] or generative replay [40, 14, 28].

Recent state-of-the-art methods designed for EFCIL often represent classes as Gaussian distributions in the latent space of the feature extractor. That enables an inference using Bayes classifier [10, 35] or training a linear classifier using pseudo-prototypes sampled from these distributions [24, 31, 51, 37]. However, we present in this work that these methods have multiple shortcomings and can be improved. First, they assume that covariance matrices of past classes are constant across incremental training.

*Code: https://github.com/grypesc/AdaGauss

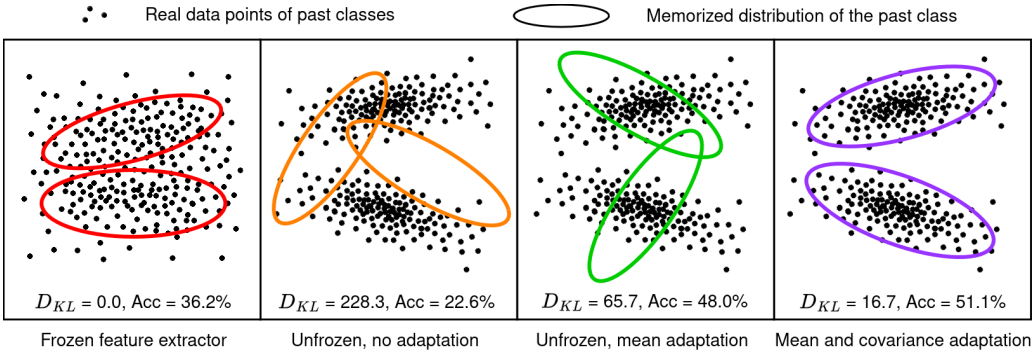

Figure 1: Latent space visualization, average accuracy after the last task, and symmetrical KL divergence between memorized and ground truth distributions for ResNet18 trained sequentially on ImagenetSubset dataset split into ten tasks. Freezing the feature extractor prevents changes in data distribution but results in inseparable classes. When the network is trained on incremental tasks (unfrozen), the ground truth distributions change and do not match the memorized ones. A suitable CL method should adapt the mean and covariance of distributions to retain valid decision boundaries.

However, as presented in Fig. 1, when the feature extractor is updated on incremental tasks (it is unfrozen), distributions of previous classes change and no longer match the memorized ones. Suitable methods must adapt both means and covariances. EFC [24] predicts drift (change) only of the distribution mean and points out that adapting covariances is an open question. Second, the methods suffer from a dimensionality collapse [30, 16], which is more significant in early tasks. That makes old classes' covariances to be of lower rank than those from recent tasks, which introduces errors while inverting the matrices for the classification, leading to increased task-recency bias. We explain this in detail in Sec. 3.2.

This work focuses on the challenging problems of adapting classes' covariances and overcoming dimensional collapse in EFCIL. We are the first to introduce a method that adapts the mean and covariance of memorized distributions, significantly reducing the error between memorized and ground truth distributions. We also overcome the dimensionality collapse of feature representations by introducing a novel anti-collapse loss, which alleviates the problem of task-recency bias. We dub the resulting method *AdaGauss* - Adapting Gaussians. Our contributions are as follows:

- We analyze dimensionality collapse in EFCIL settings and explain that it leads to task-recency bias. We introduce a novel anti-collapse loss to prevent it.
- We show that knowledge distillation techniques in EFCIL provide different representation strengths of the feature extractor. We are the first to utilize knowledge distillation through a learnable projector network in EFCIL.
- Based on these findings, we propose *AdaGauss*, a novel method to adapt both means and covariances of memorized class distributions, which results in state-of-the-art results when the model is trained from scratch or starting from a pre-trained weights.

2 Related works

Semantic drift. We investigate offline, EFCIL setting [26] focusing on keeping the network size constant, where no task information is available at test time. Regularization-based approaches penalize changes to important neural network parameters [17, 6, 47, 22] or use distillation techniques to regularize neuron activations [21, 45, 51, 24]. However, even with knowledge distillation, the features from old classes will change, causing catastrophic forgetting [9, 27]. Therefore, few works tried to predict these changes by approximating their semantic drift [45, 24, 15, 36]. However, those strategies' limitations are that they adapt only prototypes, ignoring changes in covariance matrices, which we experimentally show is suboptimal. As predicting the drift is challenging, many methods focus on scenarios where the backbone is frozen after the first task [4, 31, 23, 29, 10]. However, this prevents the feature extractor to adapt to new tasks [24]. We show that it is possible to change the feature extractor and adapt the covariance matrices of classes.

Task-recency bias. Another challenge in CL is a task-recency bias, where the model is biased towards classifying classes from new tasks [13, 26, 50]. While some works approached this problem using exemplars [1, 43, 13, 48] the problem is amplified in an exemplar-free setting. Some works considered prototype replay, which maintains the decision boundary between classes [31, 51, 37, 39, 38, 53]. To improve this strategy, PASS [52] included prototype augmentation, and EFC [24] updates their prototypes after each task. In this work, we point out that the cause for task-recency bias in the EFCIL scenario is the dimensionality collapse of the feature extractor, leading to numerical instabilities when inverting covariance matrices.

Dimensionality collapse. Recent works revealed that supervised learning exhibits signs of neural collapse [30, 16], where a large fraction of features' variance is described only by a small fraction of their dimensions. Since then, several studies [5, 7, 46, 16] showed that utilizing additional MLP projector is a crucial component to alleviate the collapse of the representations and improve their transferability. Another implication of the neural collapse in CL is that it becomes challenging to invert covariance matrices. Existing methods add a constant value to the diagonal [35, 24, 51] of the covariance matrices or utilize shrinking [10] to prevent that. On the contrary, we propose an anti-collapse loss, which is more elegant and does not artificially alter covariance matrices.

3 Method

3.1 Exemplar-Free Class-Incremental Learning (EFCIL)

Class-Incremental Learning (CIL) scenario considers a dataset split into T tasks, each corresponding to the non-overlapping set of classes $C_1 \cup C_2 \cup \cdots \cup C_T = C$ such that $C_t \cap C_s = \emptyset$ for $t \neq s$. In Exemplar-Free CIL (EFCIL), during a training step t, we have only access to current task data $D_t = \{(x, y)|y \in C_t\}$ and we cannot store any exemplars from the previous steps. The objective is to train a model that discriminates between old ($< t$) and new classes combined. We assume a task-agnostic evaluation [41, 26], where the method does not know the task id during the inference.

3.2 The three observations that motivate towards *AdaGauss*

In this section, we provide an insight into problems with current EFCIL methods. We train the standard ResNet18 [11] network on the ImagenetSubset dataset divided into ten equal tasks. We point out that: **1.** covariance of class distributions during CL sessions change and must be adapted; **2.** the task recency bias comes from the differences in representational strength of the model; **3.** when training from scratch in EFCIL, the models are susceptible to dimensionality collapse.

Observation 1. As illustrated in Fig. 1, training the feature extractor on incremental tasks makes memorized distribution not match the ground truth (GT) ones. More specifically, the mean and covariance of GT change, and to keep valid decision boundaries, both memorized means and covariances must be adapted. That decreases symmetrical KL divergence between memorized and GT distributions, thus increasing average accuracy after the last task. However, existing state-of-the-art methods [36, 24, 51, 52] do not adapt covariance matrices, while others [31, 10, 55, 54] freeze the feature extractor after the initial task, which does not guarantee separability of classes from new tasks (first image in Fig. 1).

Observation 2. When training the feature extractor with different knowledge distillation methods (feature [53, 52, 45], logit [21, 33], projected [18]), representational strength of the feature extractor increases with each task, as presented in Fig. 2. That makes memorized covariance matrices of late tasks have a higher rank than those from early tasks, as presented in Fig. 3. When these matrices are inverted, the opposite happens - due to numerical instabilities, norms of inverted covariance matrices of early tasks will be greater. That causes task-recency bias as presented in Fig. 4. In the case of Bayes classification [35, 10], the Mahalanobis distance is much higher for early tasks, whereas in the case of sampling pseudo-prototypes [24, 51] the logits for recent tasks are higher, what skews classification towards recent tasks. This bias differs from already well-studied linear head bias [13, 43, 48], as it occurs at the level of the representations, where no linear head and no exemplars are utilized.

Observation 3. Fig. 3 also presents that feature extractor suffers from dimensionality collapse [30, 16] as ranks of covariance matrices are much lower than the latent space size (512 for ResNet18). That makes classes covariance matrices non-invertible. That, in turn, disallows the calculation of Mahalanobis distance, likelihood, and sampling from such collapsed distribution. In order

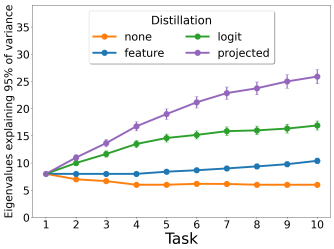

Figure 2: The representational strength of ResNet18 trained on 10 tasks of ImagenetSubset dataset split into 10 tasks for different knowledge distillation methods. After each task, we measure how many eigenvalues sum to 95% variance of all features provided.

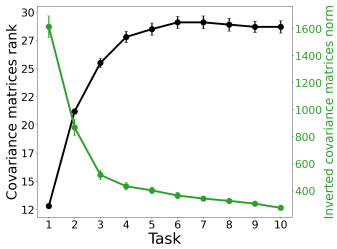

Figure 3: Average rank of memorized covariance matrices of classes after each task (black) on ImagenetSubset for logit distillation. Norm of these matrices when inverted (green). Lower rank leads to larger values in inverses of covariance matrices due to numerical instabilities.

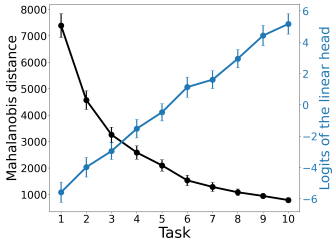

Figure 4: Average Mahalanobis distance between memorized distributions and joint dataset per each task after the last task (black) and average logit value on linear head trained by sampling prototypes from memorized distributions. There is a visible task-recency bias.

to overcome this issue, the existing methods utilize shrinking [10] or add a constant value to the diagonal [35, 24, 51] of the covariance matrices to prevent that. However, these techniques artificially alter classes' distributions, introducing additional hyperparameters and a new source of errors accumulating during long CIL sessions. A more elegant solution would directly prevent the dimensionality collapse of the feature extractor during training while preserving the class separability provided by cross-entropy.

3.3 *AdaGauss* method

Motivated by these three observations, we made the following decisions about *AdaGauss*. Based on the first observation, after training the feature extractor F on an incremental task, we train an auxiliary network (adapter), which we utilize to adapt the means and covariances of old classes to the latent space of the new feature extractor. To perform knowledge distillation and improve the representation strength of the feature extractor (second observation), we utilize feature distillation through a learnable projector. In order to overcome the dimensionality collapse and task-recency bias, showcased by the second and third observations, we utilize a novel anti-collapse loss that regularizes the features' covariance matrix and prevents dimensional collapse. *AdaGauss* memorizes each class as a mean and covariance and performs Bayes classification as in [10, 35]. We provide a pseudo-code of our method in Alg. 1. Below, we explain the motivation and details of the method.

3.3.1 Feature distillation through a learnable projector

Inspired by representational-learning [18], we utilize a feature distillation through a learnable projector to mitigate forgetting, which we refer to as projected distillation. As presented in Fig. 2, this distillation technique provides representations with a better eigenvalues distribution, thus decreasing the problem of task-recency bias compared to standard logit [21, 33] and feature [45, 53, 52] distillation techniques. As the projector, we utilize a 2-layer MLP network $\phi^{t \to t-1} : \mathbb{R}^S \to \mathbb{R}^S$ with hidden size d times bigger than the latent space S. Following existing continual learning works [21, 45, 33], when training F_t on minibatch B, we freeze F_{t-1} trained on the previous task. Finally, we calculate our knowledge distillation loss as follows:

$$L_{PKD} = \sum_{i \in B} ||\phi^{t \to t-1}\left(F_t\left(x_i\right)\right) - F_{t-1}\left(x_i\right)||^2. \tag{1}$$

3.3.2 Overcoming dimensionality collapse

As described in Sec. 3.2, existing methods for EFCIL that represent classes as Gaussian distributions suffer from dimensionality collapse, which leads to task-recency bias caused by the fact that ranks of covariance matrices are different amongst the tasks. To overcome the collapse, we encourage the

feature extractor to produce features whose dimensions are linearly independent. Therefore, in each task, we directly optimize the covariance matrices of features produced by F_t to be positive-definitive by the diagonal of the Cholesky decomposition of covariance of each training minibatch to be positive. More precisely, let S be the size of the feature vectors and a_i be i-th element of the diagonal of a Cholesky decomposition of minibatch's covariance matrix. We formulate the anti-collapse loss L_{AC} in the form:

$$L_{AC} = -\frac{1}{S}\sum_{i=1}^{S}\min(a_i, 1) \tag{2}$$

This loss forces Cholesky's decomposition of covariance of each minibatch to have diagonal entries greater than 1. Therefore, they are positive, and the covariance matrix is positive-definite due to the property of Cholesky decomposition. More on the definition of L_{AC} in Appendix, Sec. A.2).

3.3.3 Training the feature extractor

In each task t, we train all parameters of the feature extractor F_t together with additional projector ϕ used for knowledge distillation. Following most works [10, 35, 51, 31, 21], we utilize popular cross-entropy loss L_{CE} to discriminate between classes. The final loss function is:

$$L = L_{CE} + L_{AC} + \lambda L_{PKD}, \tag{3}$$

where $\lambda \in \mathcal{R}$ is a plasticity-stability trade-off hyperparameter, similar to [21].

After training the feature extractor, we represent classes C_t as multivariate Gaussian distributions in the latent space. More precisely, we represent any class $c \in C_t$ as $\mathcal{N}(\mu_c, \Sigma_c)$.

3.3.4 Adapting Gaussian distributions

After training of F_t is completed, representations of old classes drifted [45] (changed) and no longer match memorized Gaussians. Therefore, we update memorized Gaussians representing past classes to recover ground truth representations. To do that, we train an auxiliary adaptation network $\psi^{t-1\to t} : \mathbb{R}^S \to \mathbb{R}^S$ (called adapter), which maps features from the old latent space to the new one. We use only the current data from task t for that. Training loss is:

$$L_\psi = \sum_{i\in B} ||\psi^{t-1\to t}(F_{t-1}(x_i)) - F_t(x_i)||^2 + L_{AC}. \tag{4}$$

L_{AC} is the same anti-collapse loss as used during the training of the feature extractor. After training the adapter, for each old class c, we sample from $\mathcal{N}(\mu_c, \Sigma_c)$ a set of N points: $n_1, n_2, \ldots, n_N$, where $N \gg |S|$ and transform them through ψ obtaining new set: $\{\psi(n_1), \psi(n_2), \ldots, \psi(n_N)\}$. We calculate adopted mean μ_c^{new} and covariance Σ_c^{new} using new sets of data and update the old distribution as follows: $(\mu_c, \Sigma_c) = (\mu_c^{new}, \Sigma_c^{new})$ A pseudocode of the full *AdaGauss* method is presented in Alg. 1.

4 Experiments

Datasets and metrics. We evaluate our method on several well-established benchmark datasets. CIFAR100 [19] consists of 50k training and 10k testing images in resolution 32x32. TinyImageNet [20], a subset of ImageNet [8], has 100k training and 10k testing images in 64x64 resolution. ImagenetSubset contains 100 classes from ImageNet (ILSVRC 2012) [34]. We split these datasets into 10 and 20 equal tasks. Thus, each task contains the same number of classes, a standard practice in EFCIL [21, 45, 35, 24]. We also evaluate our method on fine-grained datasets: CUB200 [42] represents $11,788$ images of bird species, and FGVCAircraft [25] dataset consists of $10,200$ images of planes. We split fine-grained datasets into 5, 10, and 20 tasks. As the evaluation metric, we utilize commonly used average accuracy A_{last}, which is the accuracy after the last task, and average incremental accuracy A_{inc}, which is the average of accuracies after each task [26, 24, 10].

Baselines and hyperparameters. We compare our method to multiple EFCIL baselines. Well-established ones, like EWC [17], LwF [21], PASS [52], IL2A [51], SSRE [53], and the most recent and strong EFCIL baselines: FeTrIL [31], FeCAM [10], DS-AL [54] and EFC [24]. For the baseline

Algorithm 1 *AdaGauss*: Adapting Gaussians in EFCIL

Initialize: Training data $(D_1, D_2, \ldots, D_T)$, F_1 (feature extractor), λ, N
Train F_1 on D_1 using $L_{CE} + L_{AC}$
for $c \in C_1$ **do**
    Obtain set of features: $O = \{F_1(x) : x, c \in D_1\}$
    Store $\mu_c = \text{mean}(O)$ and $\Sigma_c = \text{covariance}(O)$
end for
for $t = 2, 3, 4, \ldots$ **do**
    Initialize $\phi^{t \to t-1}$ (distiller), $\psi^{t-1 \to t}$ (adapter)
    Train F_t on D_t using $L = L_{CE} + L_{AC} + \lambda L_{PKD}$
    for $c \in C_t$ **do**
        Obtain set of features: $O = \{F_t(x) : x, c \in D_t\}$
        Store $\mu_c = \text{mean}(O)$ and $\Sigma_c = \text{covariance}(O)$
    end for
    Train adapter $\psi^{t-1 \to t}$ on D_t using $L_\psi + L_{AC}$
    for $c \in \cup_{i=1}^{t-1} C_i$ **do**
        Sample $n_1, n_2, \ldots, n_N$ from $\mathcal{N}(\mu_c, \Sigma_c)$
        Calculate μ_c^{new}, Σ_c^{new} of set $\{\psi^{t-1 \to t}(n_1), \psi^{t-1 \to t}(n_2), \ldots, \psi^{t-1 \to t}(n_N)\}$
        $\mu_c = \mu_c^{new}$; $\Sigma_c = \Sigma_c^{new}$
    end for
end for

results on CIFAR100, TinyImageNet, and ImagenetSubset, we take the results reported in [24], while for FeCAM, we run its original implementation. For fine-grained datasets (CUB200, FGVCAircrafts), we run implementations provided in FACIL [26] and PyCIL [49] frameworks (if provided) or from the authors' repositories. We set default hyperparameters proposed in the original works. We utilize random crops and horizontal flips as data augmentation.

Implementation details and reproducibility. We utilize standard ResNet18 [11] as a feature extractor F for all methods. We train it from scratch on CIFAR100, TinyImagenetSubset, and ImagenetSubset, while for experiments on fine-grained datasets, we utilize weights pre-trained on ImageNet. We implement our method in FACIL[26] benchmark[2]. We set $\lambda = 10, N = 10000, d = 32$ and add a single linear bottleneck layer at the end of the F with S output dimensions, which define the latent space. When training from scratch, we set $S = 64$, while for fine-grained datasets, we decrease it to 32, as there are fewer examples per class. We use an SGD optimizer running for 200 epochs with a weight decay equal to 0.0005. When training from scratch, we utilize a starting learning rate (lr) of 0.1, decreased by ten times after 60, 120, and 180 epochs. We train the adapter using an SGD optimizer with weight decay of 0.0005, running for 100 epochs with a starting lr of 0.01; we decrease it ten times after 45 and 90 epochs.

We utilize a single machine with an NVIDIA RTX4080 graphics card to run experiments. The time for execution of a single experiment varied depending on the dataset type, but it was at most ten hours. We attach details of utilized hyperparameters in scripts in the code repository. We report all results as the mean and variance of five runs.

4.1 Results

Training from scratch. We present the baseline results and *AdaGauss* method when training from scratch in Tab. 1. We consider $T = 10$ and $T = 20$ equal tasks. We can see an improvement over the most recent state-of-the-art method - EFC [24]. We improve its results by 3.7% and 6.8% points in terms of average accuracy on ImagenetSubset split into 10 and 20 tasks, respectively. This improvement is also consistent in terms of average incremental accuracy - 5.1% and 7.5% points and on the other datasets. This increase can be attributed to the fact that EFC does not adapt covariance matrices from task to task (just means), which, as we showed in Sec. 3.2, is required to improve the results. Older method - IL2A [51], which does not adapt their classes representations (means and covariance matrices) method at all, achieves much lower results than our approach - 23.4% and 25.1% points lower average accuracy on ImagenetSubset.

[2]The code is provided in the Supplementary Materials and will be published upon acceptance.

Table 1: Average incremental and last accuracy in EFCIL when training the feature extractor from scratch. The mean of 5 runs is reported. Full results are in Tab. 5. We denote the best results **in bold**.

Method	CIFAR-100				TinyImageNet				ImagenetSubset			
	T=10		T=20		T=10		T=20		T=10		T=20	
	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}
EWC [17]	31.2	49.1	17.4	31.0	17.6	32.6	11.3	26.8	24.6	39.4	12.8	27.0
LwF [21]	32.8	53.9	17.4	38.4	26.1	45.1	15.0	32.9	37.7	56.4	18.6	40.2
PASS [52]	30.5	47.9	17.4	32.9	24.1	39.3	18.7	32.0	26.4	45.7	14.4	31.7
IL2A [51]	31.7	48.4	23.0	40.2	25.3	42.0	19.8	35.5	27.7	48.4	17.5	34.9
SSRE [53]	30.4	47.3	17.5	32.5	22.9	38.8	17.3	30.6	25.4	43.8	16.3	31.2
FeTrIL [31]	34.9	51.2	23.3	38.5	31.0	45.6	25.7	39.5	36.2	52.6	26.6	42.4
FeCAM [10]	32.4	48.3	20.6	34.1	30.8	44.5	25.2	38.3	38.7	54.8	29.0	44.6
DS-AL [54]	40.8	54.9	31.7	43.2	33.6	47.2	26.5	41.6	46.8	58.6	36.7	48.5
EFC [24]	43.6	58.6	32.2	47.3	34.1	48.0	28.7	42.1	47.4	59.9	35.8	49.9
AdaGauss	**46.1**	**60.2**	**37.8**	**52.4**	**36.5**	**50.6**	**31.3**	**45.1**	**51.1**	**65.0**	**42.6**	**57.4**

Table 2: Average incremental and last accuracy in EFCIL fine-grained scenarios when utilizing a pre-trained feature extractor. We report the mean of 5 runs, while variances are reported in Tab. 6.

Method	CUB200						FGVCAircraft					
	T=5		T=10		T=20		T=5		T=10		T=20	
	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}
EWC [17]	21.6	38.2	15.8	32.6	12.3	27.2	24.3	44.0	14.3	34.5	10.9	27.9
LwF [21]	44.3	57.7	30.4	46.1	19.4	34.7	39.0	55.2	28.0	46.5	14.7	30.5
PASS [52]	34.5	48.6	27.0	42.3	18.1	36.9	33.3	48.9	26.4	41.0	13.9	28.1
IL2A [51]	36.9	51.3	29.4	45.5	20.8	35.1	39.4	49.1	27.3	45.1	14.2	28.7
FeTrIL [31]	41.9	53.2	36.9	48.2	34.6	45.3	46.0	58.5	40.5	53.4	32.5	43.3
FeCAM [10]	43.5	56.0	40.2	54.9	36.2	48.9	45.3	58.0	41.4	55.2	34.0	46.0
DS-AL [54]	49.4	61.9	45.8	59.1	41.4	53.8	50.6	62.7	42.6	56.4	34.2	46.7
EFC [24]	58.3	68.9	51.0	63.3	46.1	59.3	50.1	63.2	43.1	57.6	28.1	48.2
AdaGauss	**60.4**	**69.2**	**55.8**	**66.2**	**47.4**	**60.6**	**53.3**	**64.0**	**47.5**	**58.5**	**34.8**	**48.6**

Methods such as FeTrIL [31], FeCAM [10] and DS-AL [54] overcome the problem of distribution drift by freezing the feature extractor on the first task. However, it cannot adapt well to the new incremental tasks, resulting in poor plasticity and worse results than *AdaGauss* and EFC [24]. FeTrIL achieves 14.9% and 16.0% points lower average accuracy on ImagenetSubset, while FeCAM - 12.4% and 13.6%.

Training from pre-trained model. We provide the baseline results and our method when training from a ImageNet pre-trained model in Tab. 2. Despite having a strong feature extractor from the very beginning, it still needs to be adapted to discriminate better between fine-grained classes. We report results for 5, 10, and 20 equal tasks. *AdaGauss* achieves state-of-the-art results. It improves the average accuracy of the second-best method EFC [24] by 4.8% and 4.4% points on CUB200 and StanfordCars for $T = 10$, respectively. The results are consistent for other number of tasks.

Ablation study. We perform ablation of our method on CIFAR100 and ImagenetSubset datasets split into ten equal tasks in Table 3. First, we test our method with the nearest mean classifier (NMC) instead of the Bayes classifier to verify whether considering covariance improves the results. Without covariance matrices and with NMC [33] (1st row), we get worse results: 9.7% and 9.6% points lower average accuracy on CIFAR100 and ImagenetSubset, respectively. Memorizing covariances and sampling pseudo-prototypes to adapt means (2nd row) improves NMC results only slightly. Next, we utilize the Bayes classifier instead of NMC but assume that class distributions have diagonal covariance matrices (3rd row). That decreases the average accuracy of our method by 5.0% and 3.9%, respectively, proving that ground truth test distributions have non-zero off-diagonal. Then, we test our method without adapting means (5th row) like in IL2A [51] method. That severely hurts the performance - average accuracy decreases by 21.5% and 27.2 %. On the contrary, if we adapt means but not covariances like in EFC [24], we lose far less, 3.2% and 3.1%, respectively. Lastly, we check the performance of our method without the L_{AC} component. To allow covariance matrices to be invertible, we add a shrink value of 0.5, similarly to [10]. This results in an average accuracy drop of 5.9% and 4.0%. The results are also consistent with the average incremental accuracies. This ablation proves our design choices and that all components are necessary to get the best results.

Table 3: Ablation of *AdaGauss* indicating the contribution from the different components. * signifies that we utilized covariance matrix shrinking with the value of 0.5 (chosen on the validation set) instead of anti-collapse loss to overcome the covariance matrix singularity problem.

Classifier	Cov. Matrix	Adapt mean	Adapt covariance	L_{AC}	CIFAR-100 (T=10)		ImagenetSubset (T=10)	
					A_{last}	A_{inc}	A_{last}	A_{inc}
NMC	None	✓	✓	✓	36.4	54.0	41.5	57.9
NMC	Full	✓	✓	✓	37.6	54.8	42.3	58.7
Bayes	Diagonal	✓	✓	✓	41.1	56.3	45.5	61.3
Bayes	Full	✗	✗	✗	22.9	42.8	22.5	43.4
Bayes	Full	✗	✓	✓	24.6	44.7	23.9	44.9
Bayes	Full	✓	✗	✓	42.9	57.7	48.0	62.7
Bayes	Full	✓	✓	✗*	40.2	56.2	46.7	57.1
Bayes	Full	✓	✓	✓	**46.1**	**60.2**	**51.1**	**65.0**

4.2 Adaptation results

We verify how our adaptation method improves the quality of memorized class distributions on ImagenetSubset split into ten equal tasks. For this purpose, we measure the average distances between memorized and real classes after each task. More precisely, we measure the L2 distance between means and covariances as well as symmetrical Kulbach-Leibler divergence (D_{KL}) between memorized and real distributions. We utilize projected distillation ($\lambda = 10$) and compare our method to a baseline that does not adapt distributions like in [51, 52] (No adapt) and to the prototype drift compensation introduced in EFC [24] that adapts only means. We provide results in Fig. 5. We can see that our approach allows us to better approximate ground truth distributions. More precisely, compared to EFC, it decreases the distance to real-mean by ≈29%, to real-covariance by ≈39% and D_{KL} distance by ≈72%. We can also see that the EFC approach does not improve distance to real-covariance compared to no adaptation, which is a drawback of this method.

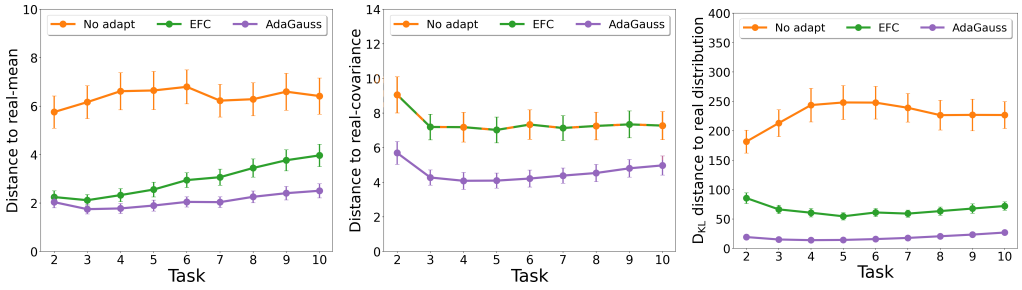

Figure 5: Distances from memorized distributions to the real ones in terms of distributions' mean, covariance and KL divergence across 10 tasks on ImagenetSubset dataset. AdaGauss greatly reduces errors and allows for better adaptation than prototype drift compensation (EFC).

4.3 Analysis of anti-collapse loss

We analyze the impact of anti-collapse L_{AC} regularization term on ImagenetSubset split to 10 equal tasks. After the last task, we verify how much L_{AC} improves the distribution of classes' covariance eigenvalues. We report results in Fig. 6. Without utilizing L_{AC}, the largest eigenvalue is $\approx 1.2 * 10^5$ times greater than the lowest, showcasing the dimensionality collapse. However, with L_{AC}, this difference equals ≈84, proving that more eigenvectors contribute towards representations, and the collapse is greatly diminished.

Next, we measure the average rank of covariance matrices memorized in each task for different knowledge distillation methods and projected distillation with L_{AC}. Here, we set $S = 64$. In Fig. 7 can see that without L_{AC}, all distillation methods present in existing methods struggle to achieve class covariance equal to latent size S, which according to Sec. 3.2 results in task-recency bias. Interestingly, when combining projected distillation with L_{AC}, the rank of covariance matrices equals 64 for each task, proving that L_{AC} is a promising approach for combating dimensionality collapse when training from scratch.

An alternative method for overcoming singularity in covariance matrices is shrinking [10]. In Fig. 8, we present results for our method with different values of shrink performed when calculating covariance matrices on CIFAR100. Intuitively, increasing the shrink value decreases the method's efficacy, as it artificially alters the covariance to be different from the ground truth representation. Without using L_{AC} and without shrink, it is impossible to invert the matrices, resulting in the crash of the method. Nevertheless, the results are the highest when utilizing L_{AC} without shrink (60.2%).

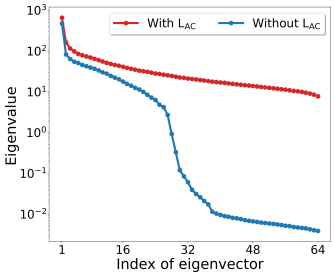

Figure 6: Distribution of eigenvalues of class representations for our method with and without L_{AC} (anti-collapse loss) term. L_{AC} greatly reduces the difference between the most and least significant eigenvalues, thus preventing dimensional collapse.

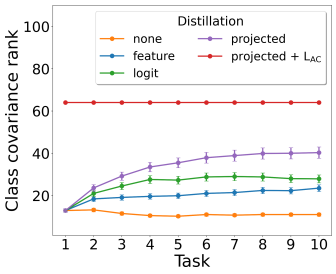

Figure 7: Ranks of classes' covariance matrices with different distillation methods and projected distillation with anti-collapse term (red) for latent space size $S = 64$. L_{AC} makes covariance ranks to be equal to S in every task.

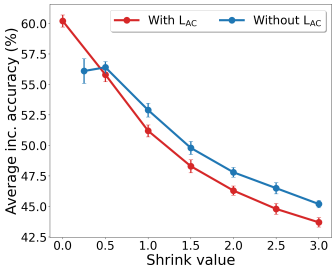

Figure 8: Average incremental accuracy for *AdaGauss* for different values of covariance shrinking, with and without anti-collapse regularization. Results without L_{AC} and shrink were not included due to the inability to invert covariance matrices after the first task.

4.4 Different distillation techniques

We test the performance of the projected distillation against other distillation techniques in *AdaGauss*. We train from scratch on CIFAR100, ImagenetSubset and utilize the pre-trained model on CUB200. We split datasets into ten equal tasks and use hyperparameters from experiments in Tab. 1 and Tab. 2. We present the results in Fig. 9. Projected distillation achieves better average accuracy than logit distillation by 1.4%, 0.9%, and 4.0% points on CIFAR100, ImagenetSubset, and CUB200, respectively. Interestingly, the gap between projected distillation and not using knowledge distillation is much lower on CUB200, which we contribute to using a strong pre-trained model.

4.5 Memory requirements

Our method does not increase the number of feature extractor's parameters. In addition, the adapter and distiller are discarded after the training, thus not increasing memory during the long CIL sessions and evaluations. *AdaGauss* requires $S + \frac{S(S-1)}{2}$ parameters to memorize the mean and covariance of a class, where S is the latent space size. Therefore, the method requires the same number of parameters as FeCAM [10] and fewer weights than EFC [24] as we do not expand the linear classifier. Additionally, S can be decreased using linear bottleneck layer before the latent space.

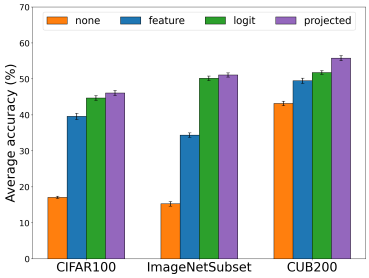

Figure 9: Average last task acc. of our method for different knowledge distillation techniques.

4.6 Time complexity of AdaGauss

We measure the training and inference time of popular EFCIL methods using their original implementations on a single machine with NVIDIA GeForce RTX 4060 and AMD Ryzen 5 5600X CPU. We repeat each experiment 5 times, train all methods for 200 epochs, use four workers, and have a batch size equal to 128. We test vanilla AdaGauss and AdaGauss, where the Bayes classifier is replaced with a trained linear head, where the classifier is trained on samples from class distributions (mean and cov. matrix). We utilize the FeTrIL version with a linear classification head.

We present results in Tab. 4. The inference of our method takes a similar amount of time as in FeCAM, as the feature extraction step is followed by performing Bayes classification. The inference

time of AdaGauss is slightly higher than that of methods with linear classification head (LwF, FeTrIL, AdaGauss with linear head) because Bayes classification requires an additional matrix multiplication when calculating the Mahalanobis distance.

The training time of AdaGauss is longer than for LwF, EFC, FeCAM, and FeTriL as we do not freeze the backbone after the initial task and additionally train the auxiliary adaptation network. Still, AdaGauss takes less time to train than its main competitor - EFC, and is much faster than SSRE. Our method does not increase the number of networks' parameters because the distiller and the adapter are disposed after training steps.

Table 4: Time complexity of EFCIL methods measured on CIFAR100 split into 10 tasks.

	LwF	FeTrIL	FeCAM	SSRE	EFC	AdaGauss	AdaGauss (lin. head)
Inference time (sec)	66.3±2.7	68.3±3.4	74.2±4.0	274.7±12.3	67.4±3.1	72.8±3.2	65.3±2.2
Training time (min)	53.8±2.9	5.3±0.3	5.9±0.4	548.1±17.4	94.±2.9	86.3±3.2	103.3±4.2

5 Conclusions and limitations

In this work, we analyze the impact of dimensionality collapse in EFCIL. We explain that it leads to differences across tasks in ranks of classes' covariance matrices, which in turn causes task-recency bias. We also present that due to distribution drift, means and covariance of classes change, and they should be adapted from task to task. Based on these findings, we propose the first EFCIL method to adapt both means and covariances, dubbed *AdaGauss*. It utilizes feature distillation through a learnable projector and a novel anti-collapse regularization term during training that prevents having degenerated, non-invertible features covariance matrices as class representations. That, in turn, alleviates the task-recency bias of the classifier in continual learning. With the series of experiments, we show that *AdaGauss* achieves state-of-the-art results in common EFCIL scenarios, both when trained from scratch and when initialized from a pre-trained model.

The limitation of our method is that the cross-entropy separates classes only from the current task. However, when training the feature extractor, old classes can begin overlapping with each other and with new classes int he latent space causing forgetting. This problem is an open question in EFCIL. We speculate it can be alleviated wit a contrastive loss. Another problem arises when there is very little data representing a single class, making high-dimensional covariance matrix impossible to calculate. We tackle it by introducing a bottleneck layer at the very end of the feature extractor. However, it can limit its representational strength.

Acknowledgments

The research of Grzegorz Rypeść was supported by the National Science Centre (Poland), PRELUDIUM 23 grant no. 2024/53/N/ST6/03018. This research was partially funded by National Science Centre, Poland, grant no: 2020/39/B/ST6/01511, 2022/45/B/ST6/02817, and 2023/51/D/ST6/02846. Bartłomiej Twardowski acknowledges the grant RYC2021-032765-I. This paper has been supported by the Horizon Europe Programme (HORIZON-CL4-2022-HUMAN-02) under the project "ELIAS: European Lighthouse of AI for Sustainability", GA no. 101120237. We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2023/017431.

References

[1] Hongjoon Ahn, Jihwan Kwak, Subin Lim, Hyeonsu Bang, Hyojun Kim, and Taesup Moon. Ss-il: Separated softmax for incremental learning. In *ICCV*, 2021.

[2] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pages 139–154, 2018.

[3] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3366–3375, 2017.

[4] Eden Belouadah and Adrian Popescu. Deesil: Deep-shallow incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[5] Florian Bordes, Randall Balestriero, Quentin Garrido, Adrien Bardes, and Pascal Vincent. Guillotine regularization: Why removing layers is needed to improve generalization in self-supervised learning. *Transactions on Machine Learning Research*, 2023.

[6] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, pages 532–547, 2018.

[7] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

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

[10] Dipam Goswami, Yuyang Liu, Bartłomiej Twardowski, and Joost van de Weijer. Fecam: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. *Advances in Neural Information Processing Systems*, 36, 2024.

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

[12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[13] 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*, pages 831–839, 2019.

[14] Wenpeng Hu, Zhou Lin, Bing Liu, Chongyang Tao, Zhengwei Tao Tao, Dongyan Zhao, Jinwen Ma, and Rui Yan. Overcoming catastrophic forgetting for continual learning via model adaptation. In *International conference on learning representations*, 2019.

[15] Ahmet Iscen, Jeffrey Zhang, Svetlana Lazebnik, and Cordelia Schmid. Memory-efficient incremental learning through feature adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 699–715. Springer, 2020.

[16] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348*, 2021.

[17] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences (PNAS)*, 2017.

[18] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR, 2019.

[19] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[20] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[21] Zhizhong Li and Derek Hoiem. Learning without forgetting. *Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2017.

[22] Xialei Liu, Marc Masana, Luis Herranz, Joost Van de Weijer, Antonio M Lopez, and Andrew D Bagdanov. Rotate your networks: Better weight consolidation and less catastrophic forgetting. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2262–2268. IEEE, 2018.

[23] Chunwei Ma, Zhanghexuan Ji, Ziyun Huang, Yan Shen, Mingchen Gao, and Jinhui Xu. Progressive voronoi diagram subdivision enables accurate data-free class-incremental learning. In *In The Eleventh International Conference on Learning Representations*, 2023.

[24] Simone Magistri, Tomaso Trinci, Albin Soutif-Cormerais, Joost van de Weijer, and Andrew D Bagdanov. Elastic feature consolidation for cold start exemplar-free incremental learning. *arXiv preprint arXiv:2402.03917*, 2024.

[25] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[26] Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost van de Weijer. Class-incremental learning: Survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2022.

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

[28] Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, Patrick Jahnichen, and Moin Nabi. Learning to remember: A synaptic plasticity driven framework for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11321–11329, 2019.

[29] Aristeidis Panos, Yuriko Kobe, Daniel Olmeda Reino, Rahaf Aljundi, and Richard E Turner. First session adaptation: A strong replay-free baseline for class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18820–18830, 2023.

[30] Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 2020.

[31] Grégoire Petit, Adrian Popescu, Hugo Schindler, David Picard, and Bertrand Delezoide. Fetril: Feature translation for exemplar-free class-incremental learning. In *Winter Conference on Applications of Computer Vision (WACV)*, 2023.

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

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

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

[35] Grzegorz Rypeść, Sebastian Cygert, Valeriya Khan, Tomasz Trzcinski, Bartosz Michał Zieliński, and Bartłomiej Twardowski. Divide and not forget: Ensemble of selectively trained experts in continual learning. In *The Twelfth International Conference on Learning Representations*, 2023.

[36] Grzegorz Rypeść, Daniel Marczak, Sebastian Cygert, Tomasz Trzciński, and Bartłomiej Twardowski. Category adaptation meets projected distillation in generalized continual category discovery. In *European Conference on Computer Vision (ECCV)*, 2024.

[37] Wuxuan Shi and Mang Ye. Prototype reminiscence and augmented asymmetric knowledge aggregation for non-exemplar class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1772–1781, 2023.

[38] 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.

[39] Marco Toldo and Mete Ozay. Bring evanescent representations to life in lifelong class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16732–16741, 2022.

[40] Gido M Van de Ven, Hava T Siegelmann, and Andreas S Tolias. Brain-inspired replay for continual learning with artificial neural networks. *Nature communications*, 11(1):4069, 2020.

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

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

[43] 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.

[44] 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.

[45] 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*, pages 6982–6991, 2020.

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

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

[48] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13208–13217, 2020.

[49] Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, and De-Chuan Zhan. Pycil: A python toolbox for class-incremental learning, 2021.

[50] Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Deep class-incremental learning: A survey. *arXiv preprint arXiv:2302.03648*, 2023.

[51] Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Cheng-lin Liu. Class-incremental learning via dual augmentation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[52] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[53] Kai Zhu, Wei Zhai, Yang Cao, Jiebo Luo, and Zheng-Jun Zha. Self-sustaining representation expansion for non-exemplar class-incremental learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[54] Huiping Zhuang, Run He, Kai Tong, Ziqian Zeng, Cen Chen, and Zhiping Lin. Ds-al: A dual-stream analytic learning for exemplar-free class-incremental learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17237–17244, 2024.

[55] Huiping Zhuang, Zhenyu Weng, Hongxin Wei, Renchunzi Xie, Kar-Ann Toh, and Zhiping Lin. Acil: Analytic class-incremental learning with absolute memorization and privacy protection. *Advances in Neural Information Processing Systems*, 35:11602–11614, 2022.

A Additional experiments

A.1 Adaptation results when starting from a pretrained model

We evaluate how our adaptation method improves the quality of memorized class distributions on CUB200 [42] split into ten equal tasks when starting from a model pre-trained on ImageNet. For this purpose, we measure the average distances between memorized and real classes after each task. More precisely, we measure the L2 distance between means and covariances as well as symmetrical Kulbach-Leibler divergence (D_{KL}) between memorized and real distributions. We utilize projected distillation ($\lambda = 10$) and compare our method to a baseline that does not adapt distributions like in [51, 52] (No adapt) and to the prototype drift compensation introduced in EFC [24] that adapts means only (EFC). We provide results in Fig. 10. Results are consistent with Fig. 10 - *AdaGauss* improves memorized distributions after every task.

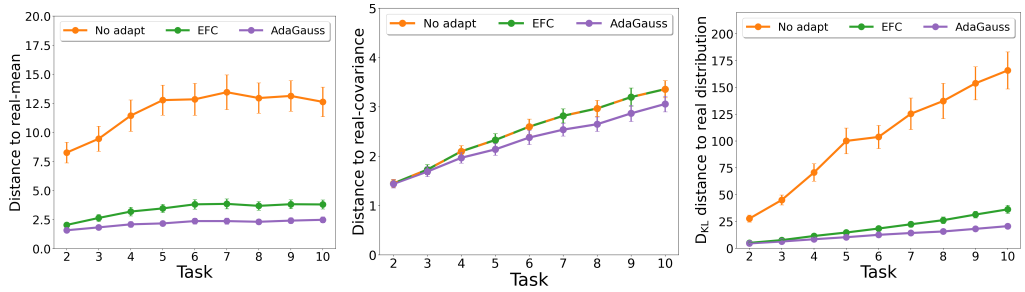

Figure 10: L2 distances from memorized distributions to the real ones in terms of distributions' mean, covariance and KL divergence across 10 tasks on CUB-200 dataset. The feature extractor was pre-trained on ImageNet.

A.2 Impact of anti-collapse loss on optimization

Training the feature extractor of *AdaGauss* combines three loss functions: cross-entropy L_{CE}, knowledge distillation through a learnable projector L_{PKD}, and anti-collapse term to prevent features from dimensional collapse L_{AC}. We analyze average values of these losses during training of our method on ImagenetSubset (we set hyperparameters as in Tab. 1). Additionally, we modify Eq. 2 to incorporate strength of covariance regularization (β hyperparameter):

$$L_{AC} = -\frac{1}{S}\sum_{i=1}^{S}\min(a_i, \beta) \tag{5}$$

In all of our experiments, we utilized $\beta = 1$ as this was sufficient to prevent dimensional collapse. Here, we test *AdaGauss* for $\beta = \{0.1, 1, 10, 100\}$.

We present results in Fig. 11. We can see that for $\beta = 1$, all losses are stable and consistently decrease. L_{AC} decreases to -1.0, a value for which it is clipped. However, when increasing β, L_{CE} and L_{KD} become bigger, underfitting our approach. This results in much lower average and incremental accuracies. On the other hand, decreasing β to 0.1 is not sufficient to prevent task-recency bias resulting in decreased accuracies.

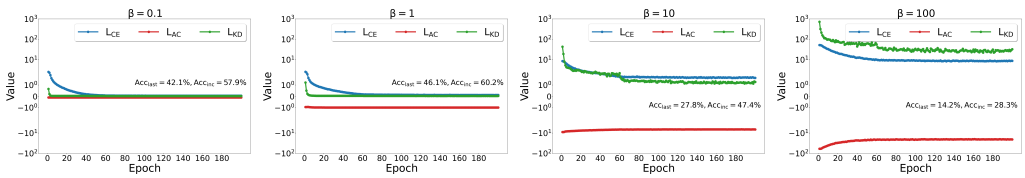

Figure 11: Value of L_{CE}, L_{PKD}, L_{AC} losses for different β parameters, last task average accuracy and average incremental accuracy.

A.3 Tab. 1 and Tab. 2 with variance

We report the mean and variance of results reported in Tab. 1 when training from scratch in Tab. 5. Also, we report results from Tab. 2 when training from a pre-trained model in Tab. 6. Although we utilize additional anti-collapse loss compared to other methods, variance of accuracies achieved by *AdaGauss* is simillar to EFC.

Table 5: Average incremental and last accuracy in EFCIL scenarios when training the feature extractor from scratch. We report means and variances of 5 runs. We denote the best results **in bold**.

Method	CIFAR-100				TinyImageNet				ImagenetSubset			
	T=10		T=20		T=10		T=20		T=10		T=20	
	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}
EWC [17]	31.2±2.9	49.1±1.3	17.4±2.4	31.0±1.2	17.6±1.5	32.6±1.2	11.3±1.2	26.8±1.1	24.6±4.1	39.4±3.1	12.8±2.0	27.0±1.0
LwF [21]	32.8±3.1	53.9±1.7	17.4±0.7	38.4±1.1	26.1±1.3	45.1±0.9	15.0±0.7	32.9±0.5	37.7±2.5	56.4±1.0	18.6±1.7	40.2±0.4
PASS [52]	30.5±1.0	47.9±1.9	17.4±0.7	32.9±1.0	24.1±0.5	39.3±0.9	18.7±1.4	32.0±1.8	26.4±1.3	45.7±0.2	14.4±1.2	31.7±0.4
IL2A [51]	31.7±1.3	48.4±2.0	23.0±0.9	40.2±1.1	25.3±0.9	42.0±1.7	19.8±1.8	35.5±2.3	27.7±1.8	48.4±1.5	17.5±1.6	34.9±0.7
SSRE [53]	30.4±0.7	47.3±1.9	17.5±0.8	32.5±1.1	22.9±1.0	38.8±2.0	17.3±1.1	30.6±2.0	25.4±1.2	43.8±1.1	16.3±1.1	31.2±1.5
FeTrIL [31]	34.9±0.5	51.2±1.1	23.3±0.8	38.5±1.1	31.0±0.9	45.6±1.7	25.7±0.6	39.5±1.2	36.2±1.2	52.6±0.6	26.6±1.5	42.4±2.1
FeCAM [10]	32.4±0.4	48.3±0.9	20.6±0.7	34.1±1.1	30.8±0.8	44.5±1.5	25.2±0.6	38.3±1.1	38.7±1.0	54.8±0.5	29.0±1.3	44.6±2.0
EFC [24]	43.6±0.7	58.6±0.9	32.2±1.3	47.3±1.4	34.1±0.8	48.0±0.6	28.7±0.4	42.1±1.0	47.4±1.4	59.9±1.4	35.8±1.7	49.9±2.1
AdaGauss	**46.1±0.8**	**60.2±0.9**	**37.8±1.5**	**52.4±1.4**	**36.5±0.9**	**50.6±0.8**	**31.3±1.0**	**45.1±1.2**	**51.1±1.2**	**65.0±1.4**	**42.6±1.6**	**57.4±1.9**

Table 6: Average incremental and last accuracy in EFCIL fine-grained scenarios when utilizing a pre-trained feature extractor on ImageNet. The means and variance of 5 runs are reported with the best results **in bold**.

Method	CUB200						StanfordCars					
	T=5		T=10		T=20		T=5		T=10		T=20	
	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}
EWC [17]	21.6±0.4	38.2±0.3	15.8±0.7	32.6±0.5	12.3±0.8	27.2±0.6	24.3±0.6	44.0±0.6	14.3±0.8	34.5±0.7	10.9±1.2	27.9±1.1
LwF [21]	44.3±0.7	57.7±0.7	30.4±1.1	46.1±1.0	19.4±1.6	34.7±1.8	39.0±0.8	55.2±0.6	28.0±1.0	46.5±1.0	14.7±1.5	30.5±1.4
PASS [52]	34.5±0.5	48.6±0.4	27.0±0.9	42.3±0.9	18.1±1.2	36.9±1.1	33.3±1.0	48.9±0.9	26.4±1.1	41.0±0.8	13.9±1.6	28.1±1.2
IL2A [51]	36.9±1.2	51.3±1.4	29.4±1.3	45.5±1.7	20.8±2.0	35.1±2.3	39.4±1.3	49.1±1.5	27.3±1.6	45.1±1.7	14.2±2.3	28.7±2.8
FeTrIL [31]	41.9±0.5	53.2±0.6	36.9±0.7	48.2±0.6	34.6±1.0	45.3±0.9	46.0±0.7	58.5±1.0	40.5±0.8	53.4±0.9	32.5±1.1	43.3±1.3
FeCAM [10]	43.5±0.5	56.0±0.7	40.2±0.8	54.9±1.0	36.2±1.1	48.9±1.3	45.3±0.5	58.0±0.6	41.4±0.8	55.2±0.9	34.0±1.1	46.0±1.2
EFC [24]	58.3±0.4	68.9±0.6	51.0±0.6	63.3±0.7	46.1±1.0	59.3±1.3	50.1±0.5	63.2±0.7	43.1±0.8	57.6±0.7	28.1±1.8	48.2±1.6
AdaGauss	**60.4±0.5**	**69.2±0.7**	**55.8±0.5**	**66.2±0.8**	**47.4±0.8**	**60.6±1.0**	**53.3±0.7**	**64.0±1.0**	**47.5±0.9**	**58.5±1.1**	**34.8±1.2**	**48.6±1.3**

A.4 Different architecture of pretrained backbone

We test AdaGauss with different feautre extractors, namely ViT small and ConvNext. The results, presented in Tab. 7, are for EFCIL setting with 10 and 20 equal tasks and weights pretrained on ImageNet (as in Tab. 2). Using more modern feature extractors architectures further improves the results of AdaGauss.

Table 7: *AdaGauss* results with different backbone architectures. We report last accuracy | average accuracy.

	CUB200				FGVCAircraft			
	T=10		T=20		T=10		T=20	
Resnet18	55.8	66.2	47.4	60.6	47.5	58.5	34.8	48.6
ConvNext (small)	63.4	73.1	47.9	64.1	49.3	62.9	37.3	51.4
ViT (small)	68.2	77.5	48.9	67.5	48.0	60.6	35.7	50.2

A.5 Half dataset results

Learning from scratch is more challenging than half-dataset setting as it requires to incrementally train feature extractor, not just the classifier. However, using the pre-trained model (or learning from half) can be considered a more practical and real-life setting. Thus, we evaluated our method with a pre-trained model in Table 2. However, we have additionally compared our method to the mentioned baselines in a half-dataset setting using the original implementations under the same data augmentations as AdaGauss. Please note that we did not have enough time to perform hyperparameter search for our method - we utilized these from the equal task setting, whereas the results for other methods were optimized by their authors. We provide results in the Tab. 8.

AdaGauss performs better than PASS, SSRE, and FeTrIL (5 tasks) in the half-dataset setting. However, it is slightly worse than most recent baselines when using default hyperparameters. FeCAM, ACIL, and DS-AL freeze the feature extractor after the initial task, which can explain their good results in the half-dataset training.

Table 8: Half dataset in the first task results. We report last accuracy | average accuracy.

	CIFAR100				ImageNetSubset			
	T = 5		T = 10		T = 5		T = 10	
PASS	54.5	61.8	53.8	60.9	57.9	64.4	58.2	61.8
SSRE	55.7	63.9	54.9	63.2	58.3	65.2	61.4	67.7
FeTrIL	58.3	65.1	56.2	64.6	65.6	72.8	65.3	72.1
FeCAM	60.2	67.2	59.9	66.9	67.3	75.3	67.6	74.9
ACIL	57.8	66.3	57.7	66.0	67.0	74.8	67.2	74.6
DS-AL	61.4	68.4	61.4	68.4	68.0	75.2	67.7	75.1
AdaGauss	58.9	65.7	55.4	63.7	66.8	74.1	62.8	68.0

A.6 Predicted semantic shift for classes

Here, we verify whether our adaptation network can predict different shift for different classes in experiments from Tab.1. We test on CIFAR100 for $T = 10$ and the answer is positive, as shown in Fig. 12.

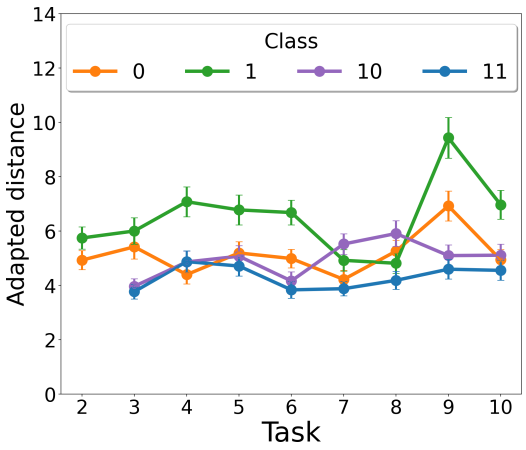

Figure 12: Predicted shift for different classes on CIFAR100 (10 equal tasks) by *AdaGauss*. The Euclidean distance is measured between old and new position in each task.

A.7 Batch norm influence on AdaGauss

We measure the accuracy achieved by AdaGauss in the EFCIL scenario from scratch (CIFAR100, ImageNetSubset) and pretrained (CUB200). We train for 10 tasks without batch norm layers and with frozen batch norm layers. Results are provided in Tab. 9. We can see that possesing batch-norm layers in Resnet18 is beneficial.

Table 9: *AdaGauss* results without or with frozen batch-norm. We report last accuracy and average accuracy separated by |.

	CIFAR100		ImageNetSubset		CUB200	
Resnet18 (no BN)	44.6±0.7	58.1±0.7	49.1±0.8	61.7±0.9	54.2±0.5	66.1±0.7
Resnet18 (freezed BN)	45.3±0.7	58.7±0.9	49.4±0.8	62.0±1.0	55.2±0.4	65.7±0.6
Resnet18	46.1±0.8	60.2±0.9	51.1±1.0	65.0±1.2	55.8±0.5	66.2±0.8

5. Conclusions

5.1. Summary of Contributions

The research presented in this dissertation addresses the fundamental challenge of Catastrophic Forgetting within the context of Exemplar-Free Class Incremental Learning and Generalized Continual Category Discovery. By operating under the constraint that past data cannot be stored—due to privacy, legal, or memory limitations, we have shifted the focus from simple data rehearsal to the preservation of knowledge through architectural, representational, and statistical innovations.

The first major contribution, the **SEED** framework, reimagines the stability-plasticity dilemma through the lens of modularity. We demonstrated that a monolithic backbone is inherently prone to interference, whereas a multi-expert ensemble can effectively isolate task-specific knowledge. By modeling classes as multivariate Gaussian distributions in a shared latent space, SEED allows for a principled selection of experts based on minimizing distribution overlap. This selective training ensures that new learning occurs with minimal disruption to previously acquired expertise, achieving state-of-the-art results on large-scale benchmarks.

The second contribution, **CAMP**, extends these principles to the more complex domain of Generalized Continual Category Discovery. Here, we addressed the dual problem of label sparsity and representation drift. By introducing projected feature distillation and a differentiable category adaptation network, we proved that it is possible to maintain consistent decision boundaries even as the latent space evolves to accommodate novel, unlabeled categories. CAMP serves as a bridge between supervised continual learning and open-world discovery, providing a powerful mechanism for autonomous systems to grow their knowledge base in non-stationary environments.

Finally, the **AdaGauss** method provides a critical diagnostic and therapeutic approach to task-recency bias. Our analysis revealed that favoritism toward recent tasks is often a symptom of dimensionality collapse in the latent space, where covariance matrices become near-singular. By implementing anti-collapse regularization and dynamic covariance adaptation, we restored the statistical validity of the classification space. This advancement ensures that models remain fair and

accurate across long learning horizons, effectively eliminating the systematic bias that has historically hindered exemplar-free methods.

5.2. Future Work

Building upon the methodologies established in this dissertation, several high-impact research directions emerge, particularly in the intersection of generative modeling and geometric deep learning.

- **Joint Training of the Adapter and Feature Extractor Networks** A compelling direction for future research involves the simultaneous optimization of the backbone feature extractor and the adaptation layers through a unified meta-learning objective. Current methodologies, including those presented in this dissertation such as *CAMP* and *AdaGauss*, primarily utilize adapters as post-hoc correction mechanisms or independent modules that operate on top of a relatively stable backbone. By transitioning to a joint training paradigm, the "adaptation error" could be backpropagated directly into the feature extractor, forcing the model to learn a representation space that is inherently "adapter-friendly." Such a system would likely require bi-level optimization strategies to ensure that the backbone learns features that minimize dimensionality collapse and representational drift for future, unseen tasks. This synergy would allow the feature extractor to develop a form of structured plasticity, potentially bridging the performance gap between exemplar-free systems and joint multi-task learning by anticipating the requirements of a continuously expanding taxonomy.
- **Generative Replay and Latent Diffusion:** A logical extension of the work on Gaussian class representations (SEED) and drift alignment (CAMP) is the integration of latent diffusion models. Instead of purely analytical distribution modeling, future systems could utilize the generative power of diffusion to "rehearse" synthesized representations of past tasks. This would provide a more expressive alternative to fixed multivariate Gaussians while remaining within the exemplar-free constraint.
- **Curvature-Aware Continual Learning:** Following the insights from AdaGauss regarding the geometry of the latent space, future research should investigate the use of Riemannian geometry and second-order optimization techniques. By considering the curvature of the loss landscape, we can potentially identify "flat minima" that are more robust to weight updates, further mitigating the stability-plasticity tradeoff without needing to store past samples.

- **Cross-Domain and Multi-Modal Adaptation:** The category discovery mechanisms developed in CAMP could be expanded to multi-modal settings where the model must continually align visual features with evolving linguistic concepts. Investigating how contrastive learning (e.g., CLIP-based architectures [73]) handles the sequential introduction of new modalities and domains remains a significant open challenge.
- **Decentralized and Federated Continual Learning:** Given the privacy motivations for exemplar-free learning, a promising direction is the application of SEED's expert selection in a federated environment. Allowing multiple edge devices to train specialized experts and selectively aggregate their knowledge could lead to highly robust, privacy-preserving global models that learn continuously from distributed data streams.

In conclusion, this work provides a comprehensive framework for building adaptive, autonomous neural systems that can learn throughout their operational lifetimes. By treating the latent space as a dynamic, geometric entity rather than a static mapping, we have paved the way for more resilient and fair artificial intelligence.

Bibliography

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[2] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[3] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[5] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[6] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017.

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

[9] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[11] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[12] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[13] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[15] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

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

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

[19] Sebastian Thrun. Lifelong learning algorithms. In *Learning to learn*, pages 181–209. Springer, 1998.

[20] Mark Bishop Ring. *Continual learning in reinforcement environments*. The University of Texas at Austin, 1994.

[21] James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419, 1995.

[22] Paul W Frankland and Bruno Bontempi. The organization of recent and remote memories. *Nature reviews neuroscience*, 6(2):119–130, 2005.

[23] Marcus K Benna and Stefano Fusi. Computational principles of synaptic memory consolidation. *Nature neuroscience*, 19(12):1697–1706, 2016.

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

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

[26] Sylvestre-Alvise Rebuffi et al. icarl: Incremental classifier and representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2001–2010, 2017.

[27] Saihui Hou et al. Learning a unified classifier incrementally via rebalancing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 831–839, 2019.

[28] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning (ICML)*, pages 3987–3995, 2017.

[29] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *Neural Information Processing Systems (NeurIPS) Deep Learning Workshop*, 2015.

[30] Andrei A Rusu et al. Progressive neural networks. In *arXiv preprint arXiv:1606.04671*, 2016.

[31] Rahaf Aljundi et al. Expert gate: Lifelong learning with a network of experts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3366–3375, 2017.

[32] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2990–2999, 2017.

[33] Grzegorz Rypeśc, Daniel Marczak, Sebastian Cygert, Tomasz Trzciński, and Bartłomiej Twardowski. Category adaptation meets projected distillation in generalized continual category discovery. In *European Conference on Computer Vision (ECCV)*, 2024.

[34] Wickliffe C Abraham and Anthony Robins. Memory retention–the synaptic stability versus plasticity dilemma. *Trends in Neurosciences*, 28(2):73–78, 2005.

[35] Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in Psychology*, 4:504, 2013.

[36] Chrisantha Fernando et al. Pathnet: Evolution channels gradient descent in super neural networks. In *arXiv preprint arXiv:1701.08734*, 2017.

[37] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. Tiny episodic memories in continual learning. In *arXiv preprint arXiv:1902.10486*, 2019.

[38] Marc Masana et al. Class-incremental learning: survey and performance evaluation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2022.

[39] 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.

[40] Grzegorz Rypeść, Sebastian Cygert, Valeriya Khan, Tomasz Trzciński, Bartosz Zieliński, and Bartłomiej Twardowski. Divide and not forget: Ensemble of selectively trained experts in continual learning. In *International Conference on Learning Representations (ICLR)*, 2024.

[41] Grzegorz Rypeść, Sebastian Cygert, Tomasz Trzciński, and Bartłomiej Twardowski. Task-recency bias strikes back: Adapting covariances in exemplar-free class incremental learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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

[43] 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*, pages 3911–3920, 2023.

[44] Dipam Goswami, Yuyang Liu, Bartłomiej Twardowski, and Joost Van De Weijer. Fecam: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. *Advances in Neural Information Processing Systems*, 36:6582–6595, 2023.

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

[46] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[47] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

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

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

[50] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[51] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object represen-

tations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013.

[52] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019.

[53] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7492–7501, 2022.

[54] Huiping Zhuang, Run He, Kai Tong, Ziqian Zeng, Cen Chen, and Zhiping Lin. Ds-al: A dual-stream analytic learning for exemplar-free class-incremental learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17237–17244, 2024.

[55] 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*, pages 6982–6991, 2020.

[56] Leonardo Ravaglia, Manuele Rusci, Davide Nadalini, Alessandro Capotondi, Francesco Conti, and Luca Benini. A tinyml platform for on-device continual learning with quantized latent replays. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 11(4):789–802, 2021.

[57] 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.

[58] 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.

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

[60] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *Conference on Computer Vision and Pattern Recognition*, CVPR, 2017.

[61] Liyuan Wang, Xingxing Zhang, Qian Li, Jun Zhu, and Yi Zhong. Coscl: Cooperation of small continual learners is stronger than a big one. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27,*

2022, Proceedings, Part XXVI, pages 254–271, 2022.

[62] Thang Doan, Seyed Iman Mirzadeh, and Mehrdad Farajtabar. Continual learning beyond a single model. *arXiv preprint arXiv:2202.09826*, 2022.

[63] Bingchen Zhao and Oisin Mac Aodha. Incremental generalized category discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19137–19147, 2023.

[64] Yanan Wu, Zhixiang Chi, Yang Wang, and Songhe Feng. Metagcd: Learning to continually learn in generalized category discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1655–1665, 2023.

[65] Hyungmin Kim, Sungho Suh, Daehwan Kim, Daun Jeong, Hansang Cho, and Junmo Kim. Proxy anchor-based unsupervised learning for continuous generalized category discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16688–16697, 2023.

[66] Subhankar Roy, Mingxuan Liu, Zhun Zhong, Nicu Sebe, and Elisa Ricci. Class-incremental novel class discovery. In *European Conference on Computer Vision*, pages 317–333. Springer, 2022.

[67] Alex Gomez-Villa, Bartlomiej Twardowski, Lu Yu, Andrew D. Bagdanov, and Joost van de Weijer. Continually learning self-supervised representations with projected functional regularization, 2022.

[68] Ahmet Iscen, Jeffrey Zhang, Svetlana Lazebnik, and Cordelia Schmid. Memory-efficient incremental learning through feature adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 699–715. Springer, 2020.

[69] Wuxuan Shi and Mang Ye. Prototype reminiscence and augmented asymmetric knowledge aggregation for non-exemplar class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1772–1781, 2023.

[70] Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Cheng-lin Liu. Class-incremental learning via dual augmentation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[71] Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 2020.

[72] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348*, 2021.

[73] 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.