mathematics

MDPI

Article

A Class-Incremental Learning Method Based on Preserving the Learned Feature Space for EEG-Based Emotion Recognition

Magdiel Jiménez-Guarneros * and Roberto Alejo-Eleuterio

Division of Postgraduate Studies and Research, National Technological of Mexico, Technological Institute of Toluca, Metepec 52149, Mexico; ralejoe@toluca.tecnm.mx

* Correspondence: mjmnzg@gmail.com

Abstract: Deep learning-based models have shown to be one of the main active research topics in emotion recognition systems from Electroencephalogram (EEG) signals. However, a significant challenge is to effectively recognize new emotions that are incorporated sequentially, as current models must perform retraining from scratch. In this paper, we propose a Class-Incremental Learning (CIL) method, named Incremental Learning preserving the Learned Feature Space (IL2FS), in order to enable deep learning models to incorporate new emotions (classes) into the already known. IL2FS performs a weight aligning to correct the bias on new classes, while it incorporates margin ranking loss and triplet loss to preserve the inter-class separation and feature space alignment on known classes. We evaluated IL2FS over two public datasets (DREAMER and DEAP) for emotion recognition and compared it with other recent and popular CIL methods reported in computer vision. Experimental results show that IL2FS outperforms other CIL methods by obtaining an average accuracy of 59.08 $\pm$ 08.26% and 79.36 $\pm$ 04.68% on DREAMER and DEAP, recognizing data from new emotions that are incorporated sequentially.

Keywords: class-incremental learning; deep learning; catastrophic forgetting; emotion recognition; electroencephalogram

Citation: Jiménez-Guarneros, M.; Alejo-Eleuterio, R. A Class-Incremental Learning Method Based on Preserving the Learned Feature Space for EEG-Based Emotion Recognition. *Mathematics* **2022**, *10*, 598. https://doi.org/10.3390/math10040598

Academic Editors: Ezequiel López-Rubio, Esteban Palomo and Enrique Domínguez

Received: 13 January 2022
Accepted: 11 February 2022
Published: 15 February 2022

1. Introduction

Emotion analysis has shown to be an important part of research fields such as human–computer interaction and health care, in order to improve the interactive experience and understand the behavior of patients [1,2]. Existing approaches in emotion recognition characterize the responses of emotions in two main modalities [3,4]: behavioral and physiological signals. The first type of modality includes those approaches based on facial expression [5,6], speech emotion recognition [7] and body language. Unlike this type of modality, the physiological signals provide a reliable way to recognize emotions since these signals are produced by the human body that may not be susceptible to subjective approaches based on behavioral signals [8]. In this sense, Electrocardiogram (ECG) [9], Electromyography (EMG) [10], Electroencephalogram (EEG) [4] or even a combination of them [11,12], have been used for emotion recognition. Among these physiological-signal-based approaches, EEG has provided a reliable and promising indicator to identify different emotional states, as it directly reflects brain activity [12]. Furthermore, EEG is a non-invasive device, easy to use, and has a low cost [4,13]. Thus, EEG has been widely used in emotion recognition systems in the last years [3,8,13–18].

Reported works have been mainly focused on extracting discriminative EEG emotional features and building more effective emotion recognition systems. The collected EEG signals are usually analyzed in three categories to extract discriminative features: time domain (e.g., statistics of signal), frequency domain (e.g., differential entropy), and time-frequency domain (e.g., Fourier transform). In this direction, many methods have been proposed via machine learning to leverage the features extracted from EEG signals [17–19]. Recently, several methods are gradually moving towards the deep learning-based approaches, becoming dominant in EEG-based emotion recognition [3,8,13–16,20,21]. For

example, different deep learning methods have been proposed to consider the spatial information, such as convolutional neural networks (CNNs) [3,14,16], capsule networks (CapsNets) [21] and graph neural networks (GNN) [8,13]. Likewise, attention mechanisms and recurrent networks [15] have been used to extract spatial and temporal information as emotion features.

Although remarkable progress has been achieved, there is a growing demand for adaptive, scalable, and responsive deep learning methods for emotion recognition tasks. Reported works are focused on recognizing emotions with fixed models while being unable to incorporate other emotions into their knowledge. New emotions may be recorded over time so that devices with pre-installed emotion recognition models may fail to recognize this new knowledge. Whenever samples from a new emotion become available, deep neural network models require retraining the whole model from scratch. This issue may be infeasible both in time or storage while using all training data or when the size of the main memory is limited [22]. Instead, the knowledge learned by a trained model should only be modified by using samples from a new emotion. In this sense, Class-incremental learning (CIL) provides a solution when new samples emerge, updating the knowledge of the model according to samples from new classes, avoiding re-configure the entire system [23].

CIL methods have been widely studied in computer vision [22] since several works have shown that deep learning models suffer from catastrophic forgetting when they are trained incrementally [24]. The catastrophic forgetting is the performance degradation of a neural network model affecting previously learned concepts whenever new ones are incorporated sequentially [25]. Different approaches have rapidly emerged to alleviate catastrophic forgetting. A first approach extends the model capacity to accommodate the latest knowledge as new data are integrated [26,27]. Although no sample is retained during incremental stages, these works may not scale well in specific scenarios since new weights are added each time. A second approach [28–31] uses a fixed model to generate feature representations across different incremental stages while multiple classifiers are trained for new classes. Although the retraining of the entire model is avoided, the performance of these methods depends on the quality of an initial representation, producing sub-optimal classification results in some cases [22]. Moreover, a third approach [25,32–40], named memory replay, stores a small set of representative samples from old classes and updates deep learning models via Fine-tuning (FT) across different incremental stages. The memory replay-based approach has shown better performance than previous approaches [35], but certainly the catastrophic forgetting is still under-studied. Mainly, in EEG-based signal recognition, Lee et al. [41] explored CIL for the imagined speech recognition task, but the authors used one of the most straightforward memory replay-based methods under an undemanding evaluation, as only a single incremental stage was tested for CIL. On the other hand, no work has been reported to study the dynamic changes in class for the EEG-based emotion recognition task. Thus, this research focuses on studying CIL for emotion recognition from EEG signals to enable deep learning models to incorporate new emotions into already known.

In this paper, we introduce *Incremental Learning preserving the Learned Feature Space* (IL2FS), a CIL method to address the catastrophic forgetting in EEG-based emotion recognition. The proposed method aims to preserve the feature space learned over past incremental stages, performing a bias correction on new classes, as well as encouraging the inter-class separation and feature space alignment over old classes. Firstly, we use Weighting Aligning (WA) [36] for bias correction on the weights at the output layer since class imbalance is present. Secondly, we use margin ranking loss to set a margin between scores of the ground-truth from old classes and their nearest score from any class (old or new), instead of only ensuring a separation between old and new classes, as reported in [33]. Finally, unlike previous CIL works for embedding networks [42–44], we propose to use triplet loss [45] to maintain the feature space alignment of old classes. IL2FS was implemented on a Capsule Network (CapsNet) architecture, which presents one of the best performances in terms of accuracy for emotion recognition [21]. We evaluate and validate our proposal on incremental learning tasks over two public datasets, DREAMER [46] and DEAP [11], using

a reduced set of samples from old classes and the maximum number of incremental stages that may be built for each dataset.

The main contributions of this work are:

1. We present a Class-incremental Learning method, named IL2FS, for emotion recognition from EEG signals, addressing the catastrophic forgetting problem.
2. IL2FS incorporates a strategy based on bias correction of the new classes while ensuring an inter-class separation and feature alignment of the old classes. This strategy allows better preservation of the learned knowledge for a greater number of incremental stages and a reduced number of reserved samples in memory.
3. We conduct experiments on two benchmarks, DEAP and DREAMER, for emotion recognition research. The proposed method achieves a significant improvement when compared with existing CIL methods.

The rest of this paper is organized as follows: in Section 2, we review previous works on class-incremental learning. Section 3 describes the proposed method in detail. Section 4 presents datasets, preprocessing procedure, neural network architecture and experimental setup. The corresponding results are reported in Section 5. Finally, the discussion and conclusions are reported in Sections 6 and 7.

2. Related Work

Existing works in EEG-based emotion recognition have focused on dynamic data distribution changes, but dynamic changes in class have not been studied yet. In [41], the authors explored CIL using a memory replay-based approach for the imagined speech task. Even so, a simple method [47] based on fine-tuning and the nearest neighbor classifier was adopted. Likewise, an undemanding evaluation was performed since only a single new class was tested, while a considerable percentage of data from old classes is reserved in memory when a new class is added. On the other hand, several CIL methods are available in computer vision to address the catastrophic forgetting problem. Among different approaches, reported in [22], we are interested in memory replay-based methods since they have shown superior performance in terms of accuracy. Thus, we describe several methods based on memory replay to deal with the catastrophic forgetting problem. We group these methods according to the problem they address.

Less forgetting. Knowledge distillation [48] was introduced as a regularizer on the outputs of a reference network and a new network in [49], in order to preserve the predictions of classes learned at previous CIL stages. For this, knowledge distillation aims to keep the new network weights close to the weights of the reference network. Moreover, Hou et al. [33] presented *Learning a Unified Classifier Incrementally via Rebalancing* (LUCIR), which introduces a less-forget constraint through the cosine distance, considering the local geometric structures of old classes in their feature space. More recently, Simon et al. [25] proposed a distillation loss, named *Geodesic*, by adopting the concept of geodesic flow between two tasks, that is, the gradual changes between tasks projected in intermediate subspaces.

Bias correction. In this group, CIL methods focus on updating the neural network weights in order to calibrate the bias produced by the class imbalance of representative samples. Wu et al. [50] proposed *Bias correction* (BiC) to rectify the weights of the model output, but a validation set is still required. In [33], authors observed that magnitudes of the weight vectors for new classes are higher than those of old classes, then, cosine normalization is used over the output layer to reduce the impact of imbalanced data. In this sense, *Incremental Learning with Dual Memory (LI2M)* [47] corrects scores of old classes storing their statistical information in an additional memory. *Classifier Weights Scaling for Class Incremental Learning* (ScaIL) [51] rectifies the weights of old classes to make them more comparable to those of new classes. Zhao et al. [36] proposed *Weight Aligning* (WA) to correct the biased weights at the output layer once the training process has ended. For this, only weight vectors of new classes are aligned to those of old classes using normalization.

Inter-class separation. The knowledge distillation loss has proven to be useful while producing more discriminative results within old classes when a bias correction is performed [36]. However, distillation loss may not be sufficient to ensure an inter-class separation between old and new classes since decision boundaries are re-configured during training over new classes. Thus, authors in [33] introduced margin ranking loss to encourage a margin that separates old and new classes. Chaudhry et al. [37] used bilevel optimization to update the model with new classes, keeping predictions intact on anchor points of old classes that lie close to the class decision boundaries.

Representative samples. Some strategies have been reported to select representative samples of old classes in order to avoid the model from overfitting to new classes. The baseline method, named Herding [32,52], selects the closest samples as most representative of a class, based on a histogram of the distances to the mean sample of that class. Authors in [53] introduced a more complex solution, named *Mnemonics*, which uses a strategy based on meta-learning to update the memory via gradient descent, selecting those samples located on boundary decisions. Generative solutions may also be found in [54,55], where artificial samples are drawn from each incremental stage, using generative adversarial networks (GANs). However, since GANs have proven to be difficult to optimize, they present scalability issues.

3. Proposed Method

In this section, we introduce the proposed method in detail. First, the Class-incremental learning setting is described. Then, we introduce an overview of the proposed method and its components. Finally, the training algorithm of the proposed method is presented.

3.1. Class Incremental Learning Setting

This research is focused on Class-Incremental Learning (CIL) based on the memory replay approach [22,32], where the neural network model complexity is maintained constant through S incremental stages, while new emotions are sequentially incorporated. In each incremental stage, samples from new emotions and a few samples from old emotions are available to retrain an existing neural network model.

Let $\mathcal{X}$ be a feature space with a label space $\mathcal{Y}$ belonging to classes (emotions) in $\mathcal{C}$. A labeled dataset is defined as $\mathcal{D} = \{(\mathbf{x}, \mathbf{y}) | \mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}\}$. We assume one initial stage and S incremental stages, where $\mathcal{C}$ is split into $S+1$ sets $\mathcal{C}^0, \mathcal{C}^1 ..., \mathcal{C}^S$ with $\mathcal{C} = \mathcal{C}^0 \cup \mathcal{C}^1 \cup ... \cup \mathcal{C}^S$ and $\mathcal{C}^i \cap \mathcal{C}^j = \emptyset$ for $i \neq j$. A budget is determined for the memory $\mathcal{M} = \{(\mathbf{x}, \mathbf{y}) | \mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}\}$, which is used to store a limited amount of representative samples from old classes. In the initial stage, a deep neural network model is trained on a labeled dataset $\mathcal{D}_0$. Next, a representative set of samples $\mathcal{E}_0$ is selected and stored in memory $\mathcal{M}$ as a replacement of $\mathcal{D}_0$, with $|\mathcal{E}_0| \ll |\mathcal{D}_0|$. In the incremental stage s, a deep network model is updated using the labeled dataset $\mathcal{D}_s$ and memory $\mathcal{M}$, that is, $\mathcal{D}_s \cup \mathcal{M}$. Notice that $\mathcal{M}$ now contains representative samples of old classes $\mathcal{E}_{0:s-1}$ from incremental stage 0 to $s-1$. We assume all training samples in $\mathcal{D}_s$ are available to train a neural network. In CIL, the main objective is to use a deep network model and $\mathcal{D}_s \cup \mathcal{M}$ to accurately classify samples belonging to old and new classes in each incremental stage s, avoiding catastrophic forgetting.

A deep neural network model is usually denoted as a labeling function f with trainable weights Φ, such that $\hat{\mathbf{y}} = f(\mathbf{x}; \Phi)$. The function f may be represented as composite of two functions, $f_{\text{enc}} \circ f_{\text{cls}}$. Here, f_{enc} represents the part of network that encodes an input $\mathbf{x}$ into a latent feature representation $\mathbf{z}$, that is, $\mathbf{z} = f_{\text{enc}}(\mathbf{x}; \theta)$; θ is the set of trainable weights. Then, latent features $\mathbf{z}$ are fed to a feature labeling function f_{cls} with weights ϕ, in order to produce a classification score $\hat{\mathbf{y}}$, i.e., $\hat{\mathbf{y}} = f_{\text{cls}}(\mathbf{z}; \phi)$. In CIL, the number of classes of the model output increases at each incremental stage. Thus, the network model f is expected to classify $|\mathcal{C}^s|$ more classes at incremental stage s than at stage $s-1$.

3.2. Overview of the Proposed Method

The proposed method, named *Incremental Learning preserving the Learned Feature Space* (IL2FS), faces the catastrophic forgetting problem aiming to preserve the learned feature

space from old classes. For this, IL2FS performs a bias correction of new classes, while the inter-class separation and feature space alignment of old classes are ensured. Firstly, a bias correction is performed on the weights at the output layer via Weight Aligning [36], as imbalanced data are present when trained over a reduced set of representative samples of old classes. Then, an inter-class separation is encouraged between scores from old classes and their nearest class (old or new) via margin ranking loss, instead of only encouraging a separation between old and new classes, as reported in [33]. Finally, since that new knowledge may modify the learned feature space at previous CIL stages, we propose to use triplet loss [45] to preserve the feature space alignment of old classes.

The complete flowchart is shown in Figure 1 and the overall objective can be written as follows

$$\mathcal{L}_{\text{inc}}(\mathcal{D}_s, \mathcal{M}, f_{(s-1)}; \Phi) = \beta \cdot \mathcal{L}_{\text{tri}} + \mathcal{L}_{\text{cls}} + \alpha \cdot \mathcal{L}_{\text{mr}}, \tag{1}$$

where $\mathcal{L}_{\text{tri}}$ is the triplet loss, $\mathcal{L}_{\text{cls}}$ is a classification loss, and $\mathcal{L}_{\text{mr}}$ is the margin ranking loss. λ, α, β are the trade-off hyper-parameters.

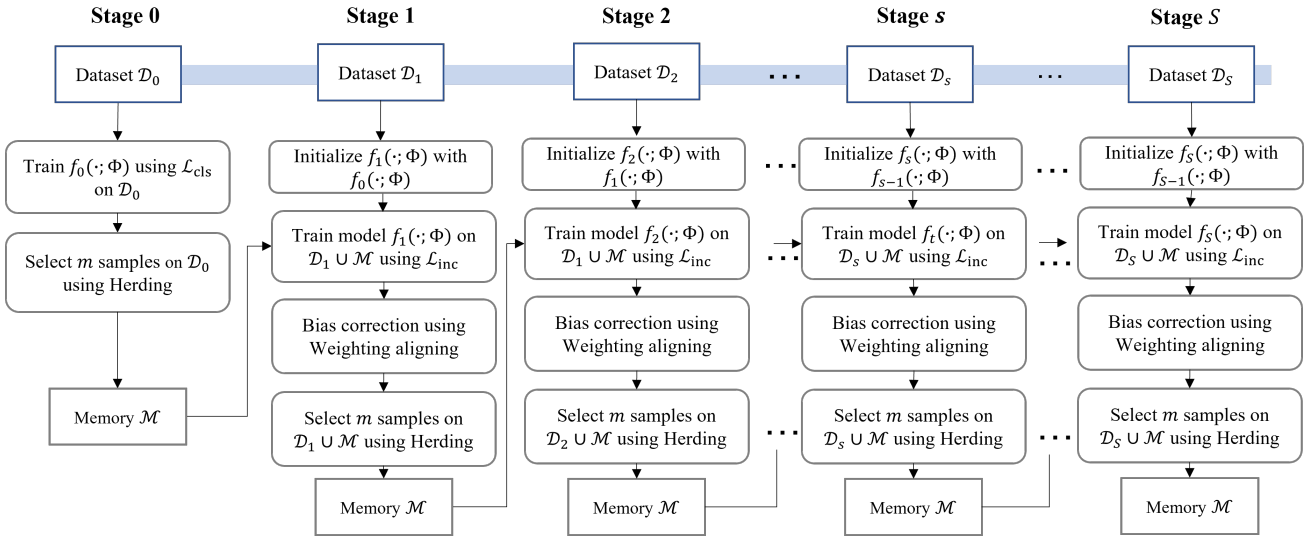


Figure 1. Flowchart of the proposed method throughout S incremental stages.

As shown in Figure 1, the network model f_0 is trained at stage 0 on $\mathcal{D}_0$, using the classification loss. Next, the Herding method [32,52] is employed to select m representative samples to be stored in memory $\mathcal{M}$. At the incremental stage s, weights Φ are initialized using those learned at stage $s-1$. Then, the network model f_s is retrained on $\mathcal{D}_s \cup \mathcal{M}$, using loss function $\mathcal{L}_{\text{inc}}$. Exponential Moving Average (EMA) [56] is also incorporated into IL2FS in order to stabilize the training of f_s over n training steps:

$$\Phi_{\text{EMA}}^{(n)} = (1 - \lambda_{\text{EMA}}) \cdot \Phi_{\text{EMA}}^{(n-1)} + \lambda_{\text{EMA}} \cdot \Phi^{(n)}, \tag{2}$$

where $\Phi_{\text{EMA}}^{(n)}$ is the EMA of successive Φ weights over n and λ_{EMA} is the decay rate or momentum. Then, at the end of the model's training, Weighting Aligning is used to align the norms of the weight vectors between old and new classes at the output layer. Likewise, m representative samples are selected on $\mathcal{D}_s \cup \mathcal{M}$, considering a balanced selection. This procedure is repeated every time new classes emerge, which must be incorporated into an existing model.

3.3. Bias Correction

Weight Aligning (WA) [36] has been used for bias correction, given that a class imbalance is produced by using a reduced set of representative samples of old classes in new incremental stages. Thus, WA rectifies the weight vectors at the output layer of a network model, aligning the norms of the weight vectors between old and new classes.

The output layer is rewritten as

$$\mathbf{W} = (\mathbf{W}_{\text{old}}, \mathbf{W}_{\text{new}}), \tag{3}$$

where

$$\begin{aligned} \mathbf{W}_{\text{old}} &= (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_{C_{\text{old}}}) \in \mathbb{R}^{d \times C_{\text{old}}}, \\ \mathbf{W}_{\text{new}} &= (\mathbf{w}_{C_{\text{old}}+1}, ..., \mathbf{w}_{C_{\text{new}}}) \in \mathbb{R}^{d \times C}, \end{aligned} \tag{4}$$

while the norms of the weight vectors are expressed as follows

$$\begin{aligned} \|\mathbf{W}_{\text{old}}\| &= (\|\mathbf{w}_1\|, \|\mathbf{w}_2\|, ..., \|\mathbf{w}_{C_{\text{old}}}\|), \\ \|\mathbf{W}_{\text{new}}\| &= (\|\mathbf{w}_{C_{\text{old}}+1}\|, ..., \|\mathbf{w}_{C_{\text{new}}}\|). \end{aligned} \tag{5}$$

Then, the weights of new classes are normalized by using

$$\bar{\mathbf{W}}_{\text{new}} = \gamma \cdot \mathbf{W}_{\text{new}}, \tag{6}$$

where

$$\gamma = \frac{M(\|\mathbf{W}_{\text{old}}\|)}{M(\|\mathbf{W}_{\text{new}}\|)}. \tag{7}$$

Here, $M(\cdot)$ computes the mean value using these weight vectors.

3.4. Inter-Class Separation

We assume that decision regions of old classes may change during model retraining, as representative samples of old classes are used for this process. Then, an inter-class separation is ensured by setting a margin over class scores throughout the different incremental learning stages.

Margin ranking loss was introduced in [33] to ensure a separation between old and new classes (see Section 2). Unlike previous work, we use a variant of the margin ranking loss to encourage an inter-class separation between the ground-truth score of an old class and its nearest score coming from any class, old or new.

For each sample $\mathbf{x}$ in memory $\mathcal{M}$, a separation is encouraged between the ground-truth old classes and their nearest class (old or new). For each sample $\mathbf{x}$ in memory $\mathcal{M}$, the score $\bar{\omega}(\mathbf{x})$ of the ground-truth old class is considered positive, while the maximum score $\bar{\omega}^k(\mathbf{x})$ among the remaining classes is considered hard negative. We have:

$$\mathcal{L}_{\text{mr}} = \sum_{\mathbf{x} \sim \{\mathcal{M}\}} \sum_{k=1}^{K} \max(b - \bar{\omega}(\mathbf{x}) + \bar{\omega}^k(\mathbf{x}), 0), \tag{8}$$

where b is the margin, $\bar{\omega}(\mathbf{x})$ is the score of the ground-truth class for the sample $\mathbf{x}$, and $\bar{\omega}^k(\mathbf{x})$ is the nearest class score for $\mathbf{x}$.

3.5. Feature Space Alignment

We incorporate triplet loss [45] to leverage the less forgetting, preserving the alignment of the feature space of classes learned at previous incremental stages. Note that existing strategies are mainly focused on maintaining the same output predictions of old classes (see Section 2). On the other hand, previous works in CIL [42–44] have mainly used triplet loss to train embedding networks and ensure an inter-class separation. However, unlike previous works, we incorporate triplet loss to preserve the feature space alignment of old samples, producing near feature representations from $f_{\text{enc}_{(S-1)}}$ and $f_{\text{enc}_{(S)}}$ for the same processed sample. Here, $f_{\text{enc}_{(S-1)}}$ is the model learned at the last incremental stage $(s-1)$ and $f_{\text{enc}_{(S)}}$ is the new model to train in the current stage s. Representations from different samples, processed by $f_{\text{enc}_{(S-1)}}$ and $f_{\text{enc}_{(S)}}$, are pushed away from each other by a small margin. Note that class labels for the processed samples are not used in our proposal, as an inter-class separation is not pursued.

More specifically, we use triplet loss to push latent feature representations $\mathbf{z}_{s-1} = f_{\text{enc}_{(S-1)}}(\mathbf{x}_1)$ and $\mathbf{z}_s = f_{\text{enc}_{(S)}}(\mathbf{x}_1)$ close to each other for the same sample $\mathbf{x}_1$. Meanwhile, the latent features $\mathbf{z}_{s-1}$ and $\mathbf{z}_s$, produced by $f_{\text{enc}_{(S-1)}}$ and $f_{\text{enc}_{(S)}}$, but coming from samples $\mathbf{x}_1$ and $\mathbf{x}_2$, are pushed away from each other by a margin.

The triplet loss is defined as follows

$$\mathcal{L}_{\text{tri}} = \sum_{\mathbf{x} \sim \{\mathcal{M}\}} \max(d(\mathbf{z}^a, \mathbf{z}^p) - d(\mathbf{z}^a, \mathbf{z}^n) + a, 0), \tag{9}$$

where $\mathbf{z}^a$ is the anchor input, $\mathbf{z}^p$ is a positive input of the same label as $\mathbf{z}^a$, while $\mathbf{z}^n$ is a negative input of a different label as $\mathbf{z}^a$; a is the margin and d is the cosine dissimilarity measure. Anchor-positive pairs are formed by latent features generated by $f_{\text{enc}_{(S-1)}}$ and $f_{\text{enc}_{(S)}}$ for the same sample, while anchor-negative pairs are formed by latent features generated by $f_{\text{enc}_{(s-1)}}$ and $f_{\text{enc}_{(S)}}$ for a pair of different samples. $f_{\text{enc}_{(S-1)}}$ processes all samples within the current batch to generate their respective latent feature representations. After, each featured sample is labeled according to its index into the batch of samples. This procedure is repeated for all samples but using $f_{\text{enc}_{(S)}}$; later, featured samples are concatenated with those obtained by $f_{\text{enc}_{(S-1)}}$. Then, the multi-similarity miner [57] is used to generate anchor-positive pairs $(\mathbf{z}^a, \mathbf{z}^p)$ and anchor-negative pairs $(\mathbf{z}^a, \mathbf{z}^n)$ over labeled feature representations in order to preserve the feature alignment of old classes.

3.6. Training of IL2FS

Algorithm 1 presents the training procedure of IL2FS at incremental stage s. First, the set of weights Φ is initialized using weights $\Phi_{(s-1)}$ (line 1). Next, we compute latent features for $\mathbf{x}$ using the reference model and current model (lines 7–8). Featured samples are labeled according to their indices into the dataset (lines 9–10). Anchor-positive and anchor-negative pairs are generated using the Multi-similarity miner (line 11) to be employed in triplet loss $\mathcal{L}_{\text{tri}}$. Then, scores for ground-truth old classes and their nearest classes are computed in order to be used in margin ranking loss $\mathcal{L}_{\text{mr}}$ (lines 12–14). After, neural network model f_s is trained using the loss function $\mathcal{L}_{\mathbf{inc}}$ (line 15). Note that $\mathcal{L}_{\mathbf{inc}}$ is composed of classification loss $\mathcal{L}_{\mathbf{cls}}$, triplet loss $\mathcal{L}_{\mathbf{tri}}$ and margin ranking loss $\mathcal{L}_{\mathbf{mr}}$. The EMA weights $\Phi^{(n)}_{\text{EMA}}$ are computed from $\Phi^{(n)}$ (line 16). After training f_s, weight vectors of the output layer are rectified employing the Weighting Aligning method (line 18). Finally, the memory $\mathcal{M}$ is updated by selecting m representative samples on $\mathcal{D}_s \bigcup \mathcal{M}$ by means of the Herding method (line 19).

Algorithm 1 Training algorithm of IL2FS at incremental stage s.

Inputs: $\mathcal{D}_s$ – training labeled dataset from new classes; $\mathcal{M}$ – memory containing representative samples from old classes; $f_{(s-1)}(\cdot;\Phi_{s-1})$ – reference model trained at incremental stage $s-1$; λ_{mr}, α, β – trade-off hyperparameters; λ_{EMA} – decay rate; η – learning rate; n – number of epochs.

Output: $f_s(\cdot;\Phi_{\text{EMA}})$ – a trained neural network model; $\mathcal{M}$ – updated memory with representative samples from old classes.

Initialize Φ_s with $\Phi_{(s-1)}$.
$\mathbf{x}_{old}, \mathbf{y}_{old} \leftarrow \mathcal{M}$
$\mathbf{x}_{new}, \mathbf{y}_{new} \leftarrow \mathcal{D}_s$
$\mathbf{x} \leftarrow \mathbf{x}_{old} \cup \mathbf{x}_{new}$
$\mathbf{y} \leftarrow \mathbf{y}_{old} \cup \mathbf{y}_{new}$
repeat
  $\mathbf{z}_{ref} \leftarrow f_{\text{enc}_{(S-1)}}(\mathbf{x})$ ▷ Compute features for samples using the reference model
  $\mathbf{z}_{cur} \leftarrow f_{\text{enc}_{(S)}}(\mathbf{x})$ ▷ Compute features for samples using the current model
  $\mathbf{v}_{ref} \leftarrow \text{GenerateLabels}(\mathbf{z}_{ref})$ ▷ Assign labels based on indices into the dataset
  $\mathbf{v}_{cur} \leftarrow \text{GenerateLabels}(\mathbf{z}_{cur})$
  $\mathbf{z}^a, \mathbf{z}^p, \mathbf{z}^n \leftarrow \text{MultiSimilartyMiner}(\mathbf{z}_{ref} \cup \mathbf{z}_{cur}, \mathbf{v}_{ref} \cup \mathbf{v}_{cur})$ ▷ Generate anchor-positive and anchor-negative pairs
  $\bar{\omega}(\mathbf{x}) \leftarrow f_{(s-1)}(\mathbf{x}_{old})$ ▷ Compute scores for samples from old classes using the reference model
  $\bar{\omega}_a(\mathbf{x}) \leftarrow f_s(\mathbf{x})$ ▷ Compute scores for all samples using the current model
  $\bar{\omega}^k(\mathbf{x}) \leftarrow \text{NearestClass}(\bar{\omega}(\mathbf{x}), \bar{\omega}_a(\mathbf{x}))$ ▷ Obtain scores from the nearest classes to old classes
  $\Phi^{(i)} \leftarrow \Phi^{(i-1)} - \eta \cdot \nabla[\beta \cdot \mathcal{L}_{\text{tri}}(\mathbf{z}^a, \mathbf{z}^p, \mathbf{z}^n; \Phi^{(i-1)}) + \mathcal{L}_{\text{cls}}(f_s(\mathbf{x}), \mathbf{y}; \Phi^{(i-1)}) + \alpha \cdot \mathcal{L}_{\text{mr}}(\bar{\omega}(\mathbf{x}), \bar{\omega}^k(\mathbf{x}); \Phi^{(i-1)})]$
  $\Phi^{(i)}_{\text{EMA}} = (1-\lambda_{\text{EMA}}) \cdot \Phi^{(i-1)}_{\text{EMA}} + \lambda_{\text{EMA}} \cdot \Phi^{(i)}$
until n epochs are reached
$\Phi_{\text{EMA}} \leftarrow \text{WeightingAligning}(\Phi_{\text{EMA}})$ ▷ Bias correction
$\mathcal{M} \leftarrow \text{Herding}(\mathcal{D}_s \bigcup \mathcal{M})$. ▷ memory is updated using the Herding method
return $f_s(\cdot;\Phi_{\text{EMA}})$, $\mathcal{M}$

4. Experimental Design

This section first describes two public datasets used in our experiments. Then, the neural network architecture, comparison methods and implementation details are introduced. (Code is available at https://github.com/mjmnzg/IL2FS. Accessed on 11 January 2022).

4.1. Datasets

Experiments were performed on two public datasets, DREAMER [46] and DEAP [11], since they are benchmarks for emotion recognition research [3,14,15,21]. DREAMER is a multi-channel dataset containing records of nine emotions from EEG signals per subject. Likewise, DEAP is a large-scale dataset containing EEG signals with different emotional evaluations. More importantly, both datasets were selected since a high number of classes may be obtained from EEG data, making it useful for the analysis of the catastrophic forgetting problem in emotion recognition.

The DREAMER dataset comprises EEG data from 23 subjects (14 male and nine female). EEG data were collected while the subjects watched 18 film clips, which contain cut-out scenes to evoke nine emotions: calmness, surprise, amusement, fear, excitement, disgust, happiness, anger, and sadness. The length of each film clip is between 65 to 393 s (M = 199 s). EEG signals were recorded at a sampling rate of 128 Hz using an Emotiv EPOC system that uses 16 electrodes, following locations according to the International 10–20 systems: AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4, M1, and M2. Sensor M1 acts as a ground reference, while M2 is a feed-forward reference; then, the remaining 14 electrodes were recorded and used for feature extraction. EEG data from all subjects have 18 experimental EEG trials, two per elicited emotion. Each EEG trial

begins with a neural film to help the subjects return to the neutral emotion state, while data serve as a baseline. EEG signals of each trial were filtered with Hamming bandpass linear phase FIR filters to extract frequencies inside the ranges of interest (4–30 Hz). Likewise, artifacts were removed by using artifact subspace reconstruction (ASR) [58]. At the final step, the Common Average Reference (CAR) method [59] was applied to compute the average value over all electrodes and subtracts it from each sample of each electrode. In our experiments, we adopt a discrete categorization instead of a dimensional categorization, with nine classes available.

The DEAP dataset contains EEG and peripheral physiological signals from 32 subjects while watching 40 music videos. EEG signals were collected using a cap of 32 electrodes, placed according to the international 10–20 system [60]. For this, a sampling rate of 512 Hz was used, then downsampled to 128 Hz. We used the pre-processed data (https://www.eecs.qmul.ac.uk/mmv/datasets/deap/readme.html. Accessed on 1 June 2021), where each trial contains 60 s of recorded signals under stimulation and 3 s of baseline signals in a relaxed state. A bandpass filter from 4.0–45.0 Hz was applied over EEG signals, and eye artifacts were removed as in [11] using independent component analysis (ICA). EEG data were averaged to the common reference. Subjects rate their levels of arousal, valence, linking, and dominance from 1 to 9 for each music video. In our experiments, we adopt a multi-class categorization scheme, combining discrete ratings of valence, arousal and dominance. Firstly, we divide each emotion dimension into two categories using a rating of 5 as threshold: low/high valence, low/high arousal and low/high dominance. Secondly, we label each EEG trial used as a combination of binary categorization in three dimensions. For instance, its label is 0 when the rating is low for the three dimensions, while the label is 1 when the rating for valence and arousal is low, but the rating for dominance is high. Finally, the recognition task is a multi-class classification composed of a maximum of 8 classes, given that not all subjects rate for every level of arousal, valence and dominance.

4.2. Preprocessing

We applied the preprocessing procedure of baseline removal on EEG signals as in the works reported by [3,15,21,61] since this method highlights the effects of stimulated emotions. We begin by using a non-overlapping window to slice baseline signals into N segments of 1 s for each trial and C electrodes. From the set of N segments, we obtain the mean segment, which represents the base emotional state without stimulation. Next, the mean segment is subtracted from the EEG signals under stimulation. The obtained differences represent the electrical changes in the brain under stimulation. Following this pre-processing, 1080 EEG samples are obtained for each subject in DREAMER, where 60 segments are obtained from each experimental trial; 18 experimental trials per subject. In this direction, each trial in DEAP is divided into 60 segments, each one containing 128 sampling points. Then, we obtain 2400 EEG samples for each subject since there are 40 trials per subject. Finally, each EEG sample in DREAMER and DEAP is a 32×128 matrix and 14×128 matrix, composed of the number of electrodes and sampling points, respectively.

4.3. Neural Network Architecture

We adopted a Capsule Network (CapsNet) architecture [21], which showed one of the best accuracy performances for EEG-based emotion recognition research. Figure 2 presents the CapsNet architecture and Table 1 describes the implementation details. Unlike the original CapsNet architecture, we add a module based on an attention mechanism, which includes a Channel-Attention block [62] into the modules from Convolutional to PrimaryCaps. In addition, the bottleneck layer proposed in [21] was removed since it dramatically increases the resources used in memory. To train CapsNet, the classification loss $\mathcal{L}_{\text{cls}}$ uses the margin and reconstruction losses, as suggested in [63]. For this purpose, CapsNet employs a separated margin loss $\mathcal{L}_k$ for each class k. On the other hand, reconstruction loss $\mathcal{L}_{\text{rec}}$ uses the sum of squared differences between the outputs of a decoder and the input EEG signal values. This decoder consists of 3 fully-connected layers that model the EEG signals.

Table 1. Specifications of the Capsule Network architecture. We include a Channel-Attention block before the PrimaryCaps module. The decoder setting for reconstruction loss is shown at the bottom.

Id	Modules	Layers (Input ID)	Hyperparameters	Output Shape
I1	Input	–	–	DREAMER: 14 × 128 DEAP: 32 × 128
C2	Convolutional	Convolution-2D (I1)	DREAMER: 64 filters, size = 6, stride = 1, activation = ReLU DEAP: 64 filters, size = 9, stride = 2, activation = ReLU	DREAMER: 64 × 123 × 9 DEAP: 64 × 60 × 12
A3	Channel-Attention	Average pooling (C2)	Size = 1, stride = 1	
C4		Convolution-2D (A3)	32 filters, size = 1, stride = 1, activation = ReLU	
C5		Convolution-2D (C4)	64 filters, size = 1, stride = 1	
M6		Maxpooling (C2)	size=1, stride=1	
C7		Convolution-2D (M6)	32 filters, size = 1, stride = 1, activation = ReLU	
C8		Convolution-2D (C7)	64 filters, size = 1, stride = 1	
S9		Sum (C5, C8)	–	
A10		Activation (S9)	Sigmoid	DREAMER: 64 × 123 × 9 DEAP: 64 × 60 × 12
C11	PrimaryCaps	Convolution-2D (A10)	DREAMER: 8 × 16 filters, size = 6, stride = 2 DEAP: 8 × 16 filters, size = 9, stride = 2	
R12		Reshape (C11)	–	DREAMER: 1088 × 8 DEAP: 832 × 8
E13	EmotionCaps	Dynamic routing (R12)	16 units	16 × 16
N14	Norm	Normalization (E13)	–	16
O15	FC	Fully connected (N14)	Dynamic outputs	DREAMER: 9 DEAP: 8
		Decoder		
F1	FC1	Fully connected (O15)	256 units	256
F2	FC2	Fully connected (F1)	512 units	512
F3	FC3	Fully connected (F2)	DREAMER: 14 × 128 units DEAP: 32 × 128 units	DREAMER: 14 × 128 DEAP: 32 × 128

4.4. Comparison Methods

We compared IL2FS with eight popular and recent CIL methods based on memory replay: Fine-tuning (FT) [51], Fine-tuning+Nearest Centroid Classifier (FT+NCC) [41,51], Less without Forgetting (LwF) [49], Incremental Classifier and Representation Learning (iCARL) [32], Mnemonics [53], ScaIL [51], Weighting Aligning (WA) [36], and Geodesic+LUCIR [25]. We selected such CIL methods in our comparison since they arise as promising solutions to address the catastrophic forgetting problem in emotion recognition. All comparison methods were downloaded from repositories of original authors and then adapted for our experiments, except FT and FT+NCC, which do not represent a challenge to implement as they are basic methods. Note that all CIL methods use the same preprocessing procedure and the CapsNet architecture described in the previous sections.

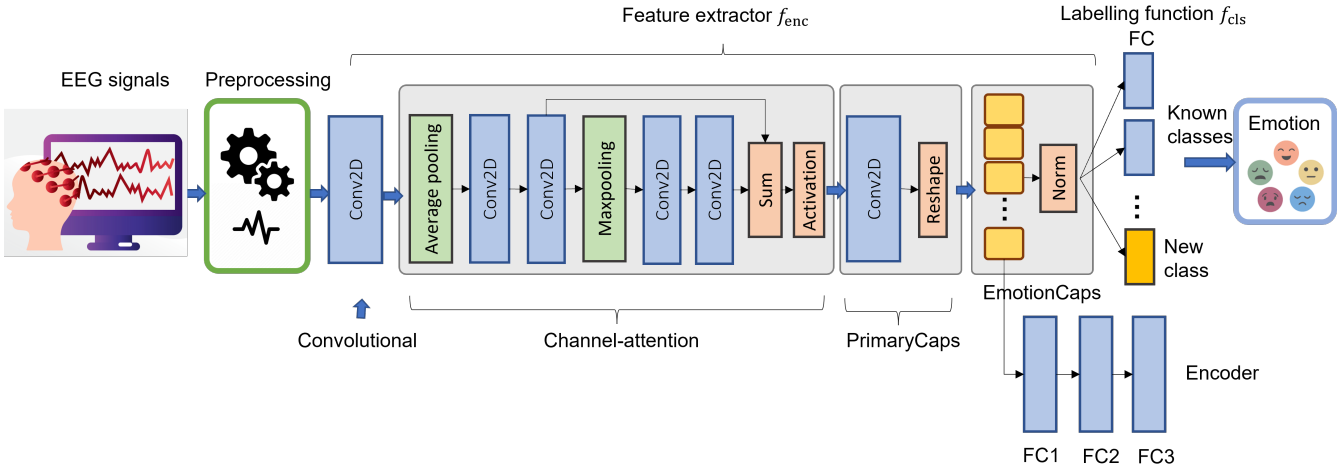


Figure 2. Diagram of the Capsule Network architecture.

4.5. Implementation Details

We first configured the hyper-parameters for the classification loss of the CapsNet architecture. Thus, the margins m^+ and m^- for the separated margin loss $\mathcal{L}_{\text{cls}}$ were set to 0.9 and 0.1, as suggested in [21,63]. Likewise, the reconstruction loss $\mathcal{L}_{\text{rec}}$ was scaled by 0.3 during training; this value was selected from {0.01, 0.1, 0.2, 0.3, and 0.5}.

Concerning the specific configuration of our proposed method, we adopted a mean layer instead of a normalization layer (N14) in the CapsNet architecture. For $\mathcal{L}_{\text{tri}}$, we used a margin $a = 0.1$ since a feature space alignment is pursued between extracted features from a reference network model and a new network model; a larger margin showed to affect the classification results negatively. To ensure an inter-class separation via margin ranking loss $\mathcal{L}_{\text{mr}}$, we used a margin b equal to 5, which was selected from {1, 3, 5, 8, and 10}. Finally, for trade-off hyper-parameters, we used $\alpha = 1$ and $\beta = 0.1$, which were selected from {0.01, 0.1, 1, and 2}. We use a momentum $\lambda_{\text{EMA}} = 0.995$ to place a greater significance on the most recent values.

Table 2 describes the specific hyper-parameters of CIL methods used in our comparison. Similar to our proposal, all hyperparameters were selected via grid search in combination with coordinated descent [64] in order to ensure the best configuration. Specifically, we select a small finite list of values for each hyper-parameter and each value is changed at a time while the rest of the hyper-parameters remains fixed.

Table 2. Hyper-parameter setting. a,b are margins; λ, λ_o, λ_{dis}, λ_{mr}, are the trade-off hyperparameters for each CIL method.

Methods	Hyper-Parameters
FT, FT+NCC, ScaIL	-
LwF, iCARL	$T = 2, \lambda_o = 1$
Mnemonics	$\lambda_{\text{dis}} = 0.5, \lambda_{\text{mr}} = 1, b = 5$
WA	$T = 2, \lambda = 0.4$
Geodesic+LUCIR	$\lambda_{\text{dis}} = 1, \lambda_{\text{mr}} = 1, b = 5$

Regarding the training algorithm of the CIL methods, we used Adam optimizer employing a mini-batch size of 10; a larger size showed to reduce the classification results using an incremental learning evaluation. For DREAMER, we set a learning rate of 0.001 up to epoch 30, when it decays to 0.0001, keeping this value until epoch 50 when the training concludes. For DEAP, we set an initial learning rate of 0.001 up to epoch 15, when it also decays by a factor of 10, and then holds this value until the end of epoch 20. Other learning rates (0.1, 0.01, 0.001, 0.0001) were evaluated, but they did not improve

the accuracy performance. An L1 regularizer was incorporated to CapsNet with a weight decay of 0.0004 for the Adam algorithm.

Our proposal and the comparison methods were implemented with PyTorch and trained on an Intel(R) Core (TM) i7 PC with an Nvidia GTX 1080 graphics card and Ubuntu v20.04 LTS.

4.6. Evaluation

As reported in [22,32], we follow the standard evaluation protocol used for the CIL setting based on the memory replay approach. The Holdout method is applied for a given dataset to build the training and testing data for each available class. Likewise, classes are arranged in a fixed random order. Each method is trained in a class-incremental way on available training data, as described in Section 3.1. At the end of each incremental stage, the resulting classifier is evaluated on testing data for already trained classes. Note that the testing dataset is not revealed to the CIL methods during training in each incremental stage to avoid overfitting. At the end of S incremental stages, we obtain S classification accuracies, averaged and reported as the final result.

We adopted an instantiation of the above protocol for each subject's data on the DREAMER and DEAP datasets, considering the most challenging scenario possible. Firstly, we start from a model trained on two classes, while remaining classes in DREAMER and DEAP come in 7 and at most six incremental stages, respectively. Secondly, we set the memory size $\mathcal{M}$ to approximately 1% of the full training set from each subject in order to store representative samples from old classes. We used 90% of the data of each class for training, while the rest of the data was used for testing. Thus, about ten samples can be stored in memory for DREAMER through 7 incremental stages, while at least 28 samples can be stored for DEAP during six stages. Note that not all subjects in DEAP rate the same levels of arousal, valence and dominance, producing an imbalanced dataset; an oversampling was applied using a random selection. Classes from incremental stages are arranged in sequence with a fixed random order. We performed five repetitions with different partitions of data and different classes, using different random seeds; a stratified sampling was performed with respect to the classes. From accuracy results by training in a class-incremental way, we compute the average and standard deviation over the incremental stages as final results. We assumed that training and testing datasets are independent and identically distributed, i.e., both datasets were drawn from the same distribution. Thus, we did not consider any change of distribution.

5. Results

Table 3 shows the average accuracy and standard deviation for all methods over DREAMER. We observed that IL2FS achieved the best average accuracy (59.08%) with one of the lowest standard deviations (8.26). Notice that IL2FS outperformed the second-best method (Mnemonics) by 8.96 percentage points (pp). Statistical differences were computed among the evaluated methods on the average accuracy of the 23 subjects. The Friedman test was applied, followed by Wilcoxon signed-rank as post hoc with the Finner correction. Friedman's test showed significant differences among CIL methods ($\chi^2(8) = 169.99$, $p = 0.0$). The Wilcoxon test indicated that the difference between IL2FS and CIL methods was statistically significant ($p < 0.05$).

Table 4 shows the average accuracy and standard deviation for all methods on DEAP. We can see that IL2FS achieved the best average accuracy (79.36%) with the lowest standard deviation (4.68). The second-best method was Geodesic+LUCIR, obtaining an average accuracy of 8.94 percentage points shorter than IL2FS. Friedman's test indicated significant differences among compared methods ($\chi^2(8) = 230.03$, $p = 0.0$). Wilcoxon signed-rank test revealed that differences between IL2FS and CIL methods are statistically significant ($p < 0.05$) for the given dataset.

Table 3. Accuracy and standard deviation for CIL methods on the DREAMER dataset using approximately 1% of the training size. The best results are in bold.

Subj.	FT	FT+NCC	LwF	iCARL	Mnemonics	ScaIL	WA	Geodesic +LUCIR	IL2FS
1	44.08 ± 13.86	48.63 ± 13.37	44.78 ± 10.25	48.26 ± 09.95	51.94 ± 14.00	40.09 ± 11.21	47.65 ± 14.92	53.29 ± 16.29	**64.58** ± 12.02
2	35.82 ± 13.87	38.31 ± 14.31	37.27 ± 15.84	39.95 ± 16.19	47.86 ± 17.64	30.56 ± 11.64	39.83 ± 19.94	42.54 ± 14.50	**53.64** ± 11.78
3	32.51 ± 11.59	34.93 ± 09.87	33.32 ± 12.91	37.31 ± 13.34	42.54 ± 12.50	29.96 ± 11.96	35.33 ± 14.45	44.37 ± 14.46	**54.43** ± 11.12
4	53.96 ± 11.88	56.55 ± 11.41	54.23 ± 14.23	56.84 ± 14.47	60.72 ± 15.30	40.09 ± 08.64	55.13 ± 16.93	60.16 ± 13.10	**64.08** ± 09.19
5	38.76 ± 14.35	41.65 ± 13.52	35.65 ± 15.67	37.73 ± 16.52	51.42 ± 16.95	29.24 ± 10.62	39.87 ± 17.65	51.13 ± 17.09	**60.74** ± 12.52
6	35.80 ± 13.39	38.47 ± 13.38	36.38 ± 15.31	40.12 ± 15.06	44.35 ± 14.42	34.98 ± 09.59	37.01 ± 16.35	42.08 ± 12.74	**53.31** ± 12.94
7	32.77 ± 11.33	35.89 ± 11.14	31.71 ± 11.98	35.87 ± 13.15	42.01 ± 13.49	27.76 ± 08.13	34.81 ± 14.29	45.17 ± 14.06	**56.34** ± 10.35
8	33.77 ± 11.41	36.90 ± 11.63	31.03 ± 11.29	33.39 ± 11.84	41.25 ± 12.33	27.18 ± 10.71	34.57 ± 12.06	43.31 ± 13.65	**49.98** ± 10.66
9	28.69 ± 08.12	32.11 ± 09.09	28.75 ± 11.89	32.67 ± 11.03	37.97 ± 13.07	25.95 ± 09.10	30.72 ± 14.50	39.73 ± 13.10	**49.07** ± 12.59
10	35.62 ± 13.81	38.06 ± 14.24	36.46 ± 15.85	38.56 ± 15.54	40.70 ± 13.61	28.70 ± 08.46	38.47 ± 13.67	40.54 ± 12.53	**51.48** ± 11.21
11	35.91 ± 10.19	39.53 ± 10.20	33.02 ± 13.70	36.61 ± 12.86	41.62 ± 12.32	34.08 ± 13.04	36.83 ± 12.10	46.29 ± 13.24	**56.08** ± 14.38
12	45.95 ± 09.95	50.54 ± 10.42	46.08 ± 12.05	49.81 ± 12.10	56.21 ± 14.50	37.23 ± 09.55	51.48 ± 15.69	57.70 ± 13.65	**66.81** ± 09.14
13	46.41 ± 15.88	48.24 ± 16.25	45.25 ± 16.26	48.39 ± 15.22	57.09 ± 14.84	39.42 ± 09.57	52.48 ± 16.78	55.51 ± 15.10	**68.07** ± 11.11
14	45.27 ± 12.84	48.52 ± 11.64	46.56 ± 18.02	49.50 ± 17.38	54.18 ± 16.59	38.57 ± 14.70	50.10 ± 21.21	51.96 ± 17.24	**60.94** ± 13.67
15	64.27 ± 11.47	66.17 ± 11.13	66.48 ± 10.64	69.18 ± 09.85	72.71 ± 12.81	56.49 ± 09.08	69.19 ± 11.46	71.30 ± 11.16	**81.16** ± 07.11
16	34.37 ± 14.03	37.87 ± 13.67	32.26 ± 11.43	35.40 ± 09.38	42.87 ± 12.67	29.04 ± 08.59	35.41 ± 14.72	43.04 ± 15.06	**50.80** ± 12.11
17	37.99 ± 12.40	41.80 ± 11.50	38.42 ± 12.16	42.11 ± 11.88	45.49 ± 14.32	31.50 ± 11.78	40.99 ± 13.49	46.73 ± 13.16	**53.91** ± 09.47
18	50.14 ± 16.92	52.06 ± 17.31	48.56 ± 15.28	51.88 ± 15.45	57.97 ± 17.41	41.17 ± 13.13	52.66 ± 17.81	56.76 ± 15.63	**62.78** ± 11.35
19	57.42 ± 12.95	61.90 ± 11.30	56.79 ± 15.52	59.20 ± 15.40	65.67 ± 11.72	50.43 ± 10.23	61.13 ± 16.25	66.94 ± 10.98	**71.56** ± 09.73
20	27.09 ± 13.44	28.79 ± 12.87	29.47 ± 15.63	31.08 ± 14.62	35.81 ± 13.15	23.91 ± 09.11	32.62 ± 15.21	35.83 ± 13.98	**49.75** ± 14.18
21	41.34 ± 10.87	43.63 ± 10.44	39.95 ± 13.04	42.68 ± 12.74	48.10 ± 12.20	31.82 ± 07.02	42.34 ± 15.87	48.22 ± 10.99	**54.25** ± 12.23
22	57.39 ± 12.71	61.14 ± 11.34	55.15 ± 14.11	57.58 ± 13.71	66.89 ± 10.40	47.36 ± 08.72	57.42 ± 16.16	64.07 ± 10.88	**68.49** ± 13.69
23	43.66 ± 15.63	45.99 ± 14.33	38.93 ± 16.62	41.39 ± 16.67	47.38 ± 16.08	32.77 ± 10.21	40.98 ± 17.76	45.17 ± 16.29	**56.66** ± 12.17
Avg.	41.69 ± 09.80	44.68 ± 09.91	41.15 ± 09.85	44.15 ± 09.85	50.12 ± 09.31	35.14 ± 08.20	44.22 ± 10.22	50.08 ± 09.31	**59.08** ± 08.26

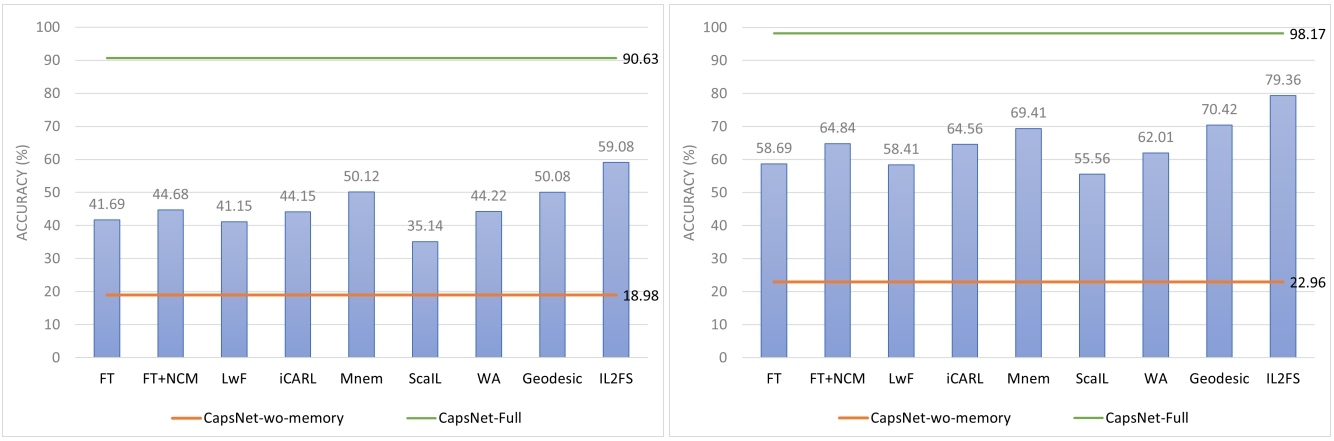
Table 4. Accuracy and standard deviation for CIL methods on the DEAP dataset using approximately 1% of the training size. The best results are in bold.

Subj.	FT	FT+NCC	LwF	iCARL	Mnemonics	ScaIL	WA	Geodesic +LUCIR	IL2FS
1	57.65 ± 13.84	64.62 ± 11.95	55.14 ± 12.12	62.57 ± 12.48	74.60 ± 17.04	53.14 ± 11.94	61.80 ± 17.93	75.06 ± 13.42	**86.47** ± 06.85
2	56.83 ± 16.68	61.35 ± 15.75	56.30 ± 18.12	58.82 ± 18.08	64.09 ± 17.77	54.10 ± 05.81	59.24 ± 19.92	62.53 ± 16.90	**74.33** ± 10.94
3	56.23 ± 09.27	60.40 ± 08.77	59.29 ± 12.84	63.56 ± 12.33	63.78 ± 15.90	52.40 ± 06.23	62.50 ± 12.58	67.24 ± 14.66	**74.31** ± 09.64
4	58.44 ± 15.77	63.74 ± 15.60	60.47 ± 15.14	65.21 ± 16.46	62.51 ± 18.69	57.31 ± 03.44	60.78 ± 17.60	69.60 ± 14.88	**75.91** ± 12.26
5	57.34 ± 13.39	63.42 ± 12.05	55.10 ± 11.90	59.98 ± 11.68	66.71 ± 17.31	51.64 ± 09.14	60.28 ± 14.03	69.12 ± 12.86	**78.55** ± 08.14
6	55.84 ± 14.10	63.04 ± 12.13	56.09 ± 15.08	63.04 ± 14.71	72.35 ± 15.76	51.28 ± 08.98	60.66 ± 17.94	72.20 ± 16.01	**83.16** ± 06.61
7	64.44 ± 16.04	70.54 ± 14.46	62.59 ± 17.69	69.65 ± 14.68	75.40 ± 15.97	59.38 ± 04.19	67.31 ± 19.10	79.58 ± 14.97	**85.28** ± 06.72
8	60.46 ± 14.56	67.00 ± 12.39	60.82 ± 14.35	66.61 ± 13.25	70.96 ± 16.66	53.71 ± 03.99	64.64 ± 15.48	71.79 ± 12.41	**82.59** ± 06.95
9	52.65 ± 15.77	59.46 ± 16.01	52.85 ± 15.62	59.02 ± 15.60	65.73 ± 17.89	48.87 ± 04.24	55.72 ± 16.74	64.45 ± 13.76	**78.22** ± 10.15
10	72.54 ± 11.89	78.08 ± 10.06	70.43 ± 11.95	74.33 ± 11.30	72.70 ± 13.22	62.91 ± 04.42	74.19 ± 12.94	82.04 ± 09.11	**85.71** ± 05.15
11	52.94 ± 19.03	59.58 ± 17.35	55.42 ± 18.49	63.22 ± 18.36	64.73 ± 21.20	52.12 ± 07.23	57.57 ± 20.20	66.36 ± 19.73	**77.96** ± 10.11
12	68.06 ± 16.98	72.76 ± 15.16	68.35 ± 16.77	74.88 ± 15.33	71.02 ± 17.65	69.16 ± 09.76	70.10 ± 15.46	76.61 ± 14.52	**79.11** ± 11.29
13	46.32 ± 12.19	52.48 ± 11.53	45.36 ± 13.13	50.33 ± 12.75	64.25 ± 16.59	38.55 ± 02.58	48.49 ± 17.13	59.71 ± 15.87	**74.74** ± 12.02
14	51.04 ± 15.11	58.49 ± 13.71	49.41 ± 16.74	55.65 ± 16.96	64.15 ± 14.57	45.08 ± 03.87	51.73 ± 15.93	63.48 ± 15.68	**75.37** ± 10.65
15	51.02 ± 13.96	57.96 ± 13.57	53.10 ± 15.41	59.35 ± 15.51	71.36 ± 15.69	53.49 ± 04.22	54.30 ± 16.34	63.75 ± 15.15	**80.50** ± 08.37
16	61.86 ± 09.81	67.57 ± 09.20	62.10 ± 09.96	67.35 ± 09.11	61.78 ± 19.88	54.16 ± 05.12	63.79 ± 11.46	70.24 ± 13.35	**82.10** ± 10.33
17	49.18 ± 10.97	55.58 ± 11.15	49.45 ± 13.83	55.88 ± 12.93	61.75 ± 16.90	42.52 ± 04.61	50.79 ± 13.57	62.45 ± 12.58	**68.84** ± 10.02
18	63.45 ± 09.97	69.54 ± 09.27	61.68 ± 08.82	67.09 ± 08.02	76.78 ± 15.31	66.03 ± 07.85	67.34 ± 14.30	74.56 ± 13.68	**79.82** ± 06.34
19	61.31 ± 12.87	66.52 ± 12.27	62.42 ± 13.45	67.94 ± 10.48	71.57 ± 17.28	59.35 ± 04.81	66.83 ± 13.87	71.54 ± 16.72	**82.06** ± 08.84
20	68.44 ± 15.69	73.53 ± 13.52	66.13 ± 17.82	70.82 ± 15.53	67.01 ± 19.20	61.15 ± 05.18	70.73 ± 16.44	74.44 ± 15.08	**82.96** ± 09.52
21	69.70 ± 14.88	74.58 ± 13.07	69.10 ± 13.62	73.16 ± 13.39	70.00 ± 21.43	67.61 ± 07.90	72.44 ± 16.90	77.39 ± 15.35	**79.11** ± 10.35
22	66.38 ± 09.98	73.02 ± 10.07	64.10 ± 12.25	69.00 ± 11.38	70.31 ± 13.68	58.79 ± 06.31	69.03 ± 14.77	72.83 ± 09.96	**80.14** ± 08.90
23	47.61 ± 12.34	55.60 ± 11.60	47.39 ± 09.88	56.89 ± 12.95	77.79 ± 15.13	48.54 ± 05.05	55.51 ± 17.85	69.97 ± 09.84	**83.30** ± 06.33
24	66.44 ± 15.95	71.95 ± 13.35	68.02 ± 18.04	73.78 ± 16.02	72.32 ± 18.86	66.30 ± 09.06	70.66 ± 18.62	74.67 ± 15.46	**84.28** ± 07.59

Table 4. *Cont.*

Subj.	FT	FT+NCC	LwF	iCARL	Mnemonics	ScaIL	WA	Geodesic +LUCIR	IL2FS
25	49.94 ± 12.30	57.68 ± 10.63	52.73 ± 10.53	60.48 ± 09.41	63.30 ± 16.39	47.68 ± 04.95	52.64 ± 13.20	62.91 ± 12.41	**76.20** ± 10.29
26	48.79 ± 08.93	54.62 ± 09.74	47.65 ± 08.71	52.85 ± 08.86	60.39 ± 15.62	48.20 ± 08.45	51.76 ± 09.78	59.99 ± 11.23	**66.78** ± 10.25
27	54.70 ± 11.18	64.47 ± 09.56	49.45 ± 07.82	66.25 ± 08.30	80.64 ± 12.15	54.49 ± 10.94	60.36 ± 15.18	74.16 ± 11.72	**80.90** ± 06.87
28	57.35 ± 10.50	64.41 ± 09.25	57.00 ± 11.46	64.69 ± 10.14	69.88 ± 14.89	55.19 ± 07.71	60.71 ± 16.05	71.67 ± 11.63	**77.46** ± 09.02
29	73.05 ± 07.99	77.94 ± 06.10	72.27 ± 07.88	77.25 ± 08.10	78.84 ± 11.28	73.64 ± 08.85	74.40 ± 08.67	82.36 ± 09.38	**85.87** ± 05.34
30	69.05 ± 13.69	74.11 ± 11.48	69.88 ± 13.77	74.64 ± 12.90	75.45 ± 15.11	67.46 ± 03.91	72.50 ± 12.96	80.69 ± 11.56	**83.45** ± 08.12
31	61.55 ± 14.27	66.49 ± 13.00	62.38 ± 12.17	66.14 ± 11.69	67.71 ± 17.99	57.06 ± 04.07	64.61 ± 14.79	69.30 ± 13.98	**77.63** ± 09.48
32	47.53 ± 09.34	54.37 ± 09.16	46.70 ± 10.90	55.44 ± 12.88	71.31 ± 14.29	46.57 ± 06.33	50.94 ± 15.02	60.83 ± 10.50	**76.49** ± 09.43
Avg.	58.69 ± 07.83	64.84 ± 07.24	58.41 ± 07.76	64.56 ± 07.00	69.41 ± 05.48	55.56 ± 08.20	62.01 ± 07.60	70.42 ± 06.50	**79.36** ± 04.68

Comparison with baseline. Figure 3 presents a comparison of IL2FS and existing CIL methods concerning the baseline approach (CapsNet-wo-memory), that is, when CapsNet did not include any data from old classes in a CIL training. In addition, we also included the average accuracy when CapsNet is trained using all training samples from old classes (CapsNet-Full) in each incremental stage. We observed that CapsNet-wo-memory obtained the worst accuracy results when samples of old classes are not available in the memory, suggesting the presence of catastrophic forgetting. However, CapsNet improved its accuracy performance when samples of old classes were employed during Fine-tuning (FT). Note that IL2FS and advanced CIL methods improved the average accuracy of FT by incorporating a specific strategy to address the catastrophic forgetting problem, except ScaIL and LwF. Finally, we observed that IL2FS is still exposed to catastrophic forgetting as CapsNet-Full achieved 90.63% and 98.17% on DEAP and DREAMER.



(**a**) (**b**)

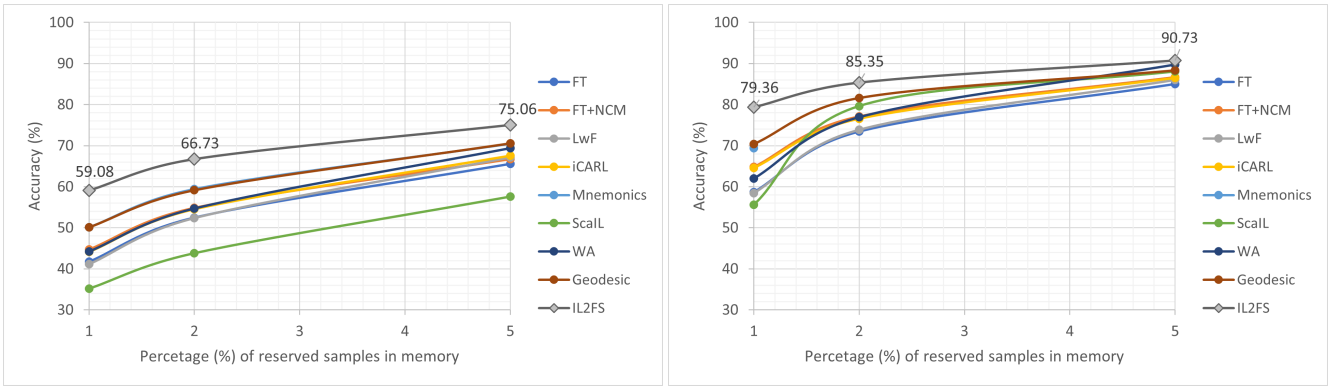
Figure 3. Comparison of CIL methods with baseline approaches on (**a**) DREAMER and (**b**) DEAP. CapsNet-wo-memory represents the average accuracy obtained by CapsNet when data from old classes are not included in the memory during the CIL training. CapsNet-Full indicates the average accuracy using all training samples from old classes in each incremental stage. Mnem and Geo+LUC indicate Mnemonics and Geodesic+LUCIR, respectively.

5.1. Ablation Studies

In this section, we present an analysis with respect to the number of reserved samples from old classes. After, we study the impact of the number of new emotions incorporated into the neural network model. Finally, we analyze the impact of each component of IL2FS.

5.1.1. Effect of the Number of Reserved Samples

Figure 4 shows the comparison of IL2FS with CIL methods, when the memory of old samples has a size close to 1%, 2%, and 5% of the size of the full training set for each subject on DREAMER and DEAP. As expected, CIL methods improved their accuracy performance when more samples were stored in the memory. However, we can see that IL2FS still maintains the best average accuracy for different sizes of the reserved samples. For DREAMER, 66.73% and 75.06% were obtained by IL2FS when the memory is close to 2% and 5% of the size of the full training set. For DEAP, IL2FS achieved average accuracies of 85.35% and 90.73% using memory sizes of 2% and 5%, respectively. Note that our proposal obtained a greater gain in average accuracy than the comparison methods when a smaller number of samples from old classes is reserved in the memory.

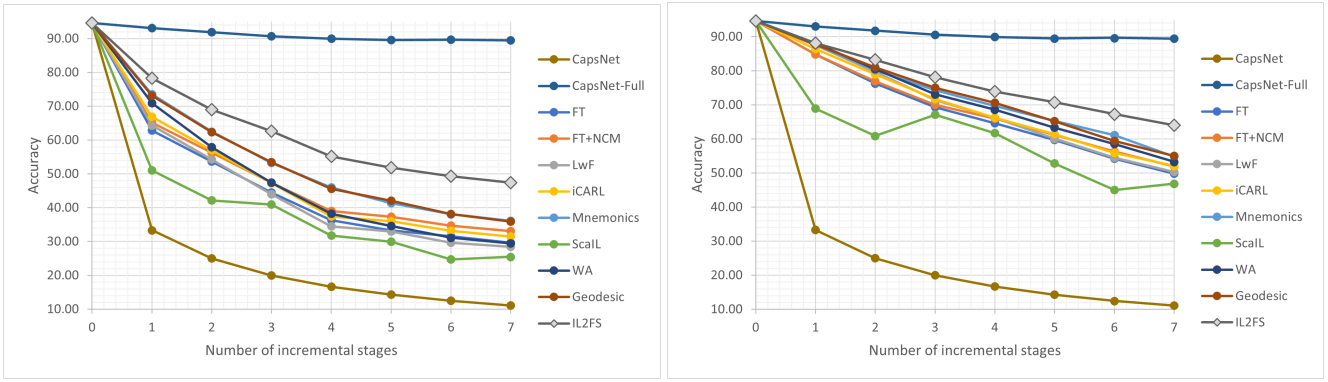


(**a**) (**b**)

Figure 4. Effect of the number of reserved samples in memory: (**a**) DREAMER and (**b**) DEAP.

5.1.2. Effect of the Number of Incremental Stages

Figure 5 shows the average accuracy of IL2FS for each incremental stage in comparison to CIL methods on DREAMER. We reported the average accuracy of CIL methods over all subjects, employing memory sizes of 1% and 5% of the size of the full training dataset. Accuracy results for CapsNet-wo-memory and CapsNet-Full were also included as baselines. We observed that a CIL strategy helps reduce catastrophic forgetting by improving the accuracy performance of CapsNet-wo-memory. However, note that CIL methods decrease their accuracy performance when the number of stages is increased. It is worth mentioning that IL2FS achieved the best average accuracies throughout different incremental stages. In addition, IL2FS obtained a greater gain than existing methods during the last incremental stages because fewer samples from old classes can be stored in the memory.



(**a**) (**b**)

Figure 5. Effect of the number of incremental stages on the DREAMER dataset, using memory sizes of (**a**) 1% and (**b**) 5% of the full training dataset of each subject. We reported average accuracy in each incremental stage over all subjects.

5.1.3. Effect of Each Component of IL2FS

The proposed method comprises three main strategies: weight aligning for bias correction, margin ranking loss for inter-class separation and triplet loss for a feature space alignment of old classes. Table 5 shows the average accuracy over all subjects for each evaluated dataset. Note that Fine-tuning achieved an average accuracy of 41.69% and 58.69% over DREAMER and DEAP, respectively. By using weight aligning for bias correction, the average accuracy is improved by 4.86 and 3.99 percentage points over the DREAMER and DEAP datasets. A variant of margin ranking loss was incorporated to encourage a separation between each ground-truth old class and its nearest class (old or new). This modification allowed IL2FS to outperform Fine-tuning+Weight aligning by 10.22 and 15.33 percentage points over DREAMER and DEAP, respectively. In addition, triplet loss was used to keep a similar alignment of the feature space of old classes. From experiments, we found that by encouraging such alignment, an improvement of 2.31 and 1.35 percentage points is observed in average accuracy on DREAMER and DEAP, respectively.

Table 5. Effect of each component of IL2FS on DREAMER and DEAP. The best results are in bold.

Method	DREAMER	DEAP
Fine-tuning (FT)	41.69 ± 09.80	58.69 ± 07.83
FT+Weight Aligning	46.55 ± 09.01	62.68 ± 07.29
FT+Weight Aligning+Margin ranking loss	56.77 ± 08.47	78.01 ± 04.25
FT+Weight Aligning+Margin ranking loss+Triplet loss (IL2FS)	**59.08** ± 08.26	**79.36** ± 04.68

6. Discussion

Experiments showed that a standard deep learning model for emotion recognition (CapsNet) degrades its accuracy performance when trained in a class-incremental way over only samples from new emotions. This problem, known as catastrophic forgetting, is presented because previously learned emotions are negatively affected when new ones are incorporated into the classifier model. Thus, unlike previous works as reported in [3,8,13–16,20,21], this research is focused on studying the catastrophic forgetting problem in EEG-based emotion recognition.

By incorporating existing CIL methods to CapsNet, we showed that classification results of the baseline approach (CapsNet-wo-memory) can be improved, suggesting that CIL methods can help mitigate the catastrophic forgetting in EEG-based emotion recognition. However, experimental results on two public datasets showed that existing CIL methods do not ensure high average accuracies. Thus, a CIL method was developed and validated to address the catastrophic forgetting problem.

Previously, Lee et al. [41] studied the CIL over the imagined speech task from EEG signals. Authors used fine-tuning and the nearest neighbor classifier to address the catastrophic forgetting, however, they stored 20% of the full data of every old class in each incremental stage. Furthermore, only one incremental stage was used for CIL evaluation, while more stages are needed to observe the negative impact of catastrophic forgetting. On the other hand, our experiments consider a rigorous evaluation over two datasets for emotion recognition, including popular and recent CIL methods in our comparison. Based on our results, we found that IL2FS outperformed existing CIL methods on two public datasets: DREAMER and DEAP. Note that we integrated a weighting aligning as the WA method for bias correction, but an inter-class separation and a feature space alignment were also considered by IL2FS, outperforming WA by 14.28 pp and 17.35 pp on DREAMER and DEAP, respectively. Like IL2FS, the Mnemonics and Geodesic+LUCIR methods ensure an inter-class separation via margin ranking loss, but IL2FS encourages the separation between old classes and their nearest one, including old or new, instead of only ensuring a separation between old and new classes. Although Mnemonics and Geodesic+LUCIR also consider strategies for bias correction and an alignment of output predictions, our proposal outperformed Mnemonics by 8.96 pp and 9.95 pp on DREAMER and DEAP, while Geodesic+LUCIR was outperformed by 9 pp and 8.94 pp, respectively. In addition, note

that comparison methods, such as LwF, iCARL, WA, Mnemonics, and Geodesic+LUCIR, use different strategies to align the output predictions of old classes to leverage the less forgetting. Unlike these works, IL2FS incorporates triplet loss to preserve the feature space alignment of old classes instead of the output predictions.

Regarding the evaluation which varies the number of reserved samples and the number of incremental stages, IL2FS showed a clear advantage compared to existing methods when the number of reserved samples in the memory is reduced. This issue is also observed when a greater number of CIL stages are achieved since a lower number of samples per class may be stored in memory. The above indicates that IL2FS preserves the learned knowledge better than compared methods throughout different incremental stages on the most challenging scenario possible for the evaluated datasets. On the other hand, as expected, every evaluated method improved its accuracy performance whenever the number of reserved samples in the memory is increased. However, by using a memory size near 5%, IL2FS still obtained the best average accuracy on the DREAMER, while it is similar with respect to the existing CIL methods for the DEAP dataset.

Concerning the effect of each component of IL2FS, weight aligning improved the average accuracy over the Fine-tuning method, which indicates that performing a bias correction is important to reduce the catastrophic forgetting problem in EEG-based emotion recognition. Then, margin ranking loss was incorporated to ensure an inter-class separation between each old class and its nearest class (old or new). Previous work in [33] showed that a separation between old and new classes might be sufficient to help reduce the catastrophic forgetting. However, we found that this strategy [33] on IL2FS obtained a similar average accuracy on DREAMER ($58.55 \pm 7.33\%$ vs. $58.74 \pm 07.56\%$), but the accuracy performance is drastically reduced on DEAP ($53.36 \pm 08.84\%$ vs. $79.36 \pm 04.68\%$). These results suggest that it is preferable to encourage an inter-class separation between each old class and its nearest class (old or new) instead of only ensuring a separation between old and new classes. Finally, unlike previous CIL works [42–44] where triplet loss is mainly used to train embedding networks and provide an inter-class separation, we used such loss function to maintain the same aligning of the feature space learned at previous incremental stages. For this, IL2FS aims to produce near feature representations coming from a reference model and a new model for the same processed sample, while features from different samples are pushed away from each other by a small margin. This strategy showed to be beneficial for the CIL task in two emotion recognition datasets.

The presented study may contribute to designing and building more adaptive and scalable classifiers, as our study showed a first Class-incremental Learning solution to avoid reconfiguring the entire system when new emotions are incorporated sequentially. For this, we consider an evaluation of the most challenging scenario that may be built over the two public datasets for emotion recognition. However, our study did not consider other CIL settings and evaluation protocols. Furthermore, other preprocessing procedures and neural network architectures were also not explored.

7. Conclusions

In this paper, we presented IL2FS, a CIL method to address the catastrophic forgetting in EEG-based emotion recognition from EEG signals. IL2FS aims to preserve the feature space learned over previous incremental stages, performing a bias correction of new classes and ensuring the inter-class separation and feature space alignment from classes learned at previous incremental stages. The proposed method was incorporated into a Capsule Network architecture for EEG-based emotion recognition. Our experiments showed that IL2FS achieved the best average accuracy over two public emotion datasets, outperforming popular and recent CIL methods under different memory sizes. Furthermore, Friedman and Wilcoxon's tests showed that IL2FS significantly outperformed existing CIL methods over the evaluated datasets, using the standard protocol for CIL methods based on memory replay. By using IL2FS, better preservation of the learned knowledge is possible when presented with a greater number of incremental stages and a reduced number of reserved samples in memory. In this direction, new emotions may be incorporated into an existing

deep neural network classifier without retraining from scratch, employing a set of representative samples of emotions previously learned in a sequential way. However, the presented results suggest that the proposed solution is still exposed to catastrophic forgetting for a high number of incremental stages and limited memory size.

As future work, we are interested in studying the negative effect of batch normalization layers since a bias may be produced over learned statistics from old classes by training over imbalanced data.

Author Contributions: Conceptualization, M.J.-G.; formal analysis, M.J.-G.; investigation, M.J.-G.; methodology, M.J.-G.; project administration, M.J.-G. and R.A.-E.; software, M.J.-G.; supervision, R.A.-E.; writing—original draft preparation, M.J.-G. and R.A.-E.; writing—review and editing, M.J.-G. and R.A.-E. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Mexican Council of Science and Technology (COMECYT), grant number CAT2021-0193.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BiC	Bias correction
CapsNet	Capsule Network
CIL	Class-incremental Learning
CNN	Convolutional Neural Network
DBN	Deep Belief Network
EEG	Electroencephalogram
FT	Fine-tuning
NCC	Nearest Centroid Classifier
GAN	Generative adversarial network
GNN	Graph Neural Network
iCARL	Incremental Classifier and Representation Learning
IL2FS	Incremental Learning preserving the Learned Feature Space
LI2M	Incremental Learning with Dual Memory
LUCIR	Learning a Unified Classifier Incrementally via Rebalancing
LwF	Learning without Forgetting
ScaIL	Classifier Weights Scaling for Class Incremental Learning
WA	Weight Aligning

References

1. Deng, J.J.; Leung, C.H.C.; Milani, A.; Chen, L. Emotional States Associated with Music: Classification, Prediction of Changes, and Consideration in Recommendation. *ACM Trans. Interact. Intell. Syst.* **2015**, *5*, 1–36.
2. Ali, M.; Mosa, A.H.; Al Machot, F.; Kyamakya, K. EEG-based emotion recognition approach for e-healthcare applications. In Proceedings of the 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), Vienna, Austria, 5–8 July 2016; pp. 946–950.
3. Huang, D.; Chen, S.; Liu, C.; Zheng, L.; Tian, Z.; Jiang, D. Differences first in asymmetric brain: A bi-hemisphere discrepancy convolutional neural network for EEG emotion recognition. *Neurocomputing* **2021**, *448*, 140–151.
4. Alarcão, S.M.; Fonseca, M.J. Emotions Recognition Using EEG Signals: A Survey. *IEEE Trans. Affect. Comput.* **2019**, *10*, 374–393.
5. Kim, J.H.; Poulose, A.; Han, D.S. The Extensive Usage of the Facial Image Threshing Machine for Facial Emotion Recognition Performance. *Sensors* **2021**, *21*, 2026.
6. Poulose, A.; Reddy, C.S.; Kim, J.H.; Han, D.S. Foreground Extraction Based Facial Emotion Recognition Using Deep Learning Xception Model. In Proceedings of the 2021 Twelfth International Conference on Ubiquitous and Future Networks (ICUFN), Barcelona, Spain, 5–8 July 2021; pp. 356–360.

7. Han, K.; Yu, D.; Tashev, I. Speech Emotion Recognition Using Deep Neural Network and Extreme Learning Machine. *Interspeech* **2014** ; pp. 223–227.
8. Song, T.; Liu, S.; Zheng, W.; Zong, Y.; Cui, Z. Instance-adaptive graph for EEG emotion recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 2–12 February 2020; Volume 34, pp. 2701–2708.
9. Agrafioti, F.; Hatzinakos, D.; Anderson, A.K. ECG pattern analysis for emotion detection. *IEEE Trans. Affect. Comput.* **2011**, *3*, 102–115.
10. Cheng, B.; Liu, G. Emotion Recognition from Surface EMG Signal Using Wavelet Transform and Neural Network. In Proceedings of the 2008 2nd International Conference on Bioinformatics and Biomedical Engineering, Shanghai, China, 16–18 May 2008; pp. 1363–1366, doi:10.1109/ICBBE.2008.670.
11. Koelstra, S.; Muhl, C.; Soleymani, M.; Lee, J.S.; Yazdani, A.; Ebrahimi, T.; Pun, T.; Nijholt, A.; Patras, I. DEAP: A Database for Emotion Analysis ;Using Physiological Signals. *IEEE Trans. Affect. Comput.* **2012**, *3*, 18–31.
12. Lan, Y.T.; Liu, W.; Lu, B.L. Multimodal Emotion Recognition Using Deep Generalized Canonical Correlation Analysis with an Attention Mechanism. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 19–24 July 2020; pp. 1–6.
13. Zhong, P.; Wang, D.; Miao, C. EEG-based emotion recognition using regularized graph neural networks. *IEEE Trans. Affect. Comput.* **2020**, 1–12 , doi:10.1109/taffc.2020.2994159.
14. Shen, L.; Zhao, W.; Shi, Y.; Qin, T.; Liu, B. Parallel Sequence-Channel Projection Convolutional Neural Network for EEG-Based Emotion Recognition. *IEEE Access* **2020**, *8*, 222966–222976.
15. Tao, W.; Li, C.; Song, R.; Cheng, J.; Liu, Y.; Wan, F.; Chen, X. EEG-based Emotion Recognition via Channel-wise Attention and Self Attention. *IEEE Trans. Affect. Comput.* **2020**, 1–12, doi:10.1109/TAFFC.2020.3025777.
16. Topic, A.; Russo, M. Emotion recognition based on EEG feature maps through deep learning network. *Eng. Sci. Technol. Int. J.* **2021**, *24*, 1442–1454.
17. Salankar, N.; Mishra, P.; Garg, L. Emotion recognition from EEG signals using empirical mode decomposition and second-order difference plot. *Biomed. Signal Process. Control* **2021**, *65*, 102389.
18. Shen, F.; Peng, Y.; Kong, W.; Dai, G. Multi-Scale Frequency Bands Ensemble Learning for EEG-Based Emotion Recognition. *Sensors* **2021**, *21*, 1262.
19. Xu, X.; Zhang, Y.; Tang, M.; Gu, H.; Yan, S.; Yang, J. Emotion Recognition Based on Double Tree Complex Wavelet Transform and Machine Learning in Internet of Things. *IEEE Access* **2019**, *7*, 154114–154120.
20. Chao, H.; Liu, Y. Emotion Recognition From Multi-Channel EEG Signals by Exploiting the Deep Belief-Conditional Random Field Framework. *IEEE Access* **2020**, *8*, 33002–33012.
21. Liu, Y.; Ding, Y.; Li, C.; Cheng, J.; Song, R.; Wan, F.; Chen, X. Multi-channel EEG-based emotion recognition via a multi-level features guided capsule network. *Comput. Biol. Med.* **2020**, *123*, 103927.
22. Belouadah, E.; Popescu, A.; Kanellos, I. A comprehensive study of class incremental learning algorithms for visual tasks. *Neural Netw.* **2021**, *135*, 38–54.
23. Geng, X.; Smith-Miles, K. *Incremental Learning*; Springer: Berlin/Heidelberg, Germany, 2009.
24. Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A.A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 3521–3526.
25. Simon, C.; Koniusz, P.; Harandi, M. On learning the geodesic path for incremental learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 1591–1600.
26. Liu, Y.; Schiele, B.; Sun, Q. Adaptive aggregation networks for class-incremental learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 2544–2553.
27. Yan, S.; Xie, J.; He, X. DER: Dynamically Expandable Representation for Class Incremental Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 3014–3023.
28. Hayes, T.L.; Kafle, K.; Shrestha, R.; Acharya, M.; Kanan, C. Remind your neural network to prevent catastrophic forgetting. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 466–483.
29. Liu, Y.; Parisot, S.; Slabaugh, G.; Jia, X.; Leonardis, A.; Tuytelaars, T. More classifiers, less forgetting: A generic multi-classifier paradigm for incremental learning. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 699–716.
30. Rajasegaran, J.; Khan, S.; Hayat, M.; Khan, F.S.; Shah, M. Itaml: An incremental task-agnostic meta-learning approach. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 13588–13597.
31. Hu, W.; Qin, Q.; Wang, M.; Ma, J.; Liu, B. Continual Learning by Using Information of Each Class Holistically. In Proceedings of the AAAI Conference on Artificial Intelligence, 2–9 February 2021; Volume 35, pp. 7797–7805. Available online: https://ojs.aaai.org/index.php/AAAI/article/view/16952 (accessed on 12 January 2022).
32. Rebuffi, S.A.; Kolesnikov, A.; Sperl, G.; Lampert, C.H. Icarl: Incremental classifier and representation learning. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2001–2010.
33. Hou, S.; Pan, X.; Loy, C.C.; Wang, Z.; Lin, D. Learning a unified classifier incrementally via rebalancing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–17 June 2019; pp. 831–839.
34. Iscen, A.; Zhang, J.; Lazebnik, S.; Schmid, C. Memory-efficient incremental learning through feature adaptation. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 699–715.

35. Prabhu, A.; Torr, P.H.; Dokania, P.K. Gdumb: A simple approach that questions our progress in continual learning. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 524–540.
36. Zhao, B.; Xiao, X.; Gan, G.; Zhang, B.; Xia, S.T. Maintaining discrimination and fairness in class incremental learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13208–13217.
37. Chaudhry, A.; Gordo, A.; Dokania, P.K.; Torr, P.H.S.; Lopez-Paz, D. Using Hindsight to Anchor Past Knowledge in Continual Learning. In Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, 2–9 February 2021; pp. 6993–7001. Available online: https://ojs.aaai.org/index.php/AAAI/article/view/16861 (accessed on 12 January 2022).
38. Bang, J.; Kim, H.; Yoo, Y.; Ha, J.W.; Choi, J. Rainbow Memory: Continual Learning with a Memory of Diverse Samples. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 8218–8227.
39. Kurmi, V.K.; Patro, B.N.; Subramanian, V.K.; Namboodiri, V.P. Do not Forget to Attend to Uncertainty while Mitigating Catastrophic Forgetting. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikola, HI, USA, 5–9 January 2021; pp. 736–745.
40. Tang, S.; Chen, D.; Zhu, J.; Yu, S.; Ouyang, W. Layerwise optimization by gradient decomposition for continual learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 9634–9643.
41. Lee, D.Y.; Lee, M.; Lee, S.W. Decoding Imagined Speech Based on Deep Metric Learning for Intuitive BCI Communication. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2021**, *29*, 1363–1374.
42. Huo, J.; Zyl, T.L.v. Comparative Analysis of Catastrophic Forgetting in Metric Learning. In Proceedings of the 2020 7th International Conference on Soft Computing Machine Intelligence (ISCMI), Stockholm, Sweden, 14–15 November 2020; pp. 68–72, doi:10.1109/ISCMI51676.2020.9311580.
43. Yu, L.; Twardowski, B.; Liu, X.; Herranz, L.; Wang, K.; Cheng, Y.; Jui, S.; Weijer, J.V.d. Semantic drift compensation for class-incremental learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 6982–6991.
44. Zhao, H.; Fu, Y.; Kang, M.; Tian, Q.; Wu, F.; Li, X. Mgsvf: Multi-grained slow vs. fast framework for few-shot class-incremental learning. *arXiv* **2021**, arXiv:2006.15524.
45. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 815–823.
46. Katsigiannis, S.; Ramzan, N. DREAMER: A Database for Emotion Recognition Through EEG and ECG Signals From Wireless Low-cost Off-the-Shelf Devices. *IEEE J. Biomed. Health Inform.* **2018**, *22*, 98–107.
47. Belouadah, E.; Popescu, A. Il2m: Class incremental learning with dual memory. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 583–592.
48. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
49. Li, Z.; Hoiem, D. Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 2935–2947.
50. Wu, Y.; Chen, Y.; Wang, L.; Ye, Y.; Liu, Z.; Guo, Y.; Fu, Y. Large scale incremental learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–17 June 2019; pp. 374–382.
51. Belouadah, E.; Popescu, A. ScaIL: Classifier Weights Scaling for Class Incremental Learning. In Proceedings of the 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass Village, CO, USA, 1–5 March 2020.
52. Welling, M. Herding Dynamic Weights for Partially Observed Random Field Models. In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2009, Montreal, QC, Canada, 18–21 June 2009; pp. 599–606.
53. Liu, Y.; Su, Y.; Liu, A.A.; Schiele, B.; Sun, Q. Mnemonics training: Multi-class incremental learning without forgetting. In Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 12245–12254.
54. Wu, C.; Herranz, L.; Liu, X.; van de Weijer, J.; Raducanu, B. Memory replay gans: Learning to generate new categories without forgetting. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 5962–5972.
55. van de Ven, G.M.; Siegelmann, H.T.; Tolias, A.S. Brain-inspired replay for continual learning with artificial neural networks. *Nat. Commun.* **2020**, *11*, 4069.
56. Tarvainen, A.; Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In Proceedings of the Advances in Neural Information Processing Systems 30, Long Beach, CA, USA, 4–9 December 2017; pp. 1195–1204.
57. Wang, X.; Han, X.; Huang, W.; Dong, D.; Scott, M.R. Multi-Similarity Loss with General Pair Weighting for Deep Metric Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–17 June 2019; pp. 5022–5030.
58. Mullen, T.R.; Kothe, C.A.; Chi, Y.M.; Ojeda, A.; Kerth, T.; Makeig, S.; Jung, T.P.; Cauwenberghs, G. Real-time neuroimaging and cognitive monitoring using wearable dry EEG. *IEEE Trans. Biomed. Eng.* **2015**, *62*, 2553–2567.
59. Davidson, R.J. Affective neuroscience and psychophysiology: Toward a synthesis. *Psychophysiology* **2003**, *40*, 655–665.
60. Sanei, S.; Chambers, J.A. *EEG Signal Processing*; John Wiley & Sons: Hoboken, NJ, USA, 2013.
61. Yang, Y.; Wu, Q.; Fu, Y.; Chen, X. Continuous Convolutional Neural Network with 3D Input for EEG-Based Emotion Recognition. In *Information Processing*; Cheng, L., Leung, A.C.S., Ozawa, S., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 433–443.

62. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the Computer Vision–ECCV 2018, Munich, Germany, 8–14 September 2018; Springer International Publishing: Cham, Switzerland, 2018; pp. 3–19.
63. Sabour, S.; Frosst, N.; Hinton, G.E. Dynamic Routing Between Capsules. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2017; Volume 30.
64. Montavon, G.; Orr, G.B.; Müller, K. (Eds.) *Neural Networks: Tricks of the Trade*, 2nd ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7700.