F-OAL: Forward-only Online Analytic Learning with Fast Training and Low Memory Footprint in Class Incremental Learning

Huiping Zhuang[1]*, **Yuchen Liu**[2]*, **Run He**[1], **Kai Tong**[1], **Ziqian Zeng**[1],
Cen Chen[1,4]†, **Yi Wang**[3], **Lap-Pui Chau**[3]
[1]South China University of Technology, China
[2]The University of Hong Kong, Hong Kong SAR
[3]The Hong Kong Polytechnic University, Hong Kong SAR
[4]Pazhou Lab, Guangzhou, China
{hpzhuang,runhe,kaitong,zqzeng,cenchen}@scut.edu.cn,
liuyuchen@connect.hku.hk,
{yi-eie.wang, lap-pui.chau}@polyu.edu.hk

Abstract

Online Class Incremental Learning (OCIL) aims to train models incrementally, where data arrive in mini-batches, and previous data are not accessible. A major challenge in OCIL is Catastrophic Forgetting, i.e., the loss of previously learned knowledge. Among existing baselines, replay-based methods show competitive results but requires extra memory for storing exemplars, while exemplar-free (i.e., data need not be stored for replay in production) methods are resource-friendly but often lack accuracy. In this paper, we propose an exemplar-free approach—Forward-only Online Analytic Learning (F-OAL). Unlike traditional methods, F-OAL does not rely on back-propagation and is forward-only, significantly reducing memory usage and computational time. Cooperating with a pre-trained frozen encoder with Feature Fusion, F-OAL only needs to update a linear classifier by recursive least square. This approach simultaneously achieves high accuracy and low resource consumption. Extensive experiments on benchmark datasets demonstrate F-OAL's robust performance in OCIL scenarios. Code is available at: https://github.com/liuyuchen-cz/F-OAL

1 Introduction

Class Incremental Learning (CIL) updates the model incrementally in a task-by-task manner with new classes in the new task. Traditional CIL most plans for static offline datasets which historical data are accessible. However, with the rapid increase of social media and mobile devices, massive amount of image data have been produced online in a streaming fashion, and render training models on static data less applicable. To address this, Online Class Incremental Learning (OCIL) is developed taking an online constraint in addition to the existing CIL setting. OCIL is a more challenging CIL setting in which data come in mini-batches, and the model is trained only in one epoch (i.e., learning from one pass data stream) [14]. The model is required to achieve high accuracy, fast training time, and low resource consumption [25].

However, CIL techniques (including OCIL) suffer from Catastrophic Forgetting (CF) [27], also known as the erosion of previous knowledge when new data are introduced. The problem becomes

*Equal contribution.
†Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-1314

more severe in online scenarios since the model can only see data once. Two major factors contribute to CF: (1) Using the loss function to update the whole network leads to uncompleted feature capturing and diminished global representation [11]. (2) Using back-propagation to adjust linear classifier results in *recency bias*, which is a significantly imbalanced weight distribution, showing preference only on current learning data [15].

To address CF in an online setting, replay-based methods [9, 21] are the mainstream solution by preserving old exemplars and revisiting them in new tasks. This strategy has strong performance but is resource consuming, while exemplar-free methods [17, 20] have lower resource consumption but show less competitive results.

Recently, Analytic Continual Learning (ACL) [49] methods emerged as an exemplar-free branch, delivering encouraging outcomes. ACL methods pinpoint the iterative back-propagation as the main factor behind catastrophic forgetting and seek to address it through linear recursive strategies. Remarkably, for the first time, these methods achieve outcomes comparable to those utilizing replay-based techniques.

There are two limitations in existing ACL methods: (1) Multiple iterations of base training are needed when the model is applied. Subsequently, the acquired knowledge is encoded into a *regularized feature autocorrelation matrix* by analytic re-alignment. The incremental learning phase then unfolds, utilizing the recursive least squares method for updates. This pattern is repeated when the dataset is switched, significantly elevating the temporal cost in an online scenario. (2) Classic ACL methods demand data aggregation from a single task, facilitating analytic learning in one fell swoop. This process increases GPU memory footprint and is unsuitable for online contexts where data for each task is presented as mini-batches.

To address those limitations, we propose Forward-only Online Analytic Learning (F-OAL) that learns online batch-wise data streams. The F-OAL consists of a frozen pre-trained encoder and an Analytic Classifier (AC). The frozen encoder is capable of countering the uncompleted feature representation caused by using the loss function to update and replace the time-consuming base training. With Feature Fusion and Smooth Projection, the encoder provide informative representation for analytic learning. The AC is updated by recursive least square rather than back-propagation to solve recency bias and decrease calculation. Therefore, F-OAL is an exemplar-free countermeasure to CF and reduces resource consumption since the encoder is frozen and only the AC is updated.

Our main contributions can be concluded as follows:

• We present the F-OAL, an exemplar-free technique that achieves high accuracy and low resource consumption together for the OCIL.

• F-OAL redefines the OCIL problem into a recursive least square manner and is updated in a mini-batch manner.

• F-OAL introduces a framework of frozen pre-trained encoder with Feature Fusion to generate representative features and Smooth Projection for recursively updating AC to counter CF.

• We conduct massive experiments on benchmark datasets with other OCIL baselines. The results demonstrate that F-OAL achieves competitive results with fasting training speed and low GPU footprint.

2 Related Works

Online Class Incremental Learning focuses on extracting knowledge from one pass data stream with new classes in the new task. Time and memory consumption requirements are particularly critical in OCIL, given the fast and large nature of online data streams [13].

Replay-based methods [33, 25, 21, 9, 31, 12, 2, 11, 15, 41, 37, 36, 1, 38, 19, 6, 39, 35, 9] are mainstream solutions for OCIL problems by preserving historical exemplars and using them in new tasks. The accuracy is better than exemplar-free methods', but the training time and memory consumption are higher.

Analytic Learning (AL), also referred to as pseudo-inverse learning [10], emerges as a solution to the pitfalls of back-propagation by recursive least square. Analytic Learning's computational intensity is demanding since the entire dataset is processed. The obstacle is solved by the block-wise recursive

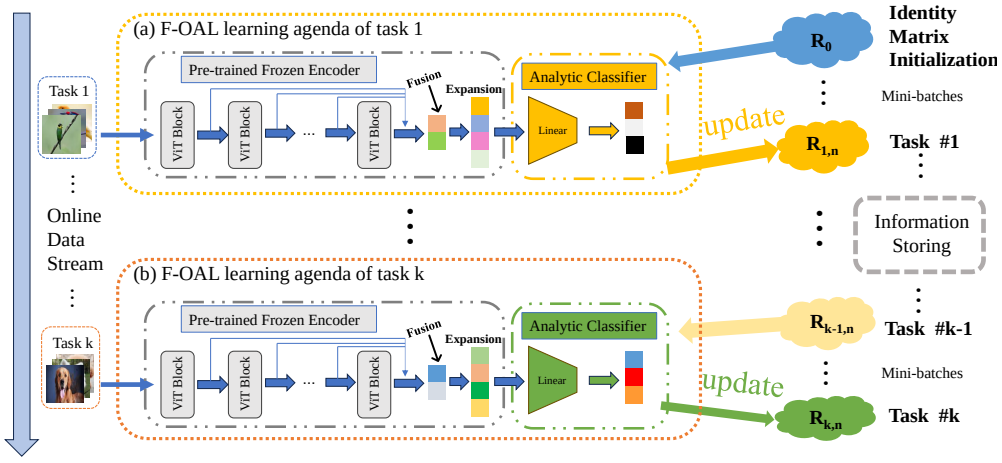

Figure 1: This diagram illustrates the learning agenda of F-OAL. In the encoder , features from each block of the ViT are extracted, summed, and averaged to form a composite feature. This feature is then randomly projected into a higher-dimensional space and normalized using the sigmoid function, serving as the activation for updating the classifier. All parameters in the encoder remain frozen. In the analytic classifier section, we introduce $\boldsymbol{R}$ to retain historical information and update the linear classifier using recursive least squares. This process is forward-only with no gradients.

Moore-Penrose algorithm [47], which achieves equivalent precision with joint-learning (i.e., training with all data). AL has already been used in online reinforcement learning [22], which shows the potential of AL in OCIL.

Exemplar-free methods can be categorized into regularization-based, prototype-based, and recently emerged ACL methods. **Regularization-based methods** [20, 17, 42, 8, 44] apply constraints on the loss function or change the gradients to preserve knowledge. These solutions reduce resource consumption but commonly can not outperform replay-based methods. **Prototype-based methods** [43, 45, 29, 40, 32, 7, 28] mitigate CF by augmenting the prototypes of features to classify old classes or generating psudo-features of old classes from new representations. **ACL methods** represent a new branch of the CIL community and show great potential in the OCIL scenario. Analytic Class Incremental Learning (ACIL) [49] is the first approach that applies AL to the CIL problem. The ACIL achieves a competitive performance in the offline CIL scenario. Gaussian Kernel Embedded Analytic Learning (GKEAL) [48], following the ACIL, focuses on solving CF in the few-shot scenario by adopting the kernel method. DS-AL [46] overcomes the under-fitting problem of AL.

3 Methodology

3.1 Proposed Framework

Our F-OAL framework consists of two modules:

Encoder. Encoder has two parts: backbone with Feature Fusion and Smooth Projection. The performance of analytic learning is highly dependent on the quality of the extracted features. Therefore, we employ Vision Transformer (ViT) [5] as the backbone instead of CNN, because ViT generates more comprehensive feature representation [30]. To further enhance the representativeness of the features, we utilize Feature Fusion. Specifically, we extract the outputs from each block of the ViT, sum them, and take their average to form the image feature.

Subsequently, we expand this feature into a higher-dimensional space using random projection and apply the sigmoid function for smoothing (i.e., Smooth Projection). This results in the final activation

https://doi.org/10.52202/079017-1314

used to update the classifier. Therefore, the encoder $\boldsymbol{\phi}(\cdot)$ is defined as:

$$\phi(x) = \underbrace{\sigma\left(LP\left(\overbrace{\frac{B_1(x) + B_1(B_2(x)) + \cdots + B_n(B_{n-1}(\cdots B_1(x)))}{n}}^{\text{Feature Fusion}}\right)\right)}_{\text{Smooth Expansion}}, \tag{1}$$

where $B_1(\cdot)\cdots B_n(\cdot)$ are blocks in ViT, $LP(\cdot)$ is linear projection and $\sigma(\cdot)$ is sigmoid activation function.

Backbone is pre-trained and frozen and weight matrix of projection is sampled from normal distribution and is frozen. Freezing the encoder avoids selective learning caused by loss function, which prioritizes features that are easiest to learn rather than the most representative [11] and significantly reduces the number of trainable parameters.

Analytic Classifier. Unlike back-propagation, we employ the least squares method to obtain the analytic solution for the linear mapping from the activation to the one-hot label. We then recursively update the weight matrix of the linear mapping. This approach is forward-only and does not require the use of gradients, resulting in low memory usage and fast computation speed.

3.2 Analytic Solution

Name	Description	Dimension
$\phi(X)$	Activation of all images.	$V \times D$
Y	One-hot label of all images.	$V \times M$
$\hat{W}$	Joint-learning result of classifier's weight matrix.	$M \times D$
$X_{k,i}^{(a)}$	Activation matrix of the i-th batch of the k-th task.	$S \times D$
$Y_{k,i}^{\text{train}}$	One-hot label matrix of the i-th batch of the k-th task.	$S \times C_s$
$X_{k,1:i}^{(a)}$	Activation matrix from the start to the i-th batch of the k-th task.	$V_s \times D$
$Y_{k,1:i}^{\text{train}}$	One-hot label matrix from the start.to the i-th batch of the k-th task.	$V_s \times C_s$
$\hat{W}^{(k,i)}$	Classifier of the i-th batch of the k-th task.	$C_s \times D$
$R_{k,i}$	Regularized feature autocorrelation matrix to the n-th batch of the k-th task.	$D \times D$

Table 1: Description of notations and their dimensions. Here, V is the number of all images, D is the encoder output dimension, M is the number of all classes, S is the batch size, C_s is the number of classes seen so far, and V_s is the number of images seen so far.

The overview is shown in Figure 1. To derive our solution, the notations needed are listed in Table 1. Unlike other methods that treat model training as a back-propagation process, our method formulates the problem using linear regression, which allows for direct computation of the optimal parameters in a closed-form solution:

$$\boldsymbol{Y} = \phi(\boldsymbol{X})\boldsymbol{W}, \tag{2}$$

where $\phi(\cdot)$ is the pre-trained and forzen encoder and $\boldsymbol{W}$ is the learnable parameter of linear classifier. The learning problem can be rewritten into an optimization form:

$$\underset{\boldsymbol{W}}{\operatorname{argmin}} \ \|\boldsymbol{Y} - \phi(\boldsymbol{X})\boldsymbol{W}\|_{\mathrm{F}}^2 + \gamma \|\boldsymbol{W}\|_{\mathrm{F}}^2, \tag{3}$$

where $\|\cdot\|_{\mathrm{F}}$ represents the Frobenius norm and γ is the regularization term. The optimal solution is defined as:

$$\hat{\boldsymbol{W}} = \left(\phi(\boldsymbol{X})^{\mathrm{T}}\phi(\boldsymbol{X}) + \gamma \boldsymbol{I}\right)^{-1} \phi(\boldsymbol{X})^{\mathrm{T}}\boldsymbol{Y}. \tag{4}$$

3.3 Learning Agenda

Given dataset $\{\boldsymbol{X}_k^{\text{train}}, \boldsymbol{Y}_k^{\text{train}}\}_1^m$ be the training set of task k (k = 1, 2, ... , m). Every task is divided into n mini-batches. We denote $\{\boldsymbol{X}_{k,i}^{\text{train}}, \boldsymbol{Y}_{k,i}^{\text{train}}\}$, as the i mini-batch (i=1, 2, ..., n) of training set of task k.

At task 1 batch k, the model aims to optimize

$$\underset{\boldsymbol{W}^{(1,i)}}{\operatorname{argmin}} \left\| \boldsymbol{Y}_{1,1:i}^{\text{train}} - (\boldsymbol{X}_{1,1:i}^{(a)})\boldsymbol{W}^{(1,i)} \right\|_{\text{F}}^2 + \gamma \left\| \boldsymbol{W}^{(1,i)} \right\|_{\text{F}}^2, \tag{5}$$

where

$$\boldsymbol{X}_{1,1:i}^{(a)} = \phi(\boldsymbol{X}_{1,1:i}^{\text{train}}). \tag{6}$$

The optimal solution to parameter $\boldsymbol{W}^{(1,i)}$ is given as:

$$\hat{\boldsymbol{W}}^{(1,i)} = \left((\boldsymbol{X}_{1,1:i}^{(a)})^{\text{T}} \boldsymbol{X}_{1,1:i}^{(a)} + \gamma \boldsymbol{I} \right)^{-1} (\boldsymbol{X}_{1,1:i}^{(a)})^{\text{T}} \boldsymbol{Y}_{1,1:i}^{\text{train}}. \tag{7}$$

Regarding $\boldsymbol{X}_{1,1:i}^{(a)}$ and $\boldsymbol{Y}_{1,1:i}^{\text{train}}$ as stacks of row activation vectors and one-hot label vectors respectively, we observe that $(\boldsymbol{X}_{1,1:i}^{(a)})^{\text{T}}(\boldsymbol{X}_{1,1:i}^{(a)})$ and $(\boldsymbol{X}_{1,1:i}^{(a)})^{\text{T}}\boldsymbol{Y}_{1,1:i}^{\text{train}}$ are both sums of outer products w.r.t feature vectors. Notice that the solution contains no data correlation. Thus we can further derive Equation 7 into

$$\begin{aligned} \hat{\boldsymbol{W}}^{1,i} &= \left(\begin{bmatrix} (\boldsymbol{X}_{1,1:i-1}^{(a)})^{\text{T}} & (\boldsymbol{X}_{1,i}^{(a)})^{\text{T}} \end{bmatrix} \begin{bmatrix} \boldsymbol{X}_{1,1:i-1}^{(a)} \\ \boldsymbol{X}_{1,i}^{(a)} \end{bmatrix} + \gamma \boldsymbol{I} \right)^{-1} \begin{bmatrix} (\boldsymbol{X}_{1,1:i-1}^{(a)})^{\text{T}} & (\boldsymbol{X}_{1,i}^{(a)})^{\text{T}} \end{bmatrix} \begin{bmatrix} \boldsymbol{Y}_{1,1:i-1}^{\text{train}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Y}_{1,i}^{\text{train}} \end{bmatrix} \\ &= \left((\boldsymbol{X}_{1,1:i-1}^{(a)})^{\text{T}}(\boldsymbol{X}_{1,1:i-1}^{(a)}) + (\boldsymbol{X}_{1,i}^{(a)})^{\text{T}}(\boldsymbol{X}_{1,i}^{(a)}) + \gamma \boldsymbol{I} \right)^{-1} \left[(\boldsymbol{X}_{1,1:i-1}^{(a)})^{\text{T}} \boldsymbol{Y}_{1,1:i-1}^{\text{train}} + (\boldsymbol{X}_{1,i}^{(a)})^{\text{T}} \boldsymbol{Y}_{1,i}^{\text{train}} \right]. \end{aligned} \tag{8}$$

Let

$$\boldsymbol{R}_{1,i-1} = \left[(\boldsymbol{X}_{1,1:i-1}^{(a)})^{\text{T}}(\boldsymbol{X}_{1,1:i-1}^{(a)}) + \gamma \boldsymbol{I} \right]^{-1}. \tag{9}$$

be the *regularized feature autocorrelation matrix* at batch i-1 of task 1, where both historical and current information is encoded in.

Therefore, we can calculate both $\boldsymbol{R}$ and $\boldsymbol{W}$ in a recursive manner by the following theorems when the serial numbers of tasks and batches are given:

Theorem 3.1 For the batch 1 of task k, Let $\hat{\boldsymbol{W}}^{(0)}$ be the null matrix. Let $\hat{\boldsymbol{W}}^{(k-1,n)'} = \begin{bmatrix} \hat{\boldsymbol{W}}^{(k-1,n)} & \boldsymbol{0} \end{bmatrix}$ and $\hat{\boldsymbol{W}}^{(k,1)}$ can be calculated via:

$$\hat{\boldsymbol{W}}^{(k,1)} = \hat{\boldsymbol{W}}^{(k-1,n)'} + \boldsymbol{R}_{k,1}\boldsymbol{X}_{k,1}^{(\text{a})\text{T}} \left(\boldsymbol{Y}_{k,1}^{\text{train}} - \boldsymbol{X}_{k,1}^{(a)} \hat{\boldsymbol{W}}^{(k-1,n)'} \right), \tag{10}$$

where

$$\boldsymbol{R}_{k,1} = \boldsymbol{R}_{k-1,n} - \boldsymbol{R}_{k-1,n}\boldsymbol{X}_{k,1}^{(\text{a})\text{T}} \left(\boldsymbol{I} + \boldsymbol{X}_{k,1}^{(\text{a})} \boldsymbol{R}_{k-1,n} \boldsymbol{X}_{k,1}^{(\text{a})\text{T}} \right)^{-1} \boldsymbol{X}_{k,1}^{(\text{a})} \boldsymbol{R}_{k-1,n}. \tag{11}$$

Theorem 3.2 For the batch i ($i > 1$) of task k, $\hat{\boldsymbol{W}}^{(k,i)}$ can be calculated via:

$$\hat{\boldsymbol{W}}^{(k,i)} = \hat{\boldsymbol{W}}^{(k,i-1)} + \boldsymbol{R}_{k,i}\boldsymbol{X}_{k,i}^{(\text{a})\text{T}} \left(\boldsymbol{Y}_{k,i}^{\text{train}} - \boldsymbol{X}_{k,i}^{(a)} \hat{\boldsymbol{W}}^{(k,i-1)} \right), \tag{12}$$

where

$$\boldsymbol{R}_{k,i} = \boldsymbol{R}_{k,i-1} - \boldsymbol{R}_{k,i-1}\boldsymbol{X}_{k,i}^{(\text{a})\text{T}} \left(\boldsymbol{I} + \boldsymbol{X}_{k,i}^{(\text{a})} \boldsymbol{R}_{k,i-1} \boldsymbol{X}_{k,i}^{(\text{a})\text{T}} \right)^{-1} \boldsymbol{X}_{k,i}^{(\text{a})} \boldsymbol{R}_{k,i-1}. \tag{13}$$

Proof. See Appendix A. □

Thus, we achieve absolute memorization in an exemplar-free way with all data used only once. The learning agenda of F-OAL is summarised in Algorithm 1.

https://doi.org/10.52202/079017-1314

Algorithm 1 Forward-Only Analytic Learning

Input: Mini-batches $\{\boldsymbol{X}_{k,i}^{\text{train}}, \boldsymbol{Y}_{k,i}^{\text{train}}\}_{1,1}^{m,n}$.
Initialization: Identity matrix $\boldsymbol{R}_0$; Null matrix $\boldsymbol{W}^{(0)}$.
for $k = 1$ **to** m **do**
  for $i = 1$ **to** n **do**
    if $i = 1$ **then**
      # **Theorem 3.1**:
      Load $\hat{\boldsymbol{W}}^{(k-1,n)}$ and $\boldsymbol{R}_{k-1,n}$;
      Expand $\hat{\boldsymbol{W}}^{(k-1,n)}$ to $\hat{\boldsymbol{W}}^{(k-1,n)'}$;
      Obtain activation $\boldsymbol{X}_{k,1}^{(\text{a})}$ based on $\boldsymbol{X}_{k,1}^{\text{train}}$;
      Update $\boldsymbol{R}_{k,1}$ by $\boldsymbol{X}_{k,1}^{(\text{a})}$ and $\boldsymbol{R}_{k-1,n}$;
      Update $\boldsymbol{W}^{(k,1)}$ by $\boldsymbol{X}_{k,1}^{(\text{a})}$; $\boldsymbol{Y}_{k,1}^{\text{train}}$, $\boldsymbol{W}^{(k-1,n)'}$ and $\boldsymbol{R}_k$;
    else
      # **Theorem 3.2**:
      Load $\boldsymbol{W}^{(k,i-1)}$ and $\boldsymbol{R}_{k,i-1}$;
      Obtain activation $\boldsymbol{X}_{k,i}^{(\text{a})}$ based on $\boldsymbol{X}_{k,i}^{\text{train}}$;
      Update $\boldsymbol{R}_{k,i}$ by $\boldsymbol{X}_{k,i}^{(\text{a})}$ and $\boldsymbol{R}_{k,i-1}$;
      Update $\boldsymbol{W}^{(k,i)}$ by $\boldsymbol{X}_{k,i}^{(\text{a})}$; $\boldsymbol{Y}_{k,i}^{\text{train}}$, $\boldsymbol{W}^{(k,i-1)}$ and $\boldsymbol{R}_{k,i-1}$;
    end if
  end for
end for

3.4 Complexity Analysis

In terms of space complexity, our trainable parameters are only $\boldsymbol{R}$ and $\boldsymbol{W}$. The $\boldsymbol{R}$ matrix is of size $D \times D$, where D is the output dimension of the encoder. In our paper, the encoder output dimension is 1000. Therefore, according to Equation 9, the size of the $\boldsymbol{R}$ matrix is $1,000 \times 1,000$. The $\boldsymbol{W}$ matrix has dimensions of $C \times D$, where C is the number of classes in the target dataset. For example, with CIFAR-100, its size is $100 \times 1,000$. The total number of trainable parameters is relatively small and does not require gradients. This results in our method using less than 2GB of GPU memory.

In terms of computational complexity, we denote the batch size as S, encoder's output size as D, and class number as C. Therefore, the dimensions of $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{R}$ and $\boldsymbol{W}$ are $S \times D$, $S \times C$, $D \times D$, and $C \times D$, respectively. Thus, the calculation is shown below:

The computational complexity for updating R is dominated by the matrix multiplications, thus:

$$\max\{\mathcal{O}(SDC), \mathcal{O}(SC), \mathcal{O}(SDC)\} \approx \max\{\mathcal{O}(SDC), \mathcal{O}(D^2C)\}. \tag{14}$$

The computational complexity for updating W is dominated by the matrix multiplications and the matrix inversion:

$$\max\{\mathcal{O}(SD), \mathcal{O}(SD^2), \mathcal{O}(S^2), \mathcal{O}(S^3), \mathcal{O}(DS^2), \mathcal{O}(D^2S), \mathcal{O}(D^2)\} \approx \max\{\mathcal{O}(S^3), \mathcal{O}(D^2S)\}. \tag{15}$$

In the OCIL setting, the batch size is relatively smaller. Therefore, the overall computational complexity is primarily controlled by D.

3.5 Overhead Analysis

In terms of space overhead, compared to the conventional backbone + classifier structure, F-OAL introduces an additional linear projection to control the output dimension D of the encoder, and a matrix $\boldsymbol{R}$, where only $\boldsymbol{R}$ is trainable. According to Equation 9, the dimension of $\boldsymbol{R}$ remains a fixed size of $D \times D$. Other methods require more extra space. For instance, LwF [20] employs knowledge distillation, necessitating the storage of additional models, while replay-based methods require extra storage to retain historical samples. In contrast, the overhead introduced by F-OAL, consisting of an additional matrix and a linear layer, is smaller.

Metric	Method	Replay?	CIFAR-100	CORe50	FGCVAircraft	DTD	Tiny-ImageNet	Country211
$A_{avg}(\%)\uparrow$	iCaRL(CVPR 2017)[31]	✓	91.6	95.6	36.4	74.1	91.3	12.2
	ER(ICRA 2019)[12]	✓	90.1	94.8	35.7	65.4	87.3	14.0
	ASER(AAAI 2021)[33]	✓	87.2	87.1	25.2	57.4	85.8	13.2
	SCR(CVPR 2021)[25]	✓	91.9	95.3	55.6	75.0	82.6	14.7
	DVC(CVPR 2022)[9]	✓	92.4	97.1	33.7	67.3	91.5	16.1
	PCR(CVPR 2023)[21]	✓	89.1	95.7	10.1	35.0	91.0	9.7
	LwF(TPAMI 2018)[20]	✗	69.3	47.0	14.2	40.2	82.5	1.4
	EWC(PNAS 2017)[17]	✗	49.9	47.9	12.0	27.6	60.5	6.1
	EASE(CVPR 2024)[42]	✗	91.1	85.0	38.2	76.0	**92.0**	15.9
	LAE(ICCV 2023)[8]	✗	79.1	73.3	13.5	63.5	86.7	14.5
	SLCA(ICCV 2023)[40]	✗	90.4	93.7	34.3	70.9	88.6	17.8
	F-OAL	✗	**91.1**	**96.3**	**62.2**	**82.8**	91.2	**24.4**
$A_{last}(\%)\uparrow$	iCaRL(CVPR 2017)	✓	87.5	93.2	29.8	66.3	87.8	6.5
	ER(ICRA 2019)	✓	84.6	92.1	28.6	54.3	81.6	6.8
	ASER(AAAI 2021)	✓	82.0	82.1	14.8	49.4	80.0	6.7
	SCR(CVPR 2021)	✓	87.7	93.6	50.3	68.7	75.8	8.0
	DVC(CVPR 2022)	✓	87.8	96.0	27.0	57.2	87.2	9.2
	PCR(CVPR 2023)	✓	81.4	93.9	9.0	34.6	86.1	6.1
	LwF(TPAMI 2018)	✗	64.8	26.3	5.8	18.3	72.5	0.5
	EWC(PNAS 2017)	✗	25.2	21.1	3.0	13.3	44.6	1.9
	EASE(CVPR 2024)	✗	85.4	78.3	29.3	67.6	**89.3**	10.5
	LAE(ICCV 2023)	✗	75.6	67.1	6.3	53.6	82.4	9.3
	SLCA(ICCV 2023)	✗	85.6	88.2	32.1	63.3	85.4	12.9
	F-OAL	✗	**86.5**	**92.5**	**54.0**	**75.9**	87.3	**17.5**
$F(\%)\downarrow$	iCaRL(CVPR 2017)	✓	3.2	2.3	7.1	7.8	2.7	6.7
	ER(ICRA 2019)	✓	20.7	4.3	34.0	29.8	13.3	21.0
	ASER(AAAI 2021)	✓	16.5	9.9	35.7	29.3	16.3	19.4
	SCR(CVPR 2021)	✓	6.2	3.8	14.5	11.6	7.7	6.4
	DVC(CVPR 2022)	✓	8.2	2.3	29.7	21.7	8.9	18.9
	PCR(CVPR 2023)	✓	9.2	4.2	2.7	1.4	8.0	1.7
	LwF(TPAMI 2018)	✗	**1.3**	**0.4**	**3.1**	**4.5**	**1.0**	**0**
	EWC(PNAS 2017)	✗	67.4	81.0	38.8	68.3	20.7	51.5
	EASE(CVPR 2024)	✗	6.1	10.7	19.2	12.5	2.8	16.8
	LAE(ICCV 2023)	✗	11.8	13.8	12.2	25.0	5.4	16.7
	SLCA(ICCV 2023)	✗	7.1	3.4	10.2	12.7	4.2	14.9
	F-OAL	✗	5.5	3.9	10.0	10.1	5.0	6.9

Table 2: This table shows the comparison results of our method with other baselines on six datasets. We select three metrics: average accuracy (A_{avg}), last task accuracy (A_{last}), and forgetting rate (F). Higher values for A_{avg} and A_{last} indicate better performance, while lower values for F indicate better performance. In **Replay?** column, replay-based methods are marked with ✓. Conversely, exemplar-free methods are marked with ✗. Data in **Bold** are the best within exemplar-free methods, and data underlined are the best considering both categories.

In terms of time overhead, our method primarily consists of a forward pass and matrix multiplication, which is determined by the output dimension of the encoder. By changing the output dimension of the encoder, we can balance the accuracy and time. According to [16], the backward pass in back-propagation (forward pass + backward pass) accounts for 70% of the time. Therefore, our method's time overhead is also relatively small.

4 Experiment

To show the effectiveness of F-OAL, we conduct extensive experiments to compare our approach with baseline methods. We build our code and reproduce relevant results based on [24, 34]. All baselines are with pre-trained ViT as the backbone.

4.1 Datasets

We focus on coarse-grained data scenarios, such as CIFAR-100, Tiny-ImageNet, and Core50, as well as fine-grained data scenarios, including DTD, FGVCAircraft, and Country211. All of these are open-source benchmark datasets. However, there are other data scenarios, such as long-tail distributions, where unbalanced data distribution will make it harder to achieve good performance by only training the classifier. We will study this case in future work.

• **CIFAR-100** [18] is constructed into 10 tasks with disjoint classes, and each task has 10 classes. Each task has 5,000 images for training and 1,000 for testing.

https://doi.org/10.52202/079017-1314

• **CORe50** [23] is a benchmark designed for class incremental learning with 9 tasks and 50 classes: 10 classes in the first task and 5 classes in the subsequent 8 tasks. Each class has 2,398 training images and 900 testing images on average.

• **FGVCAircraft** [26] contains 102 different classes of *aircraft models*. 100 classes are selected and divided into 10 tasks. Each class has 33 training images and 33 testing images on average.

• **DTD** [3] is a *texture database*, organized into 47 classes. 40 classes are selected and divided into 10 tasks. Each class has 40 training images and 40 testing images.

• **Tiny-ImageNet** is a subset of ImageNet with 200 classes for training. Each class has 500 images. The test set contains 10,000 images. The dataset is evenly divided into 10 tasks.

• **Country211** contains 211 classes of country images, with 150 train and test images per class. The dataset is evenly divided into 10 tasks with 210 classes chosen.

4.2 Evaluation Metrics

We define $a_{i,j}$ as the accuracy evaluated on the test set of task j after training the network from task 1 through to i, and the average accuracy is defined as

$$A_i = \frac{1}{i}\sum_{j=1}^{i} a_{i,j}. \tag{16}$$

When $i = m$ (i.e., the total number of tasks), A_m represents the average accuracy by the end of training.

Forgetting rate at task i is defined as Equation 17. $f_{i,j}$ represents how much the model has forgot about task j after being trained on task i. Specifically, $\max_{l\in\{1,\dots,k-1\}}(a_{l,j})$ denotes the best test accuracy the model has ever achieved on task j before learning task k, and $a_{k,j}$ is the test accuracy on task j after learning task k.

$$F_i = \frac{1}{i-1}\sum_{j=1}^{i-1} f_{i,j}, \tag{17}$$

where

$$f_{k,j} = \max_{l\in\{1,\dots,k-1\}}(a_{l,j}) - a_{k,j}, \quad \forall j < k. \tag{18}$$

4.3 Implementation Details

ViT-B [5], pre-trained on ImageNet-1K [4], serves as the backbone network for all methods. The data stream is kept identical across all experiments to ensure a fair comparison. The learning rate and batch size are set to 0.001 and 10, respectively. The optimizer is SGD. We assign 5,000 memory buffer sizes for replay-based methods. In F-OAL, the expansion size is 1,000 (i.e., the output activation size to update AC is 1,000). The regularization term is set to be 1. All experiments are conducted on a single RTX 4070ti GPU 12GB, and an average of 3 runs is reported.

4.4 Result Comparison

We tabulate the average accuracy (A_{avg}), last task accuracy (A_{last}), and the forgetting rate (F) from the compared methods in Table 2.

For fine-grained datasets such as FGCVAircraft, DTD, and Country211, F-OAL achieves the best performance, with average accuracies of 66.2%, 82.8%, and 24.4%, respectively. The second-best results are 55.6%, 76.0%, and 17.8%, respectively. Similarly, F-OAL also outperforms in last task accuracy. This demonstrates the excellent transferability of our method in the OCIL setting, effectively leveraging the feature extraction capabilities of the pre-trained encoder.

On coarse-grained datasets such as CIFAR-100, CORe50, and Tiny-ImageNet, F-OAL still demonstrates excellent performance, achieving the highest accuracy among all exemplar-free methods, except on Tiny-ImageNet where it is 0.8% behind EASE in average accuracy. Compared to the best replay-based methods, it only lags by 1.3%, 0.8%, and 0.3%, respectively.

Methods	CIFAR-100	CORe50	FGVCAircraft	DTD	Tiny-ImageNet	Country211
LwF	412	877	135	73	841	256
EWC	451	922	115	62	1,190	249
iCaRL	832	1,716	53	24	1671	513
ER	652	1,433	40	21	1,315	404
ASER	5,608	7,700	91	43	18,597	20,611
SCR	2,843	5,939	88	42	62,996	810
DVC	4,191	9,351	287	130	10,940	2,622
PCR	1,624	3,742	113	53	3,274	1,028
EASE	383	760	147	139	638	304
LAE	**252**	**458**	156	140	**500**	355
SLCA	726	1,416	289	278	1,185	551
F-OAL	261	570	**16**	**8**	507	**157**

Table 3: Training time including feature extraction is reported in seconds where replay-based methods are with 5,000 buffer size. Data in **Bold** show the fastest time.

Typically, a lower forgetting rate is better, but forgetting is based on accuracy. Therefore, when comparing forgetting rates, it is important to consider models with similar accuracy levels. When comparing F-OAL to the well-performing DVC, F-OAL exhibits a lower forgetting rate. On CIFAR-100, F-OAL's forgetting rate is 5.5%, while DVC's is 8.2%. This indicates that F-OAL not only maintains a high level of accuracy but also effectively manages forgetting.

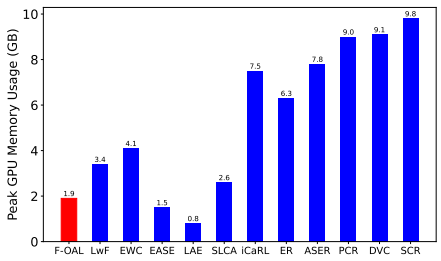

Figure 2: Peak GPU memory footprint in GB with 10 batch size on CIFAR-100. Replay-based methods are with 5,000 buffer size. F-OAL has low GPU footprint since it does not require gradients.

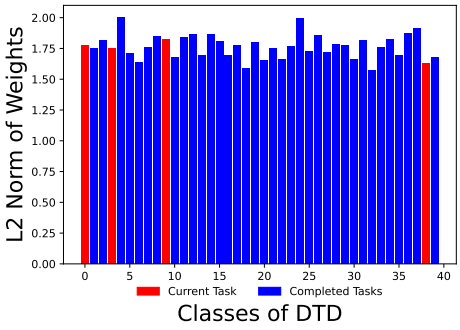

Figure 3: Visualization of the weights of a linear classifier. The result comes from F-OAL on DTD. Based on [15], when recency bias happens, L2 norm of current task is significantly larger. In F-OAL, the L2 norm of current task is in a average level.

4.5 Resource Consumption

GPU. Peak GPU memory footprint on CIFAR-100 is shown in Figure 2. As we state in Section 3.5, without using gradient and extra space for exemplars, F-OAL requires a low memory footprint while having good performance (i.e., higher accuracy than LAE and EASE on most datasets).

Training Time. Table 3 illustrates training time including feature extraction. F-OAL is fast with competitive accuracy (i.e., higher accuracy than LAE). Only the classifier and regularized feature autocorrelation matrix are updated, leading to fewer of trainable parameters and fast training speed.

4.6 Countering Recency Bias

As demonstrated in Figure 3, the linear classifier of AC obtained through F-OAL training does not exhibit recency bias. Notably, the weights corresponding to the classes in the most recent tasks do not significantly surpass those of the earlier classes. The frozen encoder and the recursive least square updating manner ensure equal treatment for all samples.

https://doi.org/10.52202/079017-1314

4.7 Ablation Study

We design the ablation study in Table 4 to verify the effectiveness of the Feature Fusion and Smooth Projection modules. Without these two components, the accuracy of F-OAL drops, especially on fine-grained datasets.

FF	SP	CIFAR-100	CORe50	FGVCAircraft	DTD	Tiny-ImageNet	Country211
✓	✓	91.1	96.3	62.2	82.8	91.2	24.4
✗	✓	90.6	95.3	60.9	80.5	91.4	21.3
✓	✗	90.7	95.4	58.7	79.3	91.2	22.8
✗	✗	90.6	95.4	56.0	71.2	91.4	21.1

Table 4: Average accuracy comparison with Feature Fusion (FF) and Smooth Projection (SP) modules across various datasets. ✓ means **with** and ✗ means **without**.

As Table 5 (See appendix B) shows, we prove analytic classifier is key to high accuracy. Herein, we define the Fully Connected Classifier (FCC) as having the identical structure to the AC, but it is updated through back-propagation rather than utilizing the $\boldsymbol{R}$ and recursive least square. Without AC, the accuracy drops from 91.1% to 32.4% on CIFAR-100.

Regularization Term. Table 6 (See appendix B) shows the effects of varying γ. For large volume dataset, F-OAL gives a robust performance in wide range of γ value (e.g., 10^2 - 10^{-3}), while small datasets need larger value (e.g., 10^2 - 1). The comprehensive results show that γ=1 is the suitable choice.

Projection Size. Figure 4 (See appendix B) demonstrates the influence of different random projection sizes. The results suggest that the setting of a 1,000-dimensional projection is appropriate.

4.8 Potential Positive and Negative Societal Impacts

The key advantage of our approach is its ability to achieve exemplar-free OCIL with fast training and low memory footprint, offering an environmentally friendly and efficient solution for this research track. However, the main limitation of our method lies in its reliance on a powerful pre-trained encoder. As a result, it is crucial for us to leverage open-source pre-trained backbone networks from the deep learning community, rather than training our own, which will otherwise lead to higher GPU usage and increased resource consumption.

5 Conclusion

In this paper, we propose Forward-only Online Analytic Learning (F-OAL), an exemplar-free method for addressing several limitations of the Online Class Incremental Learning scenario. We use Analytic Learning to acquire the optimal solution of the classifier directly instead of training for dozens of epochs by back-propagation. Leveraging the frozen pre-trained encoder with Feature Fusion and Smooth Projection and the Analytic Classifier updated by recursive least square, our approach achieves the identical solution to joint-learning on the whole dataset without preserving any historical exemplars, achieving high accuracy and reducing the resource consumption. Our experiments show the competitive performance of F-OAL.

6 Acknowledgment

This research was supported by the National Natural Science Foundation of China (62306117), the Guangzhou Basic and Applied Basic Research Foundation (2024A04J3681, 2023A04J1687), the South China University of Technology-TCL Technology Innovation Fund, the Fundamental Research Funds for the Central Universities (2023ZYGXZR023, 2024ZYGXZR074), the Guangdong Basic and Applied Basic Research Foundation (2024A1515010220), and the CAAI- MindSpore Open Fund developed on Openl Community.

References

[1] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In *Advances in Neural Information Processing Systems*, pages 11849–11860, 2019.

[2] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems*, pages 11816–11825, 2019.

[3] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[6] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, part XX 16*, pages 86–102, 2020.

[7] Qiankun Gao, Chen Zhao, Bernard Ghanem, and Jian Zhang. R-dfcil: Relation-guided representation learning for data-free class incremental learning. In *European Conference on Computer Vision*, pages 423–439, 2022.

[8] Qiankun Gao, Chen Zhao, Yifan Sun, Teng Xi, Gang Zhang, Bernard Ghanem, and Jian Zhang. A unified continual learning framework with general parameter-efficient tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11483–11493, 2023.

[9] Yanan Gu, Xu Yang, Kun Wei, and Cheng Deng. Not just selection, but exploration: Online class-incremental continual learning via dual view consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7442–7451, 2022.

[10] Ping Guo, Michael R Lyu, and NE Mastorakis. Pseudoinverse learning algorithm for feedforward neural networks. In *Advances in Neural Networks and Applications*, pages 321–326, 2001.

[11] Yiduo Guo, Bing Liu, and Dongyan Zhao. Online continual learning through mutual information maximization. In *International Conference on Machine Learning*, pages 8109–8126. PMLR, 2022.

[12] Tyler L Hayes, Nathan D Cahill, and Christopher Kanan. Memory efficient experience replay for streaming learning. In *2019 International Conference on Robotics and Automation*, pages 9769–9776, 2019.

[13] Jiangpeng He, Runyu Mao, Zeman Shao, and Fengqing Zhu. Incremental learning in online scenario. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13926–13935, 2020.

[14] Xu He and Herbert Jaeger. Overcoming catastrophic interference using conceptor-aided backpropagation. 2018.

[15] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 831–839, 2019.

[16] Zhouyuan Huo, Bin Gu, Heng Huang, et al. Decoupled parallel backpropagation with convergence guarantee. In *International Conference on Machine Learning*, pages 2098–2106. PMLR, 2018.

[17] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. In *Proceedings of the National Academy of Sciences*, pages 3521–3526, 2017.

[18] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

 https://doi.org/10.52202/079017-1314

[19] Byung Hyun Lee, Okchul Jung, Jonghyun Choi, and Se Young Chun. Online continual learning on hierarchical label expansion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11761–11770, 2023.

[20] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 2935–2947, 2017.

[21] Huiwei Lin, Baoquan Zhang, Shanshan Feng, Xutao Li, and Yunming Ye. Pcr: Proxy-based contrastive replay for online class-incremental continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24246–24255, 2023.

[22] Zichen Liu, Chao Du, Wee Sun Lee, and Min Lin. Locality sensitive sparse encoding for learning world models online. In *International Conference on Learning Representations*, 2024.

[23] Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In *Conference on Robot Learning*, pages 17–26, 2017.

[24] Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, pages 28–51, 2022.

[25] Zheda Mai, Ruiwen Li, Hyunwoo Kim, and Scott Sanner. Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3589–3599, 2021.

[26] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[27] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. 1989.

[28] Zichong Meng, Jie Zhang, Changdi Yang, Zheng Zhan, Pu Zhao, and Yanzhi WAng. Diffclass: Diffusion-based class incremental learning. *European Conference on Computer Vision*, 2024.

[29] Grégoire Petit, Adrian Popescu, Hugo Schindler, David Picard, and Bertrand Delezoide. Fetril: Feature translation for exemplar-free class-incremental learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3911–3920, 2023.

[30] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? In *Advances in Neural Information Processing Systems*, pages 12116–12128, 2021.

[31] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[32] Grzegorz Rypeść, Sebastian Cygert, Valeriya Khan, Tomasz Trzcinski, Bartosz Michał Zieliński, and Bartłomiej Twardowski. Divide and not forget: Ensemble of selectively trained experts in continual learning. In *International Conference on Learning Representations*, 2024.

[33] Dongsub Shim, Zheda Mai, Jihwan Jeong, Scott Sanner, Hyunwoo Kim, and Jongseong Jang. Online class-incremental continual learning with adversarial shapley value. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9630–9638, 2021.

[34] Hai-Long Sun, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Pilot: A pre-trained model-based continual learning toolbox. *arXiv preprint arXiv:2309.07117*, 2023.

[35] Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Foster: Feature boosting and compression for class-incremental learning. In *European conference on computer vision*, pages 398–414, 2022.

[36] Quanziang Wang, Renzhen Wang, Yichen Wu, Xixi Jia, and Deyu Meng. Cba: Improving online continual learning via continual bias adaptor. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19082–19092, 2023.

[37] Zhen Wang, Liu Liu, Yajing Kong, Jiaxian Guo, and Dacheng Tao. Online continual learning with contrastive vision transformer. In *European Conference on Computer Vision*, pages 631–650, 2022.

[38] Yujie Wei, Jiaxin Ye, Zhizhong Huang, Junping Zhang, and Hongming Shan. Online prototype learning for online continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18764–18774, 2023.

[39] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3014–3023, 2021.

[40] Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. Slca: Slow learner with classifier alignment for continual learning on a pre-trained model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19148–19158, 2023.

[41] Yaqian Zhang, Bernhard Pfahringer, Eibe Frank, Albert Bifet, Nick Jin Sean Lim, and Yunzhe Jia. A simple but strong baseline for online continual learning: Repeated augmented rehearsal. In *Advances in Neural Information Processing Systems*, pages 14771–14783, 2022.

[42] Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23554–23564, 2024.

[43] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5871–5880, 2021.

[44] Kai Zhu, Wei Zhai, Yang Cao, Jiebo Luo, and Zheng-Jun Zha. Self-sustaining representation expansion for non-exemplar class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9296–9305, 2022.

[45] Kai Zhu, Wei Zhai, Yang Cao, Jiebo Luo, and Zheng-Jun Zha. Self-sustaining representation expansion for non-exemplar class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9296–9305, 2022.

[46] Huiping Zhuang, Run He, Kai Tong, Ziqian Zeng, Cen Chen, and Zhiping Lin. Ds-al: A dual-stream analytic learning for exemplar-free class-incremental learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17237–17244, 2024.

[47] Huiping Zhuang, Zhiping Lin, and Kar-Ann Toh. Blockwise recursive moore–penrose inverse for network learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(5):3237–3250, 2021.

[48] Huiping Zhuang, Zhenyu Weng, Run He, Zhiping Lin, and Ziqian Zeng. Gkeal: Gaussian kernel embedded analytic learning for few-shot class incremental task. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7746–7755, 2023.

[49] Huiping Zhuang, Zhenyu Weng, Hongxin Wei, Renchunzi Xie, Kar-Ann Toh, and Zhiping Lin. Acil: Analytic class-incremental learning with absolute memorization and privacy protection. In *Advances in Neural Information Processing Systems*, pages 11602–11614, 2022.

 https://doi.org/10.52202/079017-1314

A Appendix

Proof of Theorem 3.1 and 3.2

For **Theorem 3.1**, we start with proving the case in batch 1 of task *k*.

According to Equation 8, of task $k-1$, we can expand the stacked activation and label matrixes:

$$\hat{\boldsymbol{W}}^{(k-1,n)} = \left(\sum_{m=1}^{k-1}\sum_{i=1}^{n} \boldsymbol{X}_{m,i}^{(\mathrm{a})\mathrm{T}}\boldsymbol{X}_{m,i}^{(\mathrm{a})} + \gamma\boldsymbol{I}\right)^{-1} \begin{bmatrix} \boldsymbol{X}_{1,1}^{(\mathrm{a})\mathrm{T}}\boldsymbol{Y}_{1,1}^{\text{train}} \\ \vdots \\ \boldsymbol{X}_{k-1,n}^{(\mathrm{a})\mathrm{T}}\boldsymbol{Y}_{k-1,n}^{\text{train}} \end{bmatrix}. \tag{19}$$

Hence, at batch 1 of task k, we have

$$\hat{\boldsymbol{W}}^{(k,1)} = \left(\sum_{m=1}^{k-1}\sum_{i=1}^{n} \boldsymbol{X}_{m,i}^{(\mathrm{a})\mathrm{T}}\boldsymbol{X}_{m,i}^{(\mathrm{a})} + \gamma\boldsymbol{I} + \boldsymbol{X}_{k,1}^{(a)\mathrm{T}}\boldsymbol{X}_{k,1}^{(a)}\right)^{-1} \begin{bmatrix} \boldsymbol{X}_{1,1}^{(\mathrm{a})\mathrm{T}}\boldsymbol{Y}_{1,1}^{\text{train}} \\ \vdots \\ \boldsymbol{X}_{k-1,n}^{(\mathrm{a})\mathrm{T}}\boldsymbol{Y}_{k-1,n}^{\text{train}} \\ \boldsymbol{X}_{k,1}^{(a)\mathrm{T}}\boldsymbol{Y}_{k,1}^{\text{train}} \end{bmatrix}. \tag{20}$$

We have defined regularized feature autocorrelation matrix $\boldsymbol{R}_{k-1,n}$ via

$$\boldsymbol{R}_{k-1,n} = \left(\sum_{m=1}^{k-1}\sum_{i=1}^{n} \boldsymbol{X}_{m,i}^{(\mathrm{a})\mathrm{T}}\boldsymbol{X}_{m,i}^{(\mathrm{a})} + \gamma\boldsymbol{I}\right)^{-1} \tag{21}$$

To facilitate subsequent calculations, here we also define a cross-correlation matrix $\boldsymbol{Q}_{k-1,n}$

$$\boldsymbol{Q}_{k-1,n} = \begin{bmatrix} \boldsymbol{X}_{1,1}^{(\mathrm{a})\mathrm{T}}\boldsymbol{Y}_{1,1}^{\text{train}} & \dots & \boldsymbol{X}_{k-1,n}^{(\mathrm{a})\mathrm{T}}\boldsymbol{Y}_{k-1,n}^{\text{train}} \end{bmatrix}. \tag{22}$$

Thus, we can rewrite Equation 20 as

$$\hat{\boldsymbol{W}}^{(k-1,n)} = \boldsymbol{R}_{k-1,n}\boldsymbol{Q}_{k-1,n}. \tag{23}$$

Therefore, at batch 1 of task *k* we have

$$\hat{\boldsymbol{W}}^{(k,1)} = \boldsymbol{R}_{k,1}\boldsymbol{Q}_{k,1}. \tag{24}$$

From Equation 21 we can recursively calculate $R_{k,1}$ from $R_{k-1,n}$

$$\boldsymbol{R}_{k,1} = \left(\boldsymbol{R}_{k-1,n}^{-1} + \boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}}\boldsymbol{X}_{k,1}\right)^{-1}. \tag{25}$$

According to the Woodbury matrix identity, we have

$$(\boldsymbol{A}+\boldsymbol{UCV})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{U}\left(\boldsymbol{C}^{-1} + \boldsymbol{V}\boldsymbol{A}^{-1}\boldsymbol{U}\right)^{-1}\boldsymbol{V}\boldsymbol{A}^{-1}. \tag{26}$$

Let $\boldsymbol{A} = \boldsymbol{R}_{k-1,n}^{-1}, \boldsymbol{U} = \boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}}$, $\boldsymbol{C} = \boldsymbol{I}$, $\boldsymbol{V} = \boldsymbol{X}_{k,1}^{(\mathrm{a})}$ in Equation 26, we have

$$\boldsymbol{R}_{k,1} = \boldsymbol{R}_{k-1,n} - \boldsymbol{R}_{k-1,n}\boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}}\left(\boldsymbol{I} + \boldsymbol{X}_{k,1}^{(\mathrm{a})}\boldsymbol{R}_{k-1,n}\boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}}\right)^{-1}\boldsymbol{X}_{k,1}^{(\mathrm{a})}\boldsymbol{R}_{k-1,n}. \tag{27}$$

Hence, $\boldsymbol{R}_k$ can be recursively updated using its last task counterpart $\boldsymbol{R}_{k-1,n}$ and current data (i.e., $\boldsymbol{X}_{k,1}$). This proves the recursive calculation in Equation 11.

Next, we derive the recursive formulation of $\hat{\boldsymbol{W}}^{(k,1)}$. To this end, we also recuse the cross-correlation matrix $\boldsymbol{Q}_{k,1}$ at the batch 1 of task *k*:

$$\boldsymbol{Q}_{k,1} = \begin{bmatrix} \boldsymbol{X}_{1,1}^{(\mathrm{a})\mathrm{T}}\boldsymbol{Y}_{1,1}^{\text{train}} & \dots & \boldsymbol{X}_{k-1,n}^{(\mathrm{a})\mathrm{T}}\boldsymbol{Y}_{k-1,n}^{\text{train}} & \boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}}\boldsymbol{Y}_{k,1}^{\text{train}} \end{bmatrix} = \boldsymbol{Q}'_{k-1,n} + \boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}}\boldsymbol{Y}_{k,1}^{\text{train}}. \tag{28}$$

where

$$\boldsymbol{Q}'_{k-1,n} = \begin{bmatrix} \boldsymbol{Q}_{k-1,n} & \boldsymbol{0}_{d_{(\mathrm{a})}\times d_{yk}} \end{bmatrix} \tag{29}$$

where $d_{(\mathrm{a})}$ is the dimension of activation and d_{yk} is dimension of total classes of task *k*. Note that the concatenation in Equation 29 is due to fact that $\boldsymbol{Y}_{k,1}$ of task *k* contains more data classes (hence more columns) than $\boldsymbol{Y}_{k-1,n}$

Let $\boldsymbol{K}_{k,1} = \left(\boldsymbol{I} + \boldsymbol{X}_{k,1}^{(\mathrm{a})}\boldsymbol{R}_{k-1,n}\boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}}\right)^{-1}$. Since $\boldsymbol{I} = \boldsymbol{K}_{k,1}\boldsymbol{K}_{k,1}^{-1} = \boldsymbol{K}_{k,1}\left(\boldsymbol{I} + \boldsymbol{X}_{k,1}^{(\mathrm{a})}\boldsymbol{R}_{k-1,n}\boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}}\right)$, we have $\boldsymbol{K}_{k,1} = \boldsymbol{I} - \boldsymbol{K}_{k,1}\boldsymbol{X}_{k,1}^{(\mathrm{a})}\boldsymbol{R}_{k-1,n}\boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}}$. Therefore,

$$
\begin{aligned}
\boldsymbol{R}_{k-1,n}\boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}}(\boldsymbol{I}+\boldsymbol{X}_{k,1}^{(\mathrm{a})}\boldsymbol{R}_{k-1,n}\boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}})^{-1} &= \boldsymbol{R}_{k,1}\boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}}\boldsymbol{K}_{k,1} \\
&= \boldsymbol{R}_{k,1}\boldsymbol{X}_{k}^{(\mathrm{a})\mathrm{T}}(\boldsymbol{I}-\boldsymbol{K}_{k,1}\boldsymbol{X}_{k,1}^{(\mathrm{a})}\boldsymbol{R}_{k-1,n}\boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}}) \\
&= (\boldsymbol{R}_{k-1,n}-\boldsymbol{R}_{k-1,n}\boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}}\boldsymbol{K}_{k,1}\boldsymbol{X}_{k,1}^{(\mathrm{a})}\boldsymbol{R}_{k-1,n})\boldsymbol{X}_{k}^{(\mathrm{a})\mathrm{T}} \\
&= \boldsymbol{R}_{k,1}\boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}}.
\end{aligned} \tag{30}
$$

As $\hat{\boldsymbol{W}}^{(k-1,n)'} = [\hat{\boldsymbol{W}}^{(k-1,n)} \quad \mathbf{0}]$ has expanded its dimension similar to what $\boldsymbol{Q}'_{k-1,n}$ does, we have

$$
\hat{\boldsymbol{W}}^{(k-1,n)'} = \boldsymbol{R}_{k-1,n}\boldsymbol{Q}'_{k-1,n}. \tag{31}
$$

Hence, $\hat{\boldsymbol{W}}^{(k,1)}$ can be rewritten as

$$
\hat{\boldsymbol{W}}^{(k,1)} = \boldsymbol{R}_{k,1}\boldsymbol{Q}_{k,1} = \boldsymbol{R}_{k,1}(\boldsymbol{Q}'_{k-1,n}+\boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}}\boldsymbol{Y}_{k,1}^{\text{train}}) = \boldsymbol{R}_{k,1}\boldsymbol{Q}'_{k-1,n}+\boldsymbol{R}_{k,1}\boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}}\boldsymbol{Y}_{k,1}^{\text{train}}. \tag{32}
$$

By substituting Equation 27 into $\boldsymbol{R}_{k,1}\boldsymbol{Q}'_{k-1,n}$, we have

$$
\begin{aligned}
\boldsymbol{R}_{k,1}\boldsymbol{Q}'_{k-1,n} &= \boldsymbol{R}_{k-1,n}\boldsymbol{Q}'_{k-1,n}-\boldsymbol{R}_{k-1,n}\boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}}(\boldsymbol{I}+\boldsymbol{X}_{k,1}^{(\mathrm{a})}\boldsymbol{R}_{k-1,n}\boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}})^{-1}\boldsymbol{X}_{k,1}^{(\mathrm{a})}\boldsymbol{R}_{k-1,n}\boldsymbol{Q}'_{k-1,n} \\
&= \boldsymbol{W}^{(k-1,n)'}-\boldsymbol{R}_{k-1,n}\boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}}(\boldsymbol{I}+\boldsymbol{X}_{k,1}\boldsymbol{R}_{k-1,n}\boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}})^{-1}\boldsymbol{X}_{k,1}^{(\mathrm{a})}\boldsymbol{W}^{(k-1,n)'}.
\end{aligned} \tag{33}
$$

According to Equation 30, Equation 33 can be rewritten as

$$
\boldsymbol{R}_{k,1}\boldsymbol{Q}'_{k-1,n} = \hat{\boldsymbol{W}}^{(k-1,n)'}-\boldsymbol{R}_{k,1}\boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}}\boldsymbol{X}_{k,1}^{(\mathrm{a})}\hat{\boldsymbol{W}}^{(k-1,n)'}. \tag{34}
$$

By inserting Equation 33 into Equation 32, we have

$$
\begin{aligned}
\hat{\boldsymbol{W}}^{(k,1)} &= \hat{\boldsymbol{W}}^{(k-1,n)'}-\boldsymbol{R}_{k,1}\boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}}\boldsymbol{X}_{k,1}^{(\mathrm{a})}\hat{\boldsymbol{W}}^{(k-1,n)'}+\boldsymbol{R}_{k,1}\boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}}\boldsymbol{Y}_{k,1}^{\text{train}} \\
&= \hat{\boldsymbol{W}}^{(k-1,n)'}+\boldsymbol{R}_{k,1}\boldsymbol{X}_{k,1}^{(\mathrm{a})\mathrm{T}}(\boldsymbol{Y}_{k,1}^{\text{train}}-\boldsymbol{X}_{k,1}^{(\mathrm{a})}\hat{\boldsymbol{W}}^{(k-1,n)'}),
\end{aligned} \tag{35}
$$

which proves the case in the batch 1 of task *k*.

For **Theorem 3.2**, we consider the case at rest batches of task *k*. In the rest of batches, the number of classes maintain unchanged compared with batch 1, which means no column expansion is required. According to Equation 35, we substitute $\hat{\boldsymbol{W}}^{(k-1,n)'}$ by $\hat{\boldsymbol{W}}^{(k,i-1)}$. Similar substitution is applied to $\boldsymbol{R}_{k-1,n}$ with $\boldsymbol{R}_{k,i-1}$ according to Equation 27 since the shape of regularized feature autocorrelation matrix is remained through whole learning agenda. Then we have

$$
\hat{\boldsymbol{W}}^{(k,i)} = \hat{\boldsymbol{W}}^{(k,i-1)}+\boldsymbol{R}_{k,i}\boldsymbol{X}_{k,i}^{(\mathrm{a})\mathrm{T}}\left(\boldsymbol{Y}_{k,i}^{\text{train}}-\boldsymbol{X}_{k,i}^{(a)}\hat{\boldsymbol{W}}^{(k,i-1)}\right), \tag{36}
$$

$$
\boldsymbol{R}_{k,i} = \boldsymbol{R}_{k,i-1}-\boldsymbol{R}_{k,i-1}\boldsymbol{X}_{k,i}^{(\mathrm{a})\mathrm{T}}\left(\boldsymbol{I}+\boldsymbol{X}_{k,i}^{(\mathrm{a})}\boldsymbol{R}_{k,i-1}\boldsymbol{X}_{k,i}^{(\mathrm{a})\mathrm{T}}\right)^{-1}\boldsymbol{X}_{k,i}^{(\mathrm{a})}\boldsymbol{R}_{k,i-1}, \tag{37}
$$

which complete the proof. □

https://doi.org/10.52202/079017-1314

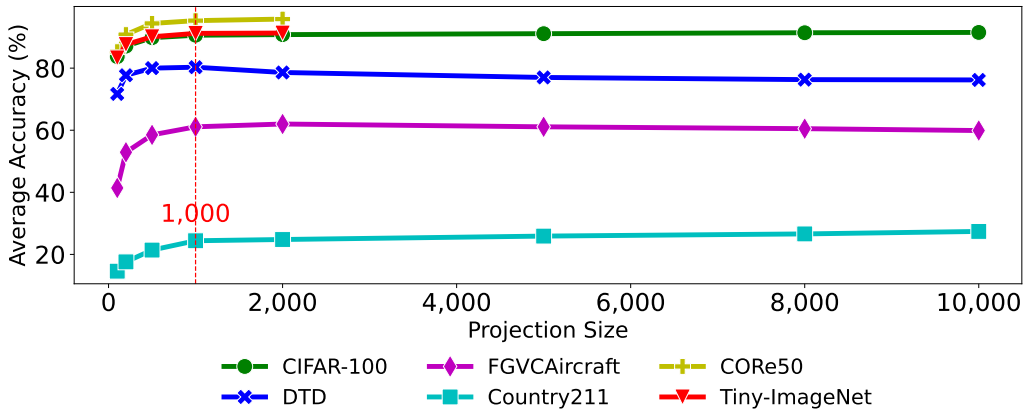

Figure 4: Average accuracy measured in different projection sizes. Due to the time complexity in computing Equation 11, higher projection sizes escalate training time while increase little performance. On small datasets, higher projection sizes may even lead to reduction.

B Appendix

Figure 4 demonstrates the influence of different random projection sizes. These experiments reveal that higher-dimensional projections may not always improve accuracy, suggesting that projection size must fit the dataset's volume. If the volume of the dataset is small, a high-dimensional projection results in over-fitting to the training set. The maximum accuracy on DTD is at 1,000 projection size, while the accuracy slightly increases after 1,000. When the projection size is 2,000, the results of FGVCAircrft and CORe50 are a little higher than the results on 1,000, but time consumption is twice higher due to the cubic time complexity in the matrix inverse. Thus, we limit the projection size measurement to 2,000 for CORe50 and Tiny-ImageNet due to the significant increase in time consumption.

Table 5 justify the contribution of AC. The reduction from 91.1% to 32.4% without AL suggesting that one epoch of simple back-propagation is unable to handle the OCIL.

Table 6 varifies the choice of 1 as regularization term is suitable for OCIL. Fine-grained datasets suffer a lot from the wrong choice of γ. The average accuracy on FGVCAircraft drops from 61.1% to 22.4% when γ is tuned from 1 to 10^{-3}.

AC	FCC	Result
✓	✗	**91.1**
✗	✓	32.4

Table 5: Average accuracy of CIFAR-100 with updating manner in classifier.

Dataset	10^2	10^1	1	10^{-1}	10^{-2}	10^{-3}
CIFAR-100	91.1	91.1	**91.1**	90.8	90.3	87.2
CORe50	95.3	95.3	**95.3**	95.2	93.6	89.9
FGVCAircraft	55.1	59.5	**61.1**	47.1	34.9	22.4
DTD	70.3	75.2	**81.6**	81.5	79.1	70.8
Tiny-ImageNet	91.2	91.2	**91.2**	90.9	90.3	89.8
Country211	23.2	23.3	**24.4**	22.6	21.3	17.4

Table 6: Average accuracy measured with different regularization terms. γ=1 shows robust results across all datasets.

https://doi.org/10.52202/079017-1314

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The most important idea of this paper is claimed in substract(i.e., forward-only learning).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Main limitation is F-OAL needs a powerful pre-trained backbone.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We have detailed proof in Apendix.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

https://doi.org/10.52202/079017-1314

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The results can be reproduced in released code.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We release the code.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have detailed information in Experiment section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification:We do not include statistical significance.

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have this in Experiment section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

https://doi.org/10.52202/079017-1314

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We follow the rules.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have this content in section 4.8.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We algorithm will not generate harmful content directly.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the source code, which is allowed by code author.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

https://doi.org/10.52202/079017-1314

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.