CARNEGIE MELLON UNIVERSITY

Continual Learning in Speech and Audio Applications

Thesis Proposal

by

Muqiao Yang

Thesis proposal submitted in partial fulfillment
for the degree of Doctor of Philosophy

Thesis committee:
Prof. Bhiksha Ramakrishnan
Prof. Shinji Watanabe
Prof. Rita Singh
Dr. Anurag Kumar

May 2023

Abstract

In recent years, the community has witnessed the enormous progress of deep neural network models in matching or even surpassing human performance on a variety of speech and audio tasks, including Automatic Speech Recognition (ASR), Spoken Language Understanding (SLU), Text-to-Speech (TTS), etc. However, their impressive and powerful achievement is predominantly dependent on training with a large set of data defined by a particular and rigid task. In such a paradigm, the model is expected to learn universal knowledge from a static entity of data and stationary environments. In contrast, the real world is inherently ever-changing and non-stationary. New data is often generated and collected every second in a stream format, and novel classes may also emerge from time to time. Without proper adaptation techniques, the knowledge learned in the past might be erased easily when the model is learning subsequent tasks, thus resulting in overall performance degradation. Such a phenomenon is called catastrophic forgetting, which limits the practical use and expansion of many deep neural network models.

In recent years, continual learning has emerged as a new machine learning paradigm that enables artificial intelligence (AI) systems to learn from a continuous stream of data and incrementally improve their performance over time. By adapting to changing environments and user needs, continual learning aims to address the catastrophic forgetting effect, so that the model can gradually extend the knowledge it acquires without drastically forgetting the knowledge that has been learned in the past. Such a property is crucial in practical applications to enable artificial systems to learn from the infinite streams of data of the changing world in a lifelong manner.

This thesis mainly focuses on the underexplored area of how continual learning techniques can be applied in speech and audio tasks. We will introduce the background and formulations of multiple continual learning scenarios, including data-incremental, class-incremental and task-incremental settings. Then we will present how different continual learning categories can be applied to different modules of the modeling pipeline. We propose methods to apply continual learning algorithms in three speech and audio tasks, including automatic speech recognition, acoustic scene classification, and spoken language understanding. We then propose to extend the existing work to the speech enhancement task. We believe that this thesis provides an overall exploration of continual learning scenarios in various speech and audio tasks, and makes an important step towards realizing lifelong learning of speech interfaces.

Contents

Chapter 1

Introduction

In the real world, human perception typically has the ability to continually learn new information and skills based on their prior experience, and apply it to later scenarios. In other words, humans can accumulate new knowledge together with their old knowledge without abruptly forgetting about it. However, such a property is non-trivial for artificial systems and machine learning (ML) models, where the phenomenon of catastrophic forgetting may happen (French, 1999). The recommender system in Netflix or other similar websites would be a simple example for this case. Recommender systems collect user data periodically over time when the users are interacting with the systems. As time elapses, customer interest may change across different categories of movies and shows, e.g., from action to romance. Well-performing recommender systems are expected to adjust themselves according to specific user behaviors so that they can give more accurate recommendations on the movies or shows that the customer may be interested in. During the continual updating process of user data, the system should not care about only the newly collected data, but should have a comprehensive analysis of the interest of the customers by combining it with the history data as well. This is an effortless task for humans, but remains a challenge for machines if no proper adaptation techniques are applied.

Typically, for artificial models like recommender systems, when learning on some new additional data, the knowledge of previously learned content tends to be forgotten if it is not recalled again, especially the knowledge that is learned a very long time ago. On one hand, if the old customer data is recalled every time when new data arises, some issues about computation efficiency, memory saving or even privacy concerns might be problematic. On the other hand, if the system only has access to limited data every time when new data is incoming, instead of training the model with all past data at one time, the generalizability of the whole model might be downgraded.

Continual learning, or *lifelong learning*, aims to circumvent the catastrophic forgetting effect with the goal of continually adapting the model by learning from infinite streams of data (Kirkpatrick et al., 2017). In contrast, *isolated learning* refers to the common paradigm where models are trained on a fixed and rigid set of data, where no other related information and previously learned knowledge are taken into consideration (Chen and Liu, 2018). Human perception differs from isolated learning by retaining past knowledge to find its similar aspects to the new situation, so that they can adapt to the ever-changing environments in the practical world. The inspiration for continual learning is to mimic such a human recognition process for ML models.

The catastrophic forgetting phenomenon refers to the fact that the adaptation of ML models to an unseen data distribution generally leads to a degradation of its ability to capture the characteristics of the old distributions. This trade-off is also called the stability-plasticity dilemma (Mermillod et al., 2013). Assuming a model is continuously witnessing new tasks in chronological order. Memory stability prefers the model to be as stable as possible so that it will not erase the knowledge learned in the past tasks, while learning plasticity expects the model to have a high generalizability so that it could accommodate the new data distribution from the knowledge of the old ones. As a formulation, let's denote the current state of a system as x, and the previous state as x_0. f is the system response. Then, a simple form of the new system response can be expressed as:

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) \tag{1.1}$$

where a smaller $f'(x_0)$ indicates a higher stability, as the model will be relatively resistant to changes brought by the new state. Similarly, a larger $f'(x_0)$ represents a higher plasticity.

To address the stability-plasticity dilemma, most existing research in continual learning of machine learning models has focused on tasks such as computer vision and reinforcement learning (Chen and Liu, 2018; Aljundi et al., 2017; Shin et al., 2017). However, limited research has been done on speech and audio applications and it still remains an underexplored area. As one of the most widely used and informative signals that play a crucial role in human communication, a continual learning manner of speech and audio models would take full advantage of the plentiful amount of data collected in daily life, including meetings, talks, conversations, environmental sounds, etc. Meanwhile, with the increasing use of voice assistants and speech-enabled devices, it has also become increasingly necessary to investigate continual learning in speech and audio applications so that the interaction system can effectively satisfy the personalized demand of users. Moreover, continual learning can help speech and audio models adapt to new tasks or environments as they arise, without requiring a large amount of new training data or retraining from scratch. This can help save a considerable amount of time and resources while enabling the development of more flexible and adaptive speech and audio technologies.

1.1 Thesis Overview

This thesis begins by introducing the general definition and objective of continual learning in Chapter 1. In Chapter 2, we will first define the formulation for continual learning problems, and then introduce various types of continual learning scenarios under different settings, including data-incremental, class-incremental, and task-incremental learning. After that, we will also present the family of continual learning methods, including rehearsal, architecture, and regularization methods. Lastly, we will summarize the main challenges of continual learning in speech and audio tasks. Different scenarios in various speech and audio tasks enable us to choose different categories of continual learning methods to solve the corresponding problems. From Chapter 3 to Chapter 5, we demonstrate three continual learning applications in speech and audio tasks, including Automatic Speech Recognition (ASR), Acoustic Scene Classification (ASC), and Spoken Language Understanding (SLU).

First, we define a data-incremental learning setting in end-to-end ASR, where continuous streams of speech become available in sequential order and there are no explicitly defined task boundaries, and the small sets of new unseen data are incoming in an online manner. This setting is challenging because we only have limited access to the training data each time. In this case, the ASR model is intrinsically trained on the same task but different domains of speech data, where the domain shift can refer to speech contents with different languages, accents, speakers, and topics, etc, therefore it is called data-incremental learning. To address the incremental data under this setting, one common approach is to combine both the past and current data, and retrain the model with the accumulated training set. However, this procedure takes significant time and computational resources, and becomes increasingly unrealistic when the amount of data is continuously growing over time. Instead, we lead an investigation into rehearsal continual learning methods, which address the trade-off between computation efficiency and online performance in data-incremental end-to-end ASR by saving an episodic memory for the past data.

In the next part, we will focus on the class-incremental learning scenario in ASC. The challenge of class-incremental learning is mainly from the unseen classes emerging streamingly. The learner has to incorporate knowledge of new classes incrementally and build a holistic classifier among all seen classes. Therefore, we apply ASC as a task to demonstrate the effect of class-incremental continual learning in the audio domain. To recognize unseen acoustic scenes, humans can associate new sound categories with specific acoustic scenes based on their extensive life experiences. In our ASC modeling, we also base on rehearsal continual learning approaches, which save a small subset of past data into a memory buffer for each of the acoustic scene classes. When new classes of acoustic scenes arise in the subsequent training process, the memory buffer is used to replay samples. In the class-incremental learning scenario of ASC, by encouraging the encoder and the classifier to learn task-agnostic and task-specific knowledge, we design a mutual information-based sampling strategy to select the most representative and informative samples from the past.

Finally, we will formulate the class-incremental learning setting in SLU by focusing on two relevant tasks, intent classification and slot filling. As a class-incremental learning scenario, new types of intents and entities are emerging over time in SLU. Different from ASC, the lexical richness enables us to treat the problems of intent classification and slot filling as a sequence-to-sequence (seq2seq) problem, where the intents and entities are generated along with the transcriptions. In the seq2seq architecture, the decoder will also be affected by catastrophic forgetting as the encoder. Under this setting, we combine both rehearsal and regularization continual learning methods via knowledge distillation (KD) to moderate the forgetting effect at both the encoder and decoder levels. By regarding the previous model as a teacher and the current model as a student, we apply KD to transfer knowledge from the teacher network to the student network, thus achieving the objective of continual learning. We will propose three KD techniques at different levels of the architecture of the SLU model, including audio-KD, token-KD, and sequence-KD, and conduct a study to demonstrate that adding more KDs at different levels will increasingly boost the overall SLU performance.

In summary, this thesis provides a comprehensive study of the application of continual learning approaches in various speech and audio tasks. We design different continual learning scenarios for corresponding tasks, including data-incremental and class-incremental learning. We then develop different categories of continual learning methods under specific scenarios. By demonstrating that appropriate continual learning methods can alleviate the catastrophic forgetting effect of ML models, we make a step forward in the direction of addressing the stability-plasticity dilemma in speech and audio applications, thus increasing their practicability in the real world.

1.2 Thesis Organization and Contributions

We describe the organization and contributions of the thesis as follows.

Organization:

- **1. Overview** We begin the thesis by providing an overview of the necessity and motivation of continual learning, and then introduce its general definition and a thesis overview in Chapter 1
- **2. Background** In Chapter 2, we will illustrate basic background knowledge of continual learning, including defining its formulation, introducing the family of various continual learning scenarios and methods, and explaining the challenges of continual learning in speech and audio applications.
- **3. Completed Work** Chapter 3 to Chapter 5 are organized into three parts, which introduce how continual learning can be applied in three speech and audio tasks with different strategies.
- **4. Proposed Work** Based on the completed work, this chapter proposes to investigate how to select different objectives of rehearsal methods for rehearsal samples and ongoing samples respectively. Furthermore, we propose to explore the task-incremental learning

setting aside from the data-incremental and class-incremental ones in the completed work.

Contributions:

- We define a novel data-incremental learning formulation in end-to-end ASR, where continuous streams of data become available to the model. The task boundary is agnostic to the model under this setting, which differs from traditional continual learning scenarios where clear boundaries are typically delineated. We then propose an online rehearsal method and a selective sampling strategy that is calculated on the gradients of rehearsal samples. Our proposed method is demonstrated to improve the trade-off between the efficiency and effectiveness of continually learned ASR models.
- We propose a novel rehearsal continual learning approach for the class-incremental setting of ASC from the perspective of mutual information optimization. Our approach optimizes the mutual information between different levels of the model, enabling it to learn both task-agnostic and task-specific knowledge by selecting representative and informative samples. This optimization encourages the model to both utilize past knowledge effectively and learn from new information. By combining a mutual information-based objective with a memory selection mechanism, we demonstrate that our performance exceeds that of existing methods on multiple continual learning evaluation metrics.
- We illustrate a class-incremental learning setting of SLU, focusing on intent classification and slot filling with an increasing number of intents and entities. By formulating it as a seq2seq problem, we propose three knowledge distillation approaches at both the encoder and decoder levels of the transformer architecture, combined with rehearsal approaches. We show that a combination of these knowledge distillation techniques could effectively mitigate the forgetting effect of the SLU model across different domain scenarios.

Chapter 2

Technical Background

In this chapter, we will present background information about the background concepts related to continual learning. We will first introduce the formulation of continual learning algorithms, then present the family of continual learning scenarios and methods, and summarize the main challenges that its application faces in speech and audio tasks in the end.

2.1 Formulations of Continual Learning

We begin this chapter by giving a formulation of continual learning first, and we will follow the notations in this section throughout the thesis. As presented in Chapter 1, continual learning is a type of machine learning paradigm that aims to enable an AI system to learn from a continuous stream of data or tasks, with the objective of solving future learning without forgetting the knowledge that it has already acquired. Therefore, a general formulation for continual learning involves the use of a sequence of tasks or domains, where each task or domain is associated with a set of data points. Figure 2.1 illustrates the idea that continual learning is to solve the trade-off in the stability-plasticity dilemma (Mermillod et al., 2013) introduced in Chapter 1. Along the time axis t, the model might witness an increasing number of tasks in chronological order, where the data distributions among tasks could be dynamic and changing. The red arrow denotes the memory stability, which has a tendency to maintain more about knowledge in the former tasks. In contrast, the blue arrow means the learning plasticity, which pushes the model to quickly adapt the new patterns in the future tasks. Between stability and plasticity, continual learning aims to balance the trade-off with using the data from the current task.

Let us consider a sequence of T tasks or domains, and the data sets associated with each task are denoted by $\{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_T\}$. Let $\mathcal{X}$ be the input space, $\mathcal{Y}$ be the output space, and f_θ be the model with parameters θ. For each specific task i, we can denote the available data $X_{\mathcal{D}_i}$ and the corresponding ground-truth label set as $Y_{\mathcal{D}_i}$. The goal of continual learning is to learn the model parameters θ such that the model can perform well on all T tasks in the sequence

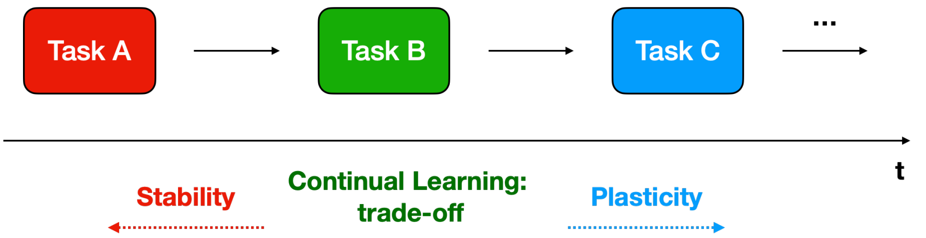


FIGURE 2.1: An example of the stability-plasticity dilemma that continual learning aims to address, where the model witnesses task A, B, and C in chronological order.

in the end, while preventing catastrophic forgetting of previously learned tasks. This can be formulated as the following optimization problem:

$$\text{minimize}_\theta \sum_{i=1}^{T} \ell(f_\theta(X_{\mathcal{D}_i}), Y_{\mathcal{D}_i}) \tag{2.1}$$

subject to:

- The model should perform well on the current task: $\ell(f_\theta(X_{\mathcal{D}_i}), Y_{\mathcal{D}_i}) \leq \epsilon$, where ϵ is a small threshold.
- The model should not forget the knowledge of previously learned tasks: For each $j < i$, $\ell(f_\theta(X_{\mathcal{D}_j}), Y_{\mathcal{D}_j}) \leq \epsilon$.

Here, ℓ is the loss function, which measures the discrepancy between the predicted output of the model and the true output. The first constraint corresponds to the current task, while the second one corresponds to the previously learned tasks. The constraints ensure that the model does not forget the previously learned tasks, while still being able to learn the current task.

2.2 Family of Continual Learning Scenarios

Given the complexity of various real-world applications, continual learning may exhibit very different scenarios across different settings. In this thesis, we mainly categorize the family of continual learning scenarios in speech and audio tasks into three types: data-incremental, class-incremental, and task-incremental learning.

Data-incremental learning typically refers to the continual learning setting where there are no assumptions of explicit task boundaries and the model is continuously learning on incremental data streams (De Lange and Tuytelaars, 2021). Sequence modeling classification tasks like automatic speech recognition are good examples of data-incremental learning settings, where the model is trained to solve the same or similar problem defined on a label space of an entire shared vocabulary, while continuous streams of data may become available over time. Implicit task boundaries indicate that the model has no information about which new stream

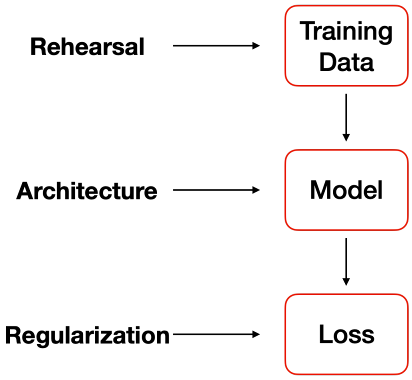


FIGURE 2.2: An overview of the categories of continual learning methods in a general ML pipeline.

of data it is currently observing. Therefore, under this setting, it needs to either process the observed data in an online fashion, or infer an implicit task identifier from statistics in the current stream.

Class-incremental learning means that the model learns from evolutive streams of data where new unseen classes are continuously appearing (Zhou et al., 2023). Many utterance modeling classification tasks like acoustic scene classification and intent classification can be demonstrated as class-incremental learning, where an utterance is mapped to one single class label. In this case, each individual task may contain a different label space that is a subset of the entire label space $\mathcal{Y}$. The ultimate goal of class-incremental learning is to build a holistic classifier that can handle all of the seen classes in the past.

Task-incremental learning in our context usually has a more general definition. There are many similar and related speech and audio tasks in applications, like intent classification and slot filling. The task incremental setting could refer to the case where the model is continually being added new capabilities, and expanding itself to be able to deal with increasingly more new tasks by transferring the knowledge learned from other tasks (Hossain et al., 2022). Another example is regression tasks like speech enhancement. Unlike data-incremental and class-incremental learning, the model does not explicitly map the data space to a label space, but improves its overall robustness when the model continually adapts to changes in the underlying data distribution, like unseen noise types in speech enhancement.

2.3 Family of Continual Learning Methods

In order to tackle various continual learning scenarios in real applications, multiple types of continual learning methods have been proposed. According to the different modules in the general ML pipeline that they operate on, continual learning methods can be mainly divided into three categories: rehearsal, architecture, and regularization-based methods. An overview of the family of continual learning methods is provided in Figure 2.2.

Rehearsal approaches typically retain examples from past data to a small set of memory buffer and replay these examples when the model is learning on new tasks, to alleviate the issue of catastrophic forgetting (Lopez-Paz and Ranzato, 2017; Rolnick et al., 2018; Chaudhry et al., 2018). Some methods are restoring the raw samples or their representations, while other methods involve applying generative models to generate pseudo data rather than replaying in memory examples (Shin et al., 2017; Lavda et al., 2018). Compared to other methods, rehearsal methods usually have a better performance with the cost of saving a rehearsal memory $\mathcal{M}$. At each individual task t, samples from the rehearsal memory from the previous task $\mathcal{M}_{t-1}$ and the current data stream $\mathcal{D}_t$ are combined to form the training data to be used in the current task t. The rehearsal samples are expected to alleviate the catastrophic forgetting effect of the model by recalling samples from the past tasks, so it improves the training data phase in the continual learning pipeline.

Architecture-based approaches are expanding the model parameters in the ML pipeline by isolating parameters for different tasks (Rusu et al., 2016; Aljundi et al., 2017; Rebuffi et al., 2017a). The objective is to make sure that there is limited interference between different subsets of modules within a model. Generally, the model parameters for each specific task t are composed of two parts: task-specific parameters and task-sharing parameters. The task-sharing parameters are shared among all tasks and continually updated, while the task-specific parameters are updated only when the model is learning this individual task and frozen when learning other tasks. Architectural methods typically would require a task oracle, which activates corresponding masks or task branches during the inference.

Regularization methods impose an additional regularization term when new tasks are encountered in order to adjust parameter importance, therefore it is improving the continual learning from the loss level (Kirkpatrick et al., 2017; Zenke et al., 2017; Aljundi et al., 2018; Li and Hoiem, 2017). The goal is to regularize the change of the model update, so that the knowledge stored in the learned model can be consolidated instead of forgotten. Typically, the importance of all the model parameters is estimated to ensure feasibility. Denoting the importance matrix for the model parameters as W_t for task t, the updated loss function can be represented as:

$$\ell(\theta_t) \leftarrow \ell(\theta_t) + \frac{\lambda}{2}(\theta_t - \theta_{t-1})^\top W_{t-1}(\theta_t - \theta_{t-1}) \tag{2.2}$$

where λ is the regularization coefficient to control the weight of regularization.

2.4 Challenges of Continual Learning in Speech and Audio Applications

As a learning paradigm focusing on improving the model performance across different domains, continual learning may face challenges from multiple facets. We scaffold the problem of continual learning in speech and audio applications into three challenges: vague task boundaries, multiple dimensions of variability, and resource constraints.

First, the task boundaries in audio and speech could be vague, and the tasks might not be delineated by clear boundaries. For example, an ASR model trained on Wall Street Journal (WSJ) (Paul and Baker, 1992) may not generalize well on LibriSpeech (Panayotov et al., 2015). Their label spaces may share commonalities, which is unlike images that usually have more disjoint label spaces, e.g. Split-MNIST (Van de Ven and Tolias, 2019) that is splitting 10 digits as 5 different tasks, while the dictionaries of different ASR corpus may share some common words or phones.

Second, speech and audio data may inherently involve high variability across multiple dimensions, e.g. new speakers, new acoustic environments, new accents, new languages, etc. This variability can make it difficult for models to generalize well across different speech and audio inputs, particularly when the amount of training data is limited.

Finally, speech and audio signals are often high-dimensional and require large amounts of data and computation resources to train a well-performing model. This may pose challenges for continual learning, especially when the data distribution changes over time or when the model needs to be adapted to new environments.

Chapter 3

Continual Learning - Automatic Speech Recognition

In this chapter, we will focus on continual learning methods in automatic speech recognition (ASR), which is basically a sequential modeling classification task to transcribe spoken language into written text. We will first present a data-incremental setting for ASR, and then propose a rehearsal method to address the trade-off between the computation resources and performance in the continual learning problem in ASR.

3.1 Background and Objective

In our previous work, we have investigated the online continual learning setting in end-to-end ASR (Yang et al., 2022c). The problem setting is as follows. First, we assume we have an initial model (M_0) that has been trained on a given dataset ($\mathcal{D}_0$). This model is treated as a seed model on which a sequence of continual learning updates is applied. Second, we have a set of labeled datasets $\{\mathcal{D}_i\}_{i=1}^{T}$ which become available sequentially over time for model training. T is the total number of subsets and the size of $\mathcal{D}_i$ is in general significantly smaller than $\mathcal{D}_0$.

Note that $\mathcal{D}_i$ could be any subset randomly split from a larger dataset, which means that different $\mathcal{D}_i$s might be different splits from the same dataset, and they might also come from different datasets. Retraining the ASR model from scratch on $\{\mathcal{D}_i\}_{i=0}^{t}$ each time a new dataset $\mathcal{D}_t$ becomes available requires significant computational cost, especially when the amount of data in $\mathcal{D}_0$ is large and the amount of data in each $\mathcal{D}_i$ is small.

Therefore, for online continual learning of ASR models, at each time-step t, a well-performing method would:

- Obtain better (or not degrade) accuracy compared to the previous model (θ_{t-1});

- Obtain better accuracy compared to continually training the previous model (θ_{t-1}) on only the new data $\mathcal{D}_n$;
- Obtain comparable accuracy to a model trained from scratch on all the data available at this time $\{\mathcal{D}_i\}_{i=0}^n$;
- Require significantly less computational cost compared to training a model from scratch on $\{\mathcal{D}_i\}_{i=0}^n$.

3.2 Methodology

The method is basically motivated by Gradient Episodic Memory (GEM) (Lopez-Paz and Ranzato, 2017). GEM is a replay-based continual learning method, which stores samples from past data as its memory. When the model encounters the data of a new task, it requires the minimization of the L2 distance between the gradients of the new data and old data, i.e.,

$$\begin{aligned} &\min_{\boldsymbol{\omega}} \frac{1}{2}||\boldsymbol{\omega} - \boldsymbol{g}||_2^2 \\ &\text{s.t. } \langle \boldsymbol{\omega}, \tilde{\boldsymbol{g}}_i \rangle \geq 0, \forall i \in (0, \ldots, t-1). \end{aligned} \tag{3.1}$$

where $\boldsymbol{g}, \tilde{\boldsymbol{g}}_i \in \mathbb{R}^{|\theta|}$ and $|\theta|$ is the number of parameters of the model. t is the number of previous tasks and $(0, \ldots, t-1)$ represents the sequence of tasks. $\boldsymbol{g}$ is the gradient over the new data in the current task and $\tilde{\boldsymbol{g}}_i$ is the gradient for the sampled data from ith task in the past, and it assumes that the positive inner product between the gradients will prevent catastrophic forgetting. $\boldsymbol{\omega} \in \mathbb{R}^{|\theta|}$ is the target gradient that we want to solve. This quadratic programming problem (3.1) is then transformed into its dual form,

$$\begin{aligned} &\min_{\boldsymbol{v}} \frac{1}{2}\boldsymbol{v}^\top \boldsymbol{G}\boldsymbol{G}^\top \boldsymbol{v} - \boldsymbol{g}^\top \boldsymbol{G}^\top \boldsymbol{v} \\ &\text{s.t. } \boldsymbol{v} \geq \mathbf{0}. \end{aligned} \tag{3.2}$$

where $\boldsymbol{G} = (\tilde{\boldsymbol{g}}_1, ..., \tilde{\boldsymbol{g}}_{t-1})$, and $\boldsymbol{v} \in \mathbb{R}^{|\theta|}$. The solution of the primal form could be recovered by $\boldsymbol{\omega} = \boldsymbol{G}^\top \boldsymbol{v} + \boldsymbol{g}$. After that, $\boldsymbol{\omega}$ will be used as the final gradient to update the parameters in the network. The derivation is as this:

$$\begin{aligned} L(\boldsymbol{\omega}, \boldsymbol{v}) &= \frac{1}{2}\boldsymbol{\omega}^\top \boldsymbol{\omega} - \boldsymbol{g}^\top \boldsymbol{\omega} + \boldsymbol{v}(\boldsymbol{G}\boldsymbol{\omega}) \\ \boldsymbol{\omega}^* &= -(\boldsymbol{v}\boldsymbol{G}^\top - \boldsymbol{g}) \end{aligned}$$

Therefore, the dual problem becomes

$$
\begin{aligned}
&\max_{\boldsymbol{v}} \frac{1}{2}(\boldsymbol{v}\boldsymbol{G} - \boldsymbol{g}^\top)(\boldsymbol{v}\boldsymbol{G}^\top - \boldsymbol{g}) + \boldsymbol{g}^\top(\boldsymbol{v}\boldsymbol{G}^\top - \boldsymbol{g}) - \boldsymbol{v}\boldsymbol{G}(\boldsymbol{v}\boldsymbol{G}^\top - \boldsymbol{g}) \\
&= \max_{\boldsymbol{v}} \frac{1}{2}\boldsymbol{v}^2\boldsymbol{G}\boldsymbol{G}^\top - \frac{1}{2}\boldsymbol{g}^\top\boldsymbol{g} - \boldsymbol{v}^2\boldsymbol{G}\boldsymbol{G}^\top + \boldsymbol{g}^\top\boldsymbol{G}^\top\boldsymbol{v} \\
&= \min_{\boldsymbol{v}} \frac{1}{2}\boldsymbol{v}^2\boldsymbol{G}\boldsymbol{G}^\top - \boldsymbol{g}^\top\boldsymbol{G}^\top\boldsymbol{v} \\
&s.t.\boldsymbol{v} \geq 0
\end{aligned}
$$

Since we do not define clear task boundaries in our online continual learning scenario, we cannot directly follow the original training protocol directly which requires calculating gradients for all previous tasks and samples from each of the previous tasks. Instead, we perform random sampling and selective sampling respectively to sample from the memory. For the selective strategy, we assign a probability score $c_i = 1 - \frac{\langle \boldsymbol{\omega}, \tilde{\boldsymbol{g}}_i \rangle}{||\boldsymbol{\omega}|| \cdot ||\tilde{\boldsymbol{g}}_i||}$ to the ith sample, and use the weighted scores $\{c_i\}_{i=1}^{|\mathcal{M}|}$ to sample from the memory buffer $\mathcal{M}$. Such a selective strategy would select samples with less similarity to other samples in the memory more frequently, inspired by the fact that unexpected events play an important role in replayed experiences from the neuroscience perspective (Isele and Cosgun, 2018) and the gradient-based sample selection (Aljundi et al., 2019). Since we have a higher probability to select the samples with lower cosine similarity scores, we are relaxing the constraints in the original primal form of GEM. In other words, instead of the strict constraint that the direction of the gradients of old and new data must be similar, we release the constraint and develop the quadratic form in eq. 3.1 so that it can be rewritten as

$$
\begin{aligned}
&\min_{\boldsymbol{\omega}} \frac{1}{2}||\boldsymbol{\omega} - \boldsymbol{g}||_2^2 \\
&\text{s.t. } \langle \boldsymbol{\omega}, \tilde{\boldsymbol{\omega}} \rangle \geq -\xi, \xi \geq 0.
\end{aligned}
\tag{3.3}
$$

where $\tilde{\boldsymbol{\omega}} \in \mathbb{R}^{|\theta|}$ is the gradient of the selected memory and ξ is the slack variable. The basic idea is similar to the soft gradient constraint in (Chen and Lin, 2019). Note that since we do not assume the explicit definition of task boundaries in our case, the dual form of the quadratic programming can be written as

$$
\begin{aligned}
&\min_{v} \frac{1}{2}v^2\tilde{\boldsymbol{\omega}}^\top\tilde{\boldsymbol{\omega}} - \boldsymbol{g}^\top\tilde{\boldsymbol{\omega}}v \\
&\text{s.t. } v \geq 0.
\end{aligned}
\tag{3.4}
$$

where v is a scalar, which is different from the original dual form. Then the updated gradient can be calculated by $\boldsymbol{\omega} = v\tilde{\boldsymbol{\omega}} + \boldsymbol{g}$ to optimize the current model.

Characteristics	v1	v2	v3
Duration of speech	118h	207h	452h
No. of unique speakers	666	1,242	2,028
No. of words	1.7M	2.6M	4.9M

Table 3.1: The overall statistics of TED-LIUM corpus v1 to v3.

3.3 Experimental Setup

This work used TED-LIUM 1, 2 and 3 (Rousseau et al., 2012, 2014; Hernandez et al., 2018) as the datasets for the experimental evaluation. TED-LIUM is a series of datasets that consist of audios and transcripts extracted from the official TED talk website[1]. Such property benefits us by providing audios from a variety of speakers and their captions with a wide range of topics. To be more specific, training across different topics of talks will help us demonstrate and evaluate the effectiveness of our continual learning settings, which is a more realistic setting in the real world. Meanwhile, in the series of TED-LIUM corpus, each succeeding version is a comprehensive expansion and evolution of the previous one. These are the important reasons why we choose TED-LIUM as the target dataset, since it somewhat imitates the growing data over time in real-world scenarios without explicitly defined boundaries. For simplicity, we name the difference set between TED-LIUM 1 and 2 as TED-LIUM 2-1, and similar for TED-LIUM 3-2. Detailed statistics are shown in Table 3.1.

The overall comparison between the statistics of TED-LIUM 1, 2 and 3 is summarized in Table 3.1. From the table, we can observe that TED-LIUM 2 and 3 expanded the original corpus by adding more acoustic and textual contents. In our experiments, we will follow the setting that the model is first pretrained on TED-LIUM 1, and then the continual learning is performed on TED-LIUM 2-1 and 3-2 with different metrics successively.

As the preprocessing step, we use a three-fold speed perturbation with factors 0.9, 1.0, and 1.1. The tokenization type is Byte-Pair Encoding (BPE), and we remove all the existing unknown tokens in the raw TED-LIUM corpus. All the results provided are performed with transformers (Karita et al., 2019) and do not use language models in the decoding process for simplicity.

3.4 Experimental Results

First, we use TED-LIUM 1 as the baseline dataset $\mathcal{D}_0$, as introduced in Section 3.2, to train for 50 epochs. The final WER achieves the state-of-the-art performance of 11.0% on TED-LIUM 1. After that, we randomly split TED-LIUM 2-1 and 3-2 evenly into a sequence of subsets $\{\mathcal{D}_i\}_{i=1}^{N}$, and continuously train the model on the N splits in order, with 10 epochs for each split. WER and the execution time will be recorded at the end of each training step, and the results are shown in Fig. 3.2. Note that the provided validation dataset is identical for all experiments

[1] https://www.ted.com/talks

Method	WER (%) after TED-LIUM 2-1	WER (%) after TED-LIUM 3-2
All Data	9.0	8.1
New Data	10.4	9.8
EWC (Kirkpatrick et al., 2017)	9.9	9.5
MAS (Aljundi et al., 2018)	9.8	9.5
O-GEM ($N = 2$)	9.4	8.5
O-GEM ($N = 10$)	9.5	8.6

Table 3.2: WER results for different methods in the online continual learning setting.

related to TED-LIUM 1, 2 and 3, so that we could compare the results of different stages fairly during the continual learning process.

The comparison of the results of our method, the online GEM (O-GEM) on TED-LIUM corpus to other continual learning algorithms, including Elastic Weight Consolidation (EWC, (Kirkpatrick et al., 2017)) and Memory Aware Synapses (MAS, (Aljundi et al., 2018)) is also provided. Basically, EWC is regularizing the parameter importance based on the gradients of the loss function, while MAS controls the parameter importance from the gradients of the learned network output function. We show the comparison between the word error rate of the online continual learning with equal number of epochs for different methods in Table 3.2. The *All Data* method is trained from scratch with all previous data each time when the model receives a new subset of data $\mathcal{D}_n$, which is our upper bound performance. In contrast, O-GEM and the *New Data* method are only trained on the specific new set $\mathcal{D}_n$ each time, and O-GEM benefits from the extracted samples of the episodic memory. We will follow the names in all the discussions later. From the figure, we can conclude that EWC and MAS could decrease the WER compared to training with only new data, while O-GEM obtains a much better performance and is closer to the WER of training with all data.

We compare the experimental results of three methods in Fig 3.1 and 3.2: the O-GEM, *All Data* and *New Data*. All three methods are training on a sequence of subsets $\{\mathcal{D}_i\}_{i=1}^N$, as introduced in Section 3.2, from TED-LIUM 2-1 and TED-LIUM 3-2. Figure 3.1 shows the learning curve of validation accuracy of the three methods. The training starts from epoch 50 which is the number of training epochs for the baseline model. In the *New Data* scenario, we load the previous model trained on $\mathcal{D}_{n-1}$ and continually train the model on $\mathcal{D}_n$ with the same training strategy as the baseline model. From Figure 3.1, we can observe that the validation accuracy of *New Data* goes up sharply at the start of each 10 epochs while goes down during the overall process of learning on the single set of data, which indicates that the model will learn new knowledge as soon as they get access to new data, but forget what it has already learned in the past quickly. Meanwhile, the performance of *New Data* is stochastic across the 10 splits because its performance is largely dependent on the content of each specific new data set. The red curve is the best performing one which increases constantly, as it is retraining from scratch with both old and new data each time whenever it absorbs unseen data. Without the

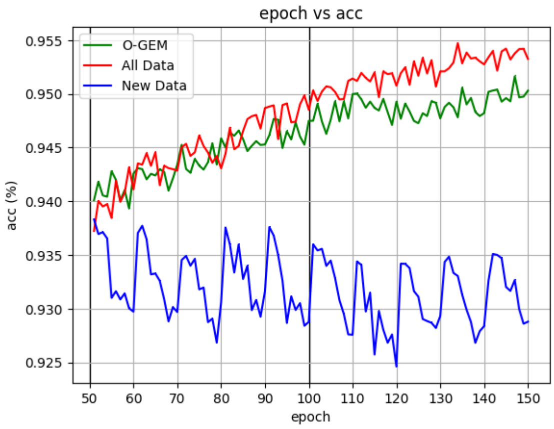


FIGURE 3.1: Validation accuracy (%) vs. epochs for different methods where $N = 10$.

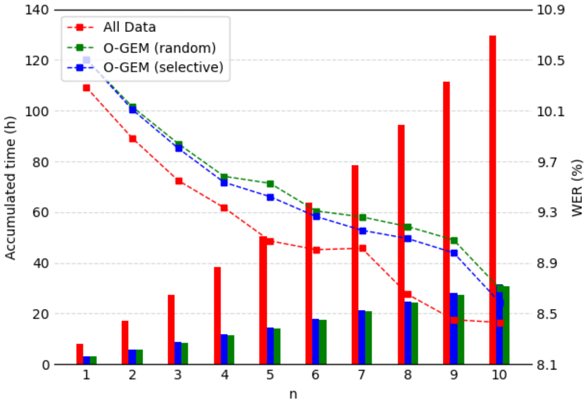


FIGURE 3.2: Word Error Rate (WER, %) & computation time (hours) vs. $\{D_n\}_{n=1}^{N}$ for different methods where $N = 10$.

time-consuming retraining, however, O-GEM can achieve competitive validation accuracy in comparison to the *All Data* case, while maintaining its increasing accuracy rather than causing catastrophic forgetting in the *New Data* method.

Figure 3.2 compares the WER results of selective O-GEM and *All Data*. Each square marker represents the WER when the continual learning on one subset $\mathcal{D}_n$ is finished. The *All Data* training requires the longest execution time, as it requires to be retrained every time when a new $\mathcal{D}_n$ is incoming. In contrast, our method could obtain a decreasing WER when it accesses more data over time, and have a close WER to the *All Data* scenario (8.6% vs. 8.4%) with more than four times less computation time for both random sampling and selective sampling, and with similar computation time, selective sampling can have a further improvement of WER compared to random sampling. Note that the computation time of O-GEM and *New Data* training is proportional to the amount of the new data only, while the training time of *All Data* will increasingly scale up since the size of old data is growing. In other words, as the model receives more information, it will be increasingly more impractical for retraining with all data since the ratio of old data size to new data size is becoming higher over time and the computation cost of retraining is more expensive compared to training on only new data. We demonstrate the experimental results with different numbers of splits N in Table 3.3, where we can observe that with more splits of new data, the ratio of computation time for O-GEM to the estimated computation time for training with all data is becoming smaller. The reason is that *All Data* would have to retrain the model from scratch for every $\mathcal{D}_n$, while our method could save more computation time, which is closer to the real-world problem setting especially when N is large.

N	Avg. hours per split	Model update time diff. vs. *All Data*	Final WER (%)
0	-	-	11.0
2	167.0	79.05%	8.5
10	33.4	23.80%	8.6
20	17.3	13.04%	8.9
100	3.3	2.86%	9.2
200	1.7	1.92%	9.5

Table 3.3: Results for the O-GEM experiments with updates of different numbers of splits N. The time for *All Data* is estimated when $N > 10$ since it is impractical. Note that $N = 0$ refers to the starting point of the online continual learning, and $N = 2$ means no splitting is performed on TED-LIUM 2-1 & 3-2.

3.5 Summary

This work is based on a novel experimental setting for online continual learning in automatic speech recognition, focusing on the case where additional training data for the same speech recognition task becomes available incrementally over time. We demonstrate that by performing incremental model updates with an online Gradient Episodic Memory (GEM) method and a selective sampling strategy, we can achieve competitive performance to retraining an ASR model from scratch each time the model has access to new data, while requiring significantly lower computation cost. We also demonstrate that our online GEM outperforms other continual learning methods under the same setting.

Chapter 4

Continual Learning - Acoustic Scene Classification

In this chapter, we will investigate how continual learning works in acoustic scene classification (ASC), which is an utterance modeling classification task to recognize the acoustic scene of one segment of utterance. Based on the class-incremental setting, we will propose a novel rehearsal method to continually learn on the unseen acoustic scene classes, where the sampling strategy of the rehearsal memory is calculated on mutual information to select the most informative and representative samples.

4.1 Problem Statement

We follow the setting of class-incremental learning, where new classes of acoustic scenes keep appearing in continuous streams of data. Compared to another category of continual learning, i.e., task-incremental learning, class-incremental learning does not have access to task identities during inference time. Therefore, its objective is to build a holistic classifier among all of the seen classes by making use of the label information only.

Typically, a task is defined by a set of train and test data that follows a similar distribution, and in practice, it usually refers to a new set of data that contains data in different classes. Consider we have the data streams $\mathcal{X} = \{X_t\}_{t=1}^{T}$ and its corresponding labels $\{Y_t\}_{t=1}^{T}$, where X_t indicates the data at task t and T is the total number of tasks. For the modeling, we use an architecture of a feature extractor Φ and a classifier Θ, where the output of the feature extractor is $\Phi(X) = Z$, which is the feature representation of X and also the input of the classifier such that $\Theta(Z) = Y$.

At the end of each task t, we inject samples of the input X_t into a memory buffer with a fixed size M, and the memory will be used to select samples for the purpose of replay when the model is learning on subsequent tasks. We will demonstrate the comparison between different sample selection strategies in Section 4.6.

4.2 Methodology

Our mutual information optimization relies on the comparisons between different augmented representations of acoustic scenes, which are also called pseudo-labeled samples. In MI estimation, the augmentations of the same input will be regarded as positive pairs, while those of different inputs are taken as negative pairs, so that the dependency between positive pairs will be maximized and that between negative pairs will be minimized (Oord et al., 2018). Following the notations in Section 4.1, we will denote Z as the feature representation of the original input X, and Z' as the encoded feature of an augmentation of input X'.

In this work, we simulate the pseudo-labeled samples through different augmentation methods. More specifically, we choose to add Gaussian noise, apply band-stop filtering to the audio input, or invert the audio along the time axis to perform multiple types of augmentations.

By deploying an arbitrary model architecture of a feature extractor followed by a classifier, we show that our mutual information optimization can be applied to both modules of the model. We would like the feature extractor to learn task-agnostic knowledge to produce effective latent representations of the input audio, while the classifier to learn task-specific knowledge to map the learned representations to specific acoustic scene classes.

Feature extractor part: To let the feature extractor learn task-agnostic knowledge, we need to guarantee that the encoded representations preserve sufficient information from the original inputs regardless of their classes. Therefore, we will want to maximize the MI between X and Z, so that Z will preserve generic information from X in the modeling of the feature extractor. To better estimate the MI, we have augmented acoustic scenes X' and its corresponding latent representation Z' described in Section 4.2. Assuming that Z and Z' are conditional independent given data X, we have

$$\begin{aligned} I(X;Z) &= H(Z) - H(Z|X) \\ &= H(Z) - H(Z|X, Z') \\ &\geq H(Z) - H(Z|Z') \\ &= I(Z;Z') \end{aligned} \tag{4.1}$$

where $I(\cdot,\cdot)$ indicates the mutual information between two variables and $H(\cdot)$ denotes the Shannon entropy or conditional entropy given another random variable. We use the property of conditional independence from Line 1 to Line 2, and the definition of conditional entropy from Line 2 to Line 3. Therefore, we have $I(X;Z) \geq I(Z;Z')$ from Eq. 4.1, which means that maximizing the MI between Z and Z' is equivalent to maximizing the lower bound of the MI between input X and the encoded features Z.

Taking a further step, as from (Oord et al., 2018), the mutual information between Z and its augmentation Z' can be estimated through the InfoNCE (NCE stands for noise-contrastive estimation) loss as the lower bound, i.e.,

$$\begin{aligned} I(Z, Z') &\geq \log N + \mathcal{L}_{\text{NCE}}(Z, Z') \\ &= \log N + \frac{1}{N} \sum_{i=1}^{N} \log \frac{(f(z_i, z_i')/\tau)}{\sum_{j=1}^{N} (f(z_i, z_j')/\tau)} \end{aligned} \tag{4.2}$$

where z_i is the representation of an individual sample in the batch. z_i and z_i' are regarded as positive sample pairs since they originated from the same sample x_i, and all other z_j's in the batch where $i \neq j$ are regarded as negative pairs. f denotes the exponential of similarity function and τ is the temperature. N is the batch size of the samples, and when it becomes larger, $I(Z, Z')$ will close its gap to the lower bound. Therefore, $I(Z, Z')$ is lower bounded by the InfoNCE loss and we will use the second term on the right-hand side as the approximation of MI in our implementations.

Overall, in addition to the task supervision loss, we add the MI estimation to the objective function to train the model as

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}}(\Theta(\Phi(X)), Y) + \lambda \mathcal{L}_{\text{NCE}}(Z, Z') \tag{4.3}$$

where $\mathcal{L}_{\text{CE}}(\cdot, \cdot)$ denotes the cross entropy loss between the predicted and the ground-truth class logits, and λ is the hyperparameter.

Classifier part: The classifier takes the latent representation Z as input and predicts the class logits Y. In contrast to the feature extractor, since we would like the classifier to learn task-specific knowledge, we need to wisely select the samples from the memory, so that they can not only bring extra information but also make sure the new information can be learned by the model. Sun et al. (Sun et al., 2022) formalize these two criteria as *surprise* and *learnability* by measuring the predictive distribution of the new sample with respect to those in the memory to decide its usefulness. In our work, we measure the information carried by the memory sample by estimating the MI between the encoded representation Z and the predicted logits Y. In this case, it differs from the original InfoNCE loss with extra information Y. We use Y here as another standard to determine positive and negative sample pairs, since we would like to train the classifier to learn task-specific knowledge given the label information. If we further incorporate the label information Y into Eq. 4.2, for each sample feature z_i, we need to consider z_k as the representation of the sample x_k for all $y_k = y_i$. All pairs of the original and augmented representations z_k and z'_k, along with the augmented representation z'_i, can be regarded as positive pairs with respect to z_i, while others are taken as negative pairs, i.e.,

$$\begin{aligned} &\mathcal{L}_{\text{NCE}}(Z, \{Z'\}, Y) \\ = &\frac{1}{N}\sum_{i=1}^{N}[\frac{1}{\sum\limits_{k=1}^{N} \mathbb{1}(y_k = y_i)} \sum_{y_k=y_i} (\sum_{\hat{z}_i \in \mathcal{S}_{z_i}} \log(f(z_i, \hat{z}_i)/\tau) \\ &- \log\sum_{j=1}^{N}(\sum_{\hat{z}_j \in \mathcal{S}_{z_j}} f(z_j, \hat{z}_j)/\tau))] \end{aligned} \tag{4.4}$$

where Y_{t-1} and Z_{t-1} are the corresponding label and features stored in memory at task $t-1$. $\mathcal{S}_{z_i}$ indicates the set of all of the original and augmented views of the sample z_k that has the same label with z_i, along with the augmented view of itself z'_i. The $\mathbb{1}(\cdot)$ function returns 1 if the condition is true and 0 otherwise. Note that we use the notations $\{Z'_t\}$ and $\{Z'_{t-1}\}$ here to indicate that more than one type of augmentation can be considered.

When we are selecting from the memory at task t, we would like to select the samples that are both representative and informative. It not only needs to carry new information to the existing model, but also should have a high learnability instead of becoming an outlier. To achieve this, we assign a score to the samples and select the samples with the highest score as

$$\begin{aligned} \text{score}_t(Y, Z) = &-\mathcal{L}_{\text{NCE}}(Z_{t-1}, \{Z'_{t-1}\}, Y_{t-1}) \\ &+ \mathcal{L}_{\text{NCE}}(Z_t, \{Z'_t\}, Y_t) \end{aligned} \tag{4.5}$$

The first term on the left-hand side of Eq. 4.4 indicates that we would like to be more likely to select samples that minimize the mutual information between Z and Z' given the class logits by the previous model at task $t-1$. In other words, the samples that are more surprising to the model are favored. Similarly, the second term indicates that the samples with higher learnability are more probably to be sampled, since they maximize the MI between Z and Z' given Y by the current model, which aligns with our objective function. In such a sense, we will be more likely to select samples that are both representative and informative, so that the model can effectively recall past knowledge and learn from new information at the same time.

4.3 Datasets and Baselines

We use the TAU Urban Acoustic Scenes 2022 (Heittola et al., 2020) and Environmental Sound Classification (ESC)-50 (Piczak) as our datasets. TAU Urban Acoustic Scenes consists of 10 classes of acoustic scenes in total, with around 1,000 samples for each class. Its acoustic scene classes are mainly about transportation and city noises. ESC-50 is a smaller dataset that is made up of 5-second-long recordings in 50 semantical classes, with 40 samples for each class. This dataset mainly covers the sounds from animals, humans and daily activities, etc. The diversity of these datasets helps us validate the effectiveness of our continual learning methods under different settings. For the task splitting, as described in Section 4.1, we split the 10 or 50 classes in the two datasets into 5 sequential tasks $\{X_t, Y_t\}_{t=1}^{5}$. Each task contains 2 or 10 different classes respectively. The order of the classes is determined by the original class indices from the raw datasets.

We evaluate the continual learning performance of acoustic scene classification with multiple baseline approaches together with our proposed method as in Tables 4.1 and 4.2. *Fine-tune* means the offline training without any continual learning approaches performed, which is the lower bound of our performance. *Random* is to randomly select samples from the memory with an equal probability. *Herding* indicates herding the embeddings of samples and selects those who are closest to the center of their corresponding class (Rebuffi et al., 2017b). *GSS* refers to gradient-based sample selection, which aims to maximize the diversity of the gradients of the samples in the memory buffer (Aljundi et al., 2019). *Uncertainty* calculates the uncertainty score of each sample based on the prototypes from the herding method, and selects the samples that the model is less confident of (Xiao et al., 2022).

4.4 Experimental Setup

For our feature extractor, we use a Temporal Convolutional Network (TCN) (Bai et al., 2018) as the feature extractor and a linear layer as the classifier. The feature extractor takes in the log-Mel spectrogram of the audio input, computed by a Hanning window with the window length of 25ms and the hop length of 10ms. The latent representation Z is represented as 100-dim embedding vectors, which are used to compute the scoring function to sample from the memory. We train the model for 50 epochs for each task with an Adam optimizer and an initial learning rate of 0.0005. We use the temperature $\tau = 1.0$ for all the experiments.

4.5 Evaluation Metrics

Aside from average classification accuracy (Acc), we use backward transfer (BWT) and forward transfer (FWT) as the evaluation metrics (Daz-Rodrguez et al., 2018) to show that our method helps not only learn task-agnostic knowledge, but also preserve the task-specific knowledge. BWT measures the influence of learning task t on the accuracies of all previous tasks $i < t$. The calculation of BWT at task t is defined as

$$\mathrm{BWT}_t = \frac{2}{t(t-1)} \sum_{i=2}^{t} \sum_{j=1}^{i-1} (a_{t,i} - a_{i,i}), t \in \{2, \cdots, T\}$$

Method	Memory Size	Acc ↑	BWT ↑	FWT ↑
fine-tune	-	20.4	-56.0	0.0
Random	0.2k	42.8	-28.5	49.8
	0.5k	49.8	-27.8	54.3
	1k	52.6	-27.0	59.2
Herding (Rebuffi et al., 2017b)	0.2k	51.6	-26.9	56.0
	0.5k	54.3	-26.3	63.3
	1k	56.2	-24.8	65.2
GSS (Aljundi et al., 2019)	0.2k	51.9	-25.3	56.5
	0.5k	54.6	-25.8	62.9
	1k	56.1	-24.6	63.7
Uncertainty (Xiao et al., 2022)	0.2k	55.9	-24.5	63.8
	0.5k	57.6	-23.7	67.5
	1k	58.9	-22.8	69.0
MIO	0.2k	58.0	-23.5	64.7
	0.5k	60.7	-22.9	69.1
	1k	**64.1**	**-22.5**	**74.8**

TABLE 4.1: Results for continual learning on TAU Urban Acoustic Scenes with different memory selection methods and sizes.

where $a_{i,j}$ denotes the accuracy of task j after learning task i. On the contrary, FWT measures the generalizability of the model by computing the influence of learning task t on the accuracies of future tasks. From (Lopez-Paz and Ranzato, 2017), we have

$$\text{FWT}_t = \frac{1}{t-1}\sum_{i=2}^{t}(a_{i-1,i} - \bar{a}_i), t \in \{2, \cdots, T\}$$

where $\bar{a}_i$ indicates the test accuracy at task i with random initialization. Overall, a higher BWT score means a smaller forgetting effect of the model on past task-specific knowledge, while a higher FWT score means a higher generalizability of the model on task-agnostic knowledge to benefit unseen tasks.

4.6 Results and Discussion

The experimental results on the TAU Urban Acoustic Scene dataset are shown in Table 4.1. The row of *fine-tune* suffers from catastrophic forgetting, and its accuracy is close to a random guess since we have 5 tasks in total. It is the lower bound of our continual learning performances. For the rest of the rows, we can observe that our proposed mutual information optimization (MIO) method achieves the highest score of Acc, BWT, FWT than other memory selection methods, which indicates that it can not only retain the task-specific knowledge in the past, but

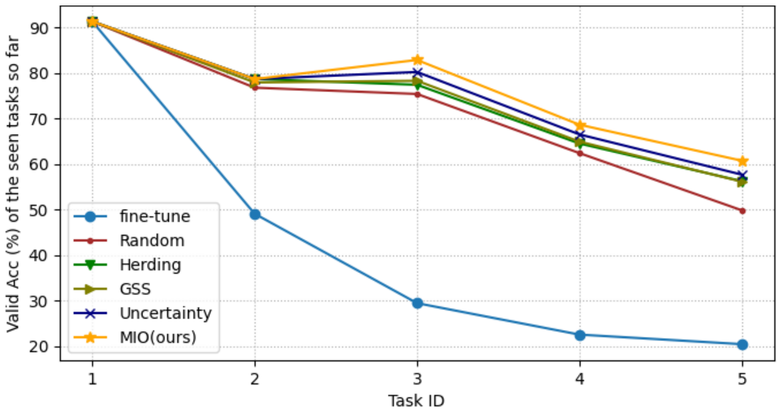


FIGURE 4.1: Average Acc (%) over tasks in sequential order for different methods. The accuracies are calculated on the test sets of the seen tasks so far.

generalize the task-agnostic knowledge to future unseen classes as well. Another observation is that our method benefits more from a larger memory size, with a higher performance gain from a smaller memory size to a larger one. This intuition aligns well with the property of the estimates of mutual information in Eq. 4.2, where the estimated MI approaches its lower bound when the number of samples N becomes larger.

Table 4.2 presents the results on the ESC-50 dataset with the same set of evaluation metrics. Overall, we can observe a similar tendency with the results in Table 4.1, except that the accuracies become lower because there are 50 classes in total and less number of samples per class. We also plot the change of the accuracies over the sequential tasks in Figure 4.1. From the figure, we can observe that our method has the least forgetting effect with a less decrease in accuracy. In contrast, fine-tuning has the largest drop with accuracy close to $\frac{1}{t}$, where t is the number of tasks that the model has experienced. It is noteworthy that from task 2 to task 3, some continual learning methods have an increased accuracy instead of a decreasing one. We speculate that this phenomenon is due to the shared properties of the acoustic scene classes that these tasks consist of.

4.7 Summary

This paper presents a replay-based continual learning approach in the acoustic scene classification task with mutual information estimation. We propose to optimize different levels of the model to learn task-agnostic and task-specific knowledge from the perspective of mutual information, and select samples from the memory buffer that are both representative and informative. We demonstrate that our proposed method outperforms other continual learning algorithms by both a lower forgetting effect and higher generalizability.

Method	Memory Size	Acc ↑	BWT ↑	FWT ↑
fine-tune	-	19.1	-58.7	0.0
Random	0.2k	22.5	-52.5	26.6
	0.5k	24.6	-49.7	27.3
	1k	26.2	-47.6	29.7
Herding (Rebuffi et al., 2017b)	0.2k	47.5	-30.8	49.3
	0.5k	49.3	-28.7	50.6
	1k	50.8	-27.9	52.2
GSS (Aljundi et al., 2019)	0.2k	48.8	-30.3	49.8
	0.5k	49.6	-29.3	50.8
	1k	50.3	-28.2	51.9
Uncertainty (Xiao et al., 2022)	0.2k	50.9	-28.9	51.6
	0.5k	51.8	-27.6	53.1
	1k	52.9	-27.1	53.9
MIO	0.2k	52.1	-28.5	53.4
	0.5k	53.7	-27.4	55.9
	1k	**55.3**	**-26.5**	**57.3**

Table 4.2: Results for continual learning on Environmental Sound Classification-50 dataset with different memory selection methods and memory sizes.

Chapter 5

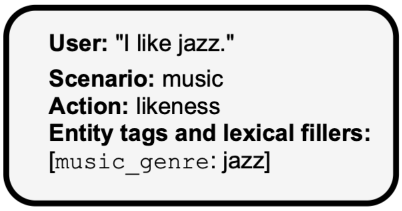
Continual Learning - Spoken Language Understanding

In this chapter, we select spoken language understanding as our task to demonstrate the effect of continual learning in the class-incremental setting. We mainly focus on intent classification and slot filling, which involve both sequence modeling and utterance modeling classification tasks. By formulating it as a sequence-to-sequence problem, we propose to combine rehearsal methods with regularization via knowledge distillation to combat forgetting at both the encoder and decoder levels.

5.1 Problem Statement

We first describe how we have defined the class-incremental learning setting for the SLURP dataset (Bastianelli et al., 2020). SLURP is a multi-domain dataset for E2E SLU comprising around 56 hours of audio of people interacting with a home assistant (*slurp_real*), with the addition of 43.5 hours of synthetic recordings (*slurp_synth*). At present this makes SLURP the biggest and the most diverse dataset in terms of lexical complexity for SLU. Each utterance is annotated with three semantics: Scenario, Action, and Entities. The pair (scenario, action) is defined as Intent. Overall, there are 18 unique scenarios, 46 actions (56 if we consider both *slurp_real* and *slurp_synth*), 55 entity types, and 69 intents. Figure 5.1 provides an example of an annotated utterance.

We have used the scenarios as a splitting criterion to define the tasks of the class-incremental setting. The complete list of scenarios is: [“alarm”, “audio”, “calendar”, “cooking”, “datetime”, “email”, “general”, “iot”, “lists”, “music”, “news”, “play”, “qa”, “recommendation”, “social”, “takeaway”, “transport”, “weather”]. Since the number of scenarios is limited and each scenario provides a high-level concept associated with each utterance, we think that they can closely resemble a practical application that must adapt to new general domains. Additionally, since the intent classification is the chief metric to assess our model against, the use of scenarios as



User: "I like jazz."

Scenario: music
Action: likeness
Entity tags and lexical fillers:
[music_genre: jazz]

FIGURE 5.1: Example of annotated utterance from the SLURP dataset. The intent in this case is the tuple (music,likeness).

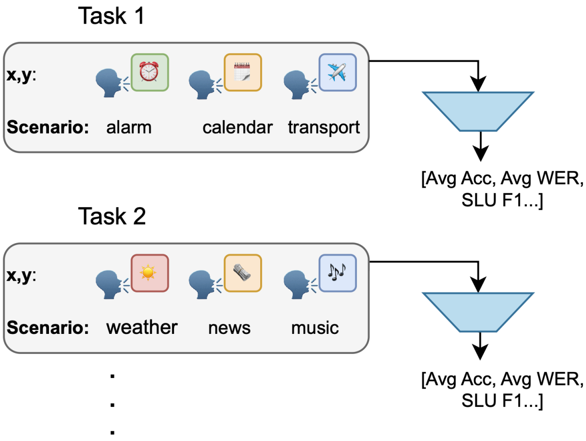

Task 1
x,y:
Scenario: alarm calendar transport
[Avg Acc, Avg WER, SLU F1...]
Task 2
x,y:
Scenario: weather news music
[Avg Acc, Avg WER, SLU F1...]

FIGURE 5.2: In the class-incremental setting for the SLURP dataset, a bunch of new scenarios (here 3) becomes available to the model for each task, while the previous ones are no longer available.

splitting criterion abides by the rule of having only intents related to scenarios available in the current task. Finally, although some actions and entities can be included in multiple scenarios, the overlap is very limited because the majority of the entities and actions are specific to a single scenario. For example, the action “taxi” is only associated with the scenario “transport”, and the entity “weather descriptor” with the scenario “weather”. Figure 5.2 shows two consecutive tasks, each introducing 3 new scenarios. Another critical aspect is the order in which the scenarios are available to the model. In our implementation, the order depends on the cardinality so that the scenarios with the highest cardinality appear first. In this way, we simulate a practical situation in which we endow the model with the sufficient general knowledge, learning the largest scenarios first, that will be useful for learning more specific scenarios. Note that we do not observe notable differences in performance for different orders.

5.2 Methodology

As discussed in the previous section, we consider a class-incremental learning setting in which we want to adapt a single model to perform well on all seen tasks. Specifically, the training dataset is divided into T distinct tasks, $\mathcal{D} = \mathcal{D}_0, ..., \mathcal{D}_{T-1}$. The dataset $\mathcal{D}_t$ of the tth task comprises audio signals $\mathcal{X}_t$ with associated transcriptions $\mathcal{Y}_t$, i.e. $\mathcal{D}_t = (\mathcal{X}_t, \mathcal{Y}_t)$. In a class-incremental learning setting, all task label sets are mutually exclusive, i.e. $\mathcal{Y}_i \cap \mathcal{Y}_j = \varnothing, i \neq j$.

We deploy a transformer-based seq2seq ASR architecture, constituted by a Wav2vec 2.0 encoder (WavEnc) (Baevski et al., 2020) followed by a transformer decoder. Let $\mathbf{x} = [x_1, \ldots, x_I]$ be an audio input sequence of length I, and $\mathbf{y} = [y_1, \cdots, y_J]$ be the corresponding output sequence of length J, with $y_j \in V$, where V is the set of all possible output subword tokens. The goal of the ASR model is to find the most probable output sequence $\hat{\mathbf{y}}$ given the input sequence $\mathbf{x}$:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{Y}^*} p(\mathbf{y}|\mathbf{x}; \theta) \tag{5.1}$$

where $\mathcal{Y}^*$ is the set of all possible token sequences and θ represents the parameters of the seq2seq model.

Suppose that $p(\mathbf{y}|\mathbf{x}; \theta_t)$ and $p(\mathbf{y}|\mathbf{x}; \theta_{t-1})$ are the output probability distributions of the transformer decoder at task t and $t-1$ parameterized by θ_t and θ_{t-1}, respectively. The model at task $t-1$ can be seen as the teacher model. Let also $\mathcal{M}$ be the set of rehearsal data at the beginning of task t. In the following equations, we use $\mathbf{x} \in \mathcal{D}_t$ in place of $(\mathbf{x}, \mathbf{y}) \in \mathcal{D}_t$ for brevity. The standard training criterion of rehearsal-based continual learning methods consists of minimizing the cross-entropy loss over $\mathcal{D}_t \cup \mathcal{M}_{t-1}$:

$$\mathcal{L}^t_{\text{CE}} = - \sum_{\mathbf{x} \in \mathcal{D}_t \cup \mathcal{M}_t} \log p(\mathbf{y}|\mathbf{x}; \theta_t) \tag{5.2}$$

The main idea of KD is to transfer knowledge from the teacher network $p(\mathbf{y}|\mathbf{x}; \theta_{t-1})$ to a student model, such that the latter mimics the former's behavior. Basically, the KD is used to force the current model to not deviate too much from the teacher, which retains the knowledge of the previous tasks. We point out that the KD, unless otherwise stated, is applied to the sole rehearsal data since the teacher can effectively predict only the data seen in the previous tasks. We propose three different types of KDs: audio-KD, token-KD, and seq-KD. The audio-KD works at the encoder's output level, whereas the other two KDs are applied to the output of the decoder. In this way, we contrast forgetting either at the encoder or at the decoder side (or both, if we combine multiple KDs).

The audio-KD forces the encoder's audio embeddings of the current task t to resemble those from the previous task $t-1$. Let $\mathit{WavEnc}(\mathbf{x}) \in \mathbb{R}^h$ be the Wav2vec 2.0 encoder output followed by a mean operation to squeeze the temporal dimension, where h is the hidden size. We define the audio-KD loss as:

$$\mathcal{L}^t_{\text{audio-KD}} = \sum_{\mathbf{x} \in \mathcal{R}_t} ||\mathit{WavEnc}_{\theta_{t-1}}(\mathbf{x}) - \mathit{WavEnc}_{\theta_t}(\mathbf{x})||^2 \tag{5.3}$$

where $||\cdot||$ is the Euclidean distance operator.

We can apply such similar reasoning to the decoder, which predicts each word of the transcription in an autoregressive way (in our case we use Byte-Pair Encoding (Sennrich et al., 2015), so we will use the term token rather than word to refer to the output units). The token-KD forces the current decoder to match the token-level distribution of the teacher. This is a kind of "local" distillation in that the student mimics the teacher for each token of the transcription. The corresponding CE criterion is defined as:

$$\mathcal{L}^t_{\text{token-KD}} = - \sum_{\mathbf{x} \in \mathcal{M}_{t-1}} \sum_{j=1}^{J} p(y_j|\mathbf{x}, \mathbf{y}_{<j}; \theta_{t-1}) \log p(y_j|\mathbf{x}, \mathbf{y}_{<j}; \theta_t) \tag{5.4}$$

where $\mathbf{y}_{<j}$ is the output sequence up to token $j-1$.

A potential flaw of this method is that if some initial token distributions are poorly estimated, their bias will be propagated until the end of the sequence. Indeed, a predicted token might be optimal at the current position in the sequence, but as we proceed through the rest of the sentence, it might turn out not to be the optimal one, given that later predicted positions are not already available.

Seq-KD is an alternative approach that trains the student to generate the same output sequence as the teacher, thus working at the sequence level. In practice, we generate a new set of automatic transcriptions with the teacher model using beam search at the end of each task ("soft transcriptions"), and then we use them to train the student network with CE criterion in the next task. Formally, we add the following CE loss:

$$\mathcal{L}^t_{\text{seq-KD}} = - \sum_{\mathbf{x} \in \mathcal{M}_{t-1}} \log p(\tilde{\mathbf{y}}|\mathbf{x}; \theta_t) \tag{5.5}$$

where $\tilde{\mathbf{y}}$ is the output sequence generated with beam search using the teacher model. Overall, the total loss to be optimized at task t is:

$$\mathcal{L}_{\text{total}} = (1 - \lambda_{\text{KD}})\mathcal{L}^t_{\text{CE}} + \sum_{k \in \mathcal{K}} \lambda_{\text{KD}} \mathcal{L}^t_k \tag{5.6}$$

where $\mathcal{K}$ = audio-KD, tok-KD, seq-KD and λ_{KD} is the weighting parameter. Depending on whether we employ a single KD or multiple ones, Eq. 5.6 changes accordingly. Figure 5.3 shows the learning process with the three KD losses applied to the transformer architecture.

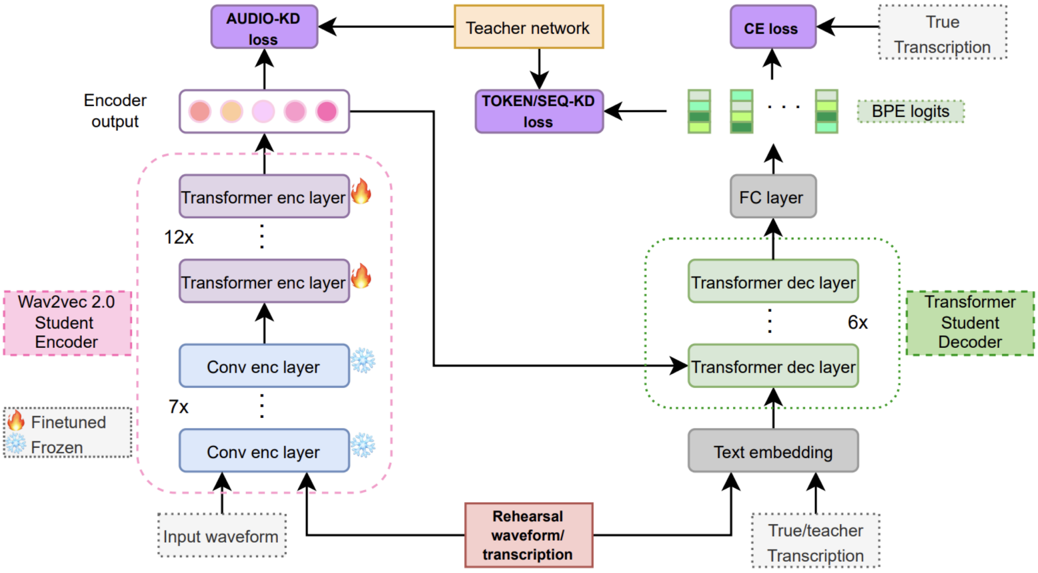


Figure 5.3: Illustration of the learning process in the proposed class-incremental learning setting. The model from the current task (student) mimics the behavior of the teacher model through audio, token, and sequence KD losses to counter forgetting.

5.3 Experimental Setup

Dataset and class-incremental learning setting. We conduct experiments on the SLURP dataset (Bastianelli et al., 2020) (see Section 5.1) using the official train, validation, and test splits, with a ratio of 70:10:20. In all experiments we also use *slurp_synth* only for training. Since very long audio data are harmful to efficient training, we remove the training samples longer than 7 seconds (around 0.004% of the total training dataset).

Concerning the definition of the class-incremental learning setting, we experiment on two configurations: 1) the dataset is partitioned into 3 tasks, each comprising 6 scenarios (denoted as SLURP-3); 2) a more challenging configuration wherein the 18 scenarios are distributed across 6 tasks (denoted as SLURP-6).

Pre-processing and model configurations. As proposed in (Arora et al., 2022), the intent and entity classification problems are treated as a sequence-to-sequence ASR task, where both intent and entities associated with an utterance are predicted alongside its transcription. In a sense, we build an augmented transcription that will be fed to the transformer decoder, prepending the intent to the original transcription, followed by the entities and the corresponding lexical values. The special token *_SEP* is used to separate the intent from the entities and the entities from the original transcription, whereas the token *_FILL* is used to separate each entity from its value. If the original transcription is the one in Fig. 5.3, then the augmented transcription becomes: *music_likeness _SEP music genre _FILL jazz _SEP I like jazz.*

Model. The encoder is the base Wav2vec 2.0 model pretrained and fine-tuned on 960 hours of Librispeech (a CNN- based feature extractor followed by 12 transformer blocks with hidden size = 768, 8 attention heads, 2048 feedforward network (FFN) hidden states). The feature extractor is kept frozen during the training, whereas the transformer blocks are fine-tuned. Then, the transformer decoder includes 6 layers with the same parameters as the encoder. We apply layer normalization to the input raw waveforms. The total number of parameters of the model is around 148M.

Training. We tokenize the transcriptions using Byte-Pair Encoding (BPE) (Sennrich et al., 2015), with a vocabulary size of 1k and BPE dropout $= 0.1$. Both at inference time and for the computation of the soft labels for the KD-seq we run beam search with beam width $= 20$. The number of epochs for each task is $\{40,25,15\}$ for SLURP-3, whereas we use $\{40,25,15,15,15,15\}$ epochs for SLURP-6. The batch size is 32. We use AdamW optimizer with learning rate = $5\mathrm{e}{-5}$ and weight decay $= 0.1$ and the same learning rate scheduler as (Vaswani et al., 2017). We use the validation set for hyperparameters tuning, and for selecting the best model for each task that is used for testing. Each experiment took approximately 1 day and a half on a Tesla V100 and a day on a Quadro RTX A5000.

CL baselines and strategies. Our upper bound is the offline method consisting of a single macro-task with all the scenarios (i.e. no incremental learning), while the naive fine-tuning approach, which retrains the same model task by task, is our lower bound. We consider two different sampling strategies for the rehearsal approach: 1) a random selection of the samples to retain, and 2) a herding-based selection (Rebuffi et al., 2017b), which selects the samples closest to their moving barycenter. We provide an example with a memory buffer of size equal to around 5% of the training dataset, and the rest of the experiments use 1%. Finally, we show the result for each KD strategy, as well as their various combinations. The KD weight in Eq. 5.6 is proportional to the fraction of rehearsal data in the mini-batch and is defined as:

$$\lambda_{\text{KD}} = \sqrt{\frac{b_{\text{mem}}}{b_{\text{all}}}} \tag{5.7}$$

where b_{mem} is the number of rehearsal data in the current mini-batch, and b_{all} is the current mini-batch size.

Metrics. We evaluate the proposed methods using 4 metrics: the average intent accuracy, **Avg Acc**, after each task; the intent accuracy after the last task **Last Acc**; the average **SLU F1** metric for slot filling (Bastianelli et al., 2020); the average word error rate, **Avg WER**, after each task.

5.4 Experimental Results

The performance for both class-incremental learning settings, SLURP-3 and SLURP-6, are reported in Table 5.1. First and foremost, we note that, as expected, the fine-tuning approach

Method	SLURP-3				SLURP-6			
	Avg Acc ↑	Last Acc ↑	Avg WER ↓	SLU F1 ↑	Avg Acc ↑	Last Acc ↑	Avg WER ↓	SLU F1 ↑
fine-tune	46.27	18.36	35.82	49.25	33.56	12.42	46.26	37.88
1% random rehearsal	71.30	61.47	29.13	60.05	66.11	59.37	34.77	55.33
1% random herding (Rebuffi et al., 2017a)	71.49	61.66	28.62	60.23	67.55	62.55	33.82	56.09
+ audio-KD	72.14	63.03	28.68	61.08	68.40	62.83	**32.04**	58.15
+ token-KD	71.79	61.54	28.82	**61.88**	68.36	62.53	32.47	58.20
+ seq-KD	**76.12**	**68.94**	**28.56**	61.50	**71.56**	**64.82**	32.50	**58.29**

Table 5.1: Results in terms of Average Accuracy, Last Accuracy, Average WER and SLU F1 for different strategies.

struggles in both settings, thus incurring catastrophic forgetting. The use of a rehearsal memory (rows 5% and 1%) proves to be very effective, even with only 1% of retained data. Therefore, in the following experiments, we consider 1% of data in the rehearsal memory. We also experiment with a more sophisticated sampling strategy based on herding (Rebuffi et al., 2017b), which achieves noteworthy improvements, in particular for SLURP-6 (+1.44% of Avg Acc).

When we focus on the proposed KDs, it is quite evident that the seq-KD leads to the most substantial improvement for both Avg and Last Acc metrics (+4.63% and +7.28% on SLURP-3). Instead, for WER and SLU F1, all three KDs behave similarly. Note that in our setting, previous intents are not seen anymore, and indeed the KDs help the model remember past scenarios. Conversely, though we expect the utterances to have some scenario-specific words, general speech tokens are spread among the tasks, making forgetting less critical for WER. Nevertheless, for the more challenging SLURP-6, KDs bring a notable enhancement also in terms of WER and SLU F1.

5.5 Summary

In this work, we define a class-incremental learning setting for a challenging SLU dataset, SLURP. To mitigate forgetting, we propose three different KD-based strategies working at different levels in the seq2seq model, with the joint use of rehearsal methods. Our extensive experiments reveal the superior performance of the proposed KD techniques, and that combining multiple KDs results in additional improvements.

Chapter 6

Proposed work

Thus far, we have demonstrated the effectiveness of continual learning methods in the data-incremental setting of ASR, and the class-incremental settings of ASC and SLU from Chapter 3 to 5 respectively. We have applied both rehearsal and regularization methods in different continual learning scenarios. Based on the completed work, we would like to first extend the work of continual learning in SLU in a multimodal formulation to investigate how different knowledge distillation techniques on different combinations of samples would affect the overall performance. Then we plan to study a more general setting aside from the completed ones, which is task-incremental learning, to continually add new capabilities to the SLU model to solve novel tasks. Furthermore, apart from the existing work on classification tasks in speech and audio, we would like to dive into a more challenging task-incremental regression task, i.e., speech enhancement, to address the continual learning problem where new noise types may appear over time.

6.1 Multimodal Contrastive Continual Learning in SLU

In the work of continual learning in SLU, we have transformed the intent and entity classification problem as a seq2seq ASR task by prepending the intent token to the original transcription. As a continuation of this idea, we propose to take into consideration of the text modality in addition to the audio modality to apply continual learning in both modalities. The overall proposed architecture is shown in Figure 6.1 with an encoder-decoder architecture.

At the encoder(s) side we would like to introduce 3 losses, two of which are applied to the two modalities alone, and one encompasses both modalities at once: 1) The Multi-modal alignment loss tries to align the encoder representations of the speech and text pair for each sample in the batch since they carry the same information but from a different perspective. This loss is computed only for the data from the current task, while the data from the rehearsal memory are not included. The loss relies on a contrastive learning approach. One important thing revolves around how to compare the two modalities since they have different numbers of tokens, and

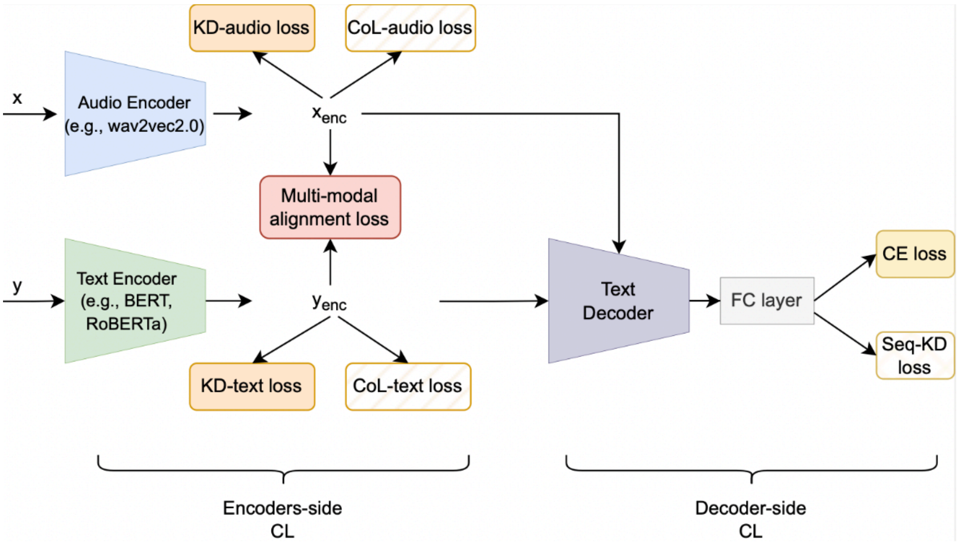


FIGURE 6.1: Proposed architecture for multimodal contrastive continual learning.

the hidden size could be the same (728 for both BERT and wav2vec 2.0). One possible way is to use just the first token of the two modalities (for example the intent token for the text and another token for the speech, although it's not straightforward to choose it as for vision where they use the [CLS] token) and project them into a shared space through two fully-connected (FC) layers and apply a standard contrastive loss as below (N is the batch size, z_{im}^k and z_{txt}^k are the projected tokens), and this approach is presented in (Kwon et al., 2022):

$$\ell_{\text{contrastive}} = -\frac{1}{N}\sum_{k=1}^{N}[\log\frac{\exp(z_{\text{im}}^k * z_{\text{txt}}^k/\tau)}{\sum_{n=1}^k \exp(z_{\text{im}}^n * z_{\text{txt}}^k/\tau)} + \log\frac{\exp(z_{\text{im}}^k * z_{\text{txt}}^k/\tau)}{\sum_{n=1}^k \exp(z_{\text{im}}^k * z_{\text{txt}}^n/\tau)}] \tag{6.1}$$

Another possibility averages the two feature representations in the time domain by means of a mean pooling operation, and then applies the same contrastive loss. In this case, no additional parameters are to be learned (i.e., no FC layers). This approach is showcased in (Ye et al., 2022).

A third approach could be a hybrid between the previous two: the intent BPE token is projected, whereas the audio signal tokens are averaged out in the temporal dimension. In this way, we don't have to consider a single token for the audio.

Overall, this loss tries to discern classes inside a specific task, thus promoting intra-task class representation learning. It does not deal with forgetting, but it seeks to learn optimal feature embeddings, which is a desideratum regardless of the setting at hand (continual learning or non-continual learning).

2) We include two KD-based losses, one for each modality, with the aim to force the current model to follow the learned features of the previous model. These two losses are only computed for the rehearsal buffer samples. In this way, our model should produce similar embeddings

for the rehearsal samples, whilst the features for the new samples (i.e., current task) should be something dissimilar. We force the model to stress this behavior even more with the next loss. This model promotes inter-task class representation learning, and tries to moderate forgetting.

3) We apply an additional contrastive loss that in principle should work with the two modalities individually. The idea here is to use the samples from the current task as anchors, whereas the data from the rehearsal memory are seen as negative samples. In this way, we hope the model will be able to push apart the embeddings belonging to the past tasks (rehearsal samples) from the ones of the current task. This idea has been presented in (Fang et al., 2021) for self-distillation KD, and then exploited and adapted in continual learning (Cha et al., 2021). In this case, the batch is composed of both rehearsal and new samples.

Note that loss 1) is applied to the new samples, loss 2) is applied only to rehearsal data, and 3) to both rehearsal and new samples.

6.2 Task-incremental SLU

Thus far, we have covered the data-incremental and class-incremental settings in continual learning for multiple speech and audio tasks. The domain of task-incremental learning still leaves underexplored. As discussed in Chapter 2, task-incremental learning is a more general setting where our goal is to continually add new capabilities to the model to solve new tasks. By transferring the knowledge learned from previous tasks, task-incremental models are expected to deal with increasingly more new tasks (Hossain et al., 2022).

We plan to expand the SLU work under the task-incremental setting, as some subtasks under SLU are correlated, which could be a good property to test the task-incremental learning scenario. One example is intent classification and slot filling. As two essential tasks for SLU, most existing works in this field are focused on joint intent classification and slot filling (Chen et al., 2019; Weld et al., 2022). In our planned work, we would like to explore the task-incremental SLU, where the model is firstly learning intent classification and then continually adding slot filling functions to itself.

From the implementation side, we propose to follow the Whisper (Radford et al., 2022) format, where a task identifier (e.g. ⟨IC⟩, ⟨SF⟩) is appended as a prompt to the beginning of the sentence, as part of the sequence of input tokens to the decoder. Similar as what we have done in the completed work, the model is expected to predict both the intent or entity class and the task type itself. From the dataset side, there are multiple datasets that involve both intent classification and slot filling tasks, including SLURP (Bastianelli et al., 2020) and SNIPS (Coucke et al., 2018). We plan to start with these datasets to reduce the complexity of initial experiments.

6.3 Continual Learning in Speech Enhancement

So far we have worked under the data-incremental and class-incremental settings, and we have demonstrated that the joint use of rehearsal and regularization methods have achieved a superior continual learning performance. The demonstrated and proposed works so far are all about classification tasks, including both sequence modeling and utterance modeling ones. Taking a further step, we would like to further investigate how continual learning works in regression tasks, where we would start with speech enhancement as an example. Such a scenario can also be regarded as task-incremental, as we do not have explicit labels to classify and data from different tasks are from different distributions.

Speech enhancement has been an important front end for many speech processing systems, aiming to improve the quality and intelligibility of speech by reconstructing and restoring the signal after degradation. In recent work, unsupervised continual self-training has been explored, which applies an iterative updating paradigm between teacher models and student models (Tzinis et al., 2022). The similar training paradigm can also be applied in continual learning, where the teacher and student model are defined by different out-of-domain (OOD) datasets. Data from previous tasks are retrieved from the rehearsal memory. Under continual learning setting, one common scenario for speech enhancement is that we are continually collecting noisy data from different noise types or environments. The goal is to enhance multiple types of noisy speech by adapting the unseen noise types without forgetting the knowledge of the seen noise types. This scenario is also useful for personalized speech enhancement, where continuous streams of data from the same speaker are collected under different environments. Another scenario is inspired by the fact that one of the challenges in speech enhancement is that speech with low Signal-to-Noise ratios (SNR) is much harder to enhance compared to those with high SNRs. Therefore, starting with enhancing high-SNR data is an easier *task*. After that, continuously lowering the SNR of the data is increasing the difficulty of the noise conditions, which makes it more challenging for the speech enhancement model.

On the fly, other related topics about continual learning in speech enhancement remain worthy of exploration. For example, our existing work about speech enhancement has shown that (Yang et al., 2022a; Zeng et al., 2023; Yang et al., 2022b) low-level acoustic features can play a significant role in improving the performance of speech enhancement, especially when considering the temporal and phonetic relationship between frames to weight the acoustic features by phoneme types. For the image classification task, prior work has shown that continual learning performance is largely affected by the spurious features that it selects (Lesort, 2022). Following this direction, we would like to further investigate the impact of various acoustic features in the continual learning of speech enhancement, and how to improve the feature selection scheme to mitigate the catastrophic forgetting effect in speech enhancement.

Chapter 7

Thesis Timeline

Thesis timeline for 2023-2024

May 2023	●	Thesis Proposal
Now-Aug 2023	●	Multimodal Contrastive Continual Learning in SLU
Now-Dec 2023	●	Investigate Task-incremental Learning in SLU
Jan-May 2024	●	Continual Learning in Speech Enhancement
Apr-May 2024	●	Thesis Completion & Thesis Defense

Bibliography

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.

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.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio, Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32. 2019.

Siddhant Arora, Siddharth Dalmia, Pavel Denisov, Xuankai Chang, Yushi Ueda, Yifan Peng, Yuekai Zhang, Sujay Kumar, Karthik Ganesan, Brian Yan, et al., Espnet-SLU: Advancing spoken language understanding through espnet. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7167–7171. IEEE. 2022.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460. 2020.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*. 2018.

Emanuele Bastianelli, Andrea Vanzo, Pawel Swietojanski, and Verena Rieser, Slurp: A spoken language understanding resource package. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020.

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

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny, Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*. 2018.

Qian Chen, Zhu Zhuo, and Wen Wang, Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*. 2019.

Zhiyi Chen and Tong Lin, Revisiting gradient episodic memory for continual learning. 2019.

Zhiyuan Chen and Bing Liu, Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 12(3):1–207. 2018.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al., Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*. 2018.

Natalia Daz-Rodrguez, Vincenzo Lomonaco, David Filliat, and Davide Maltoni, Don't forget, there is more than forgetting: new metrics for continual learning. *arXiv preprint arXiv:1810.13166*. 2018.

Matthias De Lange and Tinne Tuytelaars, Continual prototype evolution: Learning online from non-stationary data streams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8250–8259. 2021.

Zhiyuan Fang, Jianfeng Wang, Lijuan Wang, Lei Zhang, Yezhou Yang, and Zicheng Liu, Seed: Self-supervised distillation for visual representation. *arXiv preprint arXiv:2101.04731*. 2021.

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

Toni Heittola, Annamaria Mesaros, and Tuomas Virtanen, Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions. *arXiv preprint arXiv:2005.14623*. 2020.

François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia Tomashenko, and Yannick Estève, TED-LIUM 3: twice as much data and corpus repartition for experiments on speaker adaptation. In *International Conference on Speech and Computer*, pages 198–208. Springer. 2018.

Md Sazzad Hossain, Pritom Saha, Townim Faisal Chowdhury, Shafin Rahman, Fuad Rahman, and Nabeel Mohammed, Rethinking task-incremental learning baselines. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 2771–2777. IEEE. 2022.

David Isele and Akansel Cosgun, Selective experience replay for lifelong learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32. 2018.

Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, et al., A comparative study on transformer vs RNN in speech applications. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 449–456. IEEE. 2019.

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.

Gukyeong Kwon, Zhaowei Cai, Avinash Ravichandran, Erhan Bas, Rahul Bhotika, and Stefano Soatto, Masked vision and language modeling for multi-modal representation learning. *arXiv preprint arXiv:2208.02131*. 2022.

Frantzeska Lavda, Jason Ramapuram, Magda Gregorova, and Alexandros Kalousis, Continual classification learning using generative models. *arXiv preprint arXiv:1810.10612*. 2018.

Timothée Lesort, Continual feature selection: Spurious features in continual learning. *arXiv preprint arXiv:2203.01012*. 2022.

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

David Lopez-Paz and Marc'Aurelio Ranzato, Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30:6467–6476. 2017.

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

Aaron van den Oord, Yazhe Li, and Oriol Vinyals, Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*. 2018.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE. 2015.

Douglas B Paul and Janet Baker, The design for the wall street journal-based csr corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*. 1992.

Karol J. Piczak, ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pages 1015–1018. ACM Press.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever, Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*. 2022.

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. 2017a.

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. 2017b.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P Lillicrap, and Greg Wayne, Experience replay for continual learning. *arXiv preprint arXiv:1811.11682*. 2018.

Anthony Rousseau, Paul Deléglise, Yannick Esfteve, et al., Enhancing the TED-LIUM corpus with selected data for language modeling and more ted talks. In *LREC*, pages 3935–3939. 2014.

Anthony Rousseau, Paul Deléglise, and Yannick Esteve, TED-LIUM: an automatic speech recognition dedicated corpus. In *LREC*, pages 125–129. 2012.

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

Rico Sennrich, Barry Haddow, and Alexandra Birch, Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*. 2015.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim, Continual learning with deep generative replay. *arXiv preprint arXiv:1705.08690*. 2017.

Shengyang Sun, Daniele Calandriello, Huiyi Hu, Ang Li, and Michalis Titsias, Information-theoretic online memory selection for continual learning. In *International Conference on Learning Representations*. 2022.

Efthymios Tzinis, Yossi Adi, Vamsi K Ithapu, Buye Xu, Paris Smaragdis, and Anurag Kumar, Remixit: Continual self-training of speech enhancement models via bootstrapped remixing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1329–1341. 2022.

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

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

Henry Weld, Xiaoqi Huang, Siqu Long, Josiah Poon, and Soyeon Caren Han, A survey of joint intent detection and slot filling models in natural language understanding. *ACM Computing Surveys*, 55(8):1–38. 2022.

Yang Xiao, Xubo Liu, James King, Arshdeep Singh, Eng Siong Chng, Mark D Plumbley, and Wenwu Wang, Continual learning for on-device environmental sound classification. *arXiv preprint arXiv:2207.07429*. 2022.

Muqiao Yang, Joseph Konan, David Bick, Anurag Kumar, Shinji Watanabe, and Bhiksha Raj, Improving speech enhancement through fine-grained speech characteristics. In *Proceedings of the 23rd Annual Conference of the International Speech Communication Association*, pages 2953–2957. 2022a.

Muqiao Yang, Joseph Konan, David Bick, Yunyang Zeng, Shuo Han, Anurag Kumar, Shinji Watanabe, and Bhiksha Raj, Paaploss: A phonetic-aligned acoustic parameter loss for speech enhancement. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022b.

Muqiao Yang, Ian Lane, and Shinji Watanabe, Online continual learning of end-to-end speech recognition models. In *Proceedings of the 23rd Annual Conference of the International Speech Communication Association*, pages 2668–2672. 2022c.

Rong Ye, Mingxuan Wang, and Lei Li, Cross-modal contrastive learning for speech translation. *arXiv preprint arXiv:2205.02444*. 2022.

Yunyang Zeng, Joseph Konan, Shuo Han, David Bick, Muqiao Yang, Anurag Kumar, Shinji Watanabe, and Bhiksha Raj, Taploss: A temporal acoustic parameter loss for speech enhancement. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2023.

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

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.