MASIL: Towards Maximum Separable Class Representation for Few Shot Class Incremental Learning

Anant Khandelwal [1]

Abstract

Few Shot Class Incremental Learning (FSCIL) with few examples per class for each incremental session is the realistic setting of continual learning since obtaining large number of annotated samples is not feasible and cost effective. We present the framework MASIL as a step towards learning the maximal separable classifier. It addresses the common problem i.e forgetting of old classes and over-fitting to novel classes by learning the classifier weights to be maximally separable between classes forming a simplex Equiangular Tight Frame. We propose the idea of concept factorization explaining the collapsed features for base session classes in terms of concept basis and use these to induce classifier simplex for few shot classes. We further adds fine tuning to reduce any error occurred during factorization and train the classifier jointly on base and novel classes without retaining any base class samples in memory. Experimental results on miniImageNet, CIFAR-100 and CUB-200 demonstrate that MASIL outperforms all the benchmarks.

1. Introduction

The success of Convolutional Neural Networks (CNN) in wide range of computer vision tasks (Krizhevsky et al., 2012; He et al., 2016; Ren et al., 2015; Liu et al., 2017; Ma et al., 2019; Li et al., 2020) relies on the fact that the training requires large scale image datasets (Deng et al., 2009) and the train and test distributions are almost identical (Krizhevsky et al., 2017). However, when deploying them in real world environments it requires that these models to quickly adapt to changing streams of data and hence can recognize the novel classes emerged over a period of time. But the underlying bottleneck for this adaptation is that CNN requires large amount of data to be collected for each of the novel classes, this takes lot of human effort to annotate them which is infeasible. However, annotating only a few samples seems reasonable, we term this ability to adapt to novel classes (with only few examples) without forgetting the old classes as the *few shot class incremental learning* (FSCIL). Fine-tuning the pre-trained network with limited number of training examples of only novel classes cause the model to forget old classes (*catastrophic forgetting*) and overfitting on recent novel classes (French, 1999; Huang et al., 2022; Tao et al., 2020b; Dong et al., 2021). Large amount of studies has been conducted to solve the problem of *catastrophic forgetting* (Goodfellow et al., 2013). This includes approaches based on: constraining the weight changes (Kirkpatrick et al., 2017; Zenke et al., 2017; Dhar et al., 2019; Aljundi et al., 2018; Li & Hoiem, 2017), retaining the samples from previous data in a memory (Parisi et al., 2019; De Lange et al., 2019; Castro et al., 2018; Shin et al., 2017; Aljundi et al., 2018) data augmentation (Yu et al., 2020; Zhu et al., 2021a; 2022; Xiang et al., 2019), dynamic expansion based architectures (DEA) which expands the network for each new incoming task id while the weights of the base network are frozen for learning keeping both old and new information (Fernando et al., 2017; Golkar et al., 2019; Hung et al., 2019; Yan et al., 2021; Douillard et al., 2020; Li et al., 2021). All these approaches are broadly categorised into two main themes i.e. multi-task and multi-class. Multi-task approaches like DEA requires resolving the task id during inference, which is typically unavailable. Multi-class scenario refers to learning a single classifier with a aim to recognize the base and novel classes in a single task. In this paper we study the FSCIL problem under multi-class scenario, since it is more realistic and practical. Recent approaches (Zhang et al., 2021; Hersche et al., 2022; Akyürek et al., 2021) have proposed to learn the backbone network as feature extractor using data of base classes, and then use this frozen feature extractor to learn the classifier prototypes for novel classes incrementally. But this does not guarantee the maximum separability between the classifier prototypes for base and novel classes and hence can lead to the confusion between the old and new classes resulting in limited performance. Other approaches (Chen & Lee,

[*]Equal contribution [1]Applied Scientist, Amazon. Correspondence to: Anant Khandelwal <anant.iitd.2085@gmail.com>.

Proceedings of the 2nd Annual Workshop on Topology, Algebra, and Geometry in Machine Learning (TAG-ML) at the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. 2023. *Copyright 2023 by the author(s).*

2020; Akyürek et al., 2021; Hersche et al., 2022), which uses custom loss functions and regularizers to learn the classifier prototypes for novel classes along with preventing *forgetting* on base classes, are also limited by performance because of misalignment between fixed features of base classes and classifier. Recent work *NC-FSCIL*(Yang et al., 2023) proposed the use of neural collapse to learn the maximally separable classifier. They proposed to learn the two layer classifier with pre-fixed simplex weights for base and each incremental sessions. The two layer classifier training with pre-fixed weights for few shot novel classes will not be able to generalize well and hence results in overfitted class representation. Opposed to them we proposed to learn itself the maximum separable class representation using Neural Collapse properties, but we used concept factorization on backbone network to be able to represent any class in general and hence obtain the generalized classifier for novel few shot classes.

In this work, we address this problem of misalignment between fixed features of backbone network (feature extractor) and classifier prototypes to prevent forgetting of base class. Towards that we attempt to learn the maximal separable classifier to avoid confusion between base and novel classes in each incremental session. Our work is inspired from two main studies: 1) Neural Collapse (NC) for imbalanced data (Dang et al., 2023; Papyan et al., 2020) and, 2) Concept Factorization (Fel et al., 2022; Kalayeh et al., 2014). Neural collapse is the phenomenon where the network when trained beyond zero error towards zero loss, results in collapsing the last layer features of backbone network to form an Equiangular Tight Frame (ETF). The vertices of this frame denotes the feature vector representing the class and aligned with classifier prototype of the corresponding class (Papyan et al., 2020). This guarantees a maximal separable classifier since ETF is a geometric structure forming a simplex where the within class variance is minimized (because of collapse to a single vector) and between class variance is maximized lying at equal angles from each other. However, with pre-fixed simplex for base classes, the feature extractor is easy to train since the sufficient data is available for each of the class in base session ($t = 0$) resulting in collapse but for any incremental session $t \geq 1$ with few labelled samples learning the collapsed features for novel classes is challenging since with few samples (as much as 5 samples for a class) the fixed feature extractor is not able to align well with the novel class prototype. To resolve that we introduce the mechanism of concept factorization, where we dissect the collapsed feature extractor on base session to identify the concept basis in the input images. Once the concept basis ("*concept bank*") is identified from base session, we recognize them as the building block from which the incremental session classifier simplex is induced and hence the new set of coefficients can be learnt for inducing the simplex with novel classes. This is additionally fine-tuned along with base class simplex to further align this with few shot instances to reduce any irreducible error occurred during calculating optimal coefficients for the "*concept bank*". This has been illustrated in Fig.1. To summarize, our main contributions are as follows:

- We introduce a novel framework **MASIL** as an attempt to learn the maximal separable classifier for FSCIL.
- We identified the mechanism where the base session collapsed features (obtained as per Neural Collapse properties) can further be dissected in terms of "*concept bank*", which forms the basis for building classifier prototype of novel classes encountered during incremental session.
- Evaluation on three popular FSCIL benchmarks datasets demonstrating state-of-the-art performance. Extensive ablation study has been done to analyze the importance of loss function introduced using Neural collapse properties and the advantage of simplex fine-tuning to reduce the irreducible error.

2. Related Work

2.1. Few Shot Learning

The idea of few shot learning (FSL) is to adapt the model on novel classes (with only few labelled instances) without caring for the performance on base classes. Most of the works uses meta-learning (Sung et al., 2018; Finn et al., 2017; Sun et al., 2019; Snell et al., 2017) or metric learning (Snell et al., 2017; Vinyals et al., 2016; Sung et al., 2018). Recently, the approaches (Gidaris & Komodakis, 2018; Ren et al., 2019) have demonstrated the use of meta learning to recognize the base and novel classes both, by sampling "fake" few shot classification task from base classes to learn a classifier for novel classes. Finally, the learned classifier weights are combined to jointly recognize the base and novel classes. Some of the works (Ren et al., 2019) regard this as sort of incremental learning. Contrastively, FSCIL setting is much more realistic where the base dataset is not accessible during the incremental stage and we have to adapt the model for novel classes without catastrophic forgetting (Tao et al., 2020b; Dong et al., 2021). Metric learning approaches focus on learning a strong backbone network for learning transferable features across the tasks, on top which the similarity function (like k-nearest neighbours in (Vinyals et al., 2016), non linear distance metric in (Sung et al., 2018)) is learnt to demonstrate the ability to classify the novel classes with transferable features. However, this requires to train the as much similarity function as the number of incremental sessions in FSCIL but the aim of FSCIL is to train one unified classifier for the base and novel classes.

We will discuss in the next section how existing works have dealt the problem of FSCIL different from FSL.

2.2. Few Shot Class Incremental Learning

Class Incremental Learning (CIL): We start by first discussing the idea of Class Incremental Learning (CIL), it aims to learn a classifier that manages to continuously update itself to recognize all the novel classes without forgetting the base classes (Rebuffi et al., 2017; Cauwenberghs & Poggio, 2000; Li & Hoiem, 2017). To overcome this forgetting mechanism CIL studies has been categorized into three broad categories: regularization based (Kirkpatrick et al., 2017; Li & Hoiem, 2017; Dhar et al., 2019), rehearsal based (Parisi et al., 2019; De Lange et al., 2019; Aljundi et al., 2018; Castro et al., 2018), and knowledge distillation (Rebuffi et al., 2017; Hou et al., 2019; Wu et al., 2019). Regularization based methods constraint the weight changes on the novel classes thereby keeping the information previously learnt for base classes, this causes these methods to suffer for generalization on novel classes because of small allowed change in weights. Rehearsal based methods in which the model is continually be refreshed using old data reserve so that it maintains to learn the novel classes along with old classes. This is limited by the amount of old data it can retain in memory, and the how the instances from old data to be selected for maximal information with minimum memory requirements. These methods are not scalable to large number of classes because of the limited memory. For example, iCaRL(Rebuffi et al., 2017) learns the nearest neighbour classifier for novel classes while maintaining the memory of exemplars from base session. Knowledge Distillation based methods requires the use of large teacher model to guide the learning of small student model (Yang et al., 2022; Hinton et al., 2015). It works by distilling the previously learned information to new model with novel classes, so as to avoid forgetting of base classes. Recent studies (Dhar et al., 2019; Douillard et al., 2020; Hou et al., 2019) conducted the distillation on feature level rather than on output logit level at the classifier. However, these solutions suffer from a problem of distinguishing between base and novel classes leading to limitation in the performance.

Few Shot Class Incremental Learning (FSCIL): Compared to CIL setting, FSCIL aims to learns the novel classes (along with base classes) with few labelled instances (Tao et al., 2020b; Dong et al., 2021), which is much more realistic and hard, since learning from few instances of novel classes causes over-fitting on novel classes (Snell et al., 2017; Sung et al., 2018). In order to do this, some studies have focused to align base and incremental session using augmentation (Peng et al., 2022), searching for flat minima (Shi et al., 2021). However, for avoiding over-fitting to novel classes it is required that the classifier prototypes for novel classes should be maximally separable from base classes. Adjusting prototypes for base classes is not feasible since that requires the use of base session data. However, these studies (Zhang et al., 2021; Zhu et al., 2021b) have focused on evolving prototypes for novel classes. Large number of existing works have focused on building the custom loss and regularizer (Ren et al., 2019; Hou et al., 2019; Tao et al., 2020a; Joseph et al., 2022; Lu et al., 2022; Hersche et al., 2022; Akyürek et al., 2021; Yang et al., 2022). However, the same disadvantages we discussed in CIL for regularization and custom loss functions applies in FSCIL as well. In this work we focused on the optimal evolution of prototypes for novel classes which is derived from the same function using which base class prototypes have been developed and ensuring the maximal separability between old and novel classes.

3. Problem Statement and Context

In this section we will introduce the problem definition of Few Shot Class Incremental Learning in Section 3.1 and context in subsequent sections.

3.1. Few Shot Class Incremental Learning

Formally, we define Few Shot Class Incremental Learning (FSCIL) as the stream of labelled data in time sequence as $\mathcal{D}_0$, $\mathcal{D}_1$, ..., where $\mathcal{D}_t = \{(\mathbf{x}_j^t, y_j^t)\}_{j=1}^{j=|\mathcal{D}_t|}$. $\mathcal{C}_t$ be the number of classes in training set $\mathcal{D}_t$, where $\forall(i,j)\ \mathcal{C}_i \cap \mathcal{C}_j = \emptyset$. Specifically, we consider $\mathcal{D}_0$ as the base session with large label space $\mathcal{C}_0$ with each class $c \in \mathcal{C}_0$ have sufficient training images. For $t > 0$ each of the incremental session $\mathcal{D}_t$ have only few labelled images for each novel classes. FSCIL is defined as the time step incremental training of model Θ on $\mathcal{D}_t\ \forall\ t > 0$ with no access to any of the previous labelled set from $\mathcal{D}_0$ to $\mathcal{D}_{t-1}$. For $t > 0$ $\mathcal{D}_t$ we denote the setting as C classes with K training examples per class as C-way K-Shot FSCIL where $\mathcal{C}_t \cap \mathcal{C}_t' = \emptyset\ \forall t \neq t'$. After each incremental session training with $\mathcal{D}_t$, model Θ is evaluated to recognize all the training classes encountered so far i.e. $\cup_{i=0}^{i=t} \mathcal{C}^i$. Hence, FSCIL not only aims to recognize novel classes but to avoid forgetting of the old classes and the setting of learning of novel classes is highly imbalanced and suffers from data scarcity problem as well. This makes FSCIL setting more suited for real world applications.

Initialization: Assuming $\mathcal{C}_0$ as the number of base classes and we have total T incremental session and each session has k classes, so there will be total $K = \mathcal{C}_0 + Tk$ classes. To be able to perform FSCIL, we denote the model trained on base session consists of backbone feature extractor $f(;\theta_f)$ and classifier $\mathbf{W} \in \mathbb{R}^{K \times d}$, which is a MLP classifier consisting of L layers denoted as $\mathbf{W} = \mathbf{W}_1\mathbf{W}_2.....\mathbf{W}_L$. For input X we denote the features obtained from feature extractor as $\mathbf{H} = f(X, \theta_f) \in \mathbb{R}^{d \times N}$, where N is the total number of training instances. Similar to (Dang et al., 2023)

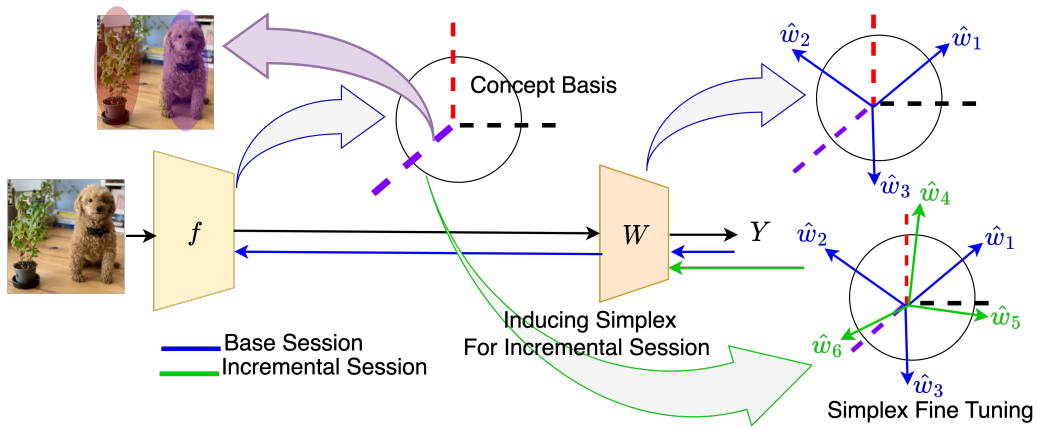

Figure 1. ***MASIL***: Illustrating the concept basis obtained from concept factorization of features after feature extractor and their relation with input images. After base session training, classifier simplex for novel classes are induced from these concept basis. Implicit memory (not shown) storing the feature mean of classes seen till current session. This is used to jointly updating the weights of classifier for both base and novel classes during Simplex Finetuning.

we also consider last layer features $\mathbf{H}$ as freely optimization variables. The optimization objective is then defined as follows:

$$\min_{\mathbf{W},\mathbf{H}} \mathcal{L}(\mathbf{W},\mathbf{H}) = \frac{1}{2N}||\mathbf{W}\mathbf{H}-\mathbf{Y}||_F^2 + \frac{\lambda_{\mathbf{W}}}{2}||\mathbf{W}||_F^2 + \frac{\lambda_{\mathbf{H}}}{2}||\mathbf{H}||_F^2 \quad (1)$$

where $\mathbf{Y} \in \mathbb{R}^{K\times N}$, is the class label for each of the training instance in N instances and λ_W, λ_H are regularization hyperparameters.

3.2. Neural Collapse

In recent works (Papyan et al., 2020; Dang et al., 2023) which have studied the practice of training the DNN beyond zero error towards zero loss. This reveals the geometric structure in the form of simplex equiangular tight frame, formed by the last layer features along with classifier weights. This has been demonstrated on the balanced data and models with various popular architectures. Neural Collapse as defined in (Papyan et al., 2020) consists of the following four properties:

- ($\mathcal{NC}1$) **Variability Collapse**: Last layer features of the backbone network for a particular class collapse to within-class mean.
- ($\mathcal{NC}2$) **Convergence**: results in optimal class-means which are equally and maximally pairwise separated forming a simplex Equiangular Tight Frame (ETF).
- ($\mathcal{NC}3$) **Classifier Convergence**: Optimal class means forming ETF are aligned to the corresponding classifier weights uptio rescaling.
- ($\mathcal{NC}4$) **Simplification to nearest class center** When ($\mathcal{NC}1$)-($\mathcal{NC}3$) holds, the model prediction using logits respects nearest class centers.

In addition to the balanced data, this (Dang et al., 2023) have derived the geometrical analysis for the imbalanced data given as:

Definition 1. *Let ($\mathbf{W}^*$, $\mathbf{H}^*$) be the global optimizer of equation 1, $r = min(K,d)$ and $\mathbf{W} = U_W S_W V_W^T$ be the SVD factorization of $\mathbf{W}$. Then the following holds for the imbalanced data:*

- *($\mathcal{NC}1$) leads to collapse of features within the same class $\mathbf{H}^* = \overline{\mathbf{H}}^*\mathbf{Y}$, where $\overline{\mathbf{H}}^* = [\mathbf{h}_1^*, \mathbf{h}_2^*, ...\mathbf{h}_K^*] \in \mathbb{R}^{d\times K}$*
- *($\mathcal{NC}3$) leads to alignment between classifier weights and corresponding class mean as $\mathbf{w}_k^* = \sqrt{\frac{n_k\lambda_H}{\lambda_W}}\mathbf{h}_k^* \; \forall k \in [K]$, where n_k is the number of instances of class k.*
- *($\mathcal{NC}2$) leads to optimal class means equally and maximally separated forming simplex Equiangular Tight Frame (ETF) $\mathbf{W}^*\mathbf{W}^{*\top} = diag\,\{s_k^2\}_{k=1}^{k=K}$, where s_k are the singular values of $\mathbf{W}^*$*

Another approach i.e. Deep Simplex Classifier(Cevikalp & Saribas, 2022), proposed the optimization problem as

minimization of features obtained from feature extractor to the vertices of simplex as:

$$\min_{\mathbf{h}_i \in \mathbf{H}^T} \frac{1}{n} \sum_{i=1}^{n} ||\mathbf{h}_i - s_{y_i}||^2 \tag{2}$$

where s_{y_i} is vertex of simplex and is treated as the class center for class y_i.

4. MASIL

Overall framework of our proposed method is illustrated in Fig. 1. FSCIL aims to learn the classifier weights $W^{K \times d}$ which works for all classes irrespective of whether they belongs to the base classes during $t = 0$ or few shot classes during $t > 0$. Traditionally, this has been achieved by first learning the classifier weights for $\mathcal{C}_0$ base classes and then learn the weights for novel classes $W^{(t)} \in \mathbb{R}^{k \times d}$ with the regularized constraint in the loss function that the old weights $W \in \mathbb{R}^{C_0 + (t-1)k}$ be preserved with little or no updates. However, this leads to misalignment between the classifier prototypes of old and novel classes causing old new confusion(ONC) (Huang et al., 2022) and catastrophic forgetting (Goodfellow et al., 2013). This causes drop in performance of FSCIL classifier as the number of incremental session grows resulting in poor generalizability even in recognizing the base classes. To mitigate this, in this work we adopted the properties obtained from Neural Collapse to learn maximally separable classifier along with concept factorization to learn classifier weights (organized as simplex) for novel classes with few samples. We restricted the feature extractor from updates during incremental session training and rely on concept factorization of the activations obtained for base classes to obtain the basis of concepts called "*concept bank*", using which we can represent maximally separable classifier weights i.e. simplex for few shot classes. To represent the classifier simplex using "*concept bank*" it requires to solve only for the coefficient matrix which can be done by just solving the Non Negative Least Squares (NNLS).

4.1. Concept Factorization

The idea of concept factorization relates to the phenomena of neural collapse, where it learns to maximally separate the classes by forming the simplex at class level on both levels of class features and classifier weights. In order to achieve this it merge the activations (during forward pass) of the same class until they all converge to the one hot class vector at the logits layer as depicted in equation 2. This allows the class wise feature vectors which are concentrated at higher layers to be recursively broken into multiple concepts moving from highest layer to lower layers tracing back to the input images where it can be explained with regions as concepts, combination of which makes it possible to be able to classify it to particular class. We adopted NMF (Non Negative Matrix Factorization) as in (Fel et al., 2022) of activations obtained at the output of feature extractor given as:

$$\min_{\mathbf{P} \geq 0, \mathbf{Q} \geq 0} \frac{1}{2} ||\mathbf{A} - \mathbf{P}\mathbf{Q}^T||_F^2 \tag{3}$$

where, $||.||_F$ is the Frobenius norm, the activations $\mathbf{A} \in \mathbb{R}^{n \times d}$ obtained from crop of images $X_i = \tau(x_i)$ $X_i \in X^{n \times p}$ with τ is a crop function. We take random crops (governed by τ) of images, this results in unique concepts across the categories to be able to build the bank of unique concept vectors called "*concept bank*". Activations at the last layer of feature extractor after global pooling for these random crops is given as $\mathbf{A} = f(X, \theta_f) \in \mathbb{R}^{n \times d}$. NMF is simply the factorization of concept activations $\mathbf{A}$ into the "*concept bank*" $\mathbf{Q} \in \mathbb{R}^{v \times d}$ (where it follows low rank factorization $v \ll \mathbf{min}(n, p)$) and coefficients $\mathbf{P} \in \mathbb{R}^{n \times v}$ denote the importance of each of the concepts in explaining the activations $\mathbf{A}$. Once the "*concept bank*" is precomputed, we can obtain the coefficients $P(x)$ for any input x using NNLS (Non-Negative Least Squares) i.e. $\underset{P \geq 0}{min} \frac{1}{2}||f(x; \theta_f) - P(x)\mathbf{Q}^T||_F^2$. Relating activation factorization in equation 3 and neural collapse in equation 2, implies that the activations when collapses to the mean features vector for each class forming the class simplex vector which is composed of concept basis vectors and the corresponding coefficients, combining for all classes which gives the overall basis called "*concept bank*".

4.2. NMF Layer

During NMF factorization of equation 3, we keep the feature extractor $f(.; \theta_f)$ frozen. We approached the NMF problem solution using **ADMM** (Alternating Direction Method of Multipliers) (Boyd et al., 2011) since NMF is non-convex, but however it can be made convex by fixing the value of either of the two factors $(\mathbf{P}, \mathbf{Q})$ which requires alternating update of either of two factors fixing one at a time, which is equivalent to solving a Non-Negative Least Squares (NNLS) problem making it convex. This alternating update mechanism called as **ADMM**, formulated as:

$$\mathbf{P}_{t+1} = \arg\min_{\mathbf{P} \geq 0} \frac{1}{2} ||\mathbf{A} - \mathbf{P}\mathbf{Q_t}^T||_F^2 \tag{4}$$

$$\mathbf{Q}_{t+1} = \arg\min_{\mathbf{Q} \geq 0} \frac{1}{2} ||\mathbf{A} - \mathbf{P_t}\mathbf{Q}^T||_F^2 \tag{5}$$

It ensures global or local minimum since each of the NNLS problem obeys *Karush–Kuhn–Tucker* (KKT) optimality conditions (Karush, 2014; Kuhn & Tucker, 2014). Using these conditions forming the implicit function (Griewank & Walther, 2008) makes the implicit differentiation (Griewank & Walther, 2008; Krantz & Parks, 2002; Bell & Burke, 2008) allows to compute the gradients $(\frac{\partial P}{\partial A}, \frac{\partial Q}{\partial A})$, but however we have to relate the concepts with the input image

regions we require to compute $(\frac{\partial P}{\partial X}, \frac{\partial Q}{\partial X})$. This can be calculated as:

$$\frac{\partial P}{\partial X} = \frac{\partial A}{\partial X}\frac{\partial P}{\partial A}, \frac{\partial Q}{\partial X} = \frac{\partial A}{\partial X}\frac{\partial Q}{\partial A} \tag{6}$$

Computation of $\frac{\partial A}{\partial X}$ is fairly straight word using *Pytorch*. More details on implementation of combining gradients from implicit differentiation in *Jax* (Huang et al., 2016; Blondel et al., 2021) and gradient from *Pytorch* computation is detailed in Section B. Once we precompute the "concept bank" for base classes using equation 5 and 6, we fixed the $\mathbf{Q}$ and allows only to compute optimal coefficients $P(x)$ for any input x using NNLS

$$\min_{\mathbf{P}\geq 0} \frac{1}{2}||f(x;\theta_f) - P(x)\mathbf{Q}^T||_F^2 \tag{7}$$

which give the optimal representation of activation for any input x in terms concept basis vectors.

4.3. Classifier Simplex Representation

Equation 2 is optimizing the feature representation for each class resulting in collapsed representation for class y_i as s_{y_i}. Similarly, equation 1 results in simplex representation for each class i.e. $\mathbf{w}_{y_i} \in \mathbf{W}$. So if we consider the normalized simplex representation on a unit hypersphere (Cevikalp & Saribas, 2022) of each class then:

$$\mathbf{w}_{y_i}^T s_{y_i} = 1 \; \forall \; y_i \in \cup_{j=0}^{j=t}\mathcal{C}_j \tag{8}$$

which results in the modified loss function of equation 2 to:

$$\min_{\mathbf{h}_i} \frac{1}{|\mathcal{D}_j|} \sum_{(x_i,y_i)\in\mathcal{D}_j} ||\mathbf{w}_{y_i}^T\mathbf{h}_i - 1||_F^2 \tag{9}$$

s.t. $\mathbf{w}_{y_i}^T s_{y_i} = 1$ which is same as in equation 1 and hence follow the neural collapse properties. Moreover, optimizing equation 2, results in the collapsed feature representation for all instances belonging to that class. Additionally, equation 3 computes the best approximation of collapsed feature representation $\mathbf{H} \approx \mathbf{P}\mathbf{Q}^T$. For any input (x_i, y_i) belongs to $\mathcal{D}_j,\ j > 0$, then optimal $\mathbf{h}_i$ obtained from equation 7 is given as:

$$\mathbf{h}_i = \mathbf{P}(x_i)\mathbf{Q}^T \tag{10}$$

From ($\mathcal{NC}1$), the collapsed feature representation of each class converge to a unique vector e.g. for class y_i the feature representation of all instances is denoted as $\mathbf{H}_{y_i} \in \mathbf{H}$, ($\mathcal{NC}1$) implies covariance $\sum_{\mathbf{H}_{y_i}} \rightarrow 0$. i.e. the features collapse to their corresponding class means i.e. $\mathbf{h}_{y_i}^* = \sum_{i=1}^{n_{y_i}} \mathbf{h}_i$, where n_{y_i} is the number of instances for class y_i, and as per loss in equation 2, this is minimum when $s_{y_i} = \mathbf{h}_{y_i}^*$, then from equation 8 and 10:

$$\hat{\mathbf{w}}_{y_i} = \frac{1}{|\mathcal{D}_j|}\left(\sum_{(x_i,y_i)\in\mathcal{D}_j} \mathbf{P}(x_i)\right)\mathbf{Q}^T \; \forall j > 0 \tag{11}$$

where the coefficients $\mathbf{P}(x_i)$ for each instance of class y_i are calculated using NNLS as per equation 7, additionally, $y_i \in \mathcal{C}_j, j > 0$ are the few shot classes and classifier weights are the optimal simplex representation for few shot classes. For base session classes ($j = 0$) the classifier simplex representation is simply $\hat{\mathbf{w}}_{y_i} = s_{y_i}^T$. Since we implemented the classifier using MLP with $L = 2$ layers, for each layer the simplex representation is $\hat{\mathbf{w}}_{l,y_i} = (\hat{\mathbf{w}}_{y_i})^{1/L}$.

4.4. Simplex Finetuning

In Section 4.3 we described the optimal simplex representation for each class belongs to the few shot class $y_i \in \mathcal{C}_j, j > 0$. But, however due to the the inherent irreducible error to NNLS, we approach the optimal representation of simplex for few shot class by further fine-tuning the classifier weights (keeping the feature extractor frozen) initialized using simplex representation as obtained in equation 11. To avoid deviating the weights to much from optimal simplex representation we add a constraint to the loss in equation 9 as:

$$\min_{\mathbf{w}_{y_i}} \mathcal{L}(\mathbf{w}_{y_i}) = \frac{1}{|\mathcal{D}_j|} \sum_{(x_i,y_i)\in\mathcal{D}_j} ||\mathbf{w}_{y_i}^T\mathbf{h}_i - 1||_F^2 + \alpha||\mathbf{w}_{y_i} - \hat{\mathbf{w}}_{y_i}||_F^2, \alpha \in [0,1] \tag{12}$$

where $y_i \in \mathcal{C}_j, j > 0$ and feature extractor is frozen and hence optimizing for the best $\mathbf{w}_{y_i}$. Since base session training (i.e. $j = 0$ and dataset $\mathcal{D}_0$) is governed using the loss function of equation 2 and hence results in collapsed representation of features at the terminal layer for each class $y_i \in \mathcal{C}_0$. For the simplex representation for each class in $\mathcal{C}_0$ to remain maximally separable with the ones obtained for few shot class we utilized the collapsed representation of features for each class to further fine tune the simplex representation, but without keeping the image instances in memory we memorized the collapsed representation (which is the mean representation of instance features for each class) in $\mathcal{M}$ given as:

$$\mathcal{M}_{y_i} = \frac{1}{n_{y_i}} \sum_{i=1}^{n_{y_i}} \mathbf{h}_i, \forall \mathcal{M}_{y_i} \in \mathcal{M} \tag{13}$$

where, n_{y_i} is the number of instances of class y_i. The updated loss function during fine tuning stage include the base session classes and few shot class is given as:

$$\min_{\mathbf{w}_{y_i}} \mathcal{L}(\mathbf{w}_{y_i}) = \frac{1}{|\mathcal{D}_j|} \sum_{(x_i,y_i)\in\mathcal{D}_j} ||\mathbf{w}_{y_i}^T\mathbf{h}_i - 1||_F^2 + \frac{1}{|\mathcal{M}|} \sum_{(\mathcal{M}_{y_i},y_i)\in\mathcal{M}} ||\mathbf{w}_{y_i}^T\mathcal{M}_{y_i} - 1||_F^2 + \alpha||\mathbf{w}_{y_i} - \hat{\mathbf{w}}_{y_i}||_F^2, \alpha \in [0,1] \tag{14}$$

Methods	Session Accuracy (%) (↑)									Average Acc. (↑)	Relative Improvement
	0	1	2	3	4	5	6	7	8		
iCaRL (Rebuffi et al., 2017)	61.31	46.32	42.94	37.63	30.49	24.00	20.89	18.80	17.21	33.29	**+41.65**
NCM (Hou et al., 2019)	61.31	47.80	39.30	31.90	25.70	21.40	18.70	17.20	14.17	30.83	**+44.69**
D-Cosine (Vinyals et al., 2016)	70.37	65.45	61.41	58.00	54.81	51.89	49.10	47.27	45.63	55.99	**+13.23**
TOPIC (Tao et al., 2020b)	61.31	50.09	45.17	41.16	37.48	35.52	32.19	29.46	24.42	39.64	**+34.44**
IDLVQ (Chen & Lee, 2020)	64.77	59.87	55.93	52.62	49.88	47.55	44.83	43.14	41.84	51.16	**+17.02**
Self-promoted (Zhu et al., 2021b)	61.45	63.80	59.53	55.53	52.50	52.50	46.69	43.79	41.92	52.76	**+16.94**
CEC (Zhang et al., 2021)	72.00	66.83	62.97	59.43	56.70	53.73	51.19	49.24	47.63	57.75	**+11.23**
LIMIT (Zhou et al., 2022b)	72.32	68.47	64.30	60.78	57.95	55.07	52.70	50.72	49.19	59.06	**+9.67**
Regularizer (Akyürek et al., 2021)	80.37	74.68	69.39	65.51	62.38	59.03	56.36	53.95	51.73	63.71	**+7.13**
MetaFSCIL (Chi et al., 2022)	72.04	67.94	63.77	60.29	57.58	55.16	52.90	50.79	49.19	58.85	**+9.67**
C-FSCIL (Hersche et al., 2022)	76.40	71.14	66.46	63.29	60.42	57.46	54.78	53.11	51.41	61.61	**+7.45**
Data-free Replay (Liu et al., 2022)	71.84	67.12	63.21	59.77	57.01	53.95	51.55	49.52	48.21	58.02	**+10.65**
ALICE (Peng et al., 2022)	80.60	70.60	67.40	64.50	62.50	60.00	57.80	56.80	55.70	63.99	**+3.16**
SSFE-Net (Pan et al., 2023)	72.06	66.17	62.25	59.74	56.36	53.85	51.96	49.55	47.73	57.74	**+11.13**
NC-FSCIL (Yang et al., 2023)	84.02	76.80	72.00	67.83	66.35	64.04	61.46	59.54	58.31	67.82	**+0.55**
MASIL(Ours)	**85.15**	**77.00**	**72.20**	**67.92**	**66.60**	**64.2**	**61.50**	**59.60**	**58.86**	**68.11**	

Table 1. Performance comparison on miniImageNet with ResNet-18 as backbone architecture under *5-way 5-shot* FSCIL setting. Table denotes the accuracy in each session, average accuracy across sessions and "Relative Improvement" denotes the improvement of our method in the last session. Methods above separating line are CIL methods for FSCIL as in (Tao et al., 2020b) and (Zhang et al., 2021)

Methods	Session Accuracy (%) (↑)									Average Acc. (↑)	Relative Improvement
	0	1	2	3	4	5	6	7	8		
iCaRL (Rebuffi et al., 2017)	64.10	53.28	41.69	34.13	27.93	25.06	20.41	15.48	13.73	32.87	**+42.42**
NCM (Hou et al., 2019)	64.10	53.05	43.96	36.97	31.61	26.73	21.23	16.78	13.54	34.22	**+42.61**
D-Cosine (Vinyals et al., 2016)	74.55	67.43	63.63	59.55	56.11	53.80	51.68	49.67	47.68	58.23	**+8.47**
TOPIC (Tao et al., 2020b)	64.10	55.88	47.07	45.16	40.11	36.38	33.96	31.55	29.37	42.62	**26.78**
Self-promoted (Chen & Lee, 2020)	64.10	65.86	61.36	57.45	53.69	50.75	48.58	45.66	43.25	54.52	**+12.9**
CEC (Zhang et al., 2021)	73.07	68.88	65.26	61.19	58.09	55.57	53.22	51.34	49.14	59.53	**+7.01**
DSN (Chi et al., 2022)	73.00	68.83	64.82	62.64	59.36	56.96	54.04	51.57	50.00	60.14	**+6.15**
LIMIT (Zhou et al., 2022b)	73.81	72.09	67.87	63.89	60.70	57.77	55.67	53.52	51.23	61.84	**+4.92**
MetaFSCIL (Akyürek et al., 2021)	74.50	70.10	66.84	62.77	59.48	56.52	54.36	52.56	49.97	60.79	**+6.18**
C-FSCIL (Hersche et al., 2022)	77.47	72.40	67.47	63.25	59.84	56.95	54.42	52.47	50.47	61.64	**+5.68**
Data-free Replay (Liu et al., 2022)	74.40	70.20	66.54	62.51	59.71	56.58	54.52	52.39	50.14	60.78	**+6.01**
ALICE (Peng et al., 2022)	79.00	70.50	67.10	63.40	61.20	59.20	58.10	56.30	54.10	63.21	**+2.05**
NC-FSCIL (Yang et al., 2023)	82.52	76.82	73.34	69.68	66.19	62.85	60.96	59.02	56.11	67.50	**+1.12**
MASIL(Ours)	**82.55**	**76.98**	**73.44**	**69.75**	**66.48**	**62.98**	**61.4**	**59.81**	**57.23**	**67.84**	

Table 2. Performance comparison on CIFAR-100 with ResNet-18 as backbone architecture under *5-way 5-shot* FSCIL setting. Table denotes the accuracy in each session, average accuracy across sessions and "Relative Improvement" denotes the improvement of our method in the last session. Methods above separating line are CIL methods for FSCIL as in (Tao et al., 2020b) and (Zhang et al., 2021)

where the constraint is now valid for base session classes as well along with few shot classes with the fact that simplex representation for each class should not deviate much (depends on the contributing factor α) from the optimal simplex representation. In each incremental session we train our classifier network using this loss function after deriving the simplex representation for each few shot classes from equation 11.

5. Experiments

We prove the effectiveness of MASIL on three well known FSCIL benchmark datasets described in Appendix A along with their corresponding FSCIL setting. Training details and hyper parameters are discussed in Appendix B.

5.1. Benchmark Evaluation

Performance comparison on miniImageNet, CIFAR-100 and CUB-200 is demonstrated in Table 1, 2 and 5 (given in Appendix C due to space limitation) respectively. Our method ***MASIL*** outperforms in all the methods in the last session with relative improvement of +3.16%, +2.05% and +0.14% on miniImageNet, CIFAR-100, CUB-200 respectively as compared to strongest baseline ALICE (Peng et al., 2022). Additionally, our method outperforms all the methods in all the sessions (except on CUB-200 session 2). Moreover, on average accuracy our method outperforms atleast by +1.79% as compared to strongest baseline, collectively is an indicator that our model helps in mitigating the forgetting issue in a realistic setting of continual learning namely *FSCIL*.

5.2. Ablation Studies

We consider variations to base model (backbone network i.e. ResNet-18 with classifier and memory as introduced in equation 13) to validate the 1) effects of loss introduced in equation 9 (ETF) as compared to cross-entropy (CE) loss with and without neural collapse induced simplex classifier

Models	miniImageNet		CIFAR-100		CUB-200	
	Final (↑)	Average (↑)	Final (↑)	Average (↑)	Final (↑)	Average (↑)
Learnable + CE	50.04	61.30	52.13	62.68	50.38	59.58
$\mathcal{NC}$ + CE	56.66	68.23	54.42	64.00	56.83	65.51
$\mathcal{NC}$ + ETF Loss	58.31	67.82	56.11	67.50	59.44	67.28
$\mathcal{NC}$ + ETF Loss + CF	58.72	68.04	56.13	67.51	59.72	67.45
***MASIL**(Ours)*	58.86	68.11	57.23	67.84	60.24	67.54

Table 3. Ablation Studies on three datasets investigating the effects of Simplex based loss, Concept Factorization and Simplex Fine Tuning

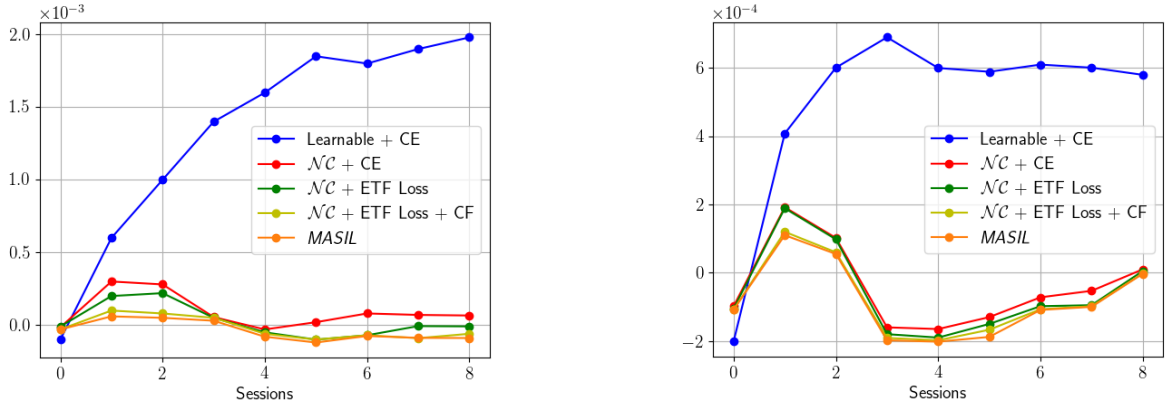

Figure 2. Average cosine similarities between different classes at each session for Train (Left) and Test (Right) on miniImageNet. Calculation of cosine similarity is done for all the classes encountered so far after the model gets trained in current session.

prototypes, 2) effects of few shot simplex induced with concept factorization and 3) effects of simplex fine tuning. To validate the first effect there are two models. The first model (Learnable + CE) uses a classifier with learning weights from CE loss, which is the most common practice. Second model ($\mathcal{NC}$ + CE) uses the CE loss with classifier weights as per the neural collapse properties but uses the CE loss instead of the loss in equation 9. To validate the second effect (third model i.e. $\mathcal{NC}$ + ETF Loss) we did not initialize the classifier weights for few shot classes and train them as per the loss in equation 9 with memory of base classes as in equation 13. For third effect i.e. fourth model ($\mathcal{NC}$ + ETF Loss + CF) we reported the performance without fine tuning and just use the classifier weights calculated from concept factorization (CF) as in equation 11. Finally, we reported performance of ***MASIL*** to compare among all of them. As shown in Table 3, adopting the loss function in equation 9 is definitely helps in mitigating performance drop as compared to CE loss even with classifier weights is assumed to be forming simplex, and it further mitigates using the weight initialized with CF and further with fine tuning. It indicates the success of CF along with neural collapse towards optimal solution for FSCIL.

5.3. Analysing Classifier Weights

We further analysed the classifier weights alignment with respect to the mean feature (collapsed feature) of each class. We used the classifier weights and mean feature from each of the models described in ablation studies to validate the effect of *MASIL* in learning the maximal separable classifier, where the separable property between classes is measured by cosine similarity. Specifically, we plotted the average cosine similarities between mean feature and the classifier weights of different classes i.e. $\text{Avg}_{k \neq k'}\{h_k \cdot w'_k\}$ for both train and test datasets. We have illustrated this for miniImageNet in Fig. 2. Clearly, on both the train and test the similarity between different classes goes on increasing for the "Learnable + CE" model. While using the loss in equation 9 (as per Neural Collapse) have no increasing trend. Incorporating concept factorization and simplex fine tuning (in *MASIL*) further reduces the similarities as the session grows on and hence mitigate the effect of forgetting and confirming the maximum separability with *MASIL*.

6. Conclusion

In this paper we propose the novel framework *MASIL* as an step towards learning the maximum separable classifier in a competitive setting of continual learning i.e. FSCIL. We propose to induce the simplex from concept factorization helps in few shot cases. We introduced novel loss function where the base and novel classes can be learnt together during fine tuning to further mitigate forgetting and overfitting. In experiments *MASIL* outperforms all the benchmarks with sufficient margin on three datasets proving its efficiency.

References

Akyürek, A. F., Akyürek, E., Wijaya, D., and Andreas, J. Subspace regularizers for few-shot class incremental learning. In *International Conference on Learning Representations*, 2021.

Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., and Tuytelaars, T. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 139–154, 2018.

Bell, B. M. and Burke, J. V. Algorithmic differentiation of implicit functions and optimal values. In *Advances in Automatic Differentiation*, pp. 67–77. Springer, 2008.

Blondel, M., Berthet, Q., Cuturi, M., Frostig, R., Hoyer, S., Llinares-López, F., Pedregosa, F., and Vert, J.-P. Efficient and modular implicit differentiation. *arXiv preprint arXiv:2105.15183*, 2021.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

Castro, F. M., Marín-Jiménez, M. J., Guil, N., Schmid, C., and Alahari, K. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 233–248, 2018.

Cauwenberghs, G. and Poggio, T. Incremental and decremental support vector machine learning. *Advances in neural information processing systems*, 13, 2000.

Cevikalp, H. and Saribas, H. Deep simplex classifier for maximizing the margin in both euclidean and angular spaces. *arXiv preprint arXiv:2212.11747*, 2022.

Chen, K. and Lee, C.-G. Incremental few-shot learning via vector quantization in deep embedded space. In *International Conference on Learning Representations*, 2020.

Cheraghian, A., Rahman, S., Ramasinghe, S., Fang, P., Simon, C., Petersson, L., and Harandi, M. Synthesized feature based few-shot class-incremental learning on a mixture of subspaces. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8661–8670, 2021.

Chi, Z., Gu, L., Liu, H., Wang, Y., Yu, Y., and Tang, J. Metafscil: A meta-learning approach for few-shot class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14166–14175, 2022.

Dang, H., Nguyen, T., Tran, T., Tran, H., and Ho, N. Neural collapse in deep linear network: From balanced to imbalanced data. *arXiv preprint arXiv:2301.00437*, 2023.

De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. Continual learning: A comparative study on how to defy forgetting in classification tasks. *arXiv preprint arXiv:1909.08383*, 2(6):2, 2019.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Dhar, P., Singh, R. V., Peng, K.-C., Wu, Z., and Chellappa, R. Learning without memorizing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5138–5146, 2019.

Dong, S., Hong, X., Tao, X., Chang, X., Wei, X., and Gong, Y. Few-shot class-incremental learning via relation knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 1255–1263, 2021.

Douillard, A., Cord, M., Ollion, C., Robert, T., and Valle, E. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *European Conference on Computer Vision*, pp. 86–102. Springer, 2020.

Fel, T., Picard, A., Bethune, L., Boissin, T., Vigouroux, D., Colin, J., Cadène, R., and Serre, T. Craft: Concept recursive activation factorization for explainability. *arXiv preprint arXiv:2211.10154*, 2022.

Fernando, C., Banarse, D., Blundell, C., Zwols, Y., Ha, D., Rusu, A. A., Pritzel, A., and Wierstra, D. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

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

Gidaris, S. and Komodakis, N. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4367–4375, 2018.

Golkar, S., Kagan, M., and Cho, K. Continual learning via neural pruning. *arXiv preprint arXiv:1903.04476*, 2019.

Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., and Bengio, Y. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

Griewank, A. and Walther, A. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. SIAM, 2008.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hersche, M., Karunaratne, G., Cherubini, G., Benini, L., Sebastian, A., and Rahimi, A. Constrained few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9057–9067, 2022.

Hinton, G., Vinyals, O., Dean, J., et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

Hou, S., Pan, X., Loy, C. C., Wang, Z., and Lin, D. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 831–839, 2019.

Huang, B., Chen, Z., Zhou, P., Chen, J., and Wu, Z. Resolving task confusion in dynamic expansion architectures for class incremental learning. *arXiv preprint arXiv:2212.14284*, 2022.

Huang, K., Sidiropoulos, N. D., and Liavas, A. P. A flexible and efficient algorithmic framework for constrained matrix and tensor factorization. *IEEE Transactions on Signal Processing*, 64(19):5052–5065, 2016.

Hung, C.-Y., Tu, C.-H., Wu, C.-E., Chen, C.-H., Chan, Y.-M., and Chen, C.-S. Compacting, picking and growing for unforgetting continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Joseph, K., Khan, S., Khan, F. S., Anwer, R. M., and Balasubramanian, V. N. Energy-based latent aligner for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7452–7461, 2022.

Kalayeh, M. M., Idrees, H., and Shah, M. Nmf-knn: Image annotation using weighted multi-view non-negative matrix factorization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 184–191, 2014.

Karush, W. Minima of functions of several variables with inequalities as side conditions. In *Traces and Emergence of Nonlinear Programming*, pp. 217–245. Springer, 2014.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Krantz, S. G. and Parks, H. R. *The implicit function theorem: history, theory, and applications*. Springer Science & Business Media, 2002.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

Kuhn, H. W. and Tucker, A. W. Nonlinear programming. In *Traces and emergence of nonlinear programming*, pp. 247–258. Springer, 2014.

Li, D., Wei, X., Hong, X., and Gong, Y. Infrared-visible cross-modal person re-identification with an x modality. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 4610–4617, 2020.

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

Li, Z., Zhong, C., Liu, S., Wang, R., and Zheng, W.-S. Preserving earlier knowledge in continual learning with the help of all previous feature extractors. *arXiv preprint arXiv:2104.13614*, 2021.

Liu, B., Cao, Y., Lin, Y., Li, Q., Zhang, Z., Long, M., and Hu, H. Negative margin matters: Understanding margin in few-shot classification. In *European conference on computer vision*, pp. 438–455. Springer, 2020.

Liu, H., Gu, L., Chi, Z., Wang, Y., Yu, Y., Chen, J., and Tang, J. Few-shot class-incremental learning via entropy-regularized data-free replay. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pp. 146–162. Springer, 2022.

Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., and Song, L. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 212–220, 2017.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Lu, B., Gan, X., Yang, L., Zhang, W., Fu, L., and Wang, X. Geometer: Graph few-shot class-incremental learning via prototype representation. *arXiv preprint arXiv:2205.13954*, 2022.

Ma, Z., Wei, X., Hong, X., and Gong, Y. Bayesian loss for crowd count estimation with point supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6142–6151, 2019.

Pan, Z., Yu, X., Zhang, M., and Gao, Y. Ssfe-net: Self-supervised feature enhancement for ultra-fine-grained few-shot class incremental learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 6275–6284, 2023.

Papyan, V., Han, X., and Donoho, D. L. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Peng, C., Zhao, K., Wang, T., Li, M., and Lovell, B. C. Few-shot class-incremental learning from an open-set perspective. In *European Conference on Computer Vision*, pp. 382–397. Springer, 2022.

Pernici, F., Bruni, M., Baecchi, C., Turchini, F., and Bimbo, A. D. Class-incremental learning with pre-allocated fixed classifiers. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 6259–6266, Los Alamitos, CA, USA, jan 2021. IEEE Computer Society. doi: 10.1109/ICPR48806.2021.9413299. URL https://doi.ieeecomputersociety.org/10.1109/ICPR48806.2021.9413299.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

Ren, M., Liao, R., Fetaya, E., and Zemel, R. Incremental few-shot learning with attention attractor networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252, 2015.

Shi, G., Chen, J., Zhang, W., Zhan, L.-M., and Wu, X.-M. Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima. *Advances in Neural Information Processing Systems*, 34:6747–6761, 2021.

Shin, H., Lee, J. K., Kim, J., and Kim, J. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Sun, Q., Liu, Y., Chua, T.-S., and Schiele, B. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 403–412, 2019.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., and Hospedales, T. M. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1199–1208, 2018.

Tao, X., Chang, X., Hong, X., Wei, X., and Gong, Y. Topology-preserving class-incremental learning. In *European Conference on Computer Vision*, pp. 254–270. Springer, 2020a.

Tao, X., Hong, X., Chang, X., Dong, S., Wei, X., and Gong, Y. Few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12183–12192, 2020b.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset. 2011.

Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., and Fu, Y. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 374–382, 2019.

Xiang, Y., Fu, Y., Ji, P., and Huang, H. Incremental learning using conditional adversarial networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6619–6628, 2019.

Yan, S., Xie, J., and He, X. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3014–3023, 2021.

Yang, D., Zhou, Y., Shi, W., Wu, D., and Wang, W. Rd-iod: Two-level residual-distillation-based triple-network for incremental object detection. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 18(1):1–23, 2022.

Yang, Y., Yuan, H., Li, X., Lin, Z., Torr, P., and Tao, D. Neural collapse inspired feature-classifier alignment for few-shot class incremental learning. *arXiv preprint arXiv:2302.03004*, 2023.

Yin, H., Molchanov, P., Alvarez, J. M., Li, Z., Mallya, A., Hoiem, D., Jha, N. K., and Kautz, J. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8715–8724, 2020.

Yu, L., Twardowski, B., Liu, X., Herranz, L., Wang, K., Cheng, Y., Jui, S., and Weijer, J. v. d. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6982–6991, 2020.

Zenke, F., Poole, B., and Ganguli, S. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pp. 3987–3995. PMLR, 2017.

Zhang, C., Cai, Y., Lin, G., and Shen, C. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12203–12213, 2020.

Zhang, C., Song, N., Lin, G., Zheng, Y., Pan, P., and Xu, Y. Few-shot incremental learning with continually evolved classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12455–12464, 2021.

Zhao, H., Fu, Y., Kang, M., Tian, Q., Wu, F., and Li, X. Mgsvf: Multi-grained slow vs. fast framework for few-shot class-incremental learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Zhou, D.-W., Wang, F.-Y., Ye, H.-J., Ma, L., Pu, S., and Zhan, D.-C. Forward compatible few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9046–9056, 2022a.

Zhou, D.-W., Ye, H.-J., and Zhan, D.-C. Few-shot class-incremental learning by sampling multi-phase tasks. *arXiv preprint arXiv:2203.17030*, 2022b.

Zhu, F., Zhang, X.-Y., Wang, C., Yin, F., and Liu, C.-L. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5871–5880, 2021a.

Zhu, K., Cao, Y., Zhai, W., Cheng, J., and Zha, Z.-J. Self-promoted prototype refinement for few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6801–6810, 2021b.

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

A. DATASET DETAILS

Following ALICE (Peng et al., 2022), we conduct evaluation of MASIL on three widely used benchmark datasets:

- **CIFAR-100** (Krizhevsky et al., 2009) consists of 100 classes in total with color images of size 32×32. Each class consists of 500 images for training and 100 images for testing. The base session ($t = 0$) consists of 60 classes and the rest 40 classes contributed for 8 incremental session with 5-way 5-shot setting (i.e. 5 images for each of the 5 classes) for $1 \leq t \leq 8$.
- **miniImageNet** (Russakovsky et al., 2015) is a variant of ImageNet() with color images of size 84×84. It also consits of same number of classes as CIFAR-100 and same number of images in train and test, resulting in the same configuration for base and incremental sessions.
- **CUB-200** (Wah et al., 2011) consists of 11,788 images (size 224×224) in total spanning across 200 classes. There are 5,994 images in train and 5,794 images in test. Base session ($t = 0$) consists of 100 classes and rest 100 classes contributed towards 10 incremental session ($1 \leq t \leq 10$) with 10-way 5-shot setting (5 images for 10 classes each).

Dataset	Base Session		Incremental Session	
	Epcohs	Learning Rate	Iterations	Learning Rate
CIFAR-100	200	0.25	20-80	0.25
miniImageNet	500	0.25	120-190	0.025
CUB-200	80	0.025	80-150	0.05

Table 4. Training Details for Base and Incremental Session

B. IMPLEMENTATION DETAILS

Backbone Architecture: Existing works in FSCIL have leveraged ResNet-18, ResNet-12, ResNet-20 (He et al., 2016) as the backbone network for feature extractor. Following ALICE (Peng et al., 2022), we use ResNet-18 as the backbone network for feature extractor on top which the two layer MLP for projecting the features as the classification layer is trained for base and incremental sessions.

Concept Factorization: We used low rank factorization variable $v = 64$ for CIFAR and miniImageNet, and $v = 72$ for CUB-200. For τ function, it corresponds to randomly choosing 10 cropped patches of size 18×18 on CIFAR-100 and miniImageNet and patch size of 64×64 for CUB-200. We didn't use the *scikit-learn implementation* (Pedregosa et al., 2011) of NMF, we leverage the work of (Fel et al., 2022; Huang et al., 2016), which uses *Jax* (Blondel et al., 2021; Pedregosa et al., 2011) implementation of ADMM(Boyd et al., 2011) using *Jaxopt* library. We convert the *Jax* array to tensor array to be able to combine with the tensor array obtained from Pytorch on the gradient $\frac{\partial A}{\partial X}$ and compute the gradient with respect to input images i.e. $\frac{\partial P}{\partial X}, \frac{\partial Q}{\partial X}$.

Training Details: Data augmentation strategies like random crop, horizontal flip, rotation, brightness variation, cutout, resizing, flipping and color jittering were all applied following the recent works (Peng et al., 2022; Tao et al., 2020b; Zhang et al., 2021). Additionally, we adopted the standard data pre-processing as in (Peng et al., 2022). With a variation of number of epochs and iterations for base and incremental session across three datasets we keep the batch size of 512 for base session and 64 for incremental session during simplex finetuning as explained in Section 4.4. Dataset wise epochs/iterations and learning rates for base and incremental session is given in Table 4. Additionally, we use SGDR (Loshchilov & Hutter, 2016) with momentum as optimizer which uses cosine annealing strategy to reduce learning rate. Our code will be publicly available upon acceptance.

C. Additional Results

We continued summarising results for CUB-200 comparing various methods with ***MASIL***. Although improvement in last session is very small +0.14% as compared to strongest baseline ALICE (Peng et al., 2022). But we are consistently better

Methods	Session Accuracy (%) (↑)											Average Acc. (↑)	Relative Improvement
	0	1	2	3	4	5	6	7	8	9	10		
iCaRL (Rebuffi et al., 2017)	68.68	52.65	48.61	44.16	36.62	29.52	27.83	26.26	24.01	23.89	21.16	36.67	**+39.08**
EEIL (Castro et al., 2018)	68.68	53.63	47.91	44.20	36.30	27.46	25.93	24.70	23.95	24.13	22.11	36.27	**+38.13**
NCM (Hou et al., 2019)	68.68	57.12	44.21	28.78	26.71	25.66	24.62	21.52	20.12	20.06	19.87	32.49	**+40.37**
Fixed classifier (Pernici et al., 2021)	68.47	51.00	45.42	40.76	35.90	33.18	27.23	24.24	21.18	17.34	16.20	34.63	**+44.04**
D-NegCosine (Liu et al., 2020)	74.96	70.57	66.62	61.32	60.09	56.06	55.03	52.78	51.50	50.08	48.47	58.86	**+11.77**
D-DeepEMD (Zhang et al., 2020)	75.35	70.69	66.68	62.34	59.76	56.54	54.61	52.52	50.73	49.20	47.60	58.73	**+12.64**
D-Cosine (Vinyals et al., 2016)	75.52	70.95	66.46	61.20	60.86	56.88	55.40	53.49	51.94	50.93	49.31	59.36	**+10.93**
DeepInv (Yin et al., 2020)	75.90	70.21	65.36	60.14	58.79	55.88	53.21	51.27	49.38	47.11	45.67	57.54	**+14.57**
TOPIC (Tao et al., 2020b)	68.68	62.49	54.81	49.99	45.25	41.40	38.35	35.36	32.22	28.31	26.28	43.92	**+33.96**
IDLVQ (Chen & Lee, 2020)	77.37	74.72	70.28	67.13	65.34	63.52	62.10	61.54	59.04	58.68	57.81	65.23	**+2.43**
SPPR (Zhu et al., 2021b)	68.68	61.85	57.43	52.68	50.19	46.88	44.65	43.07	40.17	39.63	37.33	49.32	**+22.91**
(Cheraghian et al., 2021)	68.78	59.37	59.32	54.96	52.58	49.81	48.09	46.32	44.33	43.43	43.23	51.84	**+17.01**
CEC (Zhang et al., 2021)	75.85	71.94	68.50	63.50	62.43	58.27	57.73	55.81	54.83	53.52	52.28	61.33	**+7.96**
LIMIT (Zhou et al., 2022b)	76.32	74.18	**72.68**	69.19	68.79	65.64	63.57	62.69	61.47	60.44	58.45	66.67	**+1.79**
MgSvF (Zhao et al., 2021)	72.29	70.53	67.00	64.92	62.67	61.89	59.63	59.15	57.73	55.92	54.33	62.37	**+5.91**
MetaFSCIL (Chi et al., 2022)	75.9	72.41	68.78	64.78	62.96	59.99	58.3	56.85	54.78	53.82	52.64	61.93	**+7.6**
FACT (Zhou et al., 2022a)	75.90	73.23	70.84	66.13	65.56	62.15	61.74	59.83	58.41	57.89	56.94	64.42	**+3.3**
Data-free replay(Liu et al., 2022)	75.90	72.14	68.64	63.76	62.58	59.11	57.82	55.89	54.92	53.58	52.39	61.52	**+7.85**
ALICE (Peng et al., 2022)	77.40	72.70	70.60	67.20	65.90	63.40	62.90	61.90	60.50	**60.60**	60.10	65.75	**+0.14**
SSFE-Net (Pan et al., 2023)	76.38	72.11	68.82	64.77	63.59	60.56	59.84	58.93	57.33	56.23	54.28	62.98	**+5.96**
NC-FSCIL (Yang et al., 2023)	80.45	75.98	72.30	70.28	68.17	65.16	64.43	63.25	60.66	60.01	59.44	67.28	**+0.8**
MASIL(Ours)	**80.50**	**76.02**	72.25	**70.30**	**68.85**	**65.72**	**64.45**	**63.28**	**60.80**	**60.60**	**60.24**	**67.54**	

Table 5. Performance comparison on CUB-200 with ResNet-18 as backbone architecture under *10-way 5-shot* FSCIL setting. Table denotes the accuracy in each session, average accuracy across sessions and "Relative Improvement" denotes the improvement of our method in the last session. Methods above separating line are CIL methods for FSCIL as in (Tao et al., 2020b) and (Zhang et al., 2021)

in average accuracy and session wise accuracy (except only two sessions 2 and 8). On average accuracy we outperform ALICE (Peng et al., 2022) by +1.79% as shown in the Table 5. To further analyze the underlying reason for performance improvement because of fine tuning of classifier weights obtained from concept basis, we calculated the average cosine similarity of the concept basis c_i with all c_j, where $j \neq i$ for all the three datasets as given in the Table 6. Formally it is calculated as:

$$\frac{1}{K(K-1)} \sum_{i=1}^{K} \sum_{j=1, j \neq i}^{K} c_i \cdot c_j \tag{15}$$

Dataset	Average Cosine Similarity
miniImageNet	-5.22e-4
CIFAR-100	-8.78e-3
CUB-200	-4.54e-4

Table 6. Calculated cosine similarity among concept basis for each of the three benchmark datasets

These entries are almost close to zeros resulting in the concept basis which are non-overlapping and non-repetitive and hence can induce the unique classifier weights correspond to novel classes, that can be represented in terms of their combination.