Attaining Class-level Forgetting in Pretrained Model using Few Samples

Pravendra Singh[1,⋆][0000−0003−1001−2219], Pratik Mazumder[2,⋆][0000−0003−1103−1884], Mohammed Asad Karim[3,⋆][0000−0001−9664−8706]

[1]IIT Roorkee, India [2]IIT Kanpur, India [3]Carnegie Mellon University, USA

pravendra.singh@cs.iitr.ac.in, pratikm@cse.iitk.ac.in, mkarim2@cs.cmu.edu

Abstract. In order to address real-world problems, deep learning models are jointly trained on many classes. However, in the future, some classes may become restricted due to privacy/ethical concerns, and the restricted class knowledge has to be removed from the models that have been trained on them. The available data may also be limited due to privacy/ethical concerns, and re-training the model will not be possible. We propose a novel approach to address this problem without affecting the model's prediction power for the remaining classes. Our approach identifies the model parameters that are highly relevant to the restricted classes and removes the knowledge regarding the restricted classes from them using the limited available training data. Our approach is significantly faster and performs similar to the model re-trained on the complete data of the remaining classes.

1 Introduction

There are several real-world problems in which deep learning models have exceeded human-level performance. This has led to a wide deployment of deep learning models. Deep learning models generally train jointly on a number of categories/classes of data. However, the use of some of these classes may get restricted in the future (restricted classes), and a model with the capability to identify these classes may violate legal/privacy concerns. Individuals and organizations are becoming increasingly aware of these issues leading to an increasing number of legal cases on privacy issues in recent years. In such situations, the model has to be stripped of its capability to identify these categories (Class-level Forgetting). Due to legal/privacy concerns, the available training data may also be limited. In such situations, the problem becomes even more difficult to solve in the absence of the full training data. Real world problems such as incremental and federated learning also suffer from this problem as discussed in Sec. 3. We present a "Restricted Category Removal from Model Representations with Limited Data" (RCRMR-LD) problem setting that simulates the above problem. In this paper, we propose to solve this problem in a fast and efficient manner.

⋆ All the authors contributed equally.

The objective of the RCRMR-LD problem is to remove the information regarding the restricted classes from the network representations of all layers using the limited training data available without affecting the ability of the model to identify the remaining classes. If we have access to the full training data, then we can simply exclude the restricted class examples from the training data and perform a full training of the model from scratch using the abundant data (FDR - full data retraining). However, the RCRMR-LD problem setting is based on the scenario that the directive to exclude the restricted classes is received in the future after the model has already been trained on the full data and now only a limited amount of training data is available to carry out this process. Since only limited training data is available in our RCRMR-LD problem setting, the FDR model violates our problem setting and is therefore, not a solution to our RCRMR-LD problem setting. Simply training the network from scratch on only the limited training data of the remaining classes will result in severe overfitting and significantly affect the model performance (Baseline 2, as shown in Table 1).

Another possible solution to this problem is to remove the weights of the fully-connected classification layer of the network corresponding to the excluded classes such that it can no longer classify the excluded classes. However, this approach suffers from a serious problem. Since, in this approach, we only remove some of the weights of the classification layer and the rest of the model remains unchanged, the model still contains the information required for recognizing the excluded classes. This information can be easily accessed through the features that the model extracts from the images and, therefore, we can use these features for performing classification. In this paper, we use a nearest prototype-based classifier to demonstrate that the model features still contain information regarding the restricted classes. Specifically, we use the model features of the examples from the limited training data to compute the average class prototype for each class and create a nearest class prototype-based classifier using them. Next, for any given test image, we extract its features using the model and then find the class prototype closest to the given test image. This nearest class prototype-based classifier performs close to the original fully-connected classifier on the excluded classes as shown in Table 1 (Baseline 1). Therefore, even after using this approach, the resulting model still contains information regarding the restricted classes. Another possible approach can be to apply the standard fine-tuning approach to the model using the limited available training data of the remaining classes (Baseline 8). However, fine-tuning on such limited training data is not able to sufficiently remove the restricted class information from the model representations (see Table 1), and aggressive fine-tuning on the limited training data may result in overfitting.

Considering the problems faced by the naive approaches mentioned above, we propose a novel "Efficient Removal with Preservation" (ERwP) approach to address the RCRMR-LD problem. First, we propose a novel technique to identify the model parameters that are highly relevant to the restricted classes, and to the best of our knowledge, there are no existing prior works for finding such class-specific relevant parameters. Next, we propose a novel technique that optimizes the model on the limited available training data in such a way that the restricted class information is discarded from the restricted class relevant parameters, and these parameters are reused for the remaining classes.

To the best of our knowledge, this is the first work that addresses the RCRMR-LD problem. We also propose several baseline approaches for this problem (see Sec. 6). Our proposed approach significantly outperforms all the proposed baseline approaches. Our proposed approach requires very few epochs to address the RCRMR-LD problem and is, therefore, very fast ($\sim 200\times$ on ImageNet) and efficient. The model obtained after applying our approach forgets the excluded classes to such an extent that it behaves as though it was never trained on examples from the excluded classes. The performance of our model is very similar to the full data retraining (FDR) model (see Sec. 8.1 in the manuscript and Fig. 2 in the supplementary material). We also propose the performance metrics needed to evaluate the performance of any approach for the RCRMR-LD problem.

2 Problem Setting

In this work, we present the *restricted category removal from model representations with limited data (RCRMR-LD)* problem setting, in which a deep learning model M_o trained on a specific dataset has to be modified to exclude information regarding a set of restricted/excluded classes from all layers of the deep learning model without affecting its identification power for the remaining classes (see Fig. 1). The classes that need to be excluded are referred to as the restricted/excluded classes. Let $\{C_1^e, C_2^e, ..., C_{N_e}^e\}$ be the restricted/excluded classes, where N_e refers to the number of excluded classes. The remaining classes of the dataset are the remaining/non-excluded classes. Let $\{C_1^{ne}, C_2^{ne}, ..., C_{N_{ne}}^{ne}\}$ be the non-excluded classes, where N_{ne} refers to the number of remaining/non-excluded classes. Additionally, we only have access to a limited amount of training data for the restricted classes and the remaining classes, for carrying out this process. Therefore, any approach for addressing this problem can only utilize this limited training data.

3 RCRMR-LD Problem in Real World Scenarios

A real-world scenario where our proposed RCRMR-LD problem can arise is the incremental learning setting [21,16], where the model receives training data in the form of sequentially arriving tasks. Each task contains a new set of classes. During a training session t, the model receives the task t for training and cannot access the full data of the previous tasks. Instead, the model has access to very few exemplars of the classes in the previous tasks. Suppose before training a model on training session t, it is noticed that some classes from a previous task ($< t$) have to be removed from the model since those classes have become restricted due to privacy or ethical concerns. In this case, only a limited number of exemplars are available for all these previous classes (restricted and remaining). This demonstrates that the RCRMR-LD problem is present in the incremental

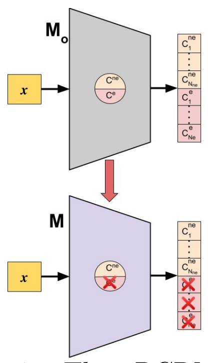

Fig. 1: The RCRMR-LD problem setting aims to remove the information regarding the restricted/excluded classes ($\{C_1^e, .., C_{N_e}^e\}$) from all layers of a trained model M_o while preserving its predictive power for the remaining classes ($\{C_1^{ne}, .., C_{N_{ne}}^{ne}\}$) using limited training data. The category removal (denoted by a red cross) has to take place at the classifier level (denoted as squares for each output logit) and at the feature/representation level (denoted as a circle)

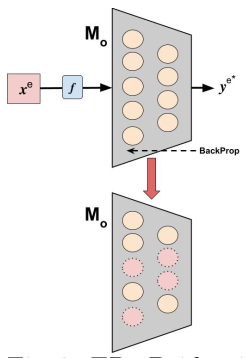

Fig. 2: ERwP identifies those parameters in the model that are highly relevant to the restricted classes. To obtain these parameters, ERwP modifies training images from a restricted class using a data augmentation f and performs backpropagation using the classification loss on these training images. ERwP then studies the gradient update that each parameter receives in this process in order to identify the highly relevant parameters for the restricted classes (denoted by dotted circles)

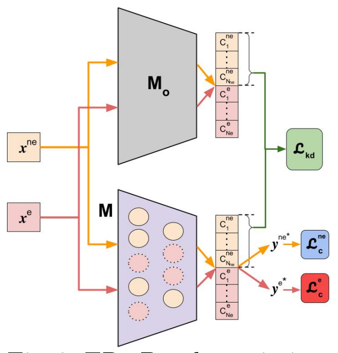

Fig. 3: ERwP only optimizes the restricted class relevant parameters in the model (denoted by dotted circles). ERwP uses $\mathcal{L}_c^e$, $\mathcal{L}_c^{ne}$ and $\mathcal{L}_{kd}$ losses to remove the restricted class information from the model while preserving its performance on the remaining classes. $\mathcal{L}_c^e$ and $\mathcal{L}_c^{ne}$ denote the classification loss on the restricted class training examples and the remaining class training example, respectively. $\mathcal{L}_{kd}$ denotes the knowledge distillation-based regularization loss that preserves the logits corresponding to only the remaining classes for all the training examples

learning setting. We experimentally demonstrate in Sec. 8.3, how our approach can address the RCRMR-LD problem in the incremental learning setting.

Let us consider another example. The EU GDPR laws require a data provider to remove information about an individual from a dataset upon that individual's request. In face recognition, this may lead to cases where the model has to be retrained from scratch, leaving out the training data for the restricted classes. In many such cases, it may be highly impractical and inefficient for the model creators to retrain the entire model from scratch. The RCRMR problem simulates this problem setting. Other examples of this problem include ethical AI concerns where protected classes (pregnant women, prisoners, children, etc.) need to be removed. There can also be other real-world scenarios, such as federated learning, where our RCRMR-LD problem can arise. Please refer to Sec. 1 in the supplementary material for more details.

4 Proposed Method

4.1 Method Description

Let, B refer to a mini-batch (of size S) from the available limited training data, and B contains training datapoints from the restricted/excluded classes $(\{(x_i^e, y_i^e)|(x_1^e, y_1^e), ..., (x_{S_e}^e, y_{S_e}^e)\})$ and from the remaining/non-excluded classes $(\{(x_j^{ne}, y_j^{ne})|(x_1^{ne}, y_1^{ne}), ..., (x_{S_{ne}}^{ne}, y_{S_{ne}}^{ne})\})$. Here, (x_i^e, y_i^e) refers to a training datapoint from the excluded classes where x_i^e is an image, y_i^e is the corresponding label and $y_i^e \in \{C_1^e, C_2^e, ..., C_{N_e}^e\}$. (x_j^{ne}, y_j^{ne}) refers to a training datapoint from the non-excluded classes where x_j^{ne} is an image, y_j^{ne} is the corresponding label and $y_j^{ne} \in \{C_1^{ne}, C_2^{ne}, ..., C_{N_{ne}}^{ne}\}$. Here, S_e and S_{ne} refer to the number of training examples in the mini-batch from the excluded and non-excluded classes, respectively, such that $S = S_e + S_{ne}$. N_e and N_{ne} refer to the number of excluded and non-excluded classes, respectively. Let M refer to the deep learning model being trained using our approach and M_o is the original trained deep learning model.

In a trained model, some of the parameters may be highly relevant to the restricted classes, and the performance of the model on the restricted classes is mainly dependent on such highly relevant parameters. Therefore, in our approach, we focus on removing the excluded class information from these restricted class relevant parameters. Since the model is trained on all the classes jointly, the parameters are shared across the different classes. Therefore identifying these class-specific relevant parameters is very difficult. Let us consider a model that is trained on color images of a class. If we now train it on grayscale images of the class, then the model has to learn to identify these new images. In order to do so, the parameters relevant to that class will receive large gradient updates as compared to the other parameters (see Sec. 5.1 in the supplementary material). We propose a novel approach for identifying the relevant parameters for the restricted classes using this idea. For each restricted class, we choose the training images belonging to that class from the limited available training data. Next, we apply a grayscale data augmentation technique/transformation f to these images so that these images become different from the images that the original model was earlier trained on (assuming that the original model has not been trained on grayscale images). We can also use other data augmentation techniques that are not seen during the training process of the original model and that do not change the class of the image (refer to Sec. 5.6 in the supplementary material). Next, we combine the predictions for each training image into a single average prediction and perform backpropagation. During the backpropagation, we study the gradients for all the parameters in each layer of the model. Accordingly, we select the parameters with the highest absolute gradient as the relevant parameters for the corresponding restricted class. Specifically, for a given restricted class, we choose all the parameters from each network layer such that pruning (zeroing out) these parameters will result in the maximum degradation of model performance on that restricted class. We provide a detailed description of the process for identifying the restricted class relevant parameters in Sec. 2 of the supplementary material. The combined set of the relevant parameters for all the excluded classes is referred to as the restricted/excluded class relevant parameters Θ_{exrel} (see Fig. 2). Please note that we use this process only to identify Θ_{exrel}, and we do not update the model parameters during this step.

Pruning the relevant parameters for a restricted class can severely impact the performance of the model for that class (see Sec. 5.1 in the supplementary material). However, this may also degrade the performance of the model on the non-excluded classes because the parameters are shared across multiple classes. Therefore, we cannot address the RCRMR-LD problem by pruning the relevant parameters of the excluded classes. Finetuning these parameters on the limited remaining class data will also not be able to sufficiently remove the restricted class information from the model. Based on this, we propose to address the RCRMR-LD problem by optimizing the relevant parameters of the restricted classes to remove the restricted class information from them and to reuse them for the remaining classes.

After identifying the restricted class relevant parameters, our ERwP approach uses a classification loss based on the cross-entropy loss function to optimize the restricted class relevant parameters of the model on each mini-batch (see Fig. 3). We know that the gradient ascent optimization algorithm can be used to maximize a loss function and encourage the model to perform badly on the given input. Therefore, we use the gradient ascent optimization on the classification loss for the limited restricted class training examples to remove the information regarding the restricted classes from Θ_{exrel}. We achieve this by multiplying the classification loss for the training examples from the excluded classes by a constant negative factor of -1. We also optimize Θ_{exrel} using the gradient descent optimization on the classification loss for the limited remaining class training example, in order to reuse these parameters for the remaining classes. We validate using this approach through various ablation experiments as shown in Sec. 5.2 in the supplementary material. The classification loss for the examples from the excluded and non-excluded classes and the overall classification loss for each mini-batch are defined as follows.

$$\mathcal{L}_c^e = \sum_{i=1}^{S_e} -1 * \ell(y_i^e, y_i^{e*}) \tag{1}$$

$$\mathcal{L}_c^{ne} = \sum_{j=1}^{S_{ne}} \ell(y_j^{ne}, y_j^{ne*}) \tag{2}$$

$$\mathcal{L}_c = \frac{1}{S}(\mathcal{L}_c^e + \mathcal{L}_c^{ne}) \tag{3}$$

Where, y_i^{e*} and y_j^{ne*} refer to the predicted class labels for x_i^e and x_j^{ne}, respectively. $\ell(.,.)$ refers to the cross-entropy loss function. $\mathcal{L}_c^e$ and $\mathcal{L}_c^{ne}$ refer to the classification loss for the examples from the excluded and non-excluded classes in the mini-batch, respectively. $\mathcal{L}_c$ refers to the overall classification loss for each mini-batch.

Since all the network parameters were jointly trained on all the classes (restricted and remaining), the restricted class relevant parameters also contain information relevant to the remaining classes. Applying the above process alone will still harm the model's predictive power for the non-excluded classes (as shown in Sec. 5.2, Table 2 in the supplementary material). This is because the gradient ascent optimization strategy will also erase some of the relevant information regarding the remaining classes. Further, applying $\mathcal{L}_c^{ne}$ on the limited training examples of the remaining classes will lead to overfitting and will not be effective enough to fully preserve the model performance on the remaining classes. In order to ensure that the model's predictive power for the non-excluded classes does not change, we use a knowledge distillation-based regularization loss. Knowledge distillation [14] ensures that the predictive power of the teacher network is replicated in the student network. In this problem setting, we want the final model to replicate the same predictive power of the original model for the remaining classes. Therefore, given any training example, we use the knowledge distillation-based regularization loss to ensure that the output logits produced by the model corresponding to only the non-excluded classes remain the same as that produced by the original model. We apply the knowledge distillation loss to the limited training examples from both the excluded and remaining classes, to preserve the non-excluded class logits of the model for any input image. We validate this knowledge distillation-based regularization loss through ablation experiments as shown in Table 2 in the supplementary material. We use the original model M_o (before applying ERwP) as the teacher network and the current model M being processed by ERwP as the student network, for the knowledge distillation process. Please note that the optimization for this loss is also carried out only for the restricted class relevant parameters of the model. Let KD refer to the knowledge distillation loss function. It computes the Kullback-Liebler (KL) divergence between the soft predictions of the teacher and the student networks and can be defined as follows:

$$KD(p_s, p_t) = KL(\sigma(p_s), \sigma(p_t)) \tag{4}$$

where, $\sigma(.)$ refers to the softmax activation function that converts logit a_i for each class i into a probability by comparing a_i with logits of other classes a_j, i.e., $\sigma(a_i) = \frac{exp^{a_i/\kappa}}{\sum_j exp^{a_j/\kappa}}$. κ refers to the temperature [14], KL refers to the KL-Divergence function. p_s, p_t refer to the logits produced by the student network and the teacher network, respectively.

The knowledge distillation-based regularization losses in our approach are defined as follows.

$$\mathcal{L}_{kd}^e = \sum_{i=1}^{S_e} KD(M(x_i^e)[C^{ne}], M_o(x_i^e)[C^{ne}]) \tag{5}$$

$$\mathcal{L}_{kd}^{ne} = \sum_{j=1}^{S_{ne}} KD(M(x_j^{ne})[C^{ne}], M_o(x_j^{ne})[C^{ne}]) \tag{6}$$

$$\mathcal{L}_{kd} = \frac{1}{S}(\mathcal{L}_{kd}^{e} + \mathcal{L}_{kd}^{ne}) \tag{7}$$

Where, $M(\#)[C^{ne}]$ and $M_o(\#)[C^{ne}]$ refer to the output logits corresponding to the remaining classes produced by M and M_o, respectively. $\#$ can be either x_i^e or x_j^{ne}. $\mathcal{L}_{kd}^{e}$ and $\mathcal{L}_{kd}^{ne}$ refer to knowledge distillation-based regularization loss for the examples from the excluded and non-excluded classes, respectively. $\mathcal{L}_{kd}$ refers to the overall knowledge distillation-based regularization loss for each mini-batch. The $\mathcal{L}_{kd}^{ne}$ loss helps in preserving the model performance for the non-excluded classes. If some of the restricted classes are similar to some of the remaining/non-excluded classes, the $\mathcal{L}_{kd}^{e}$ loss ensures that the model performance on the remaining classes is not degraded due to this similarity. This is because the $\mathcal{L}_{kd}^{e}$ loss preserves the logits corresponding to the non-excluded classes for the restricted class training examples.

The total loss $\mathcal{L}_{erwp}$ of our approach for each mini-batch is defined as follows.

$$\mathcal{L}_{erwp} = \mathcal{L}_c + \beta\mathcal{L}_{kd} \tag{8}$$

Where, β is a hyper-parameter that controls the contribution of the knowledge distillation-based regularization loss. We use this loss for fine-tuning the model for very few epochs.

5 Related Work

Pruning [1,25,11,12] involves removing redundant and unimportant weights [2,7,9] or filters [10,13,17] from a deep learning model without affecting the model performance. Pruning approaches generally identify the important parameters in the network and remove the unimportant parameters. In the RCRMR-LD problem setting, the restricted class relevant parameters are also important parameters. However, we empirically observe that pruning the restricted class relevant parameters severely affects the model performance for the remaining classes since the parameters are shared among all the classes. Therefore, pruning approaches cannot be applied in the RCRMR-LD problem setting.

In the incremental learning setting [15,24,3,18], the objective is to preserve the predictive power of the model for previously seen classes while learning a new set of classes. The work in [22] uses a topology-preserving loss to prevent catastrophic forgetting by maintaining the topology in feature space. In contrast to the incremental learning setting, our proposed RCRMR-LD problem setting involves removing the information regarding specific classes from the pre-trained model while preserving the predictive power of the model for the remaining classes.

There has been some research involving deleting individual data points from trained machine learning models such as [5,6]. The work in [5] deals with data deletion in the context of a machine learning algorithm and model. It shows how to remove the influence of a data point from a k-means clustering model. Our work focuses on restricted category removal from deep learning models with limited data. Therefore, the approaches proposed in [5] cannot be applied to RCRMR-LD. Further, the objective of data deletion is to remove a data point without affecting the model performance on any classes, including the class of the deleted data point. This is in stark contrast to our RCRMR-LD problem, where the objective is to remove the knowledge of a set of classes or categories from the model. Further, data deletion methods will require access to the entire training data of a class in order to remove the entire knowledge of a class (refer to the appendix A.1. of [6]). This is because deep learning models have a high generalization power even on unseen examples of a class on which they have been trained, and simply deleting a few data points of a class from the knowledge base of the model will not be enough to forget that class. However, in our proposed problem setting, only a limited number of training examples are present for any class. Therefore, data-deletion approaches are not solutions to our proposed RCRMR-LD problem setting. This is why we have not applied these approaches in our problem setting.

Privacy-preserving deep learning [19,4,8] involves learning representations that incorporate features from the data relevant to the given task and ignore sensitive information (such as the identity of a person). The authors in [20] propose a simple variational approach for privacy-preserving representation learning. In contrast to existing privacy preservation works, the objective of the RCRMR-LD problem setting is to achieve class-level forgetting, i.e., if a class is declared as private/restricted, then all information about this class should be removed from the model trained on it, without affecting its ability to identify the remaining classes. To the best of our knowledge, this is the first work to address the class-level forgetting problem in the limited data regime, i.e., RCRMR-LD problem setting.

6 Baselines

We propose 9 baseline models for the RCRMR-LD problem and compare our proposed approach with them. The baseline 1 involves deleting the weights of the fully-connected classification layer corresponding to the excluded classes. Baselines 2, 3, 4, 5 involve training the model on the limited training data of the remaining classes. Baselines 2 and 4 both involve training a new model from scratch using the limited training examples of only the non-excluded classes, but baseline 4 initializes the model with the weights of the original model. Baselines 3 and 5 are similar to baselines 2 and 4, respectively, but also use a knowledge distillation loss to preserve the non-excluded class logits. Baselines 6, 7, 8, 9 involve fine-tuning the model on the available limited training data. Baselines 6 and 7 fine-tune the original model on the available limited data of all the classes after mapping the restricted classes to a single excluded class, but baseline 7 also uses a knowledge distillation loss. Baselines 8 and 9 fine-tune the original model on the available limited data of only the remaining classes, but baseline 9 also uses a knowledge distillation loss. Please refer to Sec. 3 in the supplementary material for details about the baselines.

7 Performance Metrics

In the RCRMR-LD problem setting, we propose three performance metrics to validate the performance of any method: forgetting accuracy ($\texttt{FA}_\texttt{e}$), forgetting prototype accuracy ($\texttt{FPA}_\texttt{e}$), and constraint accuracy ($\texttt{CA}_\texttt{ne}$). The forgetting accuracy refers to the fully-connected classification layer accuracy of the model for the excluded classes. The forgetting prototype accuracy refers to the nearest class prototype-based classifier accuracy of the model for the excluded classes. $\texttt{CA}_\texttt{ne}$ refers to the fully-connected classification layer accuracy of the model for the non-excluded classes.

In order to judge any approach on the basis of these metrics, we follow the following sequence. First, we analyze the constraint accuracy ($\texttt{CA}_\texttt{ne}$) of the model produced by the given approach to verify if the approach has preserved the prediction power of the model for the non-excluded classes. $\texttt{CA}_\texttt{ne}$ of the model should be close to that of the original model. If this condition is not satisfied, then the approach is not suitable for this problem, and we need not analyze the other metrics. This is because if the constraint accuracy is not maintained, then the overall usability of the model is hurt significantly. Next, we analyze the forgetting accuracy ($\texttt{FA}_\texttt{e}$) of the model to verify if the excluded class information has been removed from the model at the classifier level. $\texttt{FA}_\texttt{e}$ of the model should be as close to 0% as possible. Finally, we analyze the forgetting prototype accuracy ($\texttt{FPA}_\texttt{e}$) of the model to verify if the excluded class information has been removed from the model at the feature level. $\texttt{FPA}_\texttt{e}$ of the model should be significantly less than that of the original model. However, the $\texttt{FPA}_\texttt{e}$ will not become zero since any trained model will learn to extract meaningful features, which will help the nearest class prototype-based classifier to achieve some non-negligible accuracy even on the excluded classes. Therefore, for a better analysis of the level of forgetting of the excluded classes at the feature level, we compare the $\texttt{FPA}_\texttt{e}$ of the model with the $\texttt{FPA}_\texttt{e}$ of the FDR model. The FDR model is a good candidate for this analysis since it has not been trained on the excluded classes (only trained on the complete dataset of the remaining classes), and it still achieves a non-negligible performance of the excluded classes (see Sec 8.1). However, it should be noted that this comparison is only for analysis and the comparison is not fair since the FDR model needs to train on the entire dataset (except the excluded classes).

A naive approach for measuring the capability of any approach for removing the excluded class information in this problem setting is to only consider how low the forgetting accuracy ($\texttt{FA}_\texttt{e}$) of the model for the excluded classes drops to after the excluded category removal process. However, using $\texttt{FA}_\texttt{e}$ alone may be misleading since zero or random forgetting accuracy ($\texttt{FA}_\texttt{e}$) for a excluded class does not mean that the excluded class information has been removed from all layers of the model. In order to understand this point, let us consider the weight deletion (WD) baseline (baseline 1) that simply deletes the classification layer weights corresponding to the excluded classes and achieves a forgetting accuracy ($\texttt{FA}_\texttt{e}$) of 0% for the excluded classes. However, this does not mean that the excluded class information has been removed from all the layers of the network since the rest of the network remains intact. Therefore, using only ($\texttt{FA}_\texttt{e}$) metric is not enough. Now, if we consider the forgetting prototype accuracy ($\texttt{FPA}_\texttt{e}$) of the WD model, we will observe that the $\texttt{FPA}_\texttt{e}$ of WD model is the same as that of the original model for the excluded classes. This clearly indicates that the excluded class information is still present in the layers of the network. Further, we also need to check whether the model performance for the remaining classes is maintained. We use our proposed constraint accuracy ($\texttt{CA}_\texttt{ne}$) of the non-excluded classes for this purpose. Therefore, the above discussion clearly demonstrates that a single metric is not effective in this problem setting.

8 Experiments

We have reported the experimental results for the CIFAR-100 and ImageNet-1k datasets in this section. We have also provided the results on the CUB-200 dataset in the supplementary material. Please refer to the supplementary material for the details regarding the datasets and implementation. We have provided the experimental results for the ablation experiments to validate the different components of our works in the supplementary material.

8.1 CIFAR-100 Results

We report the performance of different baselines and our proposed ERwP method on the RCRMR-LD problem using the CIFAR-100 dataset with different architectures in Table 1. We observe that the baseline 1 (weight deletion) achieves high constraint accuracy $\texttt{CA}_\texttt{ne}$ and 0% forgetting accuracy $\texttt{FA}_\texttt{e}$. But its forgetting prototype accuracy $\texttt{FPA}_\texttt{e}$ remains the same as the original model for all the three architectures, i.e., ResNet-20/56/164. Therefore, baseline 1 fails to remove the excluded class information from the model at the feature level. Baseline 2 is not able to preserve the constraint accuracy $\texttt{CA}_\texttt{ne}$ even though it performs full training on the limited excluded class data. Baseline 3 achieves higher $\texttt{CA}_\texttt{ne}$ than baseline 2, but the constraint accuracy is still too low. Baselines 4 and 5 demonstrate significantly better constraint accuracy than baseline 2 and 3, but their constraint accuracy is still significantly lower than the original model (except baseline 5 for ResNet-20). The baseline 5 with ResNet-20 maintains the constraint accuracy and achieves 0% forgetting accuracy $\texttt{FA}_\texttt{e}$ but its $\texttt{FPA}_\texttt{e}$ is still significantly high and, therefore, is unable to remove the excluded class information from the model at the feature level. The fine-tuning based baselines 6 and 7 are able to significantly reduce the forgetting accuracy $\texttt{FA}_\texttt{e}$ but their constraint accuracy $\texttt{CA}_\texttt{ne}$ drops significantly. The fine-tuning based baselines 8 and 9 only finetune the model on the limited remaining class data and as a result they are not able to sufficiently reduce either the $\texttt{FA}_\texttt{e}$ or the $\texttt{FPA}_\texttt{e}$.

Table 1: Experimental results on the CIFAR-100 dataset for RCRMR-LD

Methods	ResNet-20			ResNet-56			ResNet-164		
	$\mathtt{FA_e}$	$\mathtt{FPA_e}$	$\mathtt{CA_{ne}}$	$\mathtt{FA_e}$	$\mathtt{FPA_e}$	$\mathtt{CA_{ne}}$	$\mathtt{FA_e}$	$\mathtt{FPA_e}$	$\mathtt{CA_{ne}}$
Original	70.15%	65.25%	67.06%	70.80%	68.65%	69.88%	79.00%	76.40%	76.30%
No Training									
Baseline 1 - WD	0.00%	65.25%	69.88%	0.00%	68.65%	72.44%	0.00%	76.40%	78.23%
Full Train Schedule									
Baseline 2 - TSLNRC	0.00%	22.20%	31.55%	0.00%	20.20%	30.21%	0.00%	33.05%	40.65%
Baseline 3 - TSLNRC-KD	0.00%	27.55%	40.81%	0.00%	22.50%	32.26%	0.00%	38.55%	45.74%
Baseline 4 - TOLNRC	0.00%	50.85%	58.01%	0.00%	48.60%	57.81%	0.00%	51.55%	63.78%
Baseline 5 - TOLNRC-KD	0.00%	60.25%	67.85%	0.00%	51.25%	61.14%	0.00%	52.80%	63.75%
Only Fine-Tuning									
Baseline 6 - FOLMRCSC	24.25%	59.55%	64.03%	13.35%	60.25%	65.23%	15.40%	59.20%	71.06%
Baseline 7 - FOLMRCSC-KD	13.50%	58.80%	63.79%	12.75%	64.95%	63.41%	16.75%	65.30%	68.61%
Baseline 8 - FOLNRC	59.05%	64.30%	68.34%	66.90%	68.45%	70.11%	77.35%	75.85%	75.95%
Baseline 9 - FOLNRC-KD	57.99%	64.40%	68.40%	65.95%	68.40%	70.01%	73.30%	73.55%	75.99%
ERwP (Ours)	0.00%	48.06%	66.84%	0.00%	47.84%	69.32%	0.74%	56.23%	75.65%

Our proposed ERwP approach achieves a constraint accuracy $\mathtt{CA_{ne}}$ that is very close to the original model for all three architectures. It achieves close to 0% $\mathtt{FA_e}$. Further, it achieves a significantly lower $\mathtt{FPA_e}$ than the original model. Specifically, the $\mathtt{FPA_e}$ of our approach is significantly lower than that of the original model by absolute margins of 17.19%, 20.81%, and 20.17% for the ResNet-20, ResNet-56, and ResNet-164 architectures, respectively. The $\mathtt{FPA_e}$ for the FDR model is 44.20%, 45.40% and 51.85% for the ResNet-20, ResNet-56 and ResNet-164 architectures, respectively. Therefore, the $\mathtt{FPA_e}$ of our approach is close to that of the FDR model by absolute margins of 3.86%, 2.44% and 4.38% for the ResNet-20, ResNet-56 and ResNet-164 architectures, respectively. Therefore, our ERwP approach makes the model behave similar to the FDR model even though it was trained on only limited data from the excluded and remaining classes. Further, our ERwP requires only 10 epochs to remove the excluded class information from the model. Since the available limited training data is only 10% of the entire CIFAR-100 dataset, therefore, our ERwP approach is approximately $30 * 10 = 300\times$ faster than the FDR method that is trained on the full training data for 300 epochs.

The $\mathtt{FPA_e}$ accuracy obtained using ERwP is significantly lower than the original model, e.g., for the ResNet-56 architecture $\mathtt{FPA_e}$ of ERwP is 47.84% compared to 68.65% of the original model for the CIFAR-100 dataset using the ResNet-56 model. However, this does not indicate the presence of much restricted category information. This is because the process for obtaining the $\mathtt{FPA_e}$ accuracy involves creating prototypes from the limited training data of the restricted classes and the remaining classes and finding the nearest neighbor class. Therefore, this process is dependent on the features generated by the deep

Table 2: Experimental results on ImageNet-1k

Model	Methods	Top-1		Top-5	
		$\texttt{FA}_\texttt{e}$	$\texttt{CA}_\texttt{ne}$	$\texttt{FA}_\texttt{e}$	$\texttt{CA}_\texttt{ne}$
Res-18	Original	69.76%	69.76%	89.58%	89.02%
	ERwP	0.28%	69.13%	1.01%	88.93%
Res-50	Original	76.30%	76.11%	93.04%	92.84%
	ERwP	0.25%	75.45%	2.55%	92.39%
Mob-V2	Original	72.38%	70.83%	91.28%	90.18%
	ERwP	0.17%	70.81%	0.81%	89.95%

learning model. Deep learning models generally produce highly discriminative features that can be used to create good prototype classifiers even for classes that the models were not trained on. For example, in the few-shot learning setting, the model is generally trained only on the base classes and then evaluated on novel class episodes using a prototype-based classifier. The prototype-based classifier of the few-shot learning setting is very effective in classifying the novel classes even though the deep model, which was used to obtain the features for the prototypes, was never trained on the novel classes. The discriminative nature of the features produced by deep learning models is the main reason why ImageNet pre-trained model features are used to train classifiers for other datasets and settings, such as in zero-shot learning. In order to better appreciate the effectiveness of our approach, we also consider the FDR model, which has not seen any training data of the restricted classes and still achieves a $\texttt{FPA}_\texttt{e}$ accuracy close to that of our approach, e.g. FDR achieves a $\texttt{FPA}_\texttt{e}$ accuracy of 45.40% for the CIFAR-100 dataset using the ResNet-56 model while our approach achieves an $\texttt{FPA}_\texttt{e}$ accuracy of 47.84%. We provide this result as a reference to demonstrate that the non-zero accuracy of ERwP is due to the generalization power of deep CNNs and not due to the restricted classes information in the model. However, comparing FDR with our approach is not fair since FDR requires the full training data of the remaining classes, which violates the RCRMR-LD problem setting. Therefore, we have not provided the FDR results in the tables to maintain fairness.

8.2 ImageNet Results

Table 2 reports the experimental results for different approaches to RCRMR-LD problem over the ImageNet-1k dataset using the ResNet-18, ResNet-50 and MobileNet V2 architectures. Our proposed ERwP approach achieves a top-1 constraint accuracy $\texttt{CA}_\texttt{ne}$ that is very close to that of the original model by absolute margins of 0.63%, 0.66% and 0.02% for the ResNet-18, ResNet-50 and MobileNet V2 architectures, respectively. It achieves close to 0% top-1 forgetting accuracy $\texttt{FA}_\texttt{e}$ for all the three architectures. Therefore, our approach performs

Table 3: Performance of ERwP in incremental learning setting using ResNet-18

Model	$\texttt{FA}_{\texttt{e}}$	$\texttt{CA}_{\texttt{ne}}$
Original Model obtained after Session 4 [M4]	56.39%	58.32%
M4 modified with ERwP (Ours)	0.20%	59.93%

well even on the large-scale ImageNet-1k dataset. Further, our ERwP requires only 10 epochs to remove the excluded class information from the model. Since the available limited training data is only 5% of the entire ImageNet-1k dataset, therefore, our ERwP approach is approximately $20 * 10 = 200\times$ faster than the FDR method that is trained on the full data for 100 epochs.

8.3 RCRMR-LD Problem in Incremental Learning

In this section, we experimentally demonstrate how the RCRMR-LD problem in the incremental learning setting is addressed using our proposed approach. We consider an incremental learning setting on the CIFAR-100 dataset in which each task contains 20 classes. We use the BIC [23] method for incremental learning on this dataset. The exemplar memory size is fixed at 2000 as per the setting in [23]. In this setting, there are 5 tasks. Let us assume that the model (M4) has already been trained on 4 tasks (80 classes), and we are in the fifth training session. Suppose, at this stage, it is noticed that all the classes in the first task (20 classes) have become restricted and need to be removed before the model is trained on task 5. However, we only have a limited number of exemplars of the 80 classes seen till now, i.e., $2000/80 = 25$ per class. We apply our proposed approach to the model obtained after training session 4, and the results are reported in Table 3. The results indicate that our approach modified the model obtained after session 4, such that the forgetting accuracy of the restricted classes approaches 0% and the constraint accuracy of the remaining classes is not affected. In fact, the modified model behaves as if, it was never trained on the classes from task 1. We can now perform the incremental training of the modified model on task 5.

9 Conclusion

In this paper, we present a "Restricted Category Removal from Model Representations with Limited Data" problem in which the objective is to remove the information regarding a set of excluded/restricted classes from a trained deep learning model without hurting its predictive power for the remaining classes. We propose several baseline approaches and also the performance metrics for this setting. We propose a novel approach to identify the model parameters that are highly relevant to the restricted classes. We also propose a novel efficient approach that optimizes these model parameters in order to remove the restricted class information and re-use these parameters for the remaining classes.

References

1. Carreira-Perpinán, M.A., Idelbayev, Y.: "learning-compression" algorithms for neural net pruning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8532–8541 (2018)
2. Dong, X., Chen, S., Pan, S.J.: Learning to prune deep neural networks via layer-wise optimal brain surgeon. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 4860–4874. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
3. Douillard, A., Cord, M., Ollion, C., Robert, T., Valle, E.: Podnet: Pooled outputs distillation for small-tasks incremental learning. In: ECCV (2020)
4. Edwards, H., Storkey, A.: Censoring representations with an adversary. arXiv preprint arXiv:1511.05897 (2015)
5. Ginart, A., Guan, M.Y., Valiant, G., Zou, J.: Making ai forget you: Data deletion in machine learning. arXiv preprint arXiv:1907.05012 (2019)
6. Golatkar, A., Achille, A., Ravichandran, A., Polito, M., Soatto, S.: Mixed-privacy forgetting in deep networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 792–801 (2021)
7. Guo, Y., Yao, A., Chen, Y.: Dynamic network surgery for efficient dnns. Advances in Neural Information Processing Systems **29**, 1379–1387 (2016)
8. Hamm, J.: Minimax filter: Learning to preserve privacy from inference attacks. The Journal of Machine Learning Research **18**(1), 4704–4734 (2017)
9. Han, S., Mao, H., Dally, W.J.: Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149 (2015)
10. He, Y., Kang, G., Dong, X., Fu, Y., Yang, Y.: Soft filter pruning for accelerating deep convolutional neural networks. In: IJCAI International Joint Conference on Artificial Intelligence (2018)
11. He, Y., Dong, X., Kang, G., Fu, Y., Yan, C., Yang, Y.: Asymptotic soft filter pruning for deep convolutional neural networks. IEEE transactions on cybernetics **50**(8), 3594–3604 (2019)
12. He, Y., Liu, P., Wang, Z., Hu, Z., Yang, Y.: Filter pruning via geometric median for deep convolutional neural networks acceleration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4340–4349 (2019)
13. He, Y., Liu, P., Zhu, L., Yang, Y.: Meta filter pruning to accelerate deep convolutional neural networks. arXiv preprint arXiv:1904.03961 (2019)
14. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: NIPS Deep Learning and Representation Learning Workshop (2014), `https://fb56552f-a-62cb3a1a-s-sites.googlegroups.com/site/deeplearningworkshopnips2014/65.pdf`
15. Hou, S., Pan, X., Loy, C.C., Wang, Z., Lin, D.: Learning a unified classifier incrementally via rebalancing. In: CVPR. pp. 831–839 (2019)
16. Kemker, R., Kanan, C.: Fearnet: Brain-inspired model for incremental learning. In: International Conference on Learning Representations (2018), `https://openreview.net/forum?id=SJ1Xmf-Rb`
17. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient convnets. arXiv preprint arXiv:1608.08710 (2016)
18. Liu, Y., Schiele, B., Sun, Q.: Adaptive aggregation networks for class-incremental learning. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)

19. Louizos, C., Swersky, K., Li, Y., Welling, M., Zemel, R.: The variational fair autoencoder. arXiv preprint arXiv:1511.00830 (2015)
20. Nan, L., Tao, D.: Variational approach for privacy funnel optimization on continuous data. Journal of Parallel and Distributed Computing **137**, 17–25 (2020)
21. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2001–2010 (2017)
22. Tao, X., Chang, X., Hong, X., Wei, X., Gong, Y.: Topology-preserving class-incremental learning. In: ECCV (2020)
23. Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., Fu, Y.: Large scale incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 374–382 (2019)
24. 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: CVPR. pp. 6982–6991 (2020)
25. Zhang, T., Ye, S., Zhang, K., Tang, J., Wen, W., Fardad, M., Wang, Y.: A systematic dnn weight pruning framework using alternating direction method of multipliers. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 184–199 (2018)