# Technical Report

Jiawei Zhan*, Yi Zeng*, Tianliang Zhang, Xiaochen Chen, Bin-Bin Gao, Jun Liu
Tencent Youtu Lab, Shenzhen, China
{gavynzhan, sylviazeng, linuszhang, husonchen, danylgao, juliusliu}@tencent.com

## Abstract

*Continual Learning (CL) is actively being applied to different areas of Computer Vision such as autonomous driving, retail, etc. The main goal of this task is to handle the problem of catastrophic forgetting, where a new learned model fails to recognize older learned data. This paper proposes a collection of strategies which helps to train a robust classification CL model for EgoObjects Dataset. Our approach firstly uses a replay approach to tackle catastrophic forgetting problems. Then, Exp-Aware Buffer Management takes into account long temporal distance categories and CWR further boost our performance. Additionally, we utilize a Exp-oriented Learning Rate Decay scheduler which reduces the learning rate after each experience. These strategies helped to get 52.23% average mean class accuracy (AMCA) on the Test set.*

## 1. Introduction

In this report, we describe the approaches and strategies used to address the continual instance-level object classification track of the 3rd CLVision workshop. Our method will be described from several perspectives such as data augmentation, model training as well as general and incremental strategies.

### 1.1. Challenge

The "Continual instance-level object classification" track features a stream of **15 experiences** obtained by cropping the main object found in each video. The videos are first-view and the model needs to classify a total of **1110 categories** that are contained in a total of **200+ semantic categories** (inaccessible in training and inference), so there are quite a few semantically overlapping targets in the 1110 class instances. This track features a fully supervised (labels are given for the training set) learning scheme in which the stream of incremental experiences is modeled by following the **Class-Incremental scenario**. Therefore, each experiment requires a new classification of **74 categories** while maintaining the performance with respect to the categories which have been seen before. No task labels or other additional signals are provided at test time. Critical requirements are as follows:

- Pretraining is allowed only on ImageNet-1K.
- The size of each model obtained after each training iteration is limited to 70M parameters.
- The maximum size of replay buffer is 3500.
- Test-time training or tuning is not allowed.

### 1.2. The EgoObjects Dataset

EgoObjects dataset, provided by Meta, is used for this challenge. Critical features of this dataset are as follows:

- Videos feature a great variety of lighting conditions, scale, camera motion, and background complexity.
- Each video depicts one main object.
- The main object is used in the object classification track (by cropping the relevant portion of the image) and class labels are taken at the instance level.
- The scene complexity (amount of objects, occlusions, etc) is less than the one found in COCO, but the image quality is more varied

## 2. Methodology

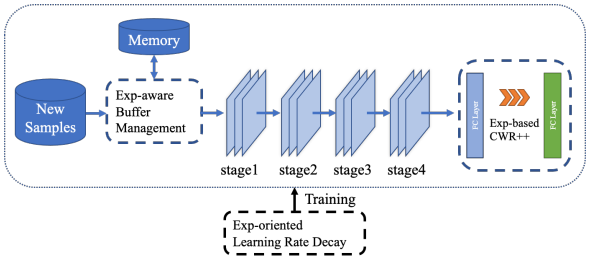


Figure 1. The overall pipeline of our methods

*These authors contributed equally to this work.

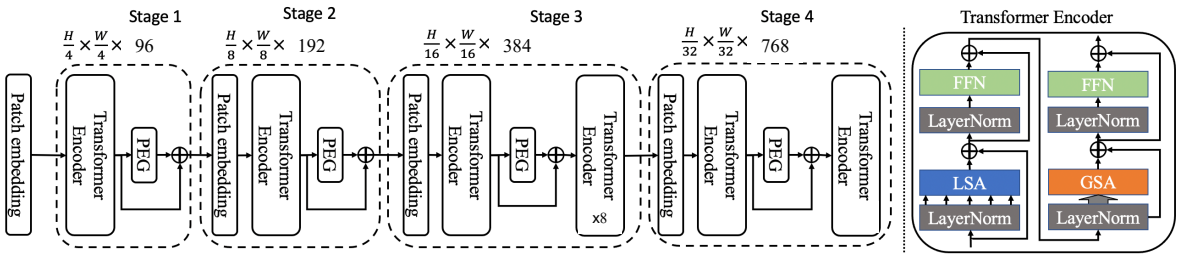


Figure 2. The Twins-SVT-Base backbone model

## 2.1. Overall Architecture

As shown in Fig.1, our overall solution consists of the following modules, firstly, the training data of the current experience is generated by Experience-aware Buffer Management, after that the training of the feature extractor and classifier is performed. The weight fusion between experiences is performed by Experience-based CWR++ after the training is finished. The training process uses Experience-oriented Learning Rate Decay to alleviate the catastrophic forgetting among experiences.

## 2.2. Backbone and Hyperparameters

We exploit the Twins-SVT-Base as backbone model [3] as shown in Fig.2, which is an improved version of the PVT model. The final fully connected layer is replaced with 1110 to match the number of categories in the competition dataset. The pretrained weights were directed obtained from Timm's official website*. The reported model top-1 accuracy on ImageNet is 83.2%, and the number of model parameters is 56M. We use the Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.03, with momentum and weight decay set to 1e-4. Within training an exp, we train a total of 25 epochs and adjust the learning rate using cosine schedule.

As loss function, we use the Cross Entropy (CE) loss, which can be expressed as:

$$\mathcal{L}(y, p) = \sum_{k=1}^{K} -y_k \log(p_k) \tag{1}$$

where $p_k$ is the prediction of the network (logits) while $y_k$ is the ground truth label. In addition to the CE loss, we trained the network with a label smoothing of parameter $\alpha$, i.e., minimizing the expected value of the cross-entropy between the modified targets $y_k^{LS}$ and the networks' output $p_k$, where $y_k^{LS}$ is defined as below:

$$y_k^{LS} = y_k(1-\alpha) + \frac{\alpha}{K} \tag{2}$$

*https://github.com/rwightman/pytorch-image-models/blob/master/timm/models/twins.py

Besides, batch size is set to be 32. The training was performed on an Nvidia Tesla V100 GPU device and could be completed in less than 10 hours. The GPU memory used for training was approximately 11 GB.

## 2.3. Data Augmentations

The input batch is first transformed in a tensor of size (b, c, w, h), where b=32 (batch size), c=3 (RGB image channel), and w=h=224 (image size). Since the training input is restricted in benchmark to be read with a forced size of 224×224, there is no significant gain from setting a larger scaling size in the data transformation. We use Albumentations [2] to perform most data augmentations, including shift scale rotate, blur, distortion, and noise.

Besides, due to the characteristics of the dataset, many samples from different categories were semantically close to each other. It was difficult to distinguish whether some cases were the same instance or not. So we considered letting the model exploit brightness and contrast hue to assist in classification. We turned off those augmentations and found that this improved the overall performance by about 1%.

In addition, fused-like augmentation methods such as cutout [4], mixup [10], cutmix [9], and mosaic [1] did not achieve performance gains in our experiments, so we deactivated these strategies in the final delivered solution.

## 2.4. Continual Learning Strategy

Regarding continual learning, which is the core objective of our competition, we mainly proposed three strategies, i.e., Exp-oriented Learning Rate Decay, Exp-aware Buffer Management and Exp-based CWR++ to alleviate catastrophic forgetting.

### 2.4.1 Exp-oriented Learning Rate Decay

When training deep neural networks, it is often useful to reduce the learning rate as the training progresses. Previously this technique was only used between different epochs within a task (experience). However, we find that after the first experience is completed, the feature distribution of the model has not changed very much. As a result, using a large

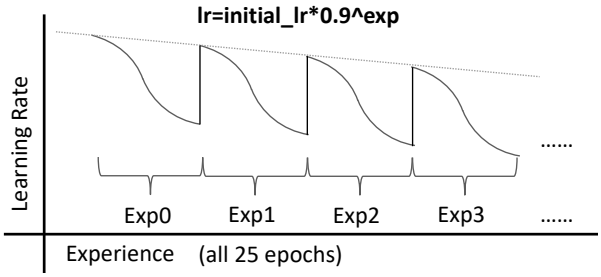


Figure 3. Exp-oriented Learning Rate Decay

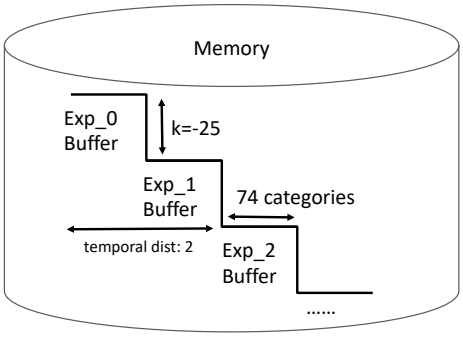


Figure 4. Exp-aware Buffer Management

learning rate in the later experience has a higher probability of forgetting old categories due to randomness, so we proposed to gradually apply a decay to the initial learning rate between experiences. This strategy leads to a little improvement in the average accuracy.

### 2.4.2 Exp-aware Buffer Management

In continuous learning, replay [5] is known to be one of the most effective techniques to tackle the catastrophic forgetting problem. Therefore, this strategy is essential to our solution.

According to the rules, the replay buffer can only contain a maximum of three thousand and 5 hundred training samples. Under this condition, we find that in the early experiences, whether we make the samples in the replay buffer balanced among experiences or classes, it is able to significantly reduce the problem of forgetting old classes. However, in the later experiences, there emerges a performance drop in the classes belonging to the earlier experiences. This implies that old class forgetting starts to show up during the training of the later experiences. Based on these observations, we propose a novel buffer management strategy termed Experience-aware Buffer Management (EBM).

Specifically, the number of samples for categories belonging to the same experience should be equal. As for classes from different experiences, we should treat them differently considering that categories from earlier experiences are more easily to be forgotten. In our experiments, between two adjacent experiences, our replay buffer will take 25 more samples from the earlier experience. We have also tried other numbers instead of 25, and find that there is little difference to the final results. Therefore, our EBM is not very sensitive to this hyper-parameter.

### 2.4.3 Exp-based CWR++

In [6], CopyWeights with Re-init (CWR) is proposed to isolate the subsets of weights that each class uses. This method performs fairly well, in spite of its simplicity. CWR maintains two sets of weights for the output classification layer, i.e., the consolidated weights (cw) used for inference and the temporary weights (tw) used for training. cw is initialized to 0 before the first experience, while tw is randomly re-initialized (e.g., Gaussian initialization with std = 0.01, mean = 0) before each training experience. At the end of each experience, the weights in tw corresponding to the classes in the current experience are scaled and copied to cw. This strategy works in scenarios where the classes between the different experiences are completely segregated.

To avoid annoyingly tuning hyper parameters, CWR+ is proposed in [7]. Two major improvements are introduced in CWR+, i.e. mean-shift and zero initialization. For more details, we refer readers to [7].

In this challenge, considering the overlapping of categories between different experiences, caused by using replay data, we improved on the basis of CWR+ and proposed experience-based CWR++. In contrast to CWR+, the weights in tw corresponding to the new classes in the current experience are also scaled and copied to cw, but for old classes present in the replay buffer, a moving average is calculated using cw and tw. The weight of the moving average is the ratio of the number of training samples when the class first appeared to the number of samples corresponded in the current replay buffer. Our experience-based CWR++ works at the end of each experience, and can greatly alleviate the

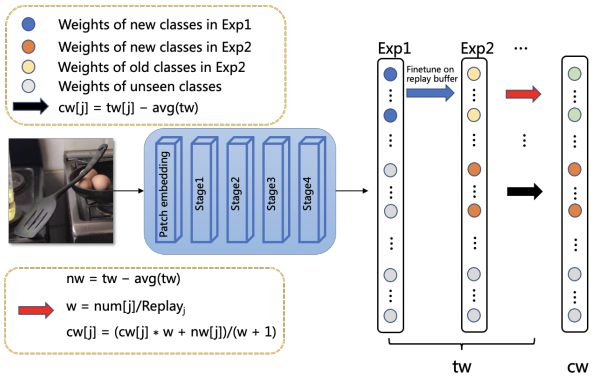


Figure 5. Exp-based CWR++

**Algorithm 1** Exp-based CWR++.

**Input:** new data (containing $s_i$ classes), replay data (containing $s_0 + s_1 + \cdots + s_{i-1}$ classes)

**cw** $= 0$
Init $\theta$ from a model pre-trained on ImageNet-1K.
**for** each training experience $E_i$ **do**
  **tw** $= 0$ (for all neurons in the output layer)
  **for** each class j **do**
    **if** j in **cw then**
      **tw**$[j]$ = **cw**$[j]$
    **end if**
  **end for**
  Train the model with SGD using both new data and replay data
  **nw** = **tw** $- avg($**tw**$)$
  **for** each class j **do**
    **if** j in **cw then**
      **if** $Replay_j > 0$ ($Replay_j$ is the number of samples of class j in current replay data) **then**
        $w = num[j]/Replay_j$
        **cw**$[j] = ($**cw**$[j] * w +$ **nw**$[j])/(w + 1)$
      **end if**
    **else**
      **cw**$[j]$ = **nw**$[j]$
      $num[j] = New_j$ ($New_j$ is the number of samples of class j in current new data)
    **end if**
  **end for**
**end for**

forgetting of old classes with little impact on the accuracy of new class recognition. It is very useful on longer sequences of tasks and can be seen as a simpler and lighter counterpart to PNN [8], with a fixed number of shared parameters. In addition, we find that it is better to not freeze the weights of the feature extractor in our experiments. The average accuracy is about 1.2 percent higher than using freezed weights. Details are shown in Algorithm 1.

## 3. Contribution of the Components

In this section, we provide some remarks regarding the contribution of each component of the proposed solution. We utilize *, **, or *** to represent three levels of importance. Components that are the most important for our solution (in terms of final accuracy on the validation set) are given ***, components that contributed in a minor way are given **, and components that do not contribute much to the final accuracy (changing them only improves the accuracy by less than 1%) are given *.

Continuous items (such as learning rate) are assessed by adjusting them in a fair manner around the discovered optimal value. Discrete items (such as the model) are compared to other similar options (e.g. models with a similar number of parameters). The difference between utilizing and not using the analyzed component is used to assess on/off components (such as cutout). Table 1 shows a summary of the contributions.

| Component | Contribution |
|---|---|
| Model | ** |
| Learning Rate | * |
| Optimizer | *** |
| Label Smooth | *** |
| Data Augmentation | ** |
| Replay | *** |
| Exp-aware Buffer Management | ** |
| Exp-based CWR++ | ** |
| Exp-oriented Learning Rate Decay | * |

Table 1. The importance of the contribution of each component of the proposed solution based on the final accuracy on the validation set. 3 stars *** indicates maximum importance, 1 star * indicates limited importance.

The contribution ratings are not completely objective, and the indicated importance should not be used to create continuous learning procedures. Nonetheless, we believe that this will be valuable to the readers as well as future research and effort.

## 4. Conclusion

In this work, we investigate the problem of continual learning from semantically similar data streams of a large number of instance classes. In addition, we propose Experience-aware buffer management to address the potential forgetting problem in long-time continuous learning. Then, we propose an improved CWR method to solve the fusion problem of categorization heads based on replay. Also, we provide data augmentation and learning rate scheduling methods to improve the performance of the model. From the results, we can see that the performance of instance classification has been further improved and the problem of catastrophic forgetting has been well solved.

## References


[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection, 2020. 2

[2] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. Albumentations: Fast and flexible image augmentations. *Information*, 11(2), 2020. 2

[3] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen.

Twins: Revisiting the design of spatial attention in vision transformers. In *NeurIPS 2021*, 2021. 2

[4] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 2

[5] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018. 3

[6] Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In *Conference on Robot Learning*, pages 17–26. PMLR, 2017. 3

[7] Davide Maltoni and Vincenzo Lomonaco. Continuous learning in single-incremental-task scenarios. *Neural Networks*, 116:56–73, 2019. 3

[8] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016. 4

[9] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *International Conference on Computer Vision (ICCV)*, 2019. 2

[10] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 2