CLEO: Continual Learning of Evolving Ontologies

Shishir Muralidhara[1], Saqib Bukhari[2], Georg Schneider[2], Didier Stricker[1,3], and René Schuster[1,3]

[1] DFKI – German Research Center for Artificial Intelligence
[2] ZF Friedrichshafen AG
[3] RPTU – University of Kaiserslautern-Landau
{firstname.lastname}@{dfki.de, zf.com}

Abstract. Continual learning (CL) addresses the problem of catastrophic forgetting in neural networks, which occurs when a trained model tends to overwrite previously learned information, when presented with a new task. CL aims to instill the lifelong learning characteristic of humans in intelligent systems, making them capable of learning continuously while retaining what was already learned. Current CL problems involve either learning new domains (domain-incremental) or new and previously unseen classes (class-incremental). However, general learning processes are not just limited to learning information, but also refinement of existing information. In this paper, we define CLEO – Continual Learning of Evolving Ontologies, as a new incremental learning setting under CL to tackle evolving classes. CLEO is motivated by the need for intelligent systems to adapt to real-world ontologies that change over time, such as those in autonomous driving. We use Cityscapes, PASCAL VOC, and Mapillary Vistas to define the task settings and demonstrate the applicability of CLEO. We highlight the shortcomings of existing CIL methods in adapting to CLEO and propose a baseline solution, called Modelling Ontologies (MoOn). CLEO is a promising new approach to CL that addresses the challenge of evolving ontologies in real-world applications. MoOn surpasses previous CL approaches in the context of CLEO.

Keywords: Continual Learning · Continual Semantic Segmentation · Class-Incremental Learning

1 Introduction

Advancements in AI have led to the emergence of systems that can match or even outperform human abilities in various domains. However, these systems are often confined to specific datasets and tasks and their efficacy is affected when faced with new tasks or data. This presents a significant hurdle in practical real-world AI applications where the circumstances are constantly evolving. For example, a self-driving car needs to adapt to new traffic regulations, signage, and weather conditions. When a trained model encounters new information or tasks, it experiences a phenomenon known as *catastrophic forgetting* [26]. This issue arises from the rigidity of neural networks, where previously learned knowledge is

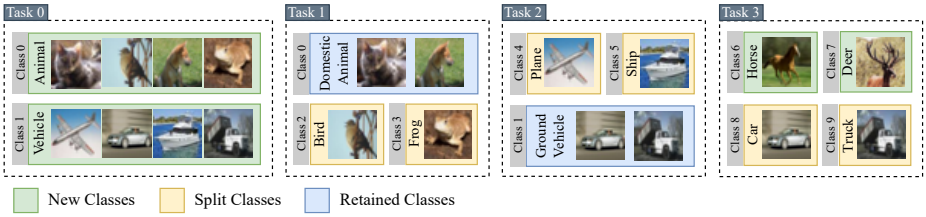


Fig. 1: CLEO extends class-incremental learning (CIL) by removing the constraint that incremental sets of classes must be non-overlapping. This allows for more flexible and nuanced learning, as existing classes can be refined into more specific classes.

overwritten when learning something new. Consequently, the model's performance significantly deteriorates, which can have severe repercussions in scenarios where the model has to acclimate to dynamic environments or critical applications where performance is imperative. Therefore, tackling forgetting is crucial when developing AI systems with lifelong learning.

In contrast, learning in humans is continuous and dynamic through which we continually assimilate new information while preserving and enriching previously learned knowledge. The goal of CL is to instill neural networks with this learning characteristic and make them capable of extending when objectives change. In incremental learning, which involves learning a sequence of tasks, the term *task* can represent changes either in the distribution of data, referred to as domain-incremental learning (DIL), or changes in the output space, which is known as class-incremental learning (CIL). However, learning is not limited only to the acquisition of new knowledge; it also involves refining and deepening our grasp of existing information. To illustrate, our initial understanding of animals might have been confined to recognizing a broad categorization, like distinguishing between birds and other animals. However, over time, our learning process matures to incorporate finer, more nuanced insights about individual animals. This highlights the necessity of broadening the scope of CL to accommodate both the introduction of novel classes, and also the enhancement of existing knowledge.

We propose a novel incremental setting termed CLEO – Continual Learning of Evolving Ontologies. CLEO represents a progressive extension of CIL, overcoming the confining assumption that all newly encountered classes are entirely novel and previously unseen. CLEO acknowledges the dynamic nature of learning, where classes may not only emerge but also evolve over time. CLEO combines elements of novelty and refinement, paving the way for more resilient AI systems. The real-world relevance of CLEO is exemplified by the Mapillary Vistas dataset [30], where an initial set of classes has evolved into more fine-grained and specific classes. An illustration of CLEO is presented in Fig. 1 using the CIFAR-10 dataset [17] as an example. In the initial task, the system learns two broad categories, namely *animals* and *vehicles*. Subsequently, more specific classes, such as *bird, frog, plane, and ship*, are split from the original classes. CLEO presents a challenge where the system must not only learn the evolved

classes, which it had previously encountered within a different semantic context, but also retain the remaining information. In this example, after *bird and frog* were separated from the *animal* class, *cat* and *dog* are left as the remaining *domestic animal* class. As in CIL, CLEO also covers the case that entirely new classes are learned, like *e.g.* the classes *horse* and *deer* in the last task of Fig. 1. CLEO embodies the fundamental idea that AI systems must possess the ability to adapt to dynamic shifts in which classes emerge, evolve, and split over time. By integrating this adaptability into AI systems, we enhance their robustness in real-world applications. This paper makes several significant contributions:

- **Formalizing Continual Learning of Evolving Ontologies:** A primary contribution is the introduction and formalization of the concept of learning evolving ontologies, from a continual learning perspective.
- **Defining Task Settings:** To provide a comprehensive understanding of CLEO, we define several task settings for semantic segmentation, using three diverse datasets: Cityscapes [8], PASCAL VOC [11], and Mapillary Vistas [30].
- **Evaluation Metrics for CLEO:** By identifying class groups and measuring their performance, we define a more nuanced evaluation of how effectively the system learns refined classes and retains information.
- **Baseline Solution – MoOn:** To operationalize the concept for semantic segmentation, we propose a baseline solution called "Modelling Ontologies" (MoOn), to handle the challenges inherent in CLEO.

2 Related Work

2.1 Continual Learning

Continual learning (CL), also known as lifelong learning, and its sub-field of incremental learning, are critical areas of machine learning that address a fundamental limitation of traditional machine learning models: Their rigidity in learning new tasks. CL is a learning paradigm that aims to train models that can learn new tasks and information without forgetting what was previously learned. This contrasts traditional machine learning, where models are typically trained on a single dataset and then remain fixed. Continual learning presents a challenge of balancing between learning new information and retaining the old [27]. Overemphasizing learning new information leads to forgetting the old while focusing too much on retaining old information hinders learning the new task. CL approaches can be grouped into three categories: Architecture-based, replay-based methods, and regularization-based methods.

Architecture-based approaches modify the architecture of neural networks to accommodate new tasks without overwriting previous information. These include methods such as parameter isolation [12, 24, 25], dynamically growing networks [1, 32, 39], or incorporation of additional components such as memory modules [2, 34]. These methods can face scalability challenges as network architecture grows, increasing memory and computational demands.

Replay-based approaches are of two types: Experience replay or rehearsal, involves storing a few instances from the previous task [3, 6, 23], which are used during training on a new task. Generative replay substitutes the memory buffer with a generative model to sample previous task instances when learning on a new task [15, 33, 35]. Challenges include privacy, data availability, memory and computational constraints in training generative models.

Regularization-based approaches [16, 20, 41] strike a balance between stability and plasticity when learning new tasks by introducing a regularization term into the loss function. One of the approaches to regularization is to determine the importance of parameters to the previous tasks and penalize overwriting them. Distillation is another approach which preserves previously learned knowledge by distilling information from the old model to the new.

2.2 Continual Semantic Segmentation

Continual semantic segmentation (CSS) [5,37,38,40] is a specialized domain within CL that focuses on the task of semantic segmentation. Incremental Learning Techniques [28] (ILT) is a regularization-based method that distils knowledge from previously trained task models and additionally freezes the encoder for the current task, which acts as an additional regularization constraint. Modeling the Background [4] (MiB) addresses the problem of background shift in incremental segmentation. It proposes a novel distillation loss that accounts for the changing background between tasks. AWT [13] addresses the background shift in classifier initialization, and transfers the most relevant weights from the previous classifiers to the new classifier. In our baseline solution for CLEO, we will build on the ideas of MiB and extend it to model the semantic shift for all classes. PLOP [10] argues that treating the background class as a single class as done in MiB, which includes both previously encountered and novel classes can lead to catastrophic forgetting of the old classes as it learns new classes. PLOP addresses this problem by pseudo-labeling old classes in the background. EWF [36] combines the weights of the old and new model trained with regularization to give a more balanced model. SDR [29] preserves latent representations of old classes through prototype matching and frees space for new classes through feature sparsity and contrastive learning. The decoder preserves previous knowledge via output-level distillation. RCIL [42] uses a representation compensation module to decouple the representation learning of old and new classes. This helps to prevent the model from forgetting old classes when learning new classes. REMINDER [31] proposes a class-similarity weighted distillation that distills knowledge from a previous model on old classes that are similar to the new classes. This reduces forgetting of old classes and improves learning of new classes.

2.3 Evolving Ontologies

An emerging area within incremental learning focuses on learning evolving classes, which involves the refinement of existing knowledge, where a broad initial grouping of classes evolves over time into more specific and nuanced classes. In contrast,

in CIL, each task consists of a non-overlapping subsets of novel classes.

Hierarchical Label Expansion [18] (HLE) for classification involves a model initially learning coarse classes, which then evolve into more fine-grained categories during each task across varying hierarchical levels. The expansion is restrictive, where the entire subset of classes at the current level expands in the next level during each task. CLEO differs from HLE in the following aspects:

- CLEO allows a more flexible evolution at any point within the sequence of tasks, in which certain classes can remain (partially) unexpanded or expand into sub-groups.
- The baseline solution for HLE is a rehearsal-based method that uses pseudo-labeling from previous classifiers – one for each level of the hierarchy – for determining the importance of data. We present a regularization-based solution that uses knowledge distillation.
- HLE considers image classification as application, while CLEO additionally deals with the more challenging problem of semantic segmentation.

Learning with Evolving Class Ontologies [22] (LECO) has been the first attempt to address evolving ontologies in semantic segmentation. However, despite its intended framing as a continual learning problem, LECO deviates from the defining principles of continual learning. We address these shortcomings and define this problem strictly within the bounds of continual learning. The primary distinctions between our framework and LECO are outlined below:

- LECO requires and stores all historical data, a constraint that contradicts the definition of CL [19], that all data is not available at once.
- LECO primarily revolves around the dilemma of whether to label new data or relabel the old data with the current label space. In contrast, CLEO confines itself to utilizing the input data and annotations within the dataset.
- Due to the availability of historical data, LECO employs naive strategies such as fine-tuning, freezing, or joint training. We highlight the limitations of existing CIL methods and propose a baseline specifically for CLEO.
- CLEO explicitly addresses catastrophic forgetting, which is one of the core problems of continual learning. Since LECO retrains entirely using all the data, the problem of catastrophic forgetting does not even arise.

3 CLEO

Deep learning has drawn inspiration from biological systems at various levels, from the way neurons form the fundamental building blocks of artificial neural networks to learning paradigms such as curriculum learning and reinforcement learning. Continual learning is another learning paradigm that aims to instill lifelong learning in intelligent systems. As discussed previously, current problems in continual learning involve adaptation to different domains, such as handling adverse conditions in autonomous driving or learning new classes over time. A

common characteristic in both these scenarios is that the task always represents new and previously unseen data. However, it is our contention that the scope of continual learning should not be limited solely to the discovery of novel classes. It should also include the enhancement and refinement of existing knowledge. This aligns closely with the way humans continually adapt and refine knowledge through neuroplasticity.

In light of this perspective, we introduce CLEO – Continual Learning of Evolving Ontologies. CLEO is centered around the idea of removing the constraints of class-incremental learning and allows classes that have a hierarchical relation (partially) overlap between tasks, thereby facilitating the refinement of already learned information. CLEO is a real and open-world problem and this is exemplified in the Mapillary Vistas dataset [30]. The initial version of the dataset defined 66 classes, which later evolved into 124 classes in the latest version. It is noteworthy to observe the transformation within class categorization. On one hand, classes that were previously categorized under *unlabeled*, such as *traffic island* emerge as novel classes. Conversely, several pre-existing classes undergo refinement, leading to the creation of new, more specific categories. In Fig. 2, we can observe a tangible example of this evolution, where the previous *billboard* class has evolved into more specialized classes, including signage classes for advertisements, information, stores, and differentiating between the front and back of the signs. In critical applications like autonomous driving, it becomes imperative for AI systems to swiftly adapt to these nuanced changes, perpetually learn, and enhance scene understanding precision.

Let $T = \{0, 1, 2, ..., n\}$ denote a sequence of $n + 1$ tasks. Each task is associated with a specific subset of classes denoted as C_t. The union of all classes across tasks is $C = \bigcup_{t=0}^{n} C_t$, representing the entire set of classes encountered during the learning process. At each task t, a subset of classes C_t is introduced incrementally similar to CIL. $C_{0:t} = C_0 \cup ... \cup C_{t-1} \cup C_t$ is the set of all classes seen until step t. CLEO acknowledges the possibility of class splitting, where an existing class c_{t-1} from $C_{0:t-1}$ can evolve into multiple new classes in C_t. Formally this can be expressed as, $C_t \subset C$, $C_i \cap C_j = \phi$ for $i \neq j$ and the class splitting is expressed as $c_{t-1} \in C_{0:t-1} \rightarrow \{c_{k+1}, c_{k+2}, ..., c_{k+m}\} \subseteq C_t$. Note that the splitting of a class is not necessarily exhaustive, *i.e.* $c_{t-1} \in C_{0:t-1}$ may persist. An example for such an evolution of classes is given in Fig. 1. The learning objective of a task remains consistent with CIL: Train a model to recognize and differentiate the current subset of classes C_t, while ensuring that the model retains knowledge from the previously encountered classes in $C_{0:t-1}$. As in CIL, the model is not provided access to data from previous tasks.

4 MoOn: Modelling Ontologies for CLEO

Class-incremental learning involves the process of training a model to incrementally learn new classes. At each increment, the model is presented with images and the corresponding ground truth which contains annotations for the new classes



Fig. 2: The Mapillary Vistas dataset [30] provides a real-world example for an evolving ontology. This figure compares ground truth annotations for the same image in two dataset versions. It illustrates the introduction of new classes, such as *traffic island* and the finer categorization of existing classes such as *billboard*.

being learned. In class-incremental semantic segmentation (CISS), at each step, all unlabeled areas are marked as *background*. Thus in CISS, the phenomenon of *semantic background shift* occurs, because in each task, the meaning of the *background* class (bg) changes. Distillation loss is a technique of transferring knowledge from an old model f_{t-1} to a new model f_t, and this is achieved by comparing the old model's prediction to the new model's predictions. The distillation loss enforces the new model's predictions to resemble those of the old model by penalizing any deviations and this helps mitigate forgetting. Consider an image x with N pixels and the corresponding ground truth y at the current task t with the class set C_t. The standard distillation loss in [4] is defined as follows:

$$\ell_{kd}^t(x,y) = -\frac{1}{N}\sum_{i\in x}\sum_{c\in C_{0:t-1}} p_x^{t-1}(i,c)\log \hat{p}_x^t(i,c) \tag{1}$$

where $p_x^{t-1}(i,c)$ is the probability for class c in pixel i given by the model f_{t-1} and $\hat{p}_x^t(i,c)$ is for class c in pixel i given by the model f_t but re-normalized across classes $C_{0:t-1}$:

$$\hat{p}_x^t(i,c) = \begin{cases} 0 & \text{if } c \in C_t \setminus \{bg\} \\ \dfrac{p_x^t(i,c)}{\sum_{k\in C_{0:t-1}} p_x^t(i,k)} & \text{if } c \in C_{0:t-1} \end{cases} \tag{2}$$

Standard distillation ignores the dynamic nature of the background in incremental semantic segmentation. The *background* class in segmentation is a catch-all class that includes all previously seen classes and potential future classes. Furthermore, the background class is present in all tasks and changes as new classes are added, leading to background shift. A class $c \in C_t$ added during the current task was previously part of the *background* in C_{t-1}. For a pixel representing a new class at the current task t, the old model f_{t-1} would predict the *background* class, and if standard distillation loss was applied naively, it would penalize the new model f_t for predicting the correct class. Modeling the Background (MiB) [4] proposes a novel distillation loss that explicitly addresses the background shift in incremental segmentation. MiB does not directly compare the prediction of the

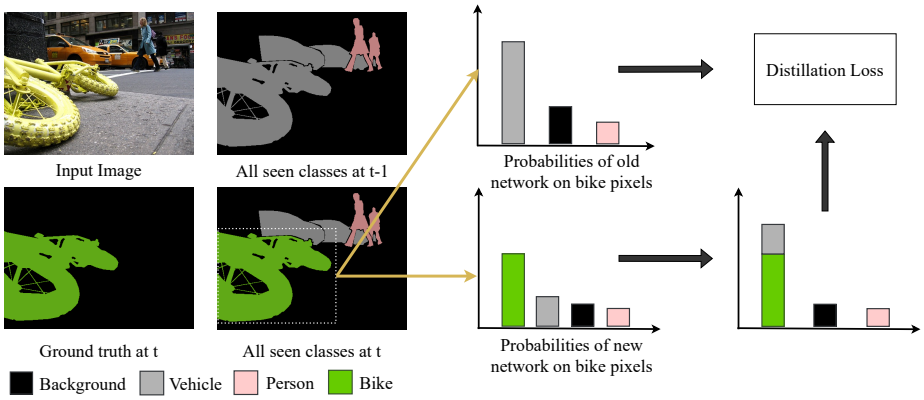


Fig. 3: Modelling Ontologies (MoOn) generalizes the idea of MiB [4] for all classes. During distillation, forgetting is mitigated, by grouping class-logits according to the evolution of the class ontology. In this example, the teacher network of the previous task predicts the *vehicle* class, while the model tries to learn *bicycle*. To avoid penalizing this, the predictions for *vehicle* and *bicycle* are combined before the distillation loss.

new model f_t with the corresponding prediction of the old model f_{t-1}. Instead, the background prediction by f_{t-1} is compared with the f_t prediction of either a new class *or* the background. Eq. (3) modifies Eq. (2) by additionally considering the whole label space instead of limiting it to $C_{0:t-1}$. MiB acknowledges that pixels that were thought to be part of the background by the old model might actually be the new classes the current model is trying to learn, thus avoiding penalizing the probabilities of new classes being learned.

$$\hat{p}_x^t(i,c) = \begin{cases} p_x^t(i,c) & \text{if } c \neq bg \\ \sum_{k \in C_t} p_x^t(i,k) & \text{if } c = bg \end{cases} \tag{3}$$

MiB is designed for CIL, where the new classes being learned at the current step, were previously unseen and are always a part of the background class. Therefore, it would suffice to take into consideration the background class, and hence modeling the background. Whereas CLEO is designed for learning of evolving classes and the new classes being learned may have been previously unseen and part of the background or been part of a broader class. As seen in Fig. 3, the class *bike* being learned at the current step may have been previously learned as a *vehicle* class grouped with other classes like *truck* and *bus*. Therefore we need to take into consideration that within CLEO, new classes may not always have been a part of the *background* class previously.

To this end, we propose Modelling Ontologies (MoOn) as a baseline solution for CLEO, which can be seen as a generalized version of MiB that accounts for the fact that new classes may emerge from any existing class, and not just the background class. Similar to how MiB considered the possibility of the new class

pixel belonging to the background class previously, MoOn additionally includes the possibility of the pixel belonging to an existing class. In MiB, the old model f_{t-1} may predict the background class for the new class being learned, whereas the old model in MoOn may predict an existing class $c \in C_{0:t-1}$. Therefore, the correct distillation would be to distill the knowledge from the old model f_{t-1} for the existing class $c \in C_{0:t-1}$ that is being split into new classes in C_t. In the illustrated scenario depicted in Fig. 3, distillation would involve transferring knowledge from the old model regarding the *vehicle* class, while training the new model to predict the *bike* class. To achieve this, at each step t, MoOn uses a set S_t of class maps $(c_{t-1} \rightarrow \{c_{k+1}, c_{k+2}, ..., c_{k+m}\} = C_{new})$ to identify the class $c_{t-1} \in C_{0:t-1}$ from which the current classes in C_t are split. If the new class c_t is completely new, then the superclass c_{t-1} in the class mapping would be the *background* class *bg*. We define $C_{e,t}$ as the set of all evolving classes c_{t-1} in S_t. The modified distillation loss for learning evolving classes in CLEO is given as:

$$\hat{p}_x^t(i,c) = \begin{cases} p_x^t(i,c) & \text{if } c \notin C_{e,t} \\ p_x^t(i,c_{t-1}) + \sum\limits_{c_n \in C_{new}} p_x^t(i,c_n) & \text{if } c \in C_{e,t} \end{cases} \tag{4}$$

4.1 Datasets

Towards the introduction of CLEO for semantic segmentation, we consider the following three diverse datasets:

- **Cityscapes:** Cityscapes [8] is a large-scale dataset of urban street scenes captured in 50 different cities under daytime and good weather conditions. The dataset provides annotations for 30 classes out of which 19 are used for evaluation and grouped into 7 categories. An overview of the hierarchy of the 19 semantic classes grouped by their parent categories is presented in the supplementary.
- **PASCAL VOC:** PASCAL VOC 2012 [11] is part of the Visual Object Classes Challenge, which ran from 2005 to 2012. PASCAL VOC is a more general dataset like COCO [21] consisting of 20 categories, including vehicles, household objects, animals, and persons. The categorization by [43] of these classes along with intermediate groupings is shown in the supplementary.
- **Mapillary Vistas:** Mapillary Vistas [30] is a large-scale street-level image dataset for understanding street scenes around the world. It covers a wider range of geographic locations and diverse weather and illumination conditions. The initial version of the dataset provided annotations for 66 classes, which later nearly doubled into 124 classes in the current version v2.0.

4.2 Baselines and Implementation

We evaluate our method's performance against standard CL baselines, namely *fine-tuning* and *joint training*. Fine-tuning involves using the previous step model

as the starting point when learning on the current task. This is similar to transfer learning and even results in positive forward transfer. However, since there is no explicit attempt to prevent catastrophic forgetting, it involves the highest amount of forgetting and forms the lower baseline. The joint training model involves learning all of the tasks together in a single step, and as there is no incremental learning there is zero forgetting and this forms the theoretical upper baseline. Since in CLEO a redefinition of classes may occur because of the evolving ontology, we define the set of classes for the *joint training* as the final set of classes after the last task. Additionally, we compare against the state-of-the-art (SOTA) methods such as MiB [4], PLOP [10] and RCIL [42].

For MiB, and our proposed approach, we adopt the framework proposed by MiB [4]. In the case of PLOP and RCIL, which involve modifications specific to their approaches, we adopt the RCIL framework [42]. All these approaches use the DeepLabV3 [7] architecture as segmentation network, utilizing a ResNet-101 [14] backbone with an output stride of 16. The backbone's initialization uses a pre-trained model from ImageNet [9]. The training methodology utilizes SGD, with a learning rate of 0.01 for the initial learning step and 0.001 for subsequent steps. Across all learning steps, we train the model using a batch size of 24, for 50 epochs on Cityscapes and for 30 epochs on PASCAL and Mapillary Vistas.

4.3 Evaluation Metrics

Mean Intersection-over-Union (mIoU) is a commonly used metric for evaluating semantic segmentation models. It measures the average overlap between the predicted segmentation and the ground truth segmentation for all classes. To calculate the mIoU, the Intersection-over-Union (IoU) is first calculated for each class, and then the IoU scores for all classes are averaged. The performance of CL models are often evaluated by measuring the amount of information forgotten by comparing the model's performance on old tasks against the joint training baseline. Within the context of CLEO, while learning evolving classes, forgetting can occur at multiple levels. We identify three types of class groups and analyze the mIoU results for each of them:

- **Unsplit Classes:** These classes are learned in the first task and are not further split. Their performance is expected to be largely unaffected, as they have not been split nor learned incrementally.
- **Split Classes:** These classes have been split either from the background or a previously learned superclass. The mIoU results for these split classes will provide insight into the model's ability to learn new classes.
- **Retained Classes:** These are classes remaining in the parent class after it has been split partially to yield new classes. Analyzing the mIoU scores for retained classes sheds light on how well the model retains knowledge.

Table 1: Results for our two CLEO settings on Cityscapes [8] after learning all tasks.

Method	CS-Ex1				CS-Ex2			
	Unsplit	*Split*	*Retained*	**All**	*Unsplit*	*Split*	*Retained*	**All**
Fine-Tuning	00.00	00.00	00.00	00.00	00.00	14.46	00.00	08.68
Joint Training	90.26	61.15	48.04	55.62	90.26	61.15	48.04	55.62
MiB [4]	90.24	03.66	07.99	09.11	90.23	18.25	11.39	18.88
PLOP [10]	87.58	34.36	13.02	28.91	86.01	31.91	15.51	28.10
RCIL [42]	89.82	00.00	07.97	06.88	89.00	00.00	07.70	06.76
MoOn (Ours)	90.21	39.50	37.00	**39.31**	90.23	42.28	35.78	**40.62**

Table 2: Task-wise results for CS-Ex2 after learning the final task.

Method	Task 0	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	All
Fine-Tuning	00.00	00.00	00.00	00.00	00.00	00.00	34.70	08.68
Joint Training	47.50	97.57	63.09	30.20	88.31	64.31	57.93	55.62
MiB [4]	19.82	02.81	09.26	00.19	74.00	45.62	15.54	18.88
PLOP [10]	22.38	95.08	41.59	00.00	84.81	53.84	13.21	28.10
RCIL [42]	16.90	00.00	00.00	00.00	00.00	00.00	00.00	06.76
MoOn (Ours)	38.11	96.89	56.26	09.19	88.16	60.97	26.10	**40.62**

4.4 Experimental Settings and Results

We introduce seven initial experimental settings, that are designed for the previously introduced datasets and resemble the complexity and diversity of CLEO beyond classical CIL. For completeness and better understanding, we present the detailed semantic evolution of all settings by listing C_t for every task in the supplementary, including the class groups as introduced in Sec. 4.3.

5 Experiments

Cityscapes From the Cityscapes class hierarchy, we derive two experimental settings. Both settings start in task 0 by learning the seven parent classes which together consist of the 19 grouped classes. The settings involve the incremental splitting of individual classes from parent classes. We do not always completely split a parent class, thus emphasizing not only learning the split class but also retaining the remaining information on the superclass.

- **CS-Ex1:** Each step involves splitting out a single class from each parent class. Since there is an unequal number of classes in the parent classes, we repeat this process until we have 19 individual classes, resulting in six tasks.
- **CS-Ex2:** At each step, we split all but one of the classes from a single parent class to form individual classes. This results in 19 classes in six evolutionary increments (seven tasks in total) for six parent classes excluding *sky*.

Table 3: Results for our three CLEO settings on PASCAL [11] after learning all tasks.

Method	VOC-Ex1				VOC-Ex2				VOC-Ex3			
	Unsplit	*Split*	*Retained*	**All**	*Unsplit*	*Split*	*Retained*	**All**	*Unsplit*	*Split*	*Retained*	**All**
Fine-Tuning	37.77	36.31	00.00	29.68	37.74	05.38	00.00	07.81	36.91	25.68	00.00	23.08
Joint Training	90.19	72.53	84.12	76.91	90.48	73.78	89.14	77.75	89.84	75.27	87.36	78.38
MiB [4]	88.91	37.28	54.99	47.05	86.99	11.28	37.87	22.84	87.01	35.75	33.28	40.28
PLOP [10]	86.85	38.67	43.56	45.62	85.77	16.18	03.67	21.26	83.41	36.74	06.61	36.88
RCIL [42]	88.08	00.84	41.11	19.29	85.29	01.43	23.85	13.18	87.45	01.99	18.63	12.51
MoOn (Ours)	88.95	55.98	68.84	**62.52**	84.45	22.08	25.42	**28.82**	83.33	47.51	42.36	**50.19**

The results from these settings are presented in Tab. 1 and the task-wise results of *CS-Ex2* in Tab. 2. We can observe the shortcomings of existing CIL methods in handling evolving ontologies, whereas our baseline solution, MoOn, outperforms these methods. All methods achieve similar results on the unsplit classes, which is *sky* in both settings. However, when learning split classes, only PLOP comes close to our approach. Notably, MoOn surpasses PLOP by a significant margin in preserving information within the existing class. In the second task setting, the last step involves learning five vehicle classes, which are well-represented in the dataset. FT and MIB methods benefit from this, whereas PLOP and RCIL achieve similar results across both settings. We provide task-wise results of *CS-Ex1* and visualizations for MoOn in the supplementary.

PASCAL VOC From the PASCAL class hierarchy, we derive three experimental settings that represent the challenges of CLEO. In the first step (task 0), the model learns all the 20 classes grouped into the four parent classes namely *animals*, *household*, *person*, and *vehicle*. The *person* class does not contain any subclasses and therefore does not undergo splitting.

- **VOC-Ex1:** This setting includes splitting into intermediate sub-groups. Task 1 includes learning the sub-groups *Farmyard, furniture,* and *2-wheeler* which in turn contain individual classes. In the next and final step, these sub-groups are split into their individual classes.
- **VOC-Ex2:** One class is split sequentially from each parent class resulting in five increments. At the end of the learning, we retain the individual classes *cat*, *bottle*, and the subgroup *4-wheeler* in their respective parent classes.
- **VOC-Ex3:** This setting involves splitting all but one of the classes from each parent class sequentially, resulting in four tasks, with three evolutionary steps for the three parent classes excluding *person* which remains unsplit.

The results on the three settings are shown in Tab. 3, while the corresponding visualizations for *VOC-Ex3* are presented in Fig. 4. The performance on the unsplit classes, *person* class in this dataset, remains consistent across the CL methods. We observe the biggest difference in performance between our approach and others for *VOC-Ex1*. This is likely due to the more challenging multiple splitting of classes into intermediate subclasses. The lowest results for all methods

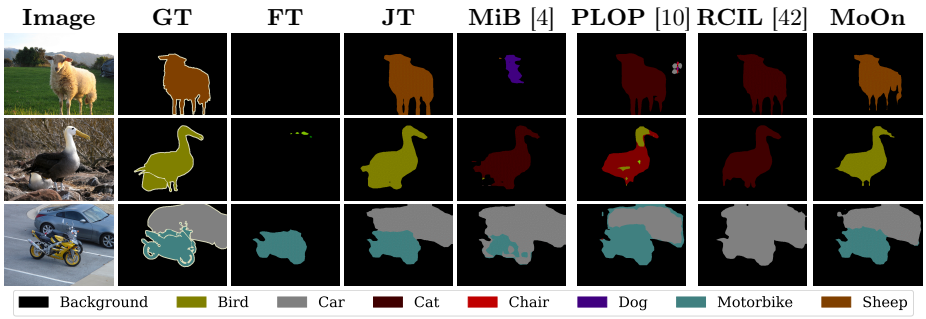


Fig. 4: Qualitative comparison on PASCAL VOC [11] *VOC-Ex3* after the final task.

Table 4: Results for CLEO settings on Mapillary Vistas [30] after learning all tasks.

Method	MV-Ex1				MV-Ex2			
	Unsplit	*Split*	*Retained*	**All**	*Unsplit*	*Split*	*Retained*	**All**
Fine-Tuning	00.00	18.59	00.00	08.69	00.00	01.47	00.00	00.69
Joint Training	28.37	22.68	25.79	25.51	28.37	22.68	25.79	25.51
MiB [4]	27.02	16.33	17.74	21.27	09.83	04.31	00.07	06.45
PLOP [10]	27.07	13.14	18.44	19.86	03.59	01.26	00.00	02.22
RCIL [42]	28.54	01.57	12.78	14.65	06.39	00.55	00.17	03.16
MoOn (Ours)	26.72	16.67	19.47	**21.44**	09.81	04.40	00.01	**06.49**

are for *VOC-Ex2*, which involves a longer sequence of tasks, and learning sets of mixed classes one from each parent class at each step.

Mapillary Vistas Using the two dataset versions of Mapillary Vistas [30], we have defined two CLEO experiments. Both settings begin by learning the original 66 classes in task 0. The newly introduced classes are grouped into their ten parent classes from the previous version resulting in the original classes. The class mapping between the new classes and their parent classes is provided in the supplementary. The subsequent steps involve learning these new classes.

- **MV-Ex1:** After learning the original classes in task 0, all of the new classes are learned in task 1, resulting in two tasks. This exemplifies a real-world application of CLEO, where the model is specifically trained to learn the new classes without a complete retraining of all the classes.
- **MV-Ex2:** This setting is more challenging with a longer sequence of tasks, which can exacerbate forgetting. In each step, one of the ten parent classes is split into its new classes, resulting in a total of eleven tasks.

Mapillary Vistas is challenging due to the large number of classes. As shown in Tab. 4, all methods exhibit suboptimal performance, including joint training which does not suffer from forgetting. The two settings are similar and have

learned the same classes after the final task. They differ in the number of steps, highlighting the effect of a longer sequence of tasks. *MV-Ex2* with a longer sequence significantly impacts the performance of all methods.

6 Limitations and Future Work

In its most general form, CLEO allows for *merging* of two or more known classes as a valid form of evolution. However, we note that the here presented formalism of CLEO only considers the separation of known parent classes. Yet, in some rare cases, it might be required to combine previous information to simplify previous tasks and free representational capacity of a model. This, however, as a version of *intended forgetting* opens up a mostly unexplored field of research in continual learning and is left for future work. Besides, semantic context can be achieved through post-processing by combining predictions of existing and new classes into a single class, preventing interference and avoiding further forgetting.

We further highlight that our baseline solution MoOn requires the information of class relations S_t about the evolution of classes. While we believe that this information is usually available to a sequentially trained model, there might be cases in which explicit knowledge about the ontology is unavailable. In such cases, S_t could be inferred from the existing model's predictions on the current task data. If the model predicts an existing class, the class being learned was previously seen and is split from this parent class. If it predicts the background, the new class is entirely new and unseen. In our experimental evaluation, we found that MoOn and other approaches notably underperform with longer sequences in CLEO, highlighting the need for future attention to long-term learning settings.

7 Conclusion

We introduce a novel framework for addressing the challenge of adapting to evolving classes, termed Continual Learning of Evolving Ontologies (CLEO). The motivation for CLEO stems from a practical scenario wherein pre-existing classes undergo transformations over time, resulting in more specialized classes. CLEO extends the functionality of CIL to not just learn new information but also refine existing information. The applicability of CLEO in the real world is exemplified through the Mapillary Vistas dataset, where an existing set of classes evolves into more detailed classes. Furthermore, we highlight the shortcomings of existing CIL methods and propose MoOn as an initial solution for CLEO. The effectiveness of MoOn for CLEO is demonstrated on seven experimental settings across three datasets, surpassing previous CIL approaches.

Acknowledgments

This work was partially funded by the Federal Ministry of Education and Research Germany under the project DECODE (01IW21001).

References

1. Aljundi, R., Chakravarty, P., Tuytelaars, T.: Expert gate: Lifelong learning with a network of experts. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
2. Asghar, N., Mou, L., Selby, K.A., Pantasdo, K.D., Poupart, P., Jiang, X.: Progressive memory banks for incremental domain adaptation. arXiv (2018)
3. Bang, J., Kim, H., Yoo, Y., Ha, J.W., Choi, J.: Rainbow memory: Continual learning with a memory of diverse samples. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
4. Cermelli, F., Mancini, M., Bulo, S.R., Ricci, E., Caputo, B.: Modeling the background for incremental learning in semantic segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
5. Cha, S., kim, b., Yoo, Y., Moon, T.: Ssul: Semantic segmentation with unknown label for exemplar-based class-incremental learning. In: Advances in Neural Information Processing Systems (2021)
6. Chaudhry, A., Rohrbach, M., Elhoseiny, M., Ajanthan, T., Dokania, P.K., Torr, P.H., Ranzato, M.: On tiny episodic memories in continual learning. arXiv (2019)
7. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv (2017)
8. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2009)
10. Douillard, A., Chen, Y., Dapogny, A., Cord, M.: Plop: Learning without forgetting for continual semantic segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
11. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision (IJCV) (2010)
12. Fernando, C., Banarse, D., Blundell, C., Zwols, Y., Ha, D., Rusu, A.A., Pritzel, A., Wierstra, D.: Pathnet: Evolution channels gradient descent in super neural networks. arXiv (2017)
13. Goswami, D., Schuster, R., van de Weijer, J., Stricker, D.: Attribution-aware weight transfer: A warm-start initialization for class-incremental semantic segmentation. In: Winter Conference on Applications of Computer Vision (2023)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
15. Kemker, R., Kanan, C.: Fearnet: Brain-inspired model for incremental learning. In: International Conference on Learning Representations (2018)
16. 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 (2017)
17. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)

18. Lee, B.H., Jung, O., Choi, J., Chun, S.Y.: Online continual learning on hierarchical label expansion. In: International Conference on Computer Vision (ICCV) (2023)
19. Lesort, T., Lomonaco, V., Stoian, A., Maltoni, D., Filliat, D., Díaz-Rodríguez, N.: Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. Information fusion (2020)
20. Li, Z., Hoiem, D.: Learning without forgetting. Transactions on Pattern Analysis and Machine Intelligence (T-PAMI) (2017)
21. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision (ECCV) (2014)
22. Lin, Z., Pathak, D., Wang, Y.X., Ramanan, D., Kong, S.: Continual learning with evolving class ontologies. In: Conference on Neural Information Processing Systems (NeurIPS) (2022)
23. Lopez-Paz, D., Ranzato, M.: Gradient episodic memory for continual learning. In: Conference on Neural Information Processing Systems (NeurIPS) (2017)
24. Mallya, A., Davis, D., Lazebnik, S.: Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In: European Conference on Computer Vision (ECCV) (2018)
25. Mallya, A., Lazebnik, S.: Packnet: Adding multiple tasks to a single network by iterative pruning. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
26. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: The sequential learning problem. Psychology of Learning and Motivation (1989)
27. Mermillod, M., Bugaiska, A., Bonin, P.: The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects (2013)
28. Michieli, U., Zanuttigh, P.: Incremental learning techniques for semantic segmentation. In: Conference on Computer Vision and Pattern Recognition Workshops (CVPR-W) (2019)
29. Michieli, U., Zanuttigh, P.: Continual semantic segmentation via repulsion-attraction of sparse and disentangled latent representations. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
30. Neuhold, G., Ollmann, T., Rota Bulo, S., Kontschieder, P.: The mapillary vistas dataset for semantic understanding of street scenes. In: International Conference on Computer Vision (ICCV) (2017)
31. Phan, M.H., Ta, T.A., Phung, S.L., Tran-Thanh, L., Bouzerdoum, A.: Class similarity weighted knowledge distillation for continual semantic segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
32. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. arXiv (2016)
33. Shin, H., Lee, J.K., Kim, J., Kim, J.: Continual learning with deep generative replay. In: Conference on Neural Information Processing Systems (NeurIPS) (2017)
34. Sprechmann, P., Jayakumar, S.M., Rae, J.W., Pritzel, A., Badia, A.P., Uria, B., Vinyals, O., Hassabis, D., Pascanu, R., Blundell, C.: Memory-based parameter adaptation. In: International Conference on Learning Representations (ICLR) (2018)
35. Van de Ven, G.M., Siegelmann, H.T., Tolias, A.S.: Brain-inspired replay for continual learning with artificial neural networks. Nature communications (2020)
36. Xiao, J.W., Zhang, C.B., Feng, J., Liu, X., van de Weijer, J., Cheng, M.M.: Endpoints weight fusion for class incremental semantic segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2023)

37. Yang, G., Fini, E., Xu, D., Rota, P., Ding, M., Nabi, M., Alameda-Pineda, X., Ricci, E.: Uncertainty-aware contrastive distillation for incremental semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022)
38. Yang, G., Fini, E., Xu, D., Rota, P., Ding, M., Tang, H., Alameda-Pineda, X., Ricci, E.: Continual attentive fusion for incremental learning in semantic segmentation. IEEE Transactions on Multimedia (2022)
39. Yoon, J., Yang, E., Lee, J., Hwang, S.J.: Lifelong learning with dynamically expandable networks. arXiv (2017)
40. Yu, L., Liu, X., Van de Weijer, J.: Self-training for class-incremental semantic segmentation. IEEE Transactions on Neural Networks and Learning Systems (2022)
41. Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. In: International Conference on Machine Learning (ICML) (2017)
42. Zhang, C.B., Xiao, J.W., Liu, X., Chen, Y.C., Cheng, M.M.: Representation compensation networks for continual semantic segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
43. Zhao, H., Hu, Q., Zhu, P., Wang, Y., Wang, P.: A recursive regularization based feature selection framework for hierarchical classification. Transactions on Knowledge and Data Engineering (2021)