Class-Incremental Learning for Action Recognition in Videos

Jaeyoo Park Minsoo Kang Bohyung Han
ECE & ASRI, Seoul National University
{bellos1203,kminsoo,bhhan}@snu.ac.kr

Abstract

We tackle catastrophic forgetting problem in the context of class-incremental learning for video recognition, which has not been explored actively despite the popularity of continual learning. Our framework addresses this challenging task by introducing time-channel importance maps and exploiting the importance maps for learning the representations of incoming examples via knowledge distillation. We also incorporate a regularization scheme in our objective function, which encourages individual features obtained from different time steps in a video to be uncorrelated and eventually improves accuracy by alleviating catastrophic forgetting. We evaluate the proposed approach on brand-new splits of class-incremental action recognition benchmarks constructed upon the UCF101, HMDB51, and Something-Something V2 datasets, and demonstrate the effectiveness of our algorithm in comparison to the existing continual learning methods that are originally designed for image data.

1. Introduction

Human activity recognition in a large-scale video dataset is a crucial step for high-level video understanding, and various approaches have been studied actively in the computer vision community [2, 17, 21, 36, 40]. If the videos containing unseen classes of actions are presented in a sequential manner, where the examples in the previously observed classes are either inaccessible or accessible in limited amounts, one needs to adapt the current model to the new data without forgetting critical knowledge of the seen examples learned in the past. The machine learning paradigm to handle such challenges is called *class-incremental learning*, and Figure 1 illustrates a training data stream for the learning framework.

While researchers have been studying action recognition problems using deep neural networks [2, 17, 21, 36, 40], continual learning in videos has not been studied actively. It is natural to claim that video-based recognition tasks are also prone to suffer from catastrophic forgetting [23] for the knowledge learned from training data provided in the past, as in the image domain. Actually, the catastrophic forgetting problem is particularly problematic in video-learning tasks because deep neural networks with shared parameters are typically applied to multiple segments or frames, resulting in acceleration of the forgetting issue and it is difficult to store many video exemplars in memory to preserve the information about the previous tasks effectively.

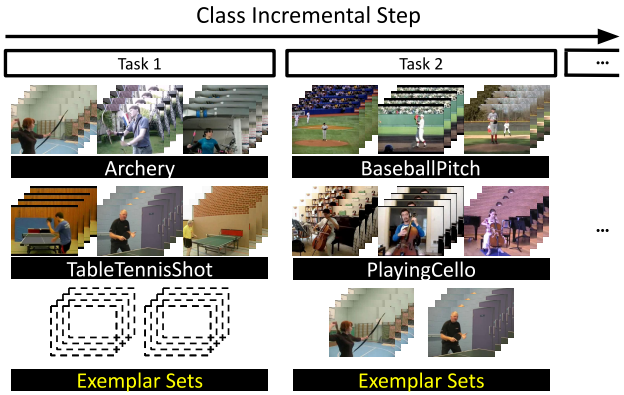
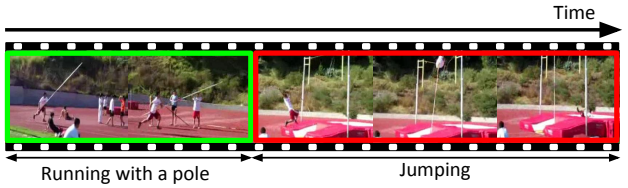
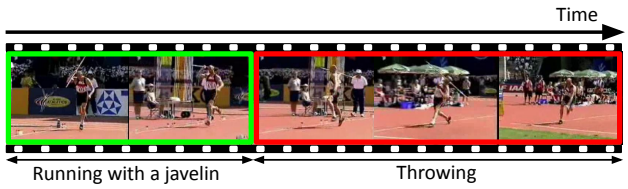


Figure 1: Illustration of class-incremental learning scenario. At each incremental step, the model learns the knowledge of new classes that are disjoint from the classes it has seen so far. Simultaneously, the model learns not to forget the knowledge of old classes that are either completely inaccessible or accessible in limited amounts.

Despite critical needs for class-incremental learning in the video domain, existing approaches [3, 6, 15, 20, 30, 43] have focused on static images only, which fails to model temporal variations and dynamics across spatial features. A single action instance is often composed of multiple subactions and the feature dynamics aligned with the subactions are indeed critical information for action recognition. For example, Figure 2 demonstrates that both of *Pole Vault* and *Javelin Throw* share a subaction of running with a long stick at the beginning but become distinct by whether the actor jumps or not at the end. This observation leads to a fundamental question about how to maintain crucial spatio-temporal information within individual videos using limited



(a) An example of Pole Vault



(b) An example of Javelin Throw

Figure 2: Subactions of action instances. Distinctive subactions are key to distinguish one action from another. Our algorithm estimates which channels are important along with the temporal dimension for class-incremental learning.

memory for continual learning.

This paper presents a novel framework for class-incremental learning for action recognition based on temporally attentive knowledge distillation. Our claim is that the representations for individual subactions should be distilled with different weights depending on their relevance and uniqueness to target classes and maintained for better utilization in the future stages. To realize this idea, we draw our attention to a joint space defined by frames in a video and channels in a feature map, and quantify importance over the space while minimizing feature redundancy across frames. Specifically, we estimate the importance in the joint space for a video by measuring how much the activation in the space affects classification losses. The computed importance provides the information about where to attend for knowledge distillation in class-incremental learning scenarios. Also, to enforce the model to learn more distinctive features across frames, we penalize the redundancy in the features extracted from the sampled frames.

The representations of video data require more computation resources for processing and storing, which makes continual learning in videos more challenging especially when some exemplars for the tasks considered earlier need to be stored in memory. So, the proposed class-incremental learning framework employs a frame-based video representation method—Temporal Shift Module (TSM) [21], and reduces computational cost for training significantly compared to 3D CNNs based on video volumes [2, 8, 9, 36, 37] and their variations [40, 47].

The contributions of this paper are summarized below:

- We introduce an efficient class-incremental learning technique for action recognition in videos by adopting a simple frame-based feature representation method to store exemplars for the tasks learned in the past.
- Our algorithm estimates time-channel importances and distills knowledge with the importance weight while encouraging the diversity of the features in each frame for regularization and enhance the performance of our target model.
- The proposed approach presents remarkable accuracy gains on the multiple standard action recognition benchmarks with brand-new splits compared to the existing methods designed in the image domain.

Our paper is organized as follows. We first discuss related works about continual learning in Section 2. Section 3 describes the proposed class-incremental learning approach in the context of action recognition. We present experimental results on the standard action recognition datasets with new splits for continual learning in Section 4, and make the conclusion in Section 5.

2. Related Works

This section reviews existing algorithms related to class-incremental learning. Most of the researches about continual learning deal with image classification problems only, so we also discuss the approaches in other tasks.

2.1. Class-Incremental Learning

Existing class-incremental learning approaches alleviate catastrophic forgetting via the following four techniques: 1) parameter regularization, 2) knowledge distillation, 3) rehearsal, and 4) bias correction.

Parameter regularization The methods in this category [1, 18, 45] estimate the importance of individual model parameters and exploit the information for model adaptation. Specifically, the learning algorithm attempts to preserve parameters with high weights while allowing unimportant ones to be flexible for update. The criteria to determine model elasticity on new tasks include Fisher information matrix [18], path integral along parameter trajectory [45], and changes in the output vectors [1]. However, these approaches empirically present poor generalization performance in class incremental learning scenarios as reported in [16, 39].

Knowledge distillation The approaches based on knowledge distillation [14, 31, 44] encourage a model to learn new tasks while mimicking the representations of the old model trained for the previous tasks without their training data. To this end, new models attempt to preserve the representations of examples by matching the outputs from the sigmoid functions [20, 30, 43], the softmax function with temperature scaling [3], and the ℓ_2-normalizations [15]. In addition, LwM [4] further minimizes the difference of the attention maps obtained from the gradients of the highest score

labels. PODNet [6] preserves the relaxed representations obtained by applying the sum pooling along the width and height dimensions to the original intermediate feature maps and controlling the balance between the previous knowledge and the new information.

Rehearsal Rehearsal-based methods store a limited number of representative examples or replay old ones using generative models while training new tasks. Incremental Classifier Representation Learning (iCaRL) [30] keeps a small number of samples per class to approximate the class centroid and makes predictions based on the nearest class mean classifiers. On the other hand, pseudo-rehearsal techniques [28, 32] generate samples in the previously observed classes using generative adversarial networks (GANs) [11, 27]. However, generating videos is too challenging to be used for class-incremental learning.

Bias correction The trained models by class-incremental learning algorithms turn out to prefer new classes partly due to the class imbalance problem, and some approaches [15, 43] aim to cope with this issue. Bias Correction (BiC) [43] corrects bias using additional scale and shift parameters for affine transformations of the logits for new classes. Zhao *et al*. [46] rescale the weight vectors for the new classes by matching the average norm of the old weight vectors.

2.2. Class-Incremental Learning in Other Domains

Although class-incremental learning has been studied for image classification, the research is also active for other applications, including person re-identification [42], 3D object classification [5], object detection [33], and semantic segmentation [24]. Continual learning in the video domain is rare [26, 41]. Despite remarkable technical advances in action recognition, catastrophic forgetting problem has not been explored actively yet. An existing approach [41] is limited to applying the iCaRL [30] based on a two-stream 3D convolutional neural network in a straightforward manner. On the other hand, our approach is based on knowledge distillation similar to [4, 6] and exploits an attention method over a time-channel space intuitively to facilitate action recognition in a class-incremental learning scenario.

2.3. Action recognition

With the great success of deep learning, various action recognition methods based on convolutional neural networks have been studied [2, 17, 21, 36, 40]. The approaches to handle this problem are grouped in two categories: 2D and 3D CNN-based methods. 2D CNN-based techniques [17, 21, 40] utilize the standard CNN models [13, 34] on each frame, and the researchers have explored how to aggregate the information from each time step [17]. For example, [10] studies how to fuse the information from two different modalities, RGB and motion, using 2D CNNs. Wang *et al*. [40] propose a strategy to learn with uniformly divided segments in multiple modalities, *i.e*. RGB difference and warped optical flow. In [47], they learn temporal dependencies across frames by exploring multiple time scales. Recently, Temporal Shift Module (TSM) [21] proposes a method to learn temporal information in an efficient way, where the feature representations of adjacent segments interact with each other during forward pass.

On the other hand, some researchers pay attention to 3D CNN [2, 8, 9, 36, 37], which is a straightforward extension of 2D CNN methods, where 3D convolution filters learn spatio-temporal information jointly. However, 3D CNN-based models are computationally expensive since they involve a large number of parameters to learn. Recent approaches handle this issue in diverse ways, for example, by applying group convolutions [37], learning 3D shift operations [7], decomposing 3D convolution filters [38], searching efficient architectures [8], *etc*. Despite remarkable advances in action recognition, the catastrophic forgetting problem is not yet studied actively. This work sheds light on this problem with a promising baseline.

3. Method

This section describes the overall framework of the proposed class-incremental learning algorithm with videos.

3.1. Problem Formulation

The goal of class-incremental learning is to train a unified deep neural network parameterized by $\boldsymbol{\Theta}$ given a sequence of tasks, $\{\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_k, \cdots\}$. We denote $\mathcal{T}_k$ as a set of videos whose labels belong to the predefined classes in $\mathcal{C}_k$, where $(\mathcal{C}_1 \cup \cdots \cup \mathcal{C}_{k-1}) \cap \mathcal{C}_k = \emptyset$. We assume that we can access a small exemplar set denoted by $\mathcal{E}_k$ such that it is a subset of $\mathcal{T}_{1:k}$, where $\mathcal{T}_{1:k} = \mathcal{T}_1 \cup \cdots \cup \mathcal{T}_k$. At each incremental step k, a model $\boldsymbol{\Theta}_k$ is trained with $\mathcal{T}'_k = \mathcal{T}_k \cup \mathcal{E}_{k-1}$. Then, the performance of the trained model is evaluated on the test examples defined by the union of all the encountered tasks without task boundaries.

3.2. Overview

Given the problem formulation, we follow the standard class-incremental learning protocol based on knowledge distillation, which is similar to the previous works [6, 15, 20, 30]. At the k^{th} incremental step, a set of model parameters, $\boldsymbol{\Theta}_k$, is learned to mimic the feature representations given by the previous model with $\boldsymbol{\Theta}_{k-1}$ while learning new classes. Our goal is to estimate desirable attention over a combination of time and channel dimensions for knowledge distillation.

Figure 3 illustrates the overall framework of our approach. Given an input video $x \in \mathcal{T}'_k$, we first divide the video into T segments, and then feed the segments to the backbone network with L layers. We adopt TSM [21] as

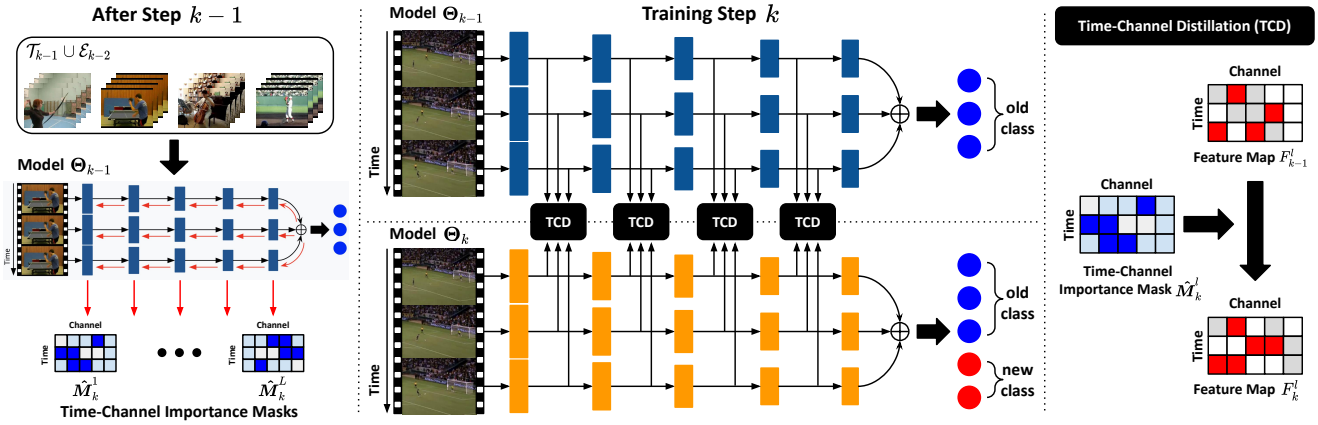


Figure 3: Illustration of the overall framework. At each incremental step k, the current model $\boldsymbol{\Theta}_k$ (bottom-center) mimics the representation of the previous model $\boldsymbol{\Theta}_{k-1}$ (top-center). The distillation process is enhanced by the time-channel importance mask, which is estimated at step $k-1$ via measuring how each feature affects the final loss. Distilling knowledge through the estimated importance map makes the model preserve important representations from the previous step; the less important representations are suppressed by the mask and updated flexibly for the new task.

our backbone model. Let $\boldsymbol{F}^l_k \in \mathbb{R}^{T\times C_l\times H_l\times W_l}$ be an intermediate feature with respect to the input x in the l^{th} layer of the model $\boldsymbol{\Theta}_k$. The distillation between $\boldsymbol{F}^l_k$ and $\boldsymbol{F}^l_{k-1}$ is weighted by the importance mask $\hat{\boldsymbol{M}}^l_k \in \mathbb{R}^{T\times C_l}$, which is the key component of our framework. The importance mask $\hat{\boldsymbol{M}}^l_k$ represents the information about which feature maps along time or channel dimensions are important to preserve knowledge for the past tasks.

After each incremental step, we select a set of video instances in $\mathcal{T}_k$ to update the exemplar memories from $\mathcal{E}_{k-1}$ to $\mathcal{E}_k$ by the herding strategy [30]. Then, we fine-tune the final classification layer while freezing other layers using $\mathcal{E}_k$, which has balanced data among the observed classes as discussed in [6, 15].

3.3. Time-Channel Importance

We focus on designing the importance mask so that it provides each feature map of a frame with the information about which channels should be preserved against the catastrophic forgetting problem. Specifically, we aim to keep the important feature maps whose update is prone to increase the final loss, and make the unimportant ones flexible for future tasks. To this end, we compute the importance of channel c at time step t for incremental step k in the l^{th} layer, which is denoted by $\boldsymbol{M}^l_{k,t,c}$, as

$$\boldsymbol{M}^l_{k,t,c} = \mathbb{E}_{(x,y)\sim\mathcal{T}_{1:k-1}} \|\nabla_{\boldsymbol{F}^l_{k-1,t,c}} \mathcal{L}^{k-1}_{\text{cls}}(x,y)\|^2_F, \quad (1)$$

where $\|\cdot\|_F$ and $\mathcal{L}^{k-1}_{\text{cls}}(x,y)$ denote the Frobenius norm and the classification error of the trained model parametrized by $\boldsymbol{\Theta_{k-1}}$ for the input video x and its label y. Since the perturbation in the feature map with the higher Frobenius norm of the gradient may result in larger increase of the final loss when feature maps are equally important, the importance mask $\boldsymbol{M}^l_{k,t,c}$ can be regarded as the sensitivity to the final loss. Thus, at the end of each incremental step, *i.e.*, $\mathcal{T}_{k-1}$, we sequentially update the important mask $\boldsymbol{M}^l_k$ for training the new model $\boldsymbol{\Theta}_k$.

However, due to the restriction of the class-incremental learning, we can access limited samples for $\mathcal{T}_{1:k-2}$ using the exemplar sets $\mathcal{E}_{k-2}$, which makes it difficult to compute $\boldsymbol{M}^l_{k,t,c}$. Hence, we approximate $\boldsymbol{M}^l_{k,t,c}$ to $\tilde{\boldsymbol{M}}^l_{k,t,c}$ by taking the expectation of the Frobenius norm over the accessible samples within $\mathcal{E}_{k-2}$ and $\mathcal{T}_{k-1}$. Furthermore, we normalize $\tilde{\boldsymbol{M}}^l_{k,t,c}$ to make the importance across layers have a similar scale as follows:

$$\hat{\boldsymbol{M}}^l_{k,t,c} = \frac{\tilde{\boldsymbol{M}}^l_{k,t,c}}{\frac{1}{TC_l}\sum_{t=1}^{T}\sum_{c=1}^{C_l}\tilde{\boldsymbol{M}}^l_{k,t,c}}. \quad (2)$$

Finally, we define the proposed distillation loss for the intermediate features in the new model based on the importance map as

$$\mathcal{L}^k_{\text{dist}} = \sum_{l=1}^{L}\sum_{t=1}^{T}\sum_{c=1}^{C_l}\hat{\boldsymbol{M}}^l_{k,t,c}\|\boldsymbol{F}^l_{k,t,c} - \boldsymbol{F}^l_{k-1,t,c}\|^2_F. \quad (3)$$

The proposed distillation loss constrains the model divergence of the sensitive feature maps not to forget the previously learned knowledge and makes the uncritical ones flexible to learn new classes.

3.4. Orthogonality between Frames

To further improve the effectiveness of the proposed knowledge distillation strategy, we adopt an additional regularization term inspired by [22], which enforces individual

features extracted from different time steps in a video to be mutually independent. The corresponding loss constrains the features at individual time steps to be orthogonal, which also makes the estimation of the importance map more distinctive. The orthogonality loss is defined by

$$\mathcal{L}^k_{\text{ortho}} = \sum_{l=1}^{L}\sum_{c=1}^{C_l} \|\boldsymbol{I}_T - \boldsymbol{F'}^l_{k,:,c}(\boldsymbol{F'}^l_{k,:,c})^\top\|_F^2, \quad (4)$$

where $\boldsymbol{I}_T \in \mathbb{R}^{T\times T}$ is an identity matrix and $\boldsymbol{F'}^l_{k,:,c}$ is given by concatenating a reshaped tensor of ℓ_2-normalized $\boldsymbol{F}^l_{k,t,c}$ along time axis t and constructing a $T \times H_l W_l$ matrix.

The orthogonality constraint is useful in continual learning scenarios since we often need to update model parameters based on limited observations of old data but a large number of examples in new tasks. Since such a challenging situation leads to unwanted representation changes of the exemplars representing previous tasks, the minimization of correlation between the representations of individual frames would help alleviate the feature drift issue.

3.5. Training Objective

The formal definition of the final objective function $\mathcal{L}^k_{\text{final}}$ at incremental step k is given by

$$\mathcal{L}^k_{\text{final}} = \mathcal{L}^k_{\text{cls}} + \alpha\mathcal{L}^k_{\text{dist}} + \beta\mathcal{L}^k_{\text{ortho}}, \quad (5)$$

where α and β are the weights for the balance between the terms. For the classification loss $\mathcal{L}^k_{\text{cls}}$, we adopt NCA loss [25] computed from the *Local Similarity Classifier* (LSC) following [6].

3.6. Exemplar Selection

After each incremental step k, we sample the most representative instances from $\mathcal{T}_k$ to construct $\mathcal{E}_k$, for the future use. We follow the herding strategy proposed by [30], for which the feature representations for all video samples are extracted from $\mathcal{T}_k$ and the class-wise mean features are computed. Then we iteratively select the instances for each of the classes until the number of selected exemplars reaches a predefined memory budget. At each iteration, we choose the exemplar that makes the mean of exemplars become closest to the real class-mean representation.

When we store videos as exemplars, we can further reduce the memory requirement by sampling frames within the video since a single video contains many repetitive and redundant frames. For each video, we have three options: storing an entire video, sampling frames randomly, or selecting frames with a uniform time interval. Among the three strategies, we choose the last one, storing the uniformly sampled T frames per video, which meets the input specification of our backbone model, TSM [21]. We further discuss this sampling strategy in Section 4.5.

4. Experiments

This section presents the experimental results of our algorithm on class-incremental action recognition benchmarks. We also demonstrate the effectiveness of our framework via several ablation studies.

4.1. Datasets

We evaluate the proposed framework on UCF101 [35], HMDB51 [19] and Something-Something V2 [12], which are the standard datasets for action recognition tasks. The UCF101 dataset consists of 13.3K videos from 101 classes. The organizers of UCF101 provide three splits of training and test datasets. The HMDB51 dataset consists of 6.8K examples from 51 action classes, and also provides three splits for training and test datasets. We adopt split 1 for both of the datasets to evaluate our approach. Something-Something V2 dataset is a large-scale motion-sensitive dataset, which contains 169K training and 25K test videos from 174 action classes. This dataset requires better temporal reasoning than UCF101 and HMDB51.

4.2. Evaluation Protocol

Since the aforementioned datasets are utilized for the class-incremental learning for the first time, we newly design the experimental protocol for the datasets. We first shuffle the classes randomly to create a sequence of classes. Following [6, 15], we assume that we initially have a trained model with half of the total classes, where the rest of the classes are provided sequentially in each incremental step. For UCF101, we trained 51 classes in the initial stage, and divided the remaining classes into groups of 10, 5, and 2 classes for class-incremental learning. For HMDB51, we learned the initial model using 26 classes and the remaining classes are equally split into 5 and 25 groups. After obtaining the initial model with 84 classes for Something-Something V2, we generate groups of 10 and 5 classes.

At each incremental step, we evaluate the model with the test data of all the seen classes until then. Following the previous works, we employ two methods for inference, CNN and NME, respectively. CNN is a standard classification protocol, where the model classifies the data using the trained fully-connected layer. NME, which is proposed by iCaRL [30], compares the feature representation of test data with the mean-of-exemplars. We report the average of the accuracies aggregated from all of the incremental steps, which is also known as *average incremental accuracy* [6, 15, 30]. Since the order of classes may affect the performance, we ran our experiments using three random class orders[1] and report the average performance. We set the memory budget for each class to 5 for UCF101

[1] Random Seeds : 1000, 1993, 2021

Table 1: Class-incremental action recognition performance on UCF101 and HMDB51 of the tested algorithms. The proposed method, TCD, achieves the best performance in all the experimental settings. NME scores for the methods without exemplars cannot be reported while iCaRL reports NME scores only since iCaRL employs NME for classification. The bold-faced numbers indicate the best performance.

	UCF101						HMDB51			
Num. of classes	10 × 5 stages		5 × 10 stages		2 × 25 stages		5 × 5 stages		1 × 25 stages	
Classifier	CNN	NME	CNN	NME	CNN	NME	CNN	NME	CNN	NME
Fine-tuning	24.97	—	13.45	—	5.78	—	16.82	—	4.83	—
LwFMC [20, 30]	42.14	—	25.59	—	11.68	—	26.82	—	16.49	—
LwM [4]	43.39	—	26.07	—	12.08	—	26.97	—	16.50	—
iCaRL [30]	—	65.34	—	64.51	—	58.73	—	40.09	—	33.77
UCIR [15]	74.31	74.09	70.42	70.50	63.22	64.00	44.90	46.53	37.04	37.15
PODNet [6]	73.26	74.37	71.58	73.75	70.28	71.87	44.32	48.78	38.76	46.62
TCD (Ours)	**74.89**	**77.16**	**73.43**	**75.35**	**72.19**	**74.01**	**45.34**	**50.36**	**40.07**	**46.66**
Oracle (Upper Bound)	84.15	83.37	83.96	83.20	83.82	83.16	55.03	55.98	54.89	55.32

Table 2: Class-incremental action recognition performance on Something-Something V2. The bold-faced numbers indicate the best performance.

Num. of classes	10 × 9 stages		5 × 18 stages	
Classifier	CNN	NME	CNN	NME
UCIR [15]	26.84	17.98	20.69	12.57
PODNet [6]	34.94	27.33	26.95	17.49
TCD (Ours)	**35.78**	**28.88**	**29.60**	**21.63**

and HMDB51, and 20 for Something-Something V2 unless specified otherwise.

4.3. Implementation Details

We construct our framework based on the official implementation of TSM[2] using the PyTorch library [29]. We follow data pre-processing protocol of TSM. For UCF101, we train a ResNet-34 TSM for 50 epochs with a batch size of 32. For HMDB51 and Something-Something V2, we adopt ResNet-50 models and train for 50 epochs with a batch size of 64. For all datasets, we use the ImageNet-pretrained weights for initialization. Note that we do not use the weights pretrained with the Kinetics dataset [2], which is common in the action recognition field. It is inappropriate to evaluate class-incremental learning with the weights pretrained with Kinetics since it shares the class information with UCF101 and HMDB51. Thus the pretrained weights already contain the class information of target datasets. Please refer to the supplementary materials for more implementation details.

4.4. Main Results

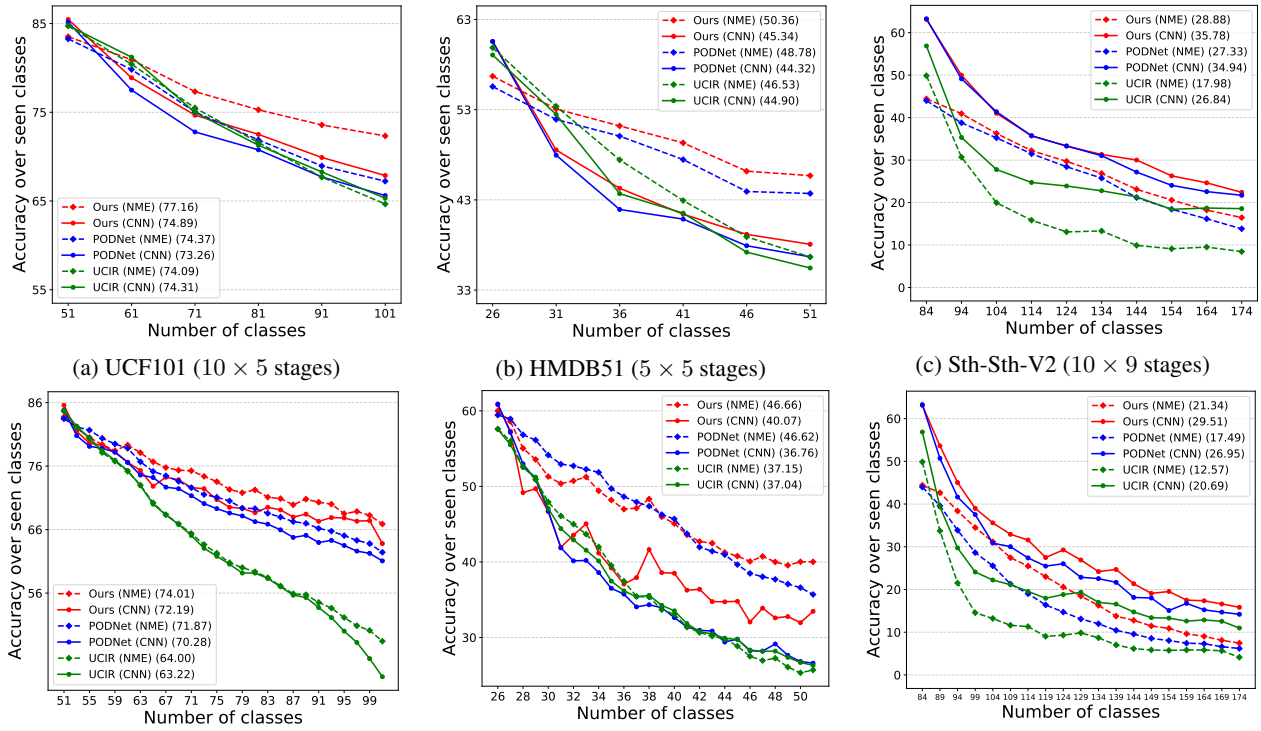
We compare the proposed method, referred to as Time-Channel Distillation (TCD), with existing class-incremental learning baselines, which are originally designed for the class-incremental image classification task. Especially, we choose the algorithms utilizing knowledge distillation as ours, including LwFMC [20, 30], LwM [4], iCaRL [30], UCIR [15], PODNet [6]. We do not report the NME results from LwFMC [20, 30] and LwM [4] since they do not use exemplars. To provide an upper-bound performance of our task, we introduce an oracle model, which is incrementally trained the model while preserving all the training data in the previous steps. We reimplement the baseline algorithms and train their models using our datasets for fair comparisons. The implementation details for the baselines are presented in the supplementary material.

Table 1 presents the overall results of the proposed algorithm and other baselines on UCF101 and HMDB51 datasets, where our approach outperforms all competing methods in all the experimental settings. As mentioned earlier, NME scores for the methods without exemplars are not reported while iCaRL has the results only because the method is originally designed for the classifier. As can be easily expected, the approaches do not exploit exemplars demonstrate poor performance.

Table 2 presents the overall results of the proposed algorithm and recent methods [6, 15] on Something-Something V2. TCD is also more effective than other methods on the large-scale motion-sensitive dataset. It is noticeable that, for Something-Something V2, the performance for NME falls behind CNN. Since Something-Something V2 needs more temporal reasoning, the strategies relying on naïve averaging of the features from all frames may not be suitable.

Figure 4 presents the average accuracy over seen classes at each incremental step. In most of the incremental steps, TCD achieves higher accuracy, which implies its great capacity to preserve the learned knowledge in the past. Note that even though the average incremental accuracy gain on

[2]https://github.com/mit-han-lab/temporal-shift-module



(a) UCF101 (10×5 stages) (b) HMDB51 (5×5 stages) (c) Sth-Sth-V2 (10×9 stages)

(d) UCF101 (2×25 stages) (e) HMDB51 (1×25 stages) (f) Sth-Sth-V2 (5×18 stages)

Figure 4: Plots for accuracy on UCF101, HMDB51 and Something-Something V2 along with the incremental steps.

HMDB51 with 25 stages is small, the accuracy at the last incremental step is better than that of PODNet.

4.5. Ablation Study and Analysis

We perform several ablation studies on UCF101 with 10 steps to analyze the effectiveness of our approach.

Effect of each component To show the effectiveness of the time-channel importance map and the frame-wise orthogonality, we conduct the experiment for variant types of our objective function, $\mathcal{L}^k_{\text{final}}$. To this end, we first define the distillation loss without importance maps, which is given by

$$\mathcal{L}'^k_{\text{dist}} = \sum_{l=1}^{L} \sum_{t=1}^{T} \sum_{c=1}^{C_l} \| \boldsymbol{F}^l_{k,t,c} - \boldsymbol{F}^l_{k-1,t,c} \|^2_F. \quad (6)$$

Table 3 presents the results from several different combinations of loss terms. The results show that all of the introduced components contribute to the performance and their combination leads to the best performance. One noticeable thing is that applying $\mathcal{L}^k_{\text{ortho}}$ without $\hat{\boldsymbol{M}}^l_{k,t,c}$ also improves the performance, where the loss alleviates the correlation between the representations across frames and help the model to address feature drift issue.

Effect of memory size To demonstrate the robustness of TCD with respect to the memory budget, we evaluate the performance of the compared methods by varying the memory budgets. Table 4 shows that TCD outperforms other baselines regardless of the memory budget.

Table 3: Ablations study results about the objective function. We demonstrate the effectiveness of the time-channel channel importance $\hat{\boldsymbol{M}}^l_{k,t,c}$ and the orthogonality among frames $\mathcal{L}^k_{\text{ortho}}$. Note that $\mathcal{L}'^k_{\text{dist}}$ denotes $\mathcal{L}^k_{\text{dist}}$ without importance map weights, $\hat{\boldsymbol{M}}^l_{k,t,c}$.

Objective function	CNN	NME
$\mathcal{L}^k_{\text{cls}} + \mathcal{L}'^k_{\text{dist}}$	71.21	73.24
$\mathcal{L}^k_{\text{cls}} + \mathcal{L}'^k_{\text{dist}} + \mathcal{L}^k_{\text{ortho}}$	72.31	74.42
$\mathcal{L}^k_{\text{cls}} + \mathcal{L}^k_{\text{dist}}$	72.61	74.81
$\mathcal{L}^k_{\text{cls}} + \mathcal{L}^k_{\text{dist}} + \mathcal{L}^k_{\text{ortho}}$ (Ours)	**73.43**	**75.35**

Sampling strategy As discussed in Section 3.6, the memory requirement for video exemplars is further reduced by storing a subset of frames in a video instead of the whole video. We conduct the experiment to show the performance variation of class-incremental learning depending on the exemplar selection strategy. We test the following three options: storing a whole video, sampling frames randomly, and selecting frames with a uniform time interval. In the setting that the whole video is stored, TSM selects a pre-

Table 4: Analysis about the memory budget for each class on UCF101 with 10 steps. The results show the robustness of our algorithm to varying memory budgets.

Memory per class	1		2		**5**		10	
	CNN	NME	CNN	NME	CNN	NME	CNN	NME
iCaRL [30]	—	58.05	—	60.50	—	64.51	—	66.94
UCIR [15]	61.92	65.52	66.43	67.58	70.42	70.50	72.47	71.69
PODNet [6]	63.18	70.96	65.93	72.78	71.58	73.75	75.44	76.39
TCD (Ours)	**64.52**	**71.96**	**68.40**	**73.30**	**73.43**	**75.35**	**76.66**	**77.09**

Table 5: Analysis about the sampling strategies for storing videos in exemplar set."All" denotes the strategy to store the entire video in the exemplar memory and sample examples randomly for training. "Random" and "Uniform" mean the strategies that sample frames randomly and with a equal time interval, respectively. The results show that storing all frames in a video does not necessarily delivers performance improvement.

Sampling strategy	All		Random		Uniform	
	CNN	NME	CNN	NME	CNN	NME
iCaRL [30]	—	64.33	—	64.68	—	64.51
UCIR [15]	70.22	70.41	70.38	70.12	70.42	70.50
PODNet [6]	71.76	73.50	72.37	73.87	71.58	73.75
TCD (Ours)	**73.89**	**75.51**	**73.17**	**75.30**	**73.43**	**75.35**

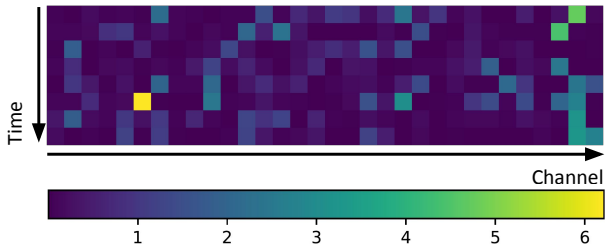


Figure 5: Visualization of the importance map obtained from the 4th ResBlock in the model trained on UCF101 with 10 stages. The colorbar indicates the magnitude of the estimated importance.

defined number of frames randomly from each exemplar at each iteration. For the random and uniform sampling strategies, we store T frames, where T is given by the hyperparameter of TSM network [21].

Table 5 demonstrates that the simple sampling strategies are as good as the methods with the whole videos in all the tested algorithms. This result implies that the diversity of sampled frames affects the overall performance marginally. In the context of class-incremental learning, a small subset of frames in exemplar videos are sufficient to maintain the knowledge about the corresponding video, which is a desirable property to continual learning. However, this experiment is limited in another aspect because our backbone model, TSM, relies only on a small number of frames.

Visualization of importance map Figure 5 illustrates an example of generated importance map after the last stage training of UCF101. The importance map for the first 32 channels of the 4th ResBlock for TSM is depicted, where the bright pixels indicate higher importance. From the figure, one can notice that the importance of each channel varies over time. The estimated mask makes the model leave critical features unaffected by knowledge distillation while providing the model with the flexibility to update unimportant features.

5. Conclusion

We presented a novel framework for class-incremental learning in the context of video action recognition, which has not been actively investigated yet. Specifically, we introduced a new knowledge distillation loss based on time-channel importance masks, which aims to preserve crucial feature maps for preventing the catastrophic forgetting problem and make trivial ones flexible for absorbing new knowledge. To effectively exploit the proposed distillation loss, we add a regularization term, which encourages individual feature maps along the time axis to be orthogonal to each other. Our algorithm achieves outstanding performance compared to existing image-specific class-incremental learning approaches on multiple standard datasets, which are newly introduced to fit class-incremental learning for videos.

Acknowledgments

This work was partly supported by Samsung Electronics Co., Ltd., the IITP grants, and the Bio & Medical Technology Development Program of the National Research Foundation (NRF) funded by the Korea government (MSIT) [2017-0-01779, 2017-0-01780, 2021M3A9E4080782].

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, 2018.

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the Kinetics dataset. In *CVPR*, 2017.

[3] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *ECCV*, 2018.

[4] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *CVPR*, 2019.

[5] Jiahua Dong, Yang Cong, Gan Sun, Bingtao Ma, and Lichen Wang. I3DOL: Incremental 3d object learning without catastrophic forgetting. In *AAAI*, 2021.

[6] Arthur Douillard, Matthieu Cord, Charles Ollion, and Thomas Robert. PODNet: Pooled outputs distillation for small-tasks incremental learning. In *ECCV*, 2020.

[7] Linxi Fan, Shyamal Buch, Guanzhi Wang, Ryan Cao, Yuke Zhu, Juan Carlos Niebles, and Li Fei-Fei. RubiksNet: learnable 3d-shift for efficient video action recognition. In *ECCV*, 2020.

[8] Christoph Feichtenhofer. X3D: Expanding architectures for efficient video recognition. In *CVPR*, 2020.

[9] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019.

[10] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016.

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative Adversarial Nets. In *NIPS*, 2014.

[12] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The "Something Something" video database for learning and evaluating visual common sense. In *ICCV*, 2017.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[14] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Workshop*, 2014.

[15] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *CVPR*, 2019.

[16] Yen-Chang Hsu, Yen-Cheng Liu, Anita Ramasamy, and Zsolt Kira. Re-evaluating continual learning scenarios: A categorization and case for strong baselines. *arXiv preprint arXiv:1810.12488*, 2018.

[17] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[18] 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. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

[19] Hilde Kuehne, Hueihan, Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: A large video database for human motion recognition. In *ICCV*, 2011.

[20] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE T-PAMI*, 40(12):2935–2947, 2017.

[21] Ji Lin, Chuang Gan, and Song Han. TSM: Temporal shift module for efficient video understanding. In *ICCV*, 2019.

[22] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. In *ICLR*, 2017.

[23] 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. Elsevier, 1989.

[24] Umberto Michieli and Pietro Zanuttigh. Incremental learning techniques for semantic segmentation. In *ICCV Workshops*, 2019.

[25] Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *ICCV*, 2017.

[26] Zihao Mu, Francisco M. Castro, Manuel J. Marín-Jiménez, Nicolás Guil, Yan-ran Li, and Shiqi Yu. iLGaCo: incremental learning of gait covariate factors. In *IEEE International Joint Conference on Biometrics (IJCB)*, 2020.

[27] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier GANs. In *ICML*, 2017.

[28] Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, Patrick Jahnichen, and Moin Nabi. Learning to remember: A synaptic plasticity driven framework for continual learning. In *CVPR*, 2019.

[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

[30] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In *CVPR*, 2017.

[31] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for thin deep nets. In *ICLR*, 2015.

[32] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *NIPS*, 2017.

[33] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *ICCV*, 2017.

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[35] K. Soomro, A. Roshan Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. In *CRCV-TR-12-01*, 2012.

[36] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[37] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *ICCV*, 2019.

[38] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018.

[39] Gido M. van de Ven and Andreas S. Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

[40] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

[41] Zhengwei Wang, Qi She, Tejo Chalasani, and Aljosa Smolic. CatNet: Class incremental 3d convnets for lifelong egocentric gesture recognition. In *CVPR Workshops*, 2020.

[42] Guile Wu and Shaogang Gong. Generalising without forgetting for lifelong person re-identification. In *AAAI*, 2021.

[43] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *CVPR*, 2019.

[44] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017.

[45] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, 2017.

[46] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *CVPR*, 2020.

[47] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, 2018.