FedET: A Communication-Efficient Federated Class-Incremental Learning Framework Based on Enhanced Transformer

Chenghao Liu[1,2] , **Xiaoyang Qu**[1] , **Jianzong Wang**[1*] and **Jing Xiao**[1]
[1]Ping An Technology (Shenzhen) Co., Ltd., Shenzhen, China
[2]The Shenzhen International Graduate School, Tsinghua University, China
liucheng21@mails.tsinghua.com.cn, {quxiaoyang343, wangjianzong347, xiaojing661}@pingan.com.cn

Abstract

Federated Learning (FL) has been widely concerned for it enables decentralized learning while ensuring data privacy. However, most existing methods unrealistically assume that the classes encountered by local clients are fixed over time. After learning new classes, this assumption will make the model's catastrophic forgetting of old classes significantly severe. Moreover, due to the limitation of communication cost, it is challenging to use large-scale models in FL, which will affect the prediction accuracy. To address these challenges, we propose a novel framework, *Federated Enhanced Transformer* (**FedET**), which simultaneously achieves high accuracy and low communication cost. Specifically, FedET uses Enhancer, a tiny module, to absorb and communicate new knowledge, and applies pre-trained Transformers combined with different Enhancers to ensure high precision on various tasks. To address local forgetting caused by new classes of new tasks and global forgetting brought by non-i.i.d (non-independent and identically distributed) class imbalance across different local clients, we proposed an Enhancer distillation method to modify the imbalance between old and new knowledge and repair the non-i.i.d. problem. Experimental results demonstrate that FedET's average accuracy on representative benchmark datasets is 14.1% higher than the state-of-the-art method, while FedET saves 90% of the communication cost compared to the previous method.

1 Introduction

Federated learning (FL) enables each participating local client to benefit from other clients' data while ensuring client's data does not leave the local [Yang *et al.*, 2019; Dong *et al.*, 2023]. On the premise of ensuring the data privacy of all clients, the problem of data silos has been successfully solved [Hong *et al.*, 2021; Qu *et al.*, 2020]. However, most existing FL methods are modelled in static scenarios, meaning the models' classes are preset and fixed, which undoubtedly reduces the model's generality. Therefore, Federated Class-Incremental Learning (FCIL) is proposed. FCIL solves the problem that FL needs to retrain the entire model when meeting the new classes, saving time and computing costs. For FCIL, how to deal with catastrophic forgetting, seek the plasticity-stability balance of the model and ensure the cooperation of multiple parties are the keys to the problem.

To date, less work has been done on FCIL studies. The research conducted by [Hendryx *et al.*, 2021] focuses on global IL by facilitating knowledge sharing among diverse clients. However, the author overlooks the non-i.i.d distribution of classes across these distinct clients. The paper [Dong *et al.*, 2022] draw on the regularization methods used in Incremental Learning (IL) and proposes two loss functions. One for addressing the issue of forgetting old classes after IL, and the other is concentrate on the global forgetting caused by the non-i.i.d (non-independent and identically distributed) distribution of classes among different clients. However, this method needs a proxy server to achieve its best performance, leading to high communication costs and some privacy issues. To raise the accuracy of the model in FCIL settings, a natural idea is to choose a more powerful backbone model. We note that there is still no work to apply transformers to FCIL, and the biggest obstacle is that the communication cost is extremely high and cannot be reduced, which makes this application unrealistic. From another perspective, the accuracy and application scope will be significantly improved if we solve the communication and non-i.i.d. problem of class distribution between different clients.

Driven by these ideas, we propose a new Federated Enhanced Transformer (**FedET**) framework. Compared with other existing FCIL methods, FedET has better prediction performance, lower communication volume, and more universality. It has achieved excellent performance in both Computer Vision (CV) and Natural Language Process (NLP) fields, also it is more efficient when dealing with catastrophic forgetting. FedET consists of four main components: Pre-trained Transformer Blocks, Enhancer Select Module, Enhancer Pool and Sample Memory Module (only the local clients have the Sample Memory Module). FedET first divides the entire label space into multiple domains, each with its corresponding Enhancer Group. When new classes need to

[*]Corresponding author: Jianzong Wang, jzwang@188.com

learn, Enhancer Select Module will determine which domain the new classes belong to and train a temporary Enhancer Group. The new Enhancer Group is obtained by performing distillation between the temporary Group and the corresponding old one. In this way, not only can the local clients have the capability of IL, but large-scale models (such as MAE [He *et al.*, 2022]) can also be used. At the same time, because only the parameters of the chosen Enhancer Group need to be updated, the communication cost is significantly reduced.

We make the following contributions:

- We introduce FedET in order to address the FCIL problem, which mitigates the issue of catastrophic forgetting in both local and global models and effectively reduces communication overhead. According to our knowledge, it is the first effort to explore the FCIL problem in a large-scale model.
- We propose the first FCIL framework used in both CV and NLP fields. Using different transformers as backbones, FedET can handle problems in multiple fields. Compared with baseline models, FedET improves the average accuracy of image classification by 3% and text classification by 1.6%.
- We develop a new loss to handle global catastrophic forgetting named entropy-aware multiple distillation. This is the first time an FCIL model incorporating entropy as a factor when setting the loss function.
- We combine the IL problem of text classification with FL for the first time. By discussing the experimental design method and baseline selection, we think it is a new challenge for both NLP and FCIL fields.

2 Preliminary

In standard IL [Rebuffi *et al.*, 2017; Simon *et al.*, 2021; Shmelkov *et al.*, 2017], the streaming task sequence is defined by $\mathcal{T} = \{\mathcal{T}^t\}_{t=1}^{T}$, in which T represents the task order, the first t tasks $\mathcal{T}^t = \{\mathbf{x}_i^t, \mathbf{y}_i^t\}_{i=1}^{N^t}$ contains N^t pairs the sample $\mathbf{x}_i^t$ and the corresponding one-hot encoded label $\mathbf{y}_i^t \in \mathcal{Y}^t$. $\mathcal{Y}^t$ represents the label space of the t-th task, which includes the new classes $M^t = \bigcup_{b=1}^{B} m_b^t$ that have not appeared in the previous $t-1$ tasks, and B represents the number of new classes. At this time, the set of all classes that the model can judge is $M^A = \bigcup_{i=1}^{t} M^i$. Inspired by [Ermis *et al.*, 2022; Liu *et al.*, 2020], based on the unique architecture of FL, we construct a Sample Memory Module $\mathcal{S}$ located on every local client to store $\frac{|\mathcal{S}|}{M^A}$ exemplars of each class at local, and it satisfies $\frac{|\mathcal{S}|}{M^A} \ll \frac{N^t}{M^t}$.

For FCIL, we give the initial setting under the FL framework [Yoon *et al.*, 2021]: we set K local clients $\mathcal{C} = \{\mathcal{C}_k\}_{k=1}^{K}$ and a global server $\mathcal{C}_G$, the model structures on all clients and server are the same, from the perspective of parameters, including the frozen parameter Φ (that is, the parameters of the selected pre-trained backbone model) and the variable parameter θ. When a clients ($a < K$) send applications to server for Class-Incremental Learning (CIL), they will access the t-th task, updated θ, and select some samples $\{\mathbf{x}_i^t, \mathbf{y}_i^t\}$ put into Sample Memory Module.

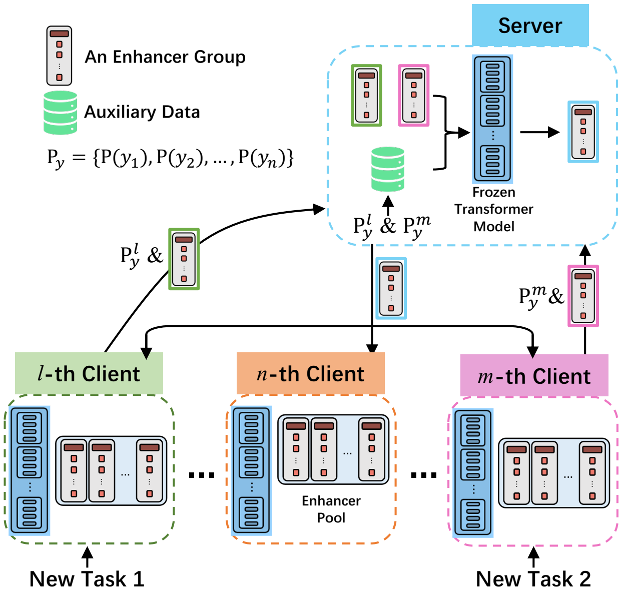


Figure 1: Simple FedET scenario when performing incremental learning. Local clients upload the weight of the selected Enhancer Group and the label distribution (P_y) of their private training data to the server after updating the group with new tasks. Then the server uses P_y to construct auxiliary data, use auxiliary data to distil upload groups, and send the updated group to all local clients.

3 Methodology

While CIL and FCIL share similarities, the key distinction between them is that FCIL involves tackling two types of forgetting: local and global. FedET addresses local forgetting through a dual distillation loss and mitigates global forgetting through auxiliary data construction and an entropy-aware multiple distillation loss. Figure 1 shows the general outline of the FedET approach.

3.1 Solution of Local Forgetting

In FedET, a local model mainly includes four parts: Pre-trained Transformer Blocks, Enhancer Select Module, Enhancer Pool and Sample Memory Module. We show the local model's specific structure and predicting process in Figure 2.

Enhancer Pool and Enhancer Group

Enhancer is the core of FedET, so it is introduced here first. An Enhancer Group contains some Enhancers and a prediction head. And the number of Enhancers is decided by the frozen Pre-trained Transformer Blocks. The mainstream methods of IL fall into three categories [Lange *et al.*, 2022]: playback, regularization, and parameter isolation. In FedET, we use a combination of three approaches: for each client, we set up an Enhancer Pool, which contains multiple Enhancer Groups $\mathcal{H} = \{\mathcal{H}^j\}_{j=1}^{J}$, each group is dedicated to being proficient in part of all existing classes. That is, for an Enhancer Group, the class it is responsible for is $M^{\mathcal{H}^j}$, and $\bigcup_{j=1}^{J} M^{\mathcal{H}^j} = M^A$. Setting the parameters of an Enhancer Group and a frozen Pre-trained transformer model as θ and Φ respectively, An Enhancer Group includes many Enhancers

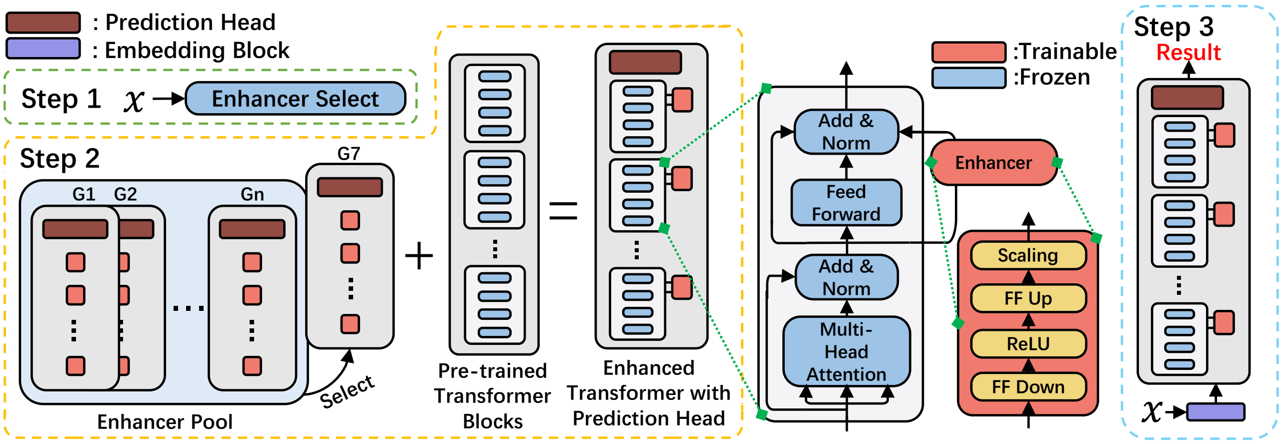


Figure 2: The workflow of making predictions by the local model in FedET. The input will first be processed by the Enhancer Select Module to decide the right Enhancer Group. Then this Group will insert into the pre-trained backbone and develop the prediction model. It should be noted that only the parameters of Enhancers are trainable, and the seventh group is used as an example in the figure.

θ_E and a prediction head θ_H. During training, only θ is modified, Φ is still frozen. Thereby greatly reducing the number of parameters that need to be adjusted without dropping accuracy. An Enhancer is comprised of three components: a down-projection with $W_{\text{down}} \in R^{n \times m}$, an activation function $f(\cdot)$, and an up-projection with $W_{\text{up}} \in R^{m \times n}$. Since the encoder structures of transformers are almost the same, after completing FedET's experiments in the NLP and CV fields, we believe this framework can be used for most of the currently known transformers.

Why choose "Enhancer + Freeze the backbone model" instead of Freeze the underlying encoder to adjust the upper encoder? We draw two perspectives from experiments and the literature [Rücklé *et al.*, 2021]. First, Enhancer is freely pluggable, and its internal structure can keep the input of the original encoder, so it can retain the maximum amount of the knowledge that the backbone model has learned during the pre-training stage. Meanwhile, through the design of the Enhancer bottleneck structure and the freezing of the pre-training model, the entire model can learn the downstream tasks better, while the number of parameters that need to be adjusted is significantly reduced. Second, we note that direct fine-tuning can easily lead to overfitting during training on downstream tasks, whereas inserting the Enhancer module performs much better. Although it can be compensated by carefully tuning hyperparameters such as learning rate and batch size, it is undoubtedly time-consuming and labour-intensive.

Enhancer Select Module and Sample Memory Module

After a sample is preprocessed, it will be input to the Enhancer Select Module $G_s(x)$. The Enhancer Select Module is a pre-trained frozen classifier. The function of this module is to select a suitable Enhancer Group to handle the input sample. The output of this classifier tells FedET which group is the right group to call up. In t-th task $\mathcal{T}^t$, the Enhancer Select Module will first select an Enhancer Group (e.g. j-th) $\mathcal{H}^j$ according to the judgement that the new class M^t is the most similar to the class $M^{\mathcal{H}^j}$, then $\mathcal{H}^j$ will participate in distillation as $\mathcal{H}^j_{\text{old}}$. There will be a randomly initialized temporary Enhancer Group $\mathcal{H}^t$ aiming to study M^t. After the study is completed, $\mathcal{H}^t$ will perform distillation with $\mathcal{H}^j_{\text{old}}$ to obtain a new Enhancer Group $\mathcal{H}^j_{\text{new}}$ which covers $\mathcal{H}^j_{\text{old}}$, and the specialized class of the group change to $M^{\mathcal{H}^j_{\text{new}}} = M^{\mathcal{H}^j} \cup M^t$. The judgment methods of the Enhancer Select Module corresponding to different task fields are also different. For CV fields, we design the Enhancer Select Module as an EfficientNet [Tan and Le, 2019], and for NLP fields, we use textRCNN [Lai *et al.*, 2015]. During the distillation process, the required old class samples $\mathcal{S}_{M^{\mathcal{H}^j_{\text{old}}}}$ are provided by the Sample Memory Module. After the new class is learned, this module will also store one typical sample of each new type $\mathcal{S}_{M^t}$, get new $\mathcal{S}_{M^{\mathcal{H}^j_{\text{new}}}} = \mathcal{S}_{M^{\mathcal{H}^j}} \cup \mathcal{S}_{M^t}$, to ensure that the subsequent distillation can proceed smoothly.

After selection, the parameters of the chosen Enhancer Group θ^{old} are linked with the frozen pre-trained model parameters Φ. The θ^{old} contains two parts: θ^{old}_E, the parameters of the Enhancers, and θ^{old}_H, the parameters of the prediction head.

Distillation of Enhancers

For a new task $\mathcal{T}^t$, the new classes need to learn is M^t. After the temporary Enhancer Group $\mathcal{H}^t$ finish studying, it will be linked with the frozen transformer model as the temporary model f_t, which contains Φ and θ^t. The frozen transformer model with $\mathcal{H}^{\text{old}}$ is called the old model f_{old}. We use the following objective to distill f_{old} and f_t:

$$f_{\text{new}}(x;\theta^{\text{new}},\Phi) = \begin{cases} f_{\text{old}}(x;\theta^{\text{old}},\Phi)[i] & 1 \le i \le m \\ f_t(x;\theta^t,\Phi)[i] & m < i \le c \end{cases} \quad (1)$$

we set

$$c = m + n \quad (2)$$

where m and n is the number of M^{old} and M^t respectively. To ensure that the consolidated model's output $f(x;\theta^{\text{new}},\Phi)$

approximates the combination of outputs from f_t and f_{old}, we utilize the output of f_t and f_{old} as supervisory signals during the joint training of the consolidated parameters θ^{new}.

To achieve this goal, we employ the double distillation loss proposed by [Zhang *et al.*, 2020] to train f_{new}. The distillation process is as follows: f_t and f_{old} are frozen, and run a feed-forward pass with every sample in training set to collect the **logits** of f_t and f_{old}:

$$\hat{y}_{\text{old}} = \{\hat{y}^1, \cdots, \hat{y}^m\}, \quad \hat{y}_t = \{\hat{y}^{m+1}, \cdots, \hat{y}^{m+n}\}$$

respectively, where the superscript is the class label. Then main requirements is to reduce the gap between the **logits** generated by f_{new} and the **logits** generated by f_t and f_{old}. Based on prior work[Zhang *et al.*, 2020], We choose L2 loss [Ba and Caruana, 2014] as the distance metric. Specifically, the training objective for consolidation is:

$$\min_{\theta^{\text{new}}} \frac{1}{|\mathcal{U}|} \sum_{x_i \in \mathcal{U}} L_{dd}(y, \dot{y}) \tag{3}$$

where $\mathcal{U}$ denotes the training samples from Sample Memory Module used for distillation. And L_{dd} is the double distillation loss:

$$L_{dd}(y, \dot{y}) = \frac{1}{m+n} \sum_{i=1}^{m+n} (y^i - \dot{y}^i)^2 \tag{4}$$

in which y^i are the **logits** produced by f_{new} for the t-th task, and

$$\dot{y}^i = \begin{cases} \hat{y}_i - \frac{1}{m} \sum\limits_{j=1}^{m} \hat{y}_j & 1 \le i \le m \\ \\ \hat{y}_i - \frac{1}{n} \sum\limits_{j=m+1}^{m+n} \hat{y}_j & m < i \le c \end{cases} \tag{5}$$

where $\hat{y}$ is the concatenation of $\hat{y}_{\text{old}}$ and $\hat{y}_{\text{new}}$. After the consolidation, the Enhancers parameters θ^{new} are used for f_{old} in the next round. The pseudo code for local forgetting solution is shown in Algorithm 1.

3.2 Solution of Global Forgetting

Global catastrophic forgetting primarily arises from the heterogeneity forgetting among local clients participating in incremental learning. Which means the non-i.i.d. class-imbalanced distributions across local clients lead to catastrophic forgetting of old classes on a global scale, further exacerbating local catastrophic forgetting. Therefore, it is necessary to solve the heterogeneity forgetting problem across clients in global perspective. To ensure precision and speed, FedET handles this problem with double distillation loss and the difference of average entropy across different clients.

Distillation of Enhancers of Different Clients

FedET changed the stereotype of having to queue up for updates and proposed a new way to update the model. The new method is more scientific, reasonable, and time-effective. When a single client uploads the new Enhancer Group obtained after distillation to the server, all the server needs to do is update the parameters of the corresponding group. When many clients upload the same Enhancer simultaneously, queuing is unscientific because only the last client's update is critical, and this is how global catastrophic forgetting happens. FedET sets a server waiting time limitation. Within a specific time, multiple schemes for an Enhancer Group will be aggregated by the server to perform global model distillation.

Algorithm 1 Local_ICL

```
Input: Enhancer Select Module $G_s(x)$
Input: Enhancer Pool $\mathcal{H} = \{\mathcal{H}^j\}_{j=1}^{J}$
Input: Sample Memory Module $\mathcal{S}$
Input: Parameters of pre-trained transformer model $\Phi$
Input: $t$-th task data $\mathcal{T}^t = \{\mathbf{x}_i^t, \mathbf{y}_i^t\}_{i=1}^{N^t}$
for $i = 1 \to N^t$ do
    Group number $j \leftarrow G_s(\mathbf{x}_i^t)$
    Put $\{\mathbf{x}_i^t, \mathbf{y}_i^t\}$ with the same $j$ into a list $L_j$
end for
for every selected $j$ do
    while Non-convergence do
        Randomly initialize temporary group $\mathcal{H}^t$
        train $f_t(x; \theta^t, \Phi)$ with $L_j$
        Sample from $L_j$ and add to $\mathcal{S}$
        $f_{\text{new}}(x; \theta^{\text{new}}, \Phi)$ = DISTILLATION$(f_t, f_{\text{old}}, \mathcal{S})$
        $\theta^j \leftarrow \theta^{\text{new}}$
        Communicate $\mathcal{H}^j$ with Server to get Global best $j$-th group in this turn
    end while
end for

function DISTILLATION$(f_t, f_{\text{old}}, \mathcal{S})$
Get $\hat{y}_{\text{old}}$ from $f_{\text{old}}$ and $\mathcal{S}$
Get $\hat{y}_t$ from $f_t$ and $\mathcal{S}$
Compute loss function as in Eq.4 and train $f_{\text{new}}$
return $f_{\text{new}}$
```

For global distillation, the server will distill some Enhancer Group at same time, which means there will be $f_t^1, f_t^2, \cdots, f_t^q (q < \textit{the number of clinets})$ and a f_{old} distill together. Note that the new classes are learned by all uploaded groups. Suppose the class which the distilled Enhancer Group major in is M^t. For every group-uploaded client, they also upload the information entropy $H(M^t)$ of M^t to the server. The server uses $H(M^t)$ to judge the importance of each f_t, in detail, the consolidated model of global distillation is:

$$f_{\text{new}}(x; \theta^{\text{new}}, \Phi) = \begin{cases} f_{\text{old}}(x; \theta^{\text{old}}, \Phi)[i] & 1 \le i \le m \\ \\ \sum\limits_{k=1}^{q} \frac{H^k}{H_{\text{sum}}} f_t^k(x; \theta^t, \Phi)[i] & m < i \le c \end{cases} \tag{6}$$

where H_{sum} is the sum of information entropy $H(M^t)$ of all uploaded clients. Noted that all output of f here are **logits**, not hard-label. To get θ^{new}, the entropy-aware multiple distillation loss L_{emd} is:

$$L_{emd}(y, \ddot{y}) = \frac{1}{m+n} \sum_{i=1}^{m+n} (y^i - \ddot{y}^i)^2 \tag{7}$$

Model	Method	Updated Paras.	Training Time
BERT	Fine-tuning	110.01×10^6	1.92 sec
	Enhancer	1.76×10^6	1.19 sec
ViT-Base	Fine-tuning	75.99×10^6	0.94 sec
	Enhancer	1.19×10^6	0.59 sec

Table 1: The communication cost and computation cost difference between whether inserting Enhancer or not. Here Updated Paras. refers to the number of updated parameters

in which $\ddot{y}$ is:

$$\ddot{y}^i = \begin{cases} \hat{y}_i - \frac{1}{m} \sum_{j=1}^{m} \hat{y}_j & 1 \le i \le m \\ \\ \hat{y}_i - \frac{1}{nH_{\text{sum}}} \sum_{j=m+1}^{m+n} \sum_{k=1}^{q} H^k \hat{y}_j^k & m < i \le c \end{cases} \tag{8}$$

Because of the nature of FL, we cannot rely solely on the sampled data to consolidate the updated Enhancers. Therefore, auxiliary data must be used. During local Enhancer distillation, we generate $\mathcal{U}$ using the Sample Memory Module, which stores one representative sample per class and utilizes data augmentation to create the dataset. For global distillation, we construct an equivalent dataset to approximate the training samples. After the local clients send the label distribution P_y to the server, the server can construct the auxiliary datasets using available data of a similar domain. Notably, these auxiliary datasets are dynamically fetched and inputted in mini-batches, reducing the storage burden, and discarded after distillation is complete.

Communication Cost Analysis

The parameter quantity of a single Enhancer is $2mn+n+m$. For a single local model, if there are D Encoder modules in one Enhancer Group, after adding a group of Enhancers, the increased parameter quantity is:

$$D \times (2mn + n + m) + n \times labels \tag{9}$$

Other parameters are frozen except for the Enhancer Group and prediction head in the model. As shown in Table 1, the network parameters that need to be transmitted are reduced by more than 70% compared with the various FCIL models previously proposed.

Computation Cost Analysis

The computation FLOPs for each Enhancer in the forward pass are $2 \times m \times n \times$ *sequence length* (normalized to a single data sample). The overhead incurred in this way is negligible compared to the original model complexity, e.g., less than 1% on BERT. In the meantime, since all other parameters are fixed during the training period, the computation during backpropagation is reduced by skipping the gradient that computes most of the weights. As shown in Table 1, the use of Enhancers reduces the training time by about 40%.

4 Experiments

As discussed in Section 1, since transformers are widely used in both NLP and CV fields, we test the performance of FedET on image and text classification tasks. The complete setup will be described in the following subsections.

Dataset	Class	Type	Train / Test
AGnews	4	News	8000 / 2000
Yelp	5	Sentiment	10000 / 2500
Amazon	5	Sentiment	10000 / 2500
DBpedia	14	Wikipedia Article	28000 / 7000
Yahoo	10	Q&A	20000 / 5000

Table 2: The text classification dataset we used includes statistics on various domains of classification tasks.

Order	Task Sequence
1	AGnews→Yelp→Yahoo
2	Yelp→Yahoo→AGnews
3	Yahoo→AGnews→Yelp
4	AG→Yelp→Amazon→Yahoo→DBpedia
5	Yelp→Yahoo→Amazon→DBpedia→AGnews
6	DBpedia→Yahoo→AGnews→Amazon→Yelp

Table 3: Six different dataset sequences for NLP experiments

4.1 Natural Language Processing (NLP)

Datasets and Baselines

Owing to the limited label space of a single dataset, we integrated five text classification datasets [Chen *et al.*, 2020] to evaluate FedET. Table 2 displays the details of the dataset. Considering the domain similarity of Yelp and Amazon, we merge their label spaces for a total of 33 classes. We followed specific task sequences as outlined in Table 3 during training. To alleviate the impact of sequence length and task order on experiment results, we test task sequences of length-3 and length-5 in different orders. The first three tasks of length-3 sequences are a cyclic shift of AGnews→Yelp→Yahoo, which belong to three distinct domains (news classification, sentiment analysis, Q&A classification). The remaining three task sequences of length-5 follow the experimental design proposed by [de Masson d'Autume *et al.*, 2019]. During validation, the validation set comprise all classes.

Currently, there is no text classification in the FCIL field, so we choose the baseline of text classification in the Class-Incremental Learning (CIL) field and federate it to form the baseline of this experiment. We compare FedET with five baselines:

- **Finetune** [Yogatama *et al.*, 2019] **+ FL**: Only new tasks are used to fine-tune the BERT model in turn.
- **Replay** [de Masson d'Autume *et al.*, 2019] **+ FL**: Replay some old tasks examples during new-tasks-learning to **Finetune** the model.
- **Regularization + Replay + FL**: On the foundation of **Replay**, add an L2 regularization term to the hidden state of the classifier following BERT.
- **IDBR** [Huang *et al.*, 2021] **+ FL**: On the basis of **Regularization + Replay + FL**, replace the L2 regularization term with an information disentanglement-based regularization term.

Model	**Length-3 Task Sequences**				**Length-5 Task Sequences**			
Order	**1**	**2**	**3**	**Average**	**4**	**5**	**6**	**Average**
Finetune + FL	25.79	36.56	41.01	34.45	32.37	32.22	26.44	30.34
Replay + FL	69.32	70.25	71.31	70.29	68.25	70.52	70.24	69.67
Regularization + Replay + FL	71.50	70.88	72.93	71.77	72.28	73.03	72.92	72.74
IDBR + FL	71.80	72.72	73.08	72.53	72.63	73.72	**73.23**	73.19
FedET	**73.12**	**73.57**	**74.28**	**73.66**	**73.83**	**74.23**	73.18	**73.75**
MTL + FL	74.16	74.16	74.16	74.16	75.09	75.09	75.09	75.09

Table 4: Performance comparisons between FedET and other incremental text classification baseline methods

Methods	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**	**Avg.**	**Communication Cost per Task**
iCaRL + FL	73.5	61.3	55.7	45.9	45.0	39.7	36.7	33.9	32.2	31.8	46.5	10.82×10^6
BiC + FL	74.3	63.0	57.7	51.3	48.3	46.0	42.7	37.7	35.3	34.0	49.0	10.82×10^6
PODNet + FL	74.3	64.0	59.0	56.7	52.7	50.3	47.0	43.3	40.0	38.3	52.6	10.82×10^6
SS-IL + FL	69.7	60.0	50.3	45.7	41.7	44.3	39.0	38.3	38.0	37.3	46.4	10.82×10^6
DDE + iCaRL + FL	76.0	57.7	58.0	56.3	53.3	50.7	47.3	44.0	40.7	39.0	52.3	10.82×10^6
GLFC	73.0	69.3	68.0	61.0	58.3	54.0	51.3	48.0	44.3	42.7	57.0	10.82×10^6
FedET($J = 10$)	**83.2**	**75.7**	**72.0**	**69.4**	**67.9**	**65.8**	**63.4**	**62.1**	**61.0**	**60.6**	**68.1**	1.19×10^6

Table 5: Comparison of FedET's performance with other CV baselines in ten incremental tasks on ImageNet-Subset. During the experiment, FedET only communicates the parameter of the changed Enhancer Group, and other methods update the entire model(ResNet18).

- **Multi-task Learning** (MTL): Train the model with all class in one task. This approach represents an upper bound on the performance achievable through incremental learning.

Implementation Details

In this NLP experiment, we set $J = 3$ Enhancer Groups and $K = 10$ local clients. The prediction head for each group is a linear layer with a Softmax activation function. For sample collection (i.e. experience replay), we stored 3% (store ratio $\gamma = 0.03$) of observed examples in the Sample Memory Module, which is used for local Enhancer distillation. We choose the pre-trained Bert-Base-Uncased from HuggingFace Transformers [Wolf *et al.*, 2020] as our backbone model. All experiments utilized a batch size of 16 and a maximum sequence length of 256. During training, ADAMW [Loshchilov and Hutter, 2019] is used as the optimizer, with a learning rate $lr = 3e^{-5}$ and a weight decay 0.01 for all parameters. For each round of global training, three clients are randomly selected for ten epochs of local training. Selected clients are randomly given 60% of the classes from the label space of its seen tasks.

Results

As shown in Table 1 and Table 4, we can directly see the importance of experience replay for FCIL in NLP. Moreover, the simple regularization approach based on experience replay consistently improves results across all six orders. In most cases, FedET achieves higher performance in incremental learning compared to other baseline methods, while significantly reducing the communication cost. Specifically, compared to IDBR+FL, FedET's Enhancer structure adds a segmentation step to the regularisation and empirical replay, further improving the performance of the model.

4.2 Computer Version(CV)

Datasets and Baselines

We use ImageNet-Subset [Deng *et al.*, 2009] and CIFAR-100 [Krizhevsky *et al.*, 2009] to evaluate our method. We follow the same protocol as iCaRL [Rebuffi *et al.*, 2017] to set incremental tasks and we use the same order generated from iCaRL for a fair comparison. In detail, we compare FedET with the following baselines in the FL scenario: **iCaRL**, **BiC** [Wu *et al.*, 2019], **PODNet** [Douillard *et al.*, 2020], **SS-IL** [Ahn *et al.*, 2021], **DDE+ iCaRL** [Hu *et al.*, 2021] and **GLFC** [Dong *et al.*, 2022].

Implementation Details

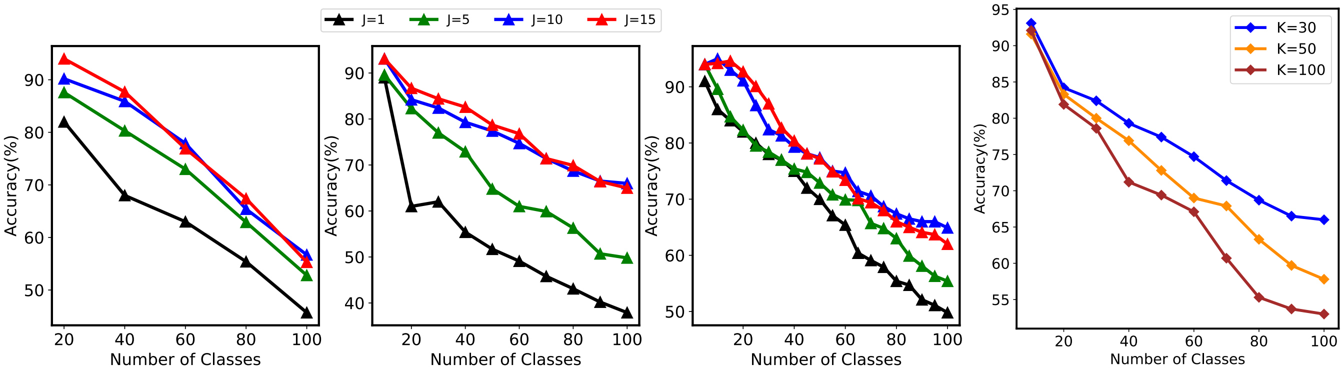
In this CV experiment, we set $J = 10$ for Enhancer Groups and $K = 30$ for local. The prediction head for each group is a linear layer with a Softmax activation function. We collected samples at a store ratio of $\gamma = 0.01$. In the CIL baselines, we choose ResNet18 [He *et al.*, 2016] to be the backbone with cross-entropy as the classification loss. On the other hand, FedET uses frozen pre-trained ViT-Base [He *et al.*, 2022] as the backbone. All experiments have a batch size of 64. The training of the Enhancer used an SGD optimizer with minimum learning rate $lr_{min} = 1e^{-5}$ and a base learning rate $lr_b = 0.1$. In each round of global training, ten clients are randomly selected for ten local-training epochs. Selected clients are randomly given 60% of the classes from the label space of its seen tasks.

Results

Table 6 and Table 5 show that FedET consistently outperform all the baselines by $3.3\% \sim 10.5\%$ in terms of average accuracy and reduces the communication cost to 11.0% of baseline models. These results demonstrate that FedET can cooperatively train a global class-incremental model in

Methods	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**	**Avg.**	**Communication Cost per Task**
iCaRL + FL	89.0	55.0	57.0	52.3	50.3	49.3	46.3	41.7	40.3	36.7	51.8	10.82×10^6
BiC + FL	88.7	63.3	61.3	56.7	53.0	51.7	48.0	44.0	42.7	40.7	55.0	10.82×10^6
PODNet + FL	89.0	71.3	69.0	63.3	59.0	55.3	50.7	48.7	45.3	45.0	59.7	10.82×10^6
SS-IL + FL	88.3	66.3	54.0	54.0	44.7	54.7	50.0	47.7	45.3	44.0	54.9	10.82×10^6
DDE + iCaRL + FL	88.0	70.0	67.3	62.0	57.3	54.7	50.3	48.3	45.7	44.3	58.8	10.82×10^6
GLFC	90.0	82.3	77.0	72.3	65.0	66.3	59.7	56.3	50.3	50.0	66.9	10.82×10^6
FedET($J = 1$)	89.0	61.0	62.0	55.4	51.7	49.1	45.8	43.1	40.2	37.9	53.2	1.19×10^6
FedET($J = 5$)	89.6	82.3	77.0	72.9	64.8	61.0	59.9	56.3	50.7	49.8	66.4	1.19×10^6
FedET($J = 10$)	**93.1**	**84.2**	**82.4**	**79.3**	**77.4**	**74.7**	**71.4**	**68.7**	**66.5**	**66.0**	**76.4**	1.19×10^6

Table 6: Comparison of FedET's performance with other CV baselines in ten incremental tasks on CIFAR-100. During the experiment, FedET only communicates the parameter of the changed Enhancer Group, and other methods update the entire model(ResNet18).



(a) Different incremental tasks with different number of Enhancer Groups J when Incremental size equals to 5 (left), 10 (middle) and 20 (right).

(b) Same incremental tasks with different number of clients K on CIFAR-100

Figure 3: Ablation study

conjunction with local clients more efficiently. Furthermore, for all incremental tasks, FedET has steady performance improvement over other methods, validating the effectiveness in addressing the forgetting problem in FCIL.

Ablation Studies

Table 6 and Figure 3 illustrate the results of our ablation experiments on the number of Enhancers and local clients.

The number of Enhancers (J) Figure 3 (a) shows the model's performance in four cases, $J = 1, J = 5, J = 10, J = 15$, respectively. Compared with $J = 10$, the performance of $J = 1$ and $J = 5$ is worse but $J = 15$ is better. Since the Enhancer Select Module is frozen, in FedET, we cannot set $J =$ *the number of classes*. It has been verified that a suitable value of J will make FedET powerful and efficient. We observed that increasing the value of J makes FedET perform better at the beginning of incremental learning but degrades faster as learning progresses. We believe that the reasons may be: (1) With the increase of J, the requirements for the Enhancer Select Module are higher. When the Enhancer Select Module cannot precisely perform rough classification, the model accuracy is bound to decrease. (2) As J increases, the learning cost on a single Enhancer Group is lower, which means that as long as the Enhancer Select Module selects the correct group, the possibility of accurate judgment will be significantly improved.

The number of clients (K) We tested the performance of FedET with client numbers of 30, 50, and 100, as depicted in Figure 3(b). It is clear that FedET's capacity declines as K increases. The performance decrease is most noticeable when $K = 100$. We believe that with an increase in K, the central server needs to perform distillation on more enhancers simultaneously, which causes the model to not fully converge within the specified number of iterations, resulting in a decrease in model performance.

5 Conclusion

FedET is an FCIL framework that can be used in many fields. Based on previous work, it introduces transformers to improve the accuracy of FCIL and increase the application field of the framework. To reduce communication costs and streamline training, only the Enhancer and its related components are designated as trainable parameters. Our experiments on datasets in NLP and CV demonstrate that FedET outperforms existing methods in FCIL while decreasing communication information by up to 90%.

Acknowledgements

This paper is supported by the Key Research and Development Program of Guangdong Province under grant No.2021B0101400003. Corresponding author is Jianzong Wang from Ping An Technology (Shenzhen) Co., Ltd.

References

[Ahn *et al.*, 2021] Hongjoon Ahn, Jihwan Kwak, Subin Lim, Hyeonsu Bang, Hyojun Kim, and Taesup Moon. SS-IL: separated softmax for incremental learning. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 824–833, 2021.

[Ba and Caruana, 2014] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2654–2662, 2014.

[Chen *et al.*, 2020] Jiaao Chen, Zichao Yang, and Diyi Yang. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2147–2157, 2020.

[de Masson d'Autume *et al.*, 2019] Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. Episodic memory in lifelong language learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13122–13131, 2019.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255, 2009.

[Dong *et al.*, 2022] Jiahua Dong, Lixu Wang, Zhen Fang, Gan Sun, Shichao Xu, Xiao Wang, and Qi Zhu. Federated class-incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10154–10163, 2022.

[Dong *et al.*, 2023] Jiahua Dong, Yang Cong, Gan Sun, Yulun Zhang, Bernt Schiele, and Dengxin Dai. No one left behind: Real-world federated class-incremental learning. abs/2302.00903, 2023.

[Douillard *et al.*, 2020] 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*, pages 86–102, 2020.

[Ermis *et al.*, 2022] Beyza Ermis, Giovanni Zappella, Martin Wistuba, and Cédric Archambeau. Memory efficient continual learning for neural text classification. abs/2203.04640, 2022.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016.

[He *et al.*, 2022] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 15979–15988, 2022.

[Hendryx *et al.*, 2021] Sean M. Hendryx, Dharma Raj KC, Bradley Walls, and Clayton T. Morrison. Federated reconnaissance: Efficient, distributed, class-incremental learning. abs/2109.00150, 2021.

[Hong *et al.*, 2021] Zhenhou Hong, Jianzong Wang, Xiaoyang Qu, Jie Liu, Chendong Zhao, and Jing Xiao. Federated learning with dynamic transformer for text to speech. In *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, pages 3590–3594, 2021.

[Hu *et al.*, 2021] Xinting Hu, Kaihua Tang, Chunyan Miao, Xian-Sheng Hua, and Hanwang Zhang. Distilling causal effect of data in class-incremental learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 3957–3966, 2021.

[Huang *et al.*, 2021] Yufan Huang, Yanzhe Zhang, Jiaao Chen, Xuezhi Wang, and Diyi Yang. Continual learning for text classification with information disentanglement based regularization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2736–2746, 2021.

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

[Lai *et al.*, 2015] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 2267–2273, 2015.

[Lange *et al.*, 2022] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory G. Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. 44(7):3366–3385, 2022.

[Liu *et al.*, 2020] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *2020*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020, pages 12242–12251, 2020.

[Loshchilov and Hutter, 2019] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[Qu *et al.*, 2020] Xiaoyang Qu, Jianzong Wang, and Jing Xiao. Quantization and knowledge distillation for efficient federated learning on edge devices. In *22nd IEEE International Conference on High Performance Computing and Communications, HPCC 2020, Yanuca Island, December 14-16, 2020*, pages 967–972, 2020.

[Rebuffi *et al.*, 2017] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5533–5542, 2017.

[Rücklé *et al.*, 2021] Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. Adapterdrop: On the efficiency of adapters in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7930–7946, 2021.

[Shmelkov *et al.*, 2017] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 3420–3429, 2017.

[Simon *et al.*, 2021] Christian Simon, Piotr Koniusz, and Mehrtash Harandi. On learning the geodesic path for incremental learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 1591–1600, 2021.

[Tan and Le, 2019] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 6105–6114, 2019.

[Wolf *et al.*, 2020] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45, 2020.

[Wu *et al.*, 2019] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 374–382, 2019.

[Yang *et al.*, 2019] Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu. *Federated Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2019.

[Yogatama *et al.*, 2019] Dani Yogatama, Cyprien de Masson d'Autume, Jerome T. Connor, Tomás Kociský, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, and Phil Blunsom. Learning and evaluating general linguistic intelligence. abs/1901.11373, 2019.

[Yoon *et al.*, 2021] Jaehong Yoon, Wonyong Jeong, Giwoong Lee, Eunho Yang, and Sung Ju Hwang. Federated continual learning with weighted inter-client transfer. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, pages 12073–12086, 2021.

[Zhang *et al.*, 2020] Junting Zhang, Jie Zhang, Shalini Ghosh, Dawei Li, Serafettin Tasci, Larry P. Heck, Heming Zhang, and C.-C. Jay Kuo. Class-incremental learning via deep model consolidation. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2020, Snowmass Village, CO, USA, March 1-5, 2020*, pages 1120–1129, 2020.