Background debiased class incremental learning for video action recognition

Le Quan Nguyen, Jinwoo Choi, L. Minh Dang, Hyeonjoon Moon

PII: S0262-8856(24)00400-1

DOI: https://doi.org/10.1016/j.imavis.2024.105295

Reference: IMAVIS 105295

To appear in: *Image and Vision Computing*

Received date: 18 October 2023

Revised date: 21 March 2024

Accepted date: 3 October 2024

Please cite this article as: L.Q. Nguyen, J. Choi, L.M. Dang, et al., Background debiased class incremental learning for video action recognition, *Image and Vision Computing* (2024), https://doi.org/10.1016/j.imavis.2024.105295

Background Debiased Class Incremental Learning for Video Action Recognition

Le Quan Nguyen[a], Jinwoo Choi[c,*], L.Minh Dang[b], Hyeonjoon Moon[a,*]

[a] *Department of Computer Science and Engineering, Sejong University, Seoul 05006, Sejong University, Seoul, Republic of Korea*

[b] *Department of Information and Communication Engineering and Convergence Engineering for Intelligent Drone, Seoul 05006, Sejong University, Seoul, Republic of Korea*

[c] *School of Computing, Kyung Hee University, Yongin, Republic of Korea*

Abstract

In this work, we tackle class incremental learning (CIL) for video action recognition, a relatively under-explored problem despite its practical importance. Directly applying image-based CIL methods does not work well in the video action recognition setting. We hypothesize the major reason is the spurious correlation between the action and background in video action recognition datasets/models. Recent literature shows that the spurious correlation hampers the generalization of models in the conventional action recognition setting. The problem is even more severe in the CIL setting due to the limited exemplars available in the rehearsal memory. We empirically show that mitigating the spurious correlation between the action and background is crucial to the CIL for video action recognition. We propose to learn background invariant action representations in the CIL setting by providing training videos with diverse backgrounds generated from background

* Co-corresponding Authors

augmentation techniques. We validate the proposed method on public benchmarks: HMDB-51, UCF-101, and Something-Something-v2.

Keywords:

action recognition, class incremental learning, debiasing, temporal shift module

---

1. Introduction

Recent advances in video action recognition show remarkable performance due to the rapid progress of deep neural network architectures [1, 2, 3, 4] and a large amount of training data [5, 6, 7, 8]. Most video action recognition approaches share an unrealistic assumption: all the labeled training data is available during a single stage. This assumption is unrealistic as it is almost impossible to define and collect videos containing all human action categories at once. In the real world, we inevitably encounter new action categories as time goes on. Storing all the growing data and retraining the model is impractical due to growing cost, privacy, and legal constraints. Therefore, we need to incrementally update the action recognition model using the videos of novel categories without access to the previous data. We formulate the aforementioned action recognition problem as a class incremental learning problem.

Figure 1: **Class incremental learning for video action recognition**. A model learns new classes in a sequential manner rather than learning all at once. To prevent catastrophic forgetting, we keep limited examples from the previous tasks in the rehearsal buffer and we reuse them in each incremental learning step.

In the class incremental learning (CIL) setting, data arrive sequentially and we do not have access to previous data or we have access to only a limited amount of previous data as shown in Figure 1. This setting is challenging due to the catastrophic forgetting problem [9, 10] in which model performance on the old tasks degrades. To overcome the catastrophic forgetting problem, there have been extensive efforts which show great progress in the image domain [11, 12, 13, 14, 15, 16, 17]. However, CIL for video action recognition is relatively under-explored. Although CIL methods working well in the image domain [11, 12, 13, 14, 15, 16], extending these methods effectively to the video domain poses non-trivial challenges. Naive extensions do not show satisfactory performance. There are only a few recent works on CIL designed for video action recognition [18, 19, 20, 21, 22]. These approaches leverage the temporal dynamics property inherent in videos. These methods either focus on spatial-temporal distillation [18, 19], key frames selection for replay buffer set [19], temporal consistency regularization [21], and learn to condensed frames with prompt [22]. Despite the great advances, they overlook the representation bias problem which is prevalent in many video action recognition datasets [23].

Representation bias is a property of a dataset, and a model trained with a biased dataset inherits this bias when making predictions. Many video action recognition datasets such as HMDB-51 [24], UCF-101 [8], Kinetics-400 [5], and ActivityNet [7] exhibit static bias [23]. This static bias is caused by the spurious correlation between static cues and the actual action. For example, most videos of the ”Make Up” class contain up-close faces, or videos of the ”Soccer Penalty” class often include scenes of grass fields. Training a video action recognition model with these static bias datasets often leads to the model prioritizing scene features over the actions being performed when making predictions. In this work, we refer to these static cues as backgrounds, and the bias toward the static cues is called background bias.

Additionally, video is more complex than image due to the additional temporal dimension. Hence training video classification models requires more data to generalize well compared to training image classification models. Unfortunately, collecting and labeling videos is more expensive than image collecting and labeling. Consequently, video datasets have even a smaller number of training examples compared to the image datasets despite their complexity: e.g., The Kinetics-700 dataset [5] has $650\,\mathrm{K}$ training examples while ImageNet [25] has $1\mathrm{M}$ training examples. Hence, even large-scale video datasets e.g., Kinetics, and ActivityNet [7] have spurious correlations between action and scene/object [23]. The class incremental learning setting makes the bias problem even more severe due to the scarcity of the previous task examples. Moreover, the inter-task confusion challenge of class incremental learning [26] promotes the bias of models. Since models are not aware of future classes, they may pick up some representation biases. While these biases are beneficial for discriminating old and current classes, they might be detrimental to distinguishing future classes. For example, in the UCF-101 dataset, videos of ”Apply Eye Makeup,” ”Apply Lipstick,” and ”Brushing Teeth” classes all contain close-up faces. In a CIL setting, these classes may not come simultaneously. If the ’Apply Lipstick’ class arrives first and the model picks up the facial features bias, this inductive bias will not be helpful in classifying later arrival videos of the ’Apply Eye Makeup’ and ’Brushing Teeth’ classes. Therefore, we hypothesize that mitigating the representation biases when training models on every incremental learning step is crucial to addressing the catastrophic forgetting problem of class incremental learning for video action recognition.

Figure 2: **Motivation of diversifying background in CIL for video action recognition.** Due to the data scarcity in CIL, a model is prone to background bias which leads to catastrophic

forgetting. With the proposed method, we diversify the videos in the rehearsal buffer. Hence, we can mitigate the forgetting problem.

In this paper, we focus on mitigating the background bias in every incremental learning step. Specifically, we employ a background augmentation to diversify the videos in the rehearsal buffer. We blend an original video from the rehearsal buffer with a background frame extracted from another video. Figure 2 shows how videos with various backgrounds diversify the training data to regularize models from learning to exploit the spurious correlation between backgrounds and actions as a shortcut. Additionally, we employ photometric/geometric augmentations to further diversify the videos from the rehearsal buffer. To verify the hypothesis and effectiveness of our approach, we conduct extensive experiments on public benchmarks: HMDB-51 [24], UCF-101 [8] and Something-Something-v2 [6]. Our debiased class incremental learning method shows consistent performance improvement over state-of-the-arts without debiasing. We make the following major contributions in this paper.

- We identify a background bias problem in class incremental learning for video action recognition (video CIL). We further analyze the background bias problem in the Video CIL setting using scene distance experiment (see Section 4.6), and Grad-CAM visualization (see Section 4.7) to confirm our hypothesis about background bias problem in video CIL

- We propose a simple, yet effective plug-and-play method for class incremental learning for video action recognition by augmenting backgrounds for every incremental learning step. The proposed background augmentation mitigates background biases and catastrophic forgetting.

- By addressing the background bias, our method achieves significant performance improvements compared to CIL baselines that do not account for it in various public benchmarks, and our proposed method achieves state-of-the-art performance.

Figure 3: **Overview of the proposed approach for class incremental learning for video action recognition.** We learn the model parameters θ_t at each incremental training step t with the training data $D_t^{'}$. $D_t^{'}$ consists of current step training data D_t and examplars from the previous tasks $E_{0:t-1}$. We randomly sample videos from $D_t^{'}$ and extract background frames from other videos in $D_t^{'}$. To extract backgrounds, we employ the temporal median filter. The videos and backgrounds are input to photometric/geometric augmentation and background augmentation. Then we feed the augmented videos into the model. Diversifying data is crucial to every incremental learning step due to limited data. We employ a knowledge distillation loss L_{KD} to mitigate forgetting and a L_{NCA} loss to learn categories.

2. Related Work

Video Action Recognition. In the past decade, there has been great progress in video action recognition thanks to deep neural networks and a massive amount of training data [24, 8, 6, 7]. Recent literature could be grouped into improving network design such as two-stream networks [27, 28], 3D CNNs [29, 1, 30, 31], 2D plus 1D CNNs [32, 33], channel shifting along temporal axis [3], and Transformers [34, 35, 36, 4, 37], improving data efficiency [38, 39], long-term temporal context modeling [31, 2, 40, 41], and multi-modal fusion of video and audio [42]. These works assume all the training data is readily available at once which is often unrealistic as we cannot

define and collect all the action categories in advance. In contrast, our focus is a more realistic setting, i.e., class incremental learning for video action recognition. **Class incremental learning.** Class incremental learning approaches can be categorized according to how to address catastrophic forgetting. To prevent catastrophic forgetting, replay memory based methods [14, 43, 44, 18] store limited number of exemplars from the previous tasks and then use them as training data during the current task training step. In knowledge distillation based methods [17, 14, 13, 45, 46, 11, 12], teacher models transfer the knowledge learned from the previous tasks to students to alleviate catastrophic forgetting. Another line of works focuses on regularizing updates of individual model parameters that are important for the previous tasks [16, 15, 47, 48, 49]. Since class incremental learning methods are often prone to class imbalance problems, recent literature proposes to correct the class bias for the improved performance [12, 13, 50]. Although these class incremental learning methods show some promising results, most of the them focus on the image domain. There are only a few works on class incremental learning for video action recognition [18, 19, 21, 22]. Time-Channel Distillation (TCD) [18] preserves the knowledge learned from previous tasks by applying knowledge distillation loss on the spatial-temporal features. TCD regularizes the knowledge transfer according to the time-channel importance mask. In this work, we complement the recent advances in class incremental learning for video action recognition by plug-and-play background debiasing.

Mitigating Biases. Many action recognition datasets e.g., UCF-101 [8], Kinetics [5], ActivityNet [7], are static biased, which means there exists a spurious correlation between actions and scenes in a dataset. Therefore, it is possible to achieve good performance on these datasets by just exploiting the spurious correlation between action and background/object types [23]. However, such a background-biased representations do not generalize across domains and tasks [51]. Recent

literature addresses the background bias problem by adversarial learning [51, 52, 53] in the context of video action recognition. Another effective approach to mitigate background bias is to incorporate data augmentation methods [39, 54], which encourage the model to become invariant to the background during training. In the class incremental learning setting, background-bias is even more severe as shown in Figure 1 as we have only limited examplars from the previous tasks. In this work, we address the background bias problem in class incremental learning by introducing background augmentations.

3. Method

3.1. Overview

Figure 3 illustrates the overview of the proposed method. We adopt a typical knowledge distillation-based class incremental learning framework [11, 18]. Specifically, we choose PODNet-Pixel baseline to test our method. There are two stages of training. The first stage is the pre-training stage where we train the model parameters θ with base classes (task 0) only. The second stage is the incremental training stage. In this stage, we learn the model parameters θ_t at each incremental training step t with the training data $D_t^{'}$ consists of K videos in total. The training data $D_t^{'}$ consists of current step training data D_t and exemplars from the previous steps $E_{0:t-1}$.

3.2. Temporal Shift Module

The Temporal Shift Module (TSM) [3] is a lightweight spatio-temporal model designed for video action recognition. TSM captures temporal information in videos by shifting feature maps

extracted from a CNN backbone along the temporal dimension. This shifting can occur either in a bi-directional manner for offline applications or a uni-directional manner for online applications (see Figure 4). Following the TCD benchmark [18], we select ResNet34 TSM and ResNet50 TSM video action recognition models, which are based on the ResNet34 and ResNet50 backbones [55].

Figure 4: **Temporal Shift Module (TSM)**. The temporal information is Incorporated into features by shifting channels in the original features along the temporal axis

Figure 5: **PODNet-Pixels baseline**. The knowledge distillation loss is applied to several CNN blocks and on embedding.

3.3. Knowledge distillation-based CIL

In Knowledge distillation-based Class Incremental Learning (CIL), the teacher model corresponds to the model trained on the preceding task θ_{t-1}, while the student model represents the current task θ_t. The knowledge from the previous task is retained by applying a Knowledge Distillation loss (KD loss), which ensures that the representations outputted by the teacher model are effectively transferred to the student model. In PODNet, the KD loss is applied to the feature maps outputted from several CNN blocks and the final embedding. One important factor when applying knowledge distillation in CIL is to balance the rigidity-plasticity trade-off. In PODNet, this trade-off is controlled by applying various pooling strategies on feature maps before applying the KD loss. The PODNet-Pixels is a variant of PODNet where no pooling is applied to the feature maps. Let $h^t_{l,c,w,h}$ be the feature maps of size (h, w, c) extracted from layer l, and h^t be the final embedding of the model θ_t. The KD loss for feature maps is:

$$L_{POD-Pixels} = \sum_{c=1}^{C}\sum_{w=1}^{W}\sum_{h=1}^{H} \| h_{l,c,w,h}^{t-1} - h_{l,c,w,h}^{t} \|, \tag{1}$$

and the KD loss for the final embedding:

$$L_{POD-flat} = \| h^{t-1} - h^{t} \|. \tag{2}$$

The total KD loss is:

$$L_{KD} = \sum_{l=1}^{L} w_l L_{POD-Pixels} + w_{flat} L_{POD-flat} \tag{3}$$

where w_l and w_{flat} are weight factors. Figure 5 illustrates the overview of PODNet-Pixels baseline.

3.4. Classifier in CIL

In many deep learning models, the classification head is a dense layer parameterized by $\theta = (\theta_i)_{i=1}^{n}$ where n is the number of classes, and θ_i is a vector with the same dimension as the final embedding h. The prediction score for a class c for a typical softmax classifier is calculated as:

$$\hat{y}_c = \frac{exp(\theta_c \cdot h)}{\sum_{i=1}^{n} exp(\theta_i \cdot h)} \tag{4}$$

In the CIL setting, one of the major causes of *catastrophic forgetting* is the class imbalance between the older classes and newer classes, since only a small number of training samples from previous tasks are kept when moving to a more recent task. Hou et al. [12] observe the manifestation of the class imbalance problem on the parameters θ of the classification head. Specifically, the magnitude of the vectors θ_i for newer classes is much larger than those for older classes; hence, the prediction scores $\hat{y}_c$ for newer classes tend to be higher than those for older

classes. In other words, the model tends to bias toward more recent classes, and this problem is called *task-recency bias* [26]. In UCIR [12], this problem is tackled by applying L2-normalization on both the final embedding h and the classifier parameter θ_i, which transforms the dot product into cosine similarity. The classification score from UCIR is as follows:

$$\hat{y}_c = \frac{exp(\eta\langle\theta_c, h\rangle)}{\sum_{i=1}^{n} exp(\eta\langle\theta_i, h\rangle)} \tag{5}$$

where $\langle .,. \rangle$ is cosine similarity and η is a learnable scale.

Despite the effect of KD loss to keep the final embedding consistent, the distribution of h often changes over time in the CIL setting. Since cosine similarity is sensitive to these changes, Douillard et al. [11] propose using K vectors $\theta_{c,k}$ to represent each class c instead of only using one proxy vector θ_c as in UCIR [12]. In PODNet [11], this classifier is called the Local Similarity Classifier (LSC), and the classification score in LSC is calculated as follows:

$$s_{c,k} = \frac{exp(\langle\theta_{c,k}, h\rangle)}{\sum_{i=1}^{n} exp(\langle\theta_{i,k}, h\rangle)}, \quad \hat{y}_c = \sum_{k=1}^{n} s_{c,k}\langle\theta_{i,k}, h\rangle \tag{6}$$

Empirically, Douillard et al. [11] found that using the NCA loss [56, 57] allows the model to converge faster than with simple cross-entropy loss. The NCA loss function is formulated to train a model using triplet data. Let x represent an anchor, y denote a positive sample, and Z represent a set of negative samples. The NCA loss encourages x to be closer to y than to any element $z \in Z$.

$$L_{NCA}(x, y, Z) = -log \frac{exp(-d(x, y))}{\sum_{z \in Z} exp(-d(x, z))}, \tag{7}$$

where $d(.)$ is a distance function.

We adopt the NCA loss L_{NCA} for training the model to classify videos, following PODNet

[11] and TCD [18]. When applying the NCA loss to the output of LSC, denoted as L_{LSC}, it is calculated as described in PODNet[11]:

$$L_{LSC} = \left[-\log \frac{\exp\left(\eta\left(\hat{\mathbf{y}}_y - \delta\right)\right)}{\sum_{i \neq y} \exp \eta \hat{\mathbf{y}}_i} \right]_+ \tag{8}$$

where $\hat{y}_y$ is the score prediction for groundtruth class y from LSC, δ is a small margin to enforce stronger class separation, η is a learnable scaling factor. A hinge $[\cdot]_+$ is used to keep the loss bounded.

We compute the final loss function as follows:

$$L = L_{LSC} + \lambda L_{KD} \tag{9}$$

where λ is the KD loss factor to control the influence of the knowledge distillation when training the model.

Figure 6: **Types of extracted backgrounds**. We categorize extracted backgrounds using the temporal median filter into 3 types. We hypothesize that when blended with other videos, the backgrounds type (a) *actor-and-scene background* is harmful to the task as the backgrounds contain humans as well. (b) *scene-only background* is beneficial as they are clean backgrounds. (c) *high-motion background* is beneficial as we can regard them as color jitter. We manually grouped each type after collecting the TMF output on the UCF-101 dataset. Best viewed with zoom and color.

3.5. Background augmentation

To mitigate background bias, we propose to employ background augmentation in class

incremental action recognition. We transform a video clip V_i into $\tilde{V}_i$ by blending V_i with a background B_j randomly selected from the background buffer BG_t as follows:

$$\tilde{V}_i(k) = (1-\alpha)V_i(k) + \alpha B_j \tag{10}$$

where, $0 < \alpha < 1$ is a blending factor.

For each incremental learning step t, firstly we extract a background B_i from each i-th video V_i with length l_i by using temporal median filter (TMF) [58] as follows where $Mdn(\cdot)$ stands for median operation.

$$B_i(x, y) = Mdn(V_i(x, y, 1), V_i(x, y, 2), ..., V_i(x, y, l_i)) \tag{11}$$

Each background pixel $B_i(x, y)$ is assigned a value equal to the median of all pixels at position (x, y) from all frames of a video V_i. Examples of TMF background are shown in Appendix .1. We store the extracted background frames $BG_t = B_{i\,i=1}^{K}$ for later use.

As shown in (11), we extract backgrounds from videos by the temporal median filter (TMF). The TMF is sensitive to the prevalent motion type in the video. We can categorize the extracted backgrounds into three types according to the prevalent motion types of videos as shown in Figure 6. Type I is low camera and human motion. In this case, visual information of both humans and scenes are preserved well by the TMF. We denote this kind of background as *actor-and-scene background*. *Scene-only background* is obtained in videos that have type II motion: the camera motion is low or zero and the humans move a lot in the video. In this case, scenes are mostly preserved well while the humans are removed from the background by the TMF. Type III motion videos have a high camera and human motions. Visual information of both scenes and humans is not preserved well by the TMF. Consequently, we get a blurry and random color blob as a background. We denote this kind of background as *high-motion background*.

Intuitively, blending videos with *scene-only backgrounds* increases the diversity of the static scenes, hence encouraging the model to focus on actors for prediction. This mechanism is similar to how ActorCutMix [39] encourages scene invariance. In ActorCutMix method [39], the first step involves obtaining the actor region for each frame. This region is defined as the bounding box of the actor, which is obtained by running an object detection model on the video frames. Once the actor region is determined, the background region is generated by removing this actor region from each frame. Finally, a training video is created by copy actor regions from one video to background region from another video frame-by-frame. Replacing ActorCutMix with our background augmentation offer a few advantages. First, we do not need extra learning to detect actors in videos. Second, an augmented video clip generated by ActorCutMix might contain high-frequency artifacts because of the copy-and-paste operation [39]. These artifacts may be detrimental because they break the temporal coherence of video clips and neural networks might pick up the high-frequency signals as a shortcut. As *high-motion background* mainly consists of random color blobs, blending a *high-motion background* to a video creates an augmentation effect similar to color jitters which is a commonly used data augmentation, hence it is likely to be beneficial for the training process as well. Augmented clips with *actor-and-scene backgrounds* might be problematic for training the model because the leftover actors in the background could confuse the model. The leftover actors might hamper the model to focus on the real actor in the video. We conduct a controlled experiment to validate the hypotheses above in Section 4.5 and Table 7.

3.6. Photometric/Geometric Augmentation

Inspired by *high-motion background* in Section 3.5, we employ photometric/geometric

augmentation as well to further debias class incremental learning. Intuitively, moderate color jitter and geometric transformations do not change the action class semantics. Therefore, adding photometric/geometric transformation might be beneficial for the class incremental learning for video action recognition as it can diversify the videos in every incremental learning step. With this motivation, we adopt RandAug [59] as our photometric/geometric augmentation. We apply the same photometric/geometric augmentation for every frame within the same video to ensure the temporally-coherent frames by following a recent work [60]. Applying different augmentations for each frame within the same video might be harmful as it could break the temporal consistency of a video.

Table 1: **Comparison with the state-of-the-arts on the UCF-101 dataset.** All the methods use a ResNet-34 TSM backbone. Column headers "Reg.", "Dist." and "Mem." stand for "Parameter Regularization", "Knowledge Distillation" and "Rehearsal" respectively. The memory size is set to 5 videos/class for all experiments. The experimental results are obtained by averaging across three random seeds. We choose PODNet-Pixel baseline to report our results. The best performance is in **bold** and the second best is underlined.

Number of classes				10×5 stages			5×10 stages			2×25 stages		
Method	Reg.	Dist.	Mem.	CNN	NME	Avg.	CNN	NME	Avg.	CNN	NME	Avg.
Fine-tuning				24.97	-	24.97	13.45	-	13.45	5.78	-	5.78
LwFMC [17]	✓			42.14	-	42.14	25.59	-	25.59	11.68	-	11.68

LwM [46]		✓		43.39	-	43.39	26.07	-	26.07	12.08	-	12.08
iCaRL [14]		✓	✓	-	65.34	65.34	-	64.51	64.51	-	58.73	58.73
UCIR [12]		✓	✓	74.31	74.09	74.20	70.42	70.50	70.46	63.22	64.00	63.61
PODNet [11]		✓	✓	73.26	74.37	73.82	71.58	73.75	72.67	70.28	71.87	71.08
TCD [18]	✓	✓	✓	74.89	77.16	76.03	73.43	75.35	74.39	72.19	74.01	73.10
FrameMaker [22]		✓	✓	<u>78.13</u>	<u>78.64</u>	<u>78.39</u>	<u>76.38</u>	<u>78.14</u>	<u>77.26</u>	<u>75.77</u>	<u>77.49</u>	<u>76.63</u>
Ours		✓	✓	**81.04**	**79.84**	**80.44**	**80.07**	**79.57**	**79.82**	**77.55**	**77.76**	**77.66**
Oracle (Upper Bound)				84.15	83.37	83.76	83.96	83.20	83.58	83.82	83.16	83.49

4. Experimental Results

4.1. Datasets

We validate the proposed method on the three publicly available video action recognition datasets: UCF-101 [8], Something-Something-v2 [6], and HMDB-51 [24]. The UCF-101 consists of 13,320 videos of 101 action categories. The HMDB-51 consists of 6,766 videos with 51 action categories. Both HMDB-51 and UCF-101 datasets come with 3 splits for training and testing. We choose split 1 following previous work [18]. The Something-Something-v2 is a large-scale dataset

with 165K training videos of 174 action classes. Something-Something-v2 requires more temporal reasoning to perform well when compared to UCF-101 and HMDB-51 datasets.

4.2. Evaluation protocol

Different task splitting results in substantial variations in the overall performance in class incremental learning settings. Therefore, we strictly follow the evaluation protocol of TCD [18] by using the same task size, random seeds[1], and the splits. We report our results using the 3 task splits scheme defined in TCD [18] for a fair comparison. Following the previous works, we use both CNN and NME evaluation protocols. Accuracy in CNN is calculated from the output of the classifier and this term comes from UCIR [12]. Nearest-mean-exemplar (NME) classifier [14] makes a prediction by comparing a feature representation of a test sample to the mean class representation of exemplars. Backward forgetting (BWF) [61, 21] measures the influence of learning from the current task affects performance on test data of the previous task. Oracle model [18] is incrementally trained while preserving all data from the previous task. Oracle model does not suffer from catastrophic forgetting, hence it serves as the upper bound.

4.3. Implementation Details

PODNet-Pixel baseline. For a fair comparison with other video CIL methods [18, 22], we use ResNet-34 TSM as the backbone for UCF-101 dataset, and ResNet-50 TSM for Something-Something-v2 dataset. All backbones are pretrained on ImageNet dataset and implemented in mmaction2 library [62]. Unless otherwise specified, we set the memory size as 5 videos/class for experiments on the HMDB-51 and UCF-101 datasets, and 20 videos/class for the

[1] Random seeds: 1000, 1993, 2021

Something-Something-v2 dataset. Knowledge distillation loss is calculated on logits and immediate features output of TSM layers in the backbone. On the incremental training stage, we split the training into 2 steps for each task. For the first step, model is trained with training data $D_t^{'}$, then we create exemplars E_t with herding strategy [14]. We finetune the model using data of the current task exemplar set E_t and the exemplar set of previous tasks $E_{0:t-1}$. Unlike previous works [18, 11, 12] where they finetune the classifier while freezing the backbone, we finetune both the backbone and classifier because they empirically improve the accuracy of the model. We use the local similarity classifier (LSC) in all experiments with only one proxy and a learnable parameter η. We set $\delta = 0.6$ in all experiments following PODNet [11] and TCD [18]. For the UCF-101 dataset [8] 10×5 stages experiment, we set KD loss factor λ to 1.0 for the feature maps of immediate layers and 0.01 for logits. For the UCF-101 5×10 stages and 2×25 stages experiments, we set λ to 0.01 for both the feature maps of immediate layers and logits. For the Something-Something-v2 dataset, we set λ to 0.5 for the feature maps of immediate layers and 1 for logits. For the HMDB-51 dataset, we set λ to 3.0 for the feature maps of immediate layers and 0.1 for logits. For all the experiments, we multiply λ with an adaptive scaling factor $\lambda_{adaptive} = \sqrt{\frac{|C_{1:k}|}{|C_k|}}$ where $C_{1:k}$ denotes the number of seen classes until task k, following the previous works [12, 11, 18]. We choose PODNet-Pixel as the main baseline to test our methods when comparing with other CIL methods due to its simplicity and good performance.

iCaRL baseline. To compare different video data augmentation methods, we use the iCaRL baseline since both ActorCutMix [39] and VideoMix [63] require label smoothing. The NCA loss used in the PODNet-Pixel baseline is not applicable for label smoothing. The iCaRL

implementation used in Table 5 and Table 8 differ slightly from the original iCaRL [14]. As suggested in the previous work [64], we train the iCaRL baseline using the cross-entropy loss, which has been shown to improve overall accuracy over the binary cross-entropy loss.

For both baselines, the model is trained with batch size of 96 on UCF-101, and the batch size of 48 for HMDB-51 and Something-Something-v2 datasets. The models are trained for 50 epochs per incremental task. We train the model using SGD with a learning rate of 10^{-3}, decay rate of 10^{-4}, and 0.9 in momentum. The learning rate is decreased by 10 times at epochs 20 and 30. In all experiments, we set the background blending factor $\alpha = 0.5$, and the background augmentation probability $p = 0.25$, unless stated otherwise. For training samples that are not background augmented, they are fed to the photometric/geometric augmentation pipeline when both pipelines are used. We use either four Tesla V100 or RTX 3090 GPUs for the model training and testing.

4.4. Comparison with state-of-the-art

Here, we compare our method with state-of-the-art on public benchmarks. Note that every method compared here is equipped with the same TSM backbone. As shown in Table 1, the proposed method achieves state-of-the-art performance on the UCF-101 dataset, consistently across all class incremental learning settings with both CNN and NME evaluation protocols. Remarkably, our method is only 3.3 points below the upper bound on average. Additionally, we show the results on the Something-Something-v2 dataset in Table 2. In all settings, our method outperforms FrameMaker [22], the former state-of-the-art method, with a significant margin ($7.78 \sim 8.45$ points) under the CNN evaluation protocol. The results show that our background

debiased class incremental learning method is effective in a dataset that requires complex temporal reasoning as well.

As shown in Table 3, the proposed method shows favorable performance on the HMDB-51 dataset [24] in terms of both CNN and NME evaluation protocol. We observe 4.09 points improvement on average compared to FrameMaker [22]. Remarkably, our method is only 2.09 points below the upper bound on average.

Figure 7 shows the average accuracy for each incremental training step. In all settings, our method achieves the highest accuracy under the CNN evaluation protocol. In the Something-Something-v2 5×18 stages setting, our method achieves the highest accuracy for the last few incremental steps under NME evaluation protocol as well.

Table 2: **Comparison with the state-of-the-arts on the Something-Something-v2 dataset.** All the methods use a ResNet-50 TSM backbone. The memory size is set to 20 videos/class for all experiments. The experimental results are obtained by averaging across three random seeds. We choose PODNet-Pixel baseline to report our results. The best performance is in **bold** and the second best is underlined.

Number of classes	10×9 stages			5×18 stages		
Method	CNN	NME	Avg.	CNN	NME	Avg.
UCIR	26.84	17.98	22.41	20.69	12.57	16.30
PODNet	34.94	27.33	31.14	26.95	17.49	22.22
TCD	35.7	<u>28.8</u>	32.3	29.6	21.6	25.6

	8	8	3	0	3	2
FrameMaker	37.2	**29.9**	33.5	30.9	22.8	26.9
	5	**2**	9	8	4	1
Ours	**45.0**	24.4	**34.7**	**39.4**	**24.1**	**31.1**
	3	3	**3**	**3**	**4**	**9**

Table 3: **Comparison with the state of the arts on the HMDB51 dataset.** All the methods use a ResNet-50 TSM backbone. The memory size is set to 5 videos/class, and use 5×5 stages setting for all experiments. The experimental results are obtained by averaging across three random seeds. We choose PODNet-Pixel baseline to report our results. The best performance is in **bold** and the second best is underlined.

Method	CNN	NME	Avg
Fine-tuning	16.8	-	16.8
	2		2
LwFMC [17]	26.8	-	26.8
	2		2
LwM [46]	26.9	-	26.9
	7		7
iCaRL [14]	-	40.0	40.0
		9	9
UCIR [12]	44.9	46.5	45.7
	0	3	2
PODNet [11]	44.3	48.7	46.5
	2	8	5

TCD [18]	45.34	50.36	47.85
FrameMaker [22]	<u>47.54</u>	<u>51.12</u>	<u>49.33</u>
Ours	**55.20**	**51.64**	**53.42**
Oracle (Upper Bound)	55.03	55.98	55.51

Figure 7: **Average accuracy across tasks on the UCF-101, Something-Something-v2, and HMDB-51 datasets.** The proposed method shows significant performance gain compared to the existing methods.

Comparison of biased start baseline with state-of-the-art. To emphasize the effectiveness of the proposed method in the context of CIL, we show the results obtained from the *biased start* baseline in Figure 8 together with other CIL and video augmentation methods. In this baseline, we disable the debiasing augmentations during the base task (task 0) and activate them during the incremental learning steps. The figure below illustrates the *average accuracy* and *absolute slope* of the accuracy curve. We observe that the *biased start* baseline still achieves favorable average accuracy compared to the other methods while the base task accuracy of this baseline is similar to that of existing methods. Moreover, the *biased start* baseline shows a smaller *absolute slope* in the accuracy curve compared to the other methods being compared. These findings indicate the effectiveness of our proposed method not only in the base task but also in the

incremental learning steps.

Figure 8: **Comparison of biased start baseline with other CIL methods on the UCF-101 dataset.** The biased start baseline achieves higher average accuracy and a smaller absolute slope compared to existing methods, with similar base task accuracy.

Effect of memory size. In Table 4, we analyze the effect of memory size (videos per class) in the UCF-101 5×10 stages setting. Following existing works [11, 18, 22], we evaluate CNN and NME performances with varying memory sizes. Our method outperforms all the existing methods with significant margins except for the memory size of 1 video/class case. Our results clearly indicate that diversifying the background is a crucial factor in achieving effective class incremental action recognition across different memory sizes.

Table 4: **Memory size analysis.** The proposed method shows favorable performance across memory budgets. We show results on the UCF-101 dataset, 5×10 stages setting. The best performance is in **bold** and the second best is underlined.

Memory size	1		2		5		10	
Method	CNN	NME	CNN	NME	CNN	NME	CNN	NME
iCaRL	-	58.05	-	60.50	-	64.51	-	66.94
UCIR	61.92	65.52	66.43	67.58	70.42	70.50	72.47	71.69
PODNet	63.1	70.9	65.9	72.7	71.5	73.7	75.4	76.3

	8	6	3	8	8	5	4	9
TCD	64.52	71.96	68.40	73.30	73.43	75.35	76.66	77.09
FrameMaker	**73.64**	**76.98**	75.19	**77.43**	76.38	78.14	-	-
Ours	66.72	72.42	**76.36**	76.38	**80.07**	**79.57**	**81.92**	**81.59**

Comparison with existing video data augmentation methods. We validate the hypothesis that background debiasing is crucial in class incremental action recognition by comparing the proposed method with existing video augmentation methods. We compare the proposed method with VideoMix[63] and ActorCutMix[39] in Table 5. VideoMix is a video cut-and-paste method without background debiasing. ActorCutMix is a background debiasing augmentation. The results show that mixing videos without background debiasing (VideoMix w/o. photometric/geometric aug.) shows inferior performance than mixing videos with debiasing (VideoMix w. photometric/geometric aug., ActorCutMix, and ours). Among the compared video data augmentations, ours shows the best performance. The reason for choosing iCaRL as a baseline in this experiment is explained in Section 4.3. We also provide the accuracy curves for these experiments in Figure Figure 9 for a more complete comparison.

Table 5: **Comparison with existing video augmentations.** Experiment on the UCF-101 dataset with the iCaRL baseline. The VideoMix method is tested with and without photometric/geometric augmentation (with aug. and without aug.). The memory size is set to 5 videos/class for all experiments. The best performance is in **bold** and the second best is underlined.

Number of classes		10×5 stages	5×10 stages
Method	Debiasing ?	NME	NME
VideoMix w/o. aug.	×	72.00	70.80
VideoMix w. aug.	✓	78.11	76.86
ActorCutMix	✓	<u>78.72</u>	<u>78.09</u>
Ours	✓	**80.51**	**79.94**

Here, we also discuss the differences between our method and other fusion methods. VideoMix and ActorCutMix replaces a region on a video frame with a patch of a frame from another training video. As we discussed in Section 3.5, an advantage of the proposed method over VideoMix and ActorCutMix is that ours does not use copy-and-paste operations which create high-frequency artifacts. The high-frequency artifacts from copy-and-paste operations may be detrimental because the operations could break the temporal coherence of video clips [39].

4.5. Ablation study

Effect of each debiasing augmentation. We validate the effectiveness of photometric/geometric and background augmentations using the UCF-101 and Something-Something-v2 datasets and present the results in Table 6. The results on the UCF-101 dataset are averaged over three random seeds. Since Something-Something-v2 is a very large dataset, we conducted the ablation study with a seed value of 1000 only. Photometric/geometric augmentation shows a significant accuracy improvement of 2.69 and 3.24 points on average compared to the respective baselines on the UCF-101 and Something-Something-v2 datasets. Similarly, background augmentation yields a

notable improvement of 2.91 and 1.42 points on average compared to the respective baselines. Finally, the combination of photometric/geometric augmentation and background augmentation results in a significant improvement of 4.64 and 4.89 points on average on the two datasets. Since UCF-101 is a static-bias dataset and Something-Something-v2 is a temporal-bias dataset, our results demonstrate that our proposed method is effective for both types of video action recognition datasets.

Table 6: **Ablation study on different augmentations.** We show results on the UCF-101 and Something-Something-v2 (Sth-Sth) datasets. We use PODNet-Pixel baseline for all experiments in the table. The memory size is set to 5 videos/class for the UCF-101 experiments, and 20 videos/class for the Something-Something-v2 experiments. The best performance is in **bold** and the second best is underlined.

Dataset	UCF-101 5×10 stages			Sth-Sth 10×9 stages		
Method	CNN	NME	Avg.	CNN	NME	Avg.
Baseline	75.61	75.99	75.80	41.53	19.63	30.58
Photo./geo. aug.	78.64	78.34	78.49	44.21	23.42	33.82
Background aug.	79.14	78.28	78.71	43.16	20.83	32.00
Both aug.	**81.04**	**79.84**	**80.44**	**45.68**	**25.26**	**35.47**

Figure 9: **Comparison of biased start baseline with other video augmentation methods on the UCF-101 dataset.** The biased start baseline achieves higher average accuracy and a smaller absolute slope compared to existing methods, with similar base task accuracy.

Effect of background type. As discussed in Section 3.5, we can categorize backgrounds extracted with temporal median filter into three types. We design several controlled experiments to study the effect of each background type. To simplify the task of categorizing ~ 10 K training videos from UCF-101, we employ pseudo background data instead. This involves creating specific data generation pipelines, as described below:

Table 7: **Ablation study on different background types.** We show experimental results on the UCF-101 dataset with a memory size of 5 videos/class. The results are from the 5×10 stages setting. The best performance is in **bold** and the second best is underlined.

Method	CNN	NME	Avg.
Photometric/geometric aug. only	78.43	78.67	78.55
+ Actor-and-scene (pseudo) bkg. aug.	78.30	77.58	77.94
+ High-motion (pseudo) bkg. aug.	79.20	79.12	79.16
+ Scene-only & high-motion bkg. aug.	**79.98**	**80.00**	**79.99**
+ All backgrounds	79.84	79.11	79.48

Type I: *Pseudo actor-and-scene background.* They have both actors and scenes well preserved. Therefore, we can get this type of pseudo backgrounds directly from a video. Given training data D_t' of task t, we randomly select a video from D_t' and then randomly select a

frame in the selected video. The randomly sampled frame contains both scene and actor(s) because UCF-101 is a temporally trimmed dataset.

Type III: *Pseudo high-motion backgrounds.* They are backgrounds from videos with high motions. We can collect a set of pseudo backgrounds by simulating camera motion. For a video from training data, we apply temporally inconsistent spatial cropping for each frame of the video, then apply the temporal median filter on all cropped frames to obtain a pseudo *high-motion background*.

Type II+III: *Scene-only* & *high-motion backgrounds.* We obtain this pseudo background set by discarding all actor-and-scene backgrounds from all backgrounds extracted by TMF. We get all actor-and-scene backgrounds by running Mask-RCNN [65] to detect humans.

Figure 10: **Analysis on the effect of background bias to the average accuracy and backward forgetting (BWF).** The model with debiasing consistently outperforms the baseline without debiasing in both average accuracy and backward forgetting across different scene distances between exemplar set and test set.

To study the effect of background type, we train three models by replacing the backgrounds BG_t with these pseudo backgrounds in our training pipeline described in Section 3.1. We turn on the photometric/geometric augmentation for all the models for a fair comparison. We show the results in Table 7. We set photometric/geometric augmentation without background augmentation as a baseline. Compared to the baseline, actor-and-scene background shows inferior performance which validates the hypothesis: Augmented clips with *actor-and-scene backgrounds* might be problematic for training the model because the leftover actors in the background could

confuse the model. As shown in the result, *high-motion background* shows superior performance (79.16%) compared to the baseline (78.55%). This result also verifies our hypothesis: blending a *high-motion background* to a video creates an augmentation effect similar to color jitters which is a commonly used data augmentation, hence it is likely to be beneficial for the training process. Scene-only & high-motion background shows the best performance (79.99%) among all the methods. The results indicate that adding scene-only background augmentation on top of high-motion background augmentation is beneficial for the class incremental learning. The controlled experimental results verify all hypotheses we set in Section 3.5. In order to use *scene-only* & *high-motion backgrounds*, we need to filter out the *actor-and-scene backgrounds* by running a human detector on every background frame. While the performance gain from filtering out *actor-and-scene backgrounds* is not significant (0.51 points), running a human detector on every background frame is computationally expensive. Therefore we use all background types for our debiasing method in all experiments except for the experiments in Table 7.

Table 8: **Compatibility with different CIL methods.** We show experimental results on the UCF-101 dataset with a memory size of 5 videos/class. The results are from the 10×5 stages setting. The best performance is in **bold** and the second best is underlined.

Number of classes	10×5 stages		5×10 stages		2×25 stages	
Method	CNN	NME	CNN	NME	CNN	NME
iCaRL baseline	-	77.74	-	76.80	-	74.64
iCaRL w. debiasing	-	**80.51**	-	**79.94**	-	<u>77.23</u>
PODNet-Pixel baseline	<u>75.61</u>	76.00	<u>75.00</u>	75.98	<u>70.37</u>	72.32

PODNet-Pixel w. debiasing **81.04** 79.84 **80.07** 79.57 **77.55** **77.76**

Compatibility with different CIL methods. In this work, we provide an effective plug-and-play background debiasing method that can be plugged into any CIL method. To validate the compatibility with different CIL methods, we plug the proposed method into two well-established CIL methods: iCaRL [14] and PODNet-pixel [11]. Table 8 shows that we can achieve significant improvement when the proposed method is plugged into iCaRL and PODNet. The proposed method achieves $2.59 \sim 3.14$ and $3.59 \sim 7.18$ points improvement compared to iCaRL and PODNet-Pixel baselines respectively. The results validate that the proposed method is compatible with different class incremental learning methods.

Ablation study on hyperparameters. We provide empirical findings regarding the blending factor α, and the background augmentation probability p, as summarized in the Table 9. Our method demonstrates the highest performance when $\alpha = 0.5$ and $p = 0.5$.

Table 9: **Ablation study on blending factor α and the background augmentation probability p.** We conduct the experiments on UCF-101 10×5 stages setting with a random seed 1000. The memory size is set to 5 videos/class for all experiments. The best performance is in **bold** and the second best is underlined.

α	p	CNN ↑	NME ↑
0.2	0.25	79.04	77.55
0.3	0.25	78.92	77.52
0.4	0.25	79.96	77.47

0.5	0.25	80.36	78.94
0.5	0.50	**80.52**	**79.15**
0.5	0.75	78.77	77.78

Figure 11: **Grad-CAM visualization on the UCF-101 10 × 5 stages setting.** Our debiased method consistently focuses and the actor instead of the background across different incremental learning steps. Green indicates correct and red indicates incorrect predictions. Best viewed with zoom and color.

4.6. Analysis on background bias in CIL

The background bias of action recognition models is even more severe in the context of CIL due to the scarcity of previous task examples. Here, we conduct a controlled experiment to demonstrate the importance of scene debiasing in the context of class-incremental action recognition. In Figure 10, we show two plots: i) the average accuracy vs. scene distance between exemplar set and test set. ii) the backward forgetting vs. scene distance between exemplar set and test set.

In Figure 10, the model with debiasing consistently outperforms the baseline without debiasing in both average accuracy and BWF across different scene distances. When scene distance is high (0.22), the proposed method shows a significantly higher performance compared to the baseline (77.05% vs. 72.63%). The results demonstrate that the proposed method improves the robustness to background bias in the context of class-incremental action recognition.

Experimental settings. To measure the scene distance between exemplar set and test set, we employ a scene classifier: ResNet-50 pre-trained on the place365 dataset [66]. We run the scene

classifier to extract scene features from all videos of the UCF-101 dataset. For each video, we uniformly sample 8 frames and feed them to the scene classifier to extract scene features. We average the 8 scene feature vectors to get a single scene feature vector per video. Then, we calculate a cosine distance matrix between videos in the training set and test set. We define the distance between a training video to the test set as the smallest distance between the training video and the test set. Similarly, we define the distance between an exemplar set and the test set as the average distance to the test set of all videos in the exemplar set. To control the scene distance between the exemplar sets to the test set, we pre-define the exemplar sets before training models. For each class, we sort the training videos by the scene distance to the test set. Then, we uniformly split the sorted training videos into 8 groups with different scene distances to the test set. We repeat this process for all classes, and form 8 exemplar sets. Then we train a baseline without debiasing and a model with debiasing with the pre-defined exemplar sets. We observe how the class-incremental learning performance changes according to the scene distance.

Implementation details. We conduct the experiments in the UCF-101 10×5 stages setting with memory size set to 5 videos/class. We use a random seed 1000 for incremental task splits. As a scene feature extractor, we employ the ResNet-50 pre-trained on the place365 dataset [66]. The baseline is PODNet-Pixel. We use both photometric/geometric augmentation and background augmentation for debiasing.

4.7. Qualitative evaluation

We present the Grad-CAM [67] visualizations of class incremental action recognition models in Figure 11. We conduct the experiment on the UCF-101 10×5 stages setting with random seed of 1000. We use PODNet-Pixel as a baseline. To highlight the impact of catastrophic

forgetting related to background bias, we select videos from the testing set of the initial task ($t = 0$) that are incorrectly classified by the baseline at task 5 but correctly classified by the baseline at task 0 and our model at task 0 and 5. In example (a), the baseline model at task 0 focuses on the background but correctly predicts TennisSwing class. However, at task 5, the baseline model incorrectly predicts the same video sample as a more recently learned FieldHockeyPenalty class by focusing on the background. This scenario exemplifies a typical case of catastrophic forgetting where the model tends to favor the more recent class. In contrast, the proposed method focuses on the actor and predicts the correct TenisSwing class in the stage 5. We observe a similar pattern in the other two examples.

5. Conclusions

In this paper, we introduce a simple, yet effective plug-and-play method for class incremental learning for video action recognition. We hypothesize mitigating representation bias is crucial for the class incremental learning for video action recognition. We propose to employ background augmentation and photometric/geometric augmentation to diversify videos in every incremental learning step to tackle the catastrophic forgetting problem. We empirically validate the effectiveness of the proposed method through extensive experiments. We show promising video action recognition performance on the public benchmarks consistently across multiple datasets, class incremental learning settings, and evaluation protocols.

6. Limitation

While the method proposed in this study leverages the prevalence of background bias in datasets, it is pertinent to acknowledge its limitations, particularly concerning fine-grain action

recognition datasets such as the Diving48 dataset [23]. Unlike more generalized action recognition datasets like UCF-101, HMDB-51, ActivityNet, and Kinetics-400, which commonly exhibit background bias, these specialized datasets may vary in their degree of background bias. This variance could potentially constrain the applicability and effectiveness of the proposed method within this specific domain.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No new data were created or analysed during this study. Data sharing is not applicable to this article. The code and pretrained models will be made publicly available upon acceptance.

Acknowledgement

This work was supported by The Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2020R1A6A1A03038540). Additionally, this work was also supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea government(MSIT) including: 1) Development of explainable AI-based diagnosis and analysis frame work using energy demand big data in multiple domains (No.2022-0-00106), and 2) the metaverse support program to nurture the best talents (IITP-2023-RS-2023-00254529).

References

[1] J. Carreira, A. Zisserman, Quo vadis, action recognition? a new model and the kinetics dataset, in: CVPR, 2017.

[2] C. Feichtenhofer, H. Fan, J. Malik, K. He, Slowfast networks for video recognition, in: ICCV, 2019.

[3] J. Lin, C. Gan, S. Han, Tsm: Temporal shift module for efficient video understanding, in: ICCV, 2019.

[4] G. Bertasius, H. Wang, L. Torresani, Is space-time attention all you need for video understanding?, in: ICML, 2021.

[5] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al., The kinetics human action video dataset, arXiv preprint arXiv:1705.06950 (2017).

[6] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, et al., The" something something" video database for learning and evaluating visual common sense, in: ICCV, 2017.

[7] F. Caba Heilbron, V. Escorcia, B. Ghanem, J. Carlos Niebles, Activitynet: A large-scale video benchmark for human activity understanding, in: CVPR, 2015.

[8] K. Soomro, A. R. Zamir, M. Shah, Ucf101: A dataset of 101 human actions classes from videos in the wild, arXiv preprint arXiv:1212.0402 (2012).

[9] M. McCloskey, N. J. Cohen, Catastrophic interference in connectionist networks: The sequential learning problem, in: Psychology of learning and motivation, Vol. 24, Elsevier, 1989, pp. 109–165.

[10] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, Y. Bengio, An empirical investigation of catastrophic forgetting in gradient-based neural networks, in: ICLR, 2014.

[11] A. Douillard, M. Cord, C. Ollion, T. Robert, E. Valle, Podnet: Pooled outputs distillation for small-tasks incremental learning, in: ECCV, 2020.

[12] S. Hou, X. Pan, C. C. Loy, Z. Wang, D. Lin, Learning a unified classifier incrementally via rebalancing, in: CVPR, 2019.

[13] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, Y. Fu, Large scale incremental learning, in: CVPR, 2019.

[14] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, C. H. Lampert, icarl: Incremental classifier and representation learning, in: CVPR, 2017.

[15] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al., Overcoming catastrophic forgetting in neural networks, Proceedings of the national academy of sciences 114 (13) (2017) 3521–3526.

[16] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, T. Tuytelaars, Memory aware synapses: Learning what (not) to forget, in: ECCV, 2018.

[17] Z. Li, D. Hoiem, Learning without forgetting, TPAMI 40 (12) (2017) 2935–2947.

[18] J. Park, M. Kang, B. Han, Class-incremental learning for action recognition in videos, in: ICCV, 2021.

[19] H. Zhao, X. Qin, S. Su, Y. Fu, Z. Lin, X. Li, When video classification meets incremental classes, in: ACM MM, 2021.

[20] J. Ma, X. Tao, J. Ma, X. Hong, Y. Gong, Class incremental learning for video action classification, in: ICIP, 2021.

[21] A. Villa, K. Alhamoud, V. Escorcia, F. Caba, J. L. Alcázar, B. Ghanem, vclimb: A novel video class incremental learning benchmark, in: CVPR, 2022.

[22] Y. Pei, Z. Qing, J. Cen, X. Wang, S. Zhang, Y. Wang, M. Tang, N. Sang, X. Qian, Learning a condensed frame for memory-efficient video class-incremental learning, in: NeurIPS, 2022.

[23] Y. Li, Y. Li, N. Vasconcelos, Resound: Towards action recognition without representation bias, in: ECCV, 2018.

[24] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, Hmdb: a large video database for human motion recognition, in: ICCV, 2011.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: CVPR, 2009.

[26] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, J. van de Weijer, Class-incremental learning: survey and performance evaluation on image classification, TPAMI (2022).

[27] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: NeurIPS, 2014.

[28] C. Feichtenhofer, A. Pinz, A. Zisserman, Convolutional two-stream network fusion for video action recognition, in: CVPR, 2016.

[29] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in: ICCV, 2015.

[30] K. Hara, H. Kataoka, Y. Satoh, Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?, in: CVPR, 2018.

[31] X. Wang, R. Girshick, A. Gupta, K. He, Non-local neural networks, in: CVPR, 2018.

[32] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, M. Paluri, A closer look at spatiotemporal convolutions for action recognition, in: CVPR, 2017.

[33] S. Xie, C. Sun, J. Huang, Z. Tu, K. Murphy, Rethinking spatiotemporal feature learning for video understanding, in: ECCV, 2018.

[34] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, C. Schmid, Vivit: A video vision transformer, in: ICCV, 2021.

[35] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, C. Feichtenhofer, Multiscale vision transformers, in: ICCV, 2021.

[36] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, H. Hu, Video swin transformer, in: CVPR, 2022.

[37] J. Wang, G. Bertasius, D. Tran, L. Torresani, Long-short temporal contrastive learning of video transformers, in: CVPR, 2022.

[38] A. Singh, O. Chakraborty, A. Varshney, R. Panda, R. Feris, K. Saenko, A. Das, Semi-supervised action recognition with temporal contrastive learning, in: CVPR, 2021.

[39] Y. Zou, J. Choi, Q. Wang, J.-B. Huang, Learning representational invariances for data-efficient action recognition, CVIU (2023).

[40] C.-Y. Wu, C. Feichtenhofer, H. Fan, K. He, P. Krahenbuhl, R. Girshick, Long-term feature banks for detailed video understanding, in: CVPR, 2019.

[41] M. M. Islam, G. Bertasius, Long movie clip classification with state-space video models, in: ECCV, 2022.

[42] F. Feng, Y. Ming, N. Hu, J. Zhou, See, move and hear: a local-to-global multi-modal interaction network for video action recognition (2023).

[43] O. Ostapenko, M. Puscas, T. Klein, P. Jahnichen, M. Nabi, Learning to remember: A

synaptic plasticity driven framework for continual learning, in: CVPR, 2019.

[44] H. Shin, J. K. Lee, J. Kim, J. Kim, Continual learning with deep generative replay, in: NeurIPS, 2017.

[45] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, K. Alahari, End-to-end incremental learning, in: ECCV, 2018.

[46] P. Dhar, R. V. Singh, K.-C. Peng, Z. Wu, R. Chellappa, Learning without memorizing, in: CVPR, 2019.

[47] F. Zenke, B. Poole, S. Ganguli, Continual learning through synaptic intelligence, in: ICML, 2017.

[48] G. M. Van de Ven, A. S. Tolias, Three scenarios for continual learning, arXiv preprint arXiv:1904.07734 (2019).

[49] Y.-C. Hsu, Y.-C. Liu, A. Ramasamy, Z. Kira, Re-evaluating continual learning scenarios: A categorization and case for strong baselines, arXiv preprint arXiv:1810.12488 (2018).

[50] B. Zhao, X. Xiao, G. Gan, B. Zhang, S.-T. Xia, Maintaining discrimination and fairness in class incremental learning, in: CVPR, 2020.

[51] J. Choi, C. Gao, J. C. Messou, J.-B. Huang, Why can't i dance in the mall? learning to mitigate scene bias in action recognition, in: NeurIPS, 2019.

[52] H. Bahng, S. Chun, S. Yun, J. Choo, S. J. Oh, Learning de-biased representations with biased representations, in: ICML, 2020.

[53] W. Bao, Q. Yu, Y. Kong, Evidential deep learning for open set action recognition, in: ICCV, 2021.

[54] S. N. Gowda, M. Rohrbach, F. Keller, L. Sevilla-Lara, Learn2augment: Learning to composite videos for data augmentation in action recognition, in: ECCV, 2022.

[55] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, 2016.

[56] J. Goldberger, G. E. Hinton, S. Roweis, R. R. Salakhutdinov, Neighbourhood components analysis (2004).

[57] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, S. Singh, No fuss distance metric learning using proxies, in: ICCV, 2017.

[58] M. Piccardi, Background subtraction techniques: a review, in: SMC, 2004.

[59] E. D. Cubuk, B. Zoph, J. Shlens, Q. V. Le, Randaugment: Practical automated data augmentation with a reduced search space, in: CVPR Workshops, 2020.

[60] R. Qian, T. Meng, B. Gong, M.-H. Yang, H. Wang, S. Belongie, Y. Cui, Spatiotemporal contrastive video representation learning, in: CVPR, 2021.

[61] D. Lopez-Paz, M. Ranzato, Gradient episodic memory for continual learning, in: NeurIPS, 2017.

[62] M. Contributors, Openmmlab's next generation video understanding toolbox and benchmark, https://github.com/open-mmlab/mmaction2 (2020).

[63] S. Yun, S. J. Oh, B. Heo, D. Han, J. Kim, Videomix: Rethinking data augmentation for video classification, arXiv preprint arXiv:2012.03457 (2020).

[64] C. He, R. Wang, X. Chen, A tale of two cils: the connections between class incremental learning and class imbalanced learning, and beyond, in: CVPR, 2021.

[65] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, R. Girshick, Detectron2, https://github.com/facebookresearch/detectron2 (2019).

[66] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, A. Torralba, Places: A 10 million image database for scene recognition, TPAMI (2017).

[67] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: ICCV, 2017.

Appendix .1. Backgrounds extracted by TMF

We show background frames extracted by TMF on the UCF-101 dataset in Figure .12. The extracted backgrounds are diverse. There are three types of backgrounds as we mention in Section 3.5: 1) *actor-and-scene*, 2) *scene-only*, and 3) *high-motion* backgrounds. We manually select some samples for each background types

Figure .12: **Visualization of backgrounds extracted by temporal median filter (TMF).** We categorize extracted backgrounds using the temporal median filter into 3 types. We hypothesize that when blended with other videos, the backgrounds type (a) *actor-and-scene background* is harmful to the task as the backgrounds contain humans as well. (b) *scene-only background* is beneficial as they are clean backgrounds. (c) *high-motion background* is beneficial as we can regard them as color jitter. We manually grouped each type after collecting the TMF output on the UCF-101 dataset. Best viewed with zoom and color.

Appendix .2. Pseudo background for experiments on different background types

As discussed in the Section 4.5, we design several controlled experiments to study the effect of each background type and present the results in Table 7. Figure .13 shows samples of the background types used for the experiments.

Figure .13: **Types of pseudo backgrounds**. We visualize a few backgrounds used for the controlled experiments in Section 4.5. (a) pseudo *actor-and-scene background* obtained by randomly sampling a frame per video. (b) pseudo *high-motion background* obtained by applying temporally inconsistent spatial cropping followed by TMF to each video. (c) *scene-only* &

high-motion background extracted with TMF followed by human detector to discard the frames with humans. Best viewed with zoom and color.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Highlights

Background Debiased Class Incremental Learning for Video Action Recognition

Le Quan Nguyen, Jinwoo Choi, L.Minh Dang, Hyeonjoon Moon

- We identify a background bias problem in class incremental learning for video action recognition (video CIL). We further analyze the background bias problem in the Video CIL setting using scene distance experiment, and Grad-CAM visualization to confirm our hypothesis about background bias problem in video CIL
- We propose a simple, yet effective plug-and-play method for class incremental learning for video action recognition by augmenting backgrounds for every incremental learning step. The proposed background augmentation mitigates background biases and catastrophic forgetting.
- By addressing the background bias, our method achieves significant performance improvements compared to CIL baselines that do not account for it in various public benchmarks, and our proposed method achieves state-of-the-art performance.

Highlights

- Video-based class incremental learning (video CIL) is important, yet under-explored.
- Identify the background bias problem in video CIL setting.
- The proposed method diversifies the scenes of training videos in the CIL setting.
- The proposed method is a plug-and-play video CIL method.
- The proposed method is tested UCF-101, HMDB-51, and Something-Something v2.

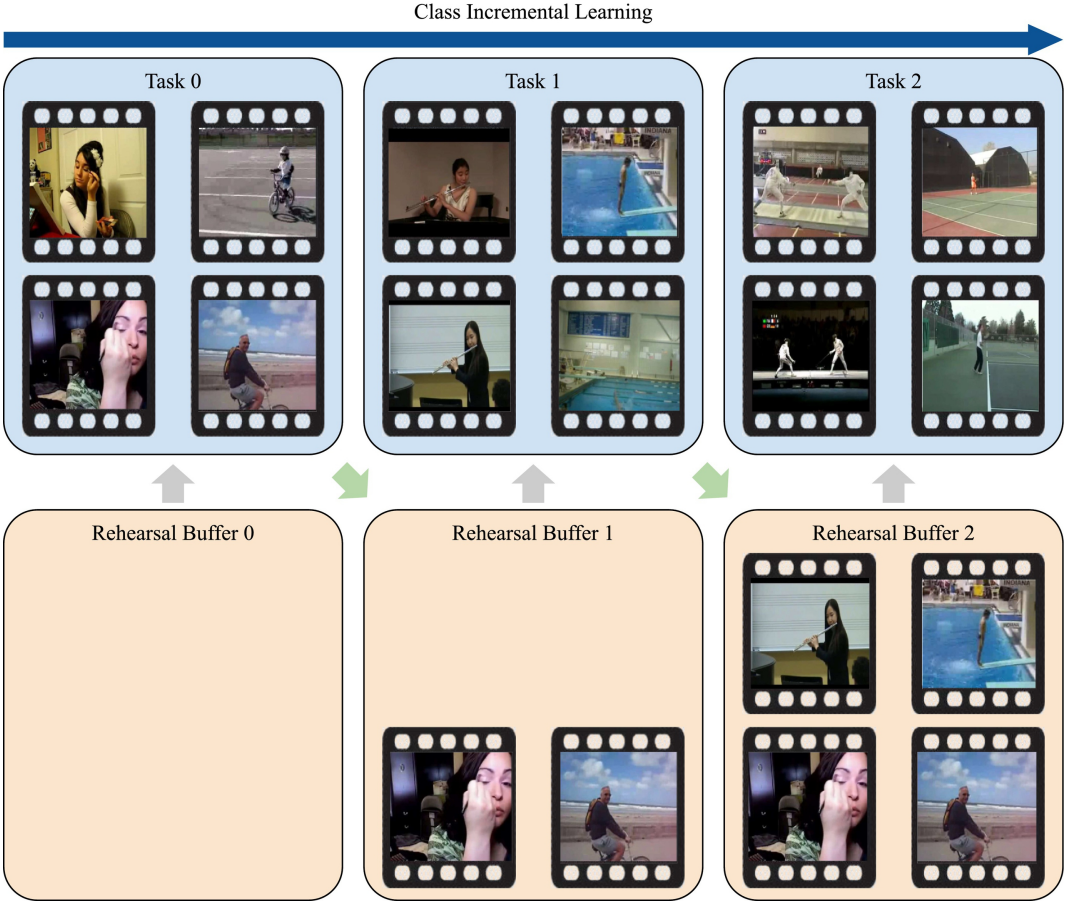

Class Incremental Learning
Task 0
Task 1
Task 2
Rehearsal Buffer 0
Rehearsal Buffer 1
Rehearsal Buffer 2

Figure 1


Backgrounds
Original video clips
Background augmented clips

Figure 2

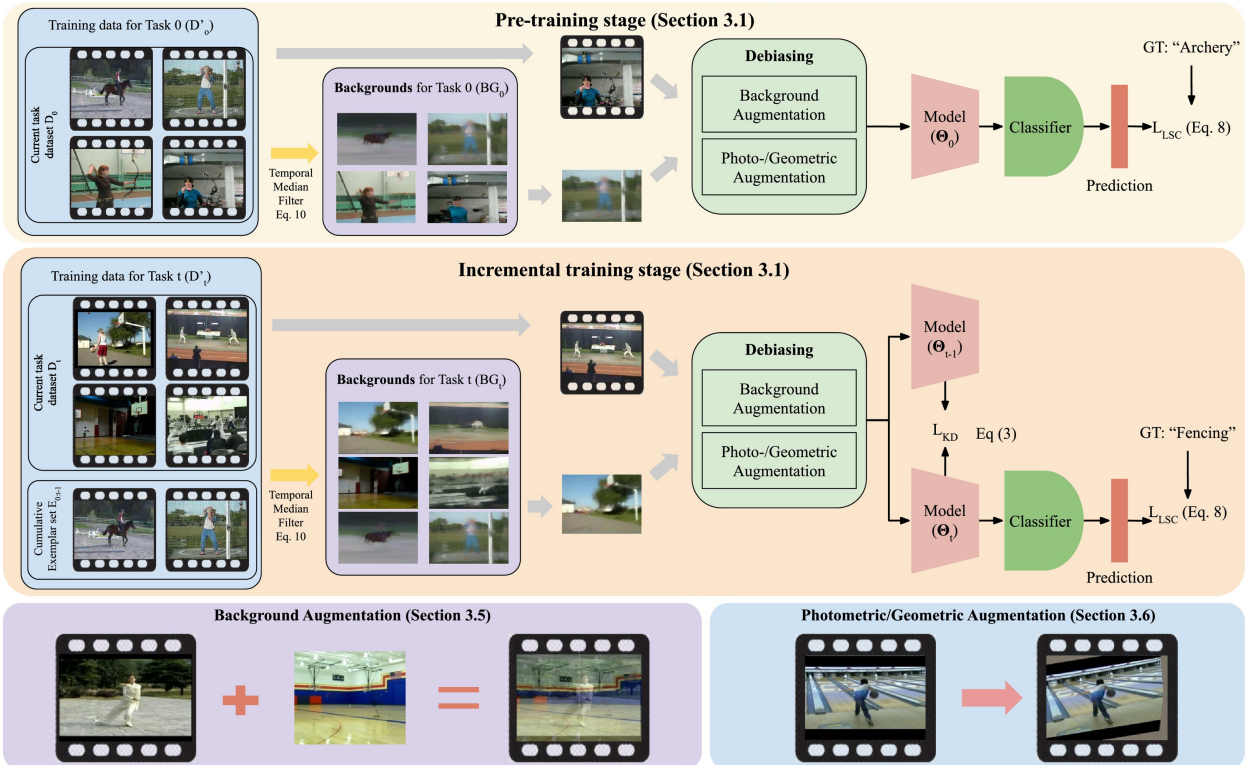

Pre-training stage (Section 3.1)
Training data for Task 0 (D'0)
Current task dataset D0
Backgrounds for Task 0 (BG0)
Temporal Median Filter Eq. 10
Debiasing
Background Augmentation
Photo-/Geometric Augmentation
Model (Θ0)
Classifier
GT: "Archery"
LLSC (Eq. 8)
Prediction
Incremental training stage (Section 3.1)
Training data for Task t (D't)
Current task dataset Dt
Cumulative Exemplar set Et-1
Backgrounds for Task t (BGt)
Temporal Median Filter Eq. 10
Debiasing
Background Augmentation
Photo-/Geometric Augmentation
Model (Θt-1)
LKD Eq (3)
Model (Θt)
Classifier
GT: "Fencing"
LLSC (Eq. 8)
Prediction
Background Augmentation (Section 3.5)
Photometric/Geometric Augmentation (Section 3.6)

Figure 3

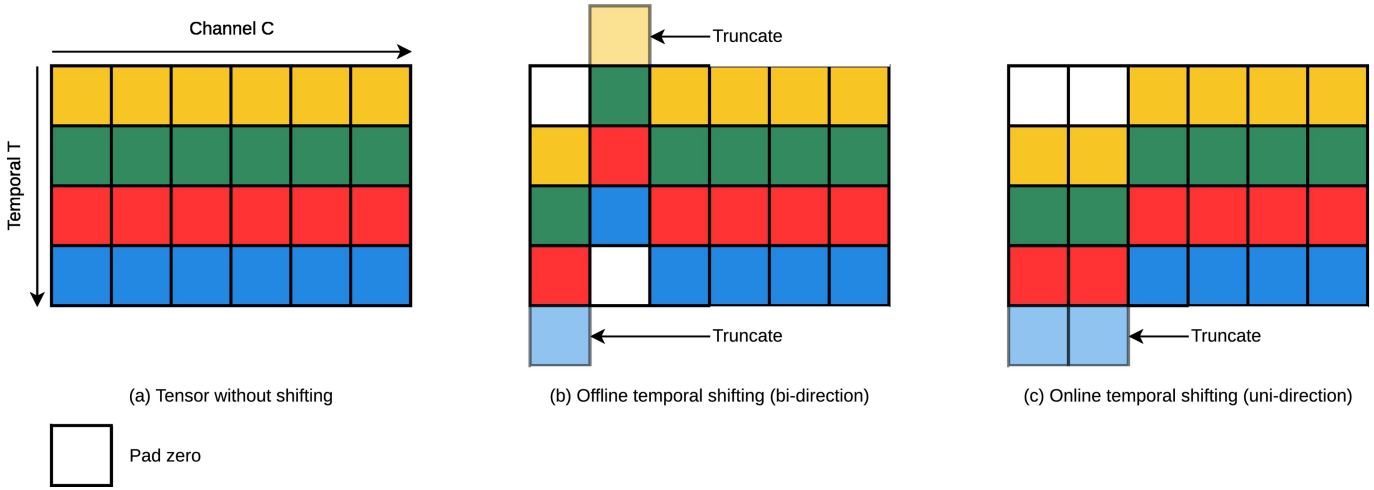

Channel C
Temporal T
Truncate
Truncate
Truncate
(a) Tensor without shifting
(b) Offline temporal shifting (bi-direction)
(c) Online temporal shifting (uni-direction)
Pad zero

Figure 4

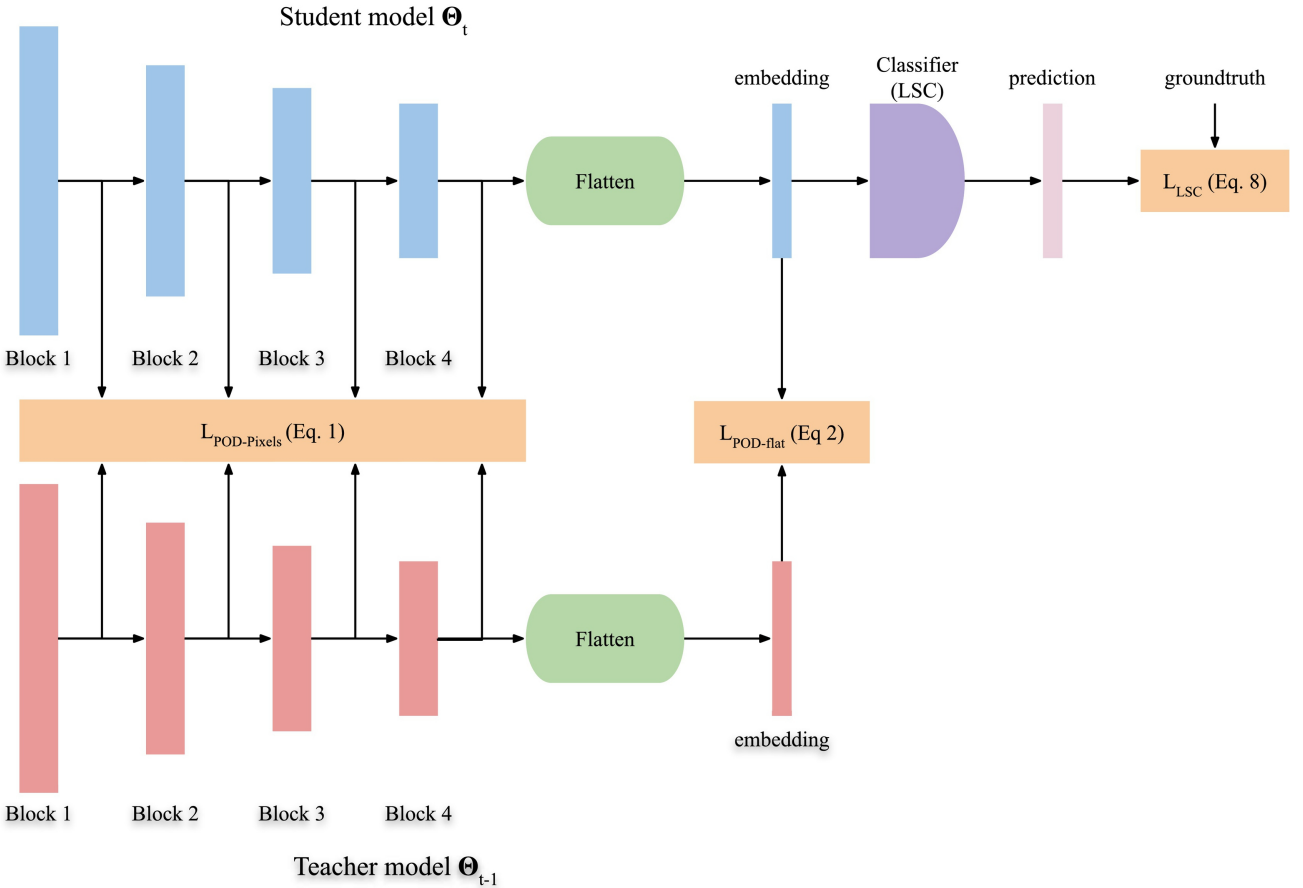
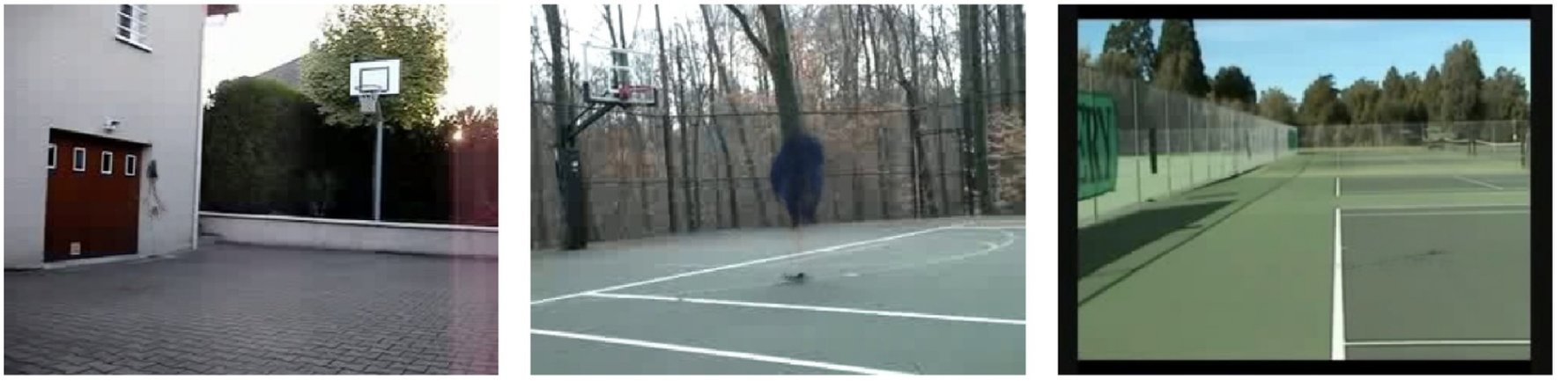

Student model Θ_t
embedding
Classifier (LSC)
prediction
groundtruth
Flatten
L_{LSC} (Eq. 8)
Block 1
Block 2
Block 3
Block 4
$L_{POD\text{-}Pixels}$ (Eq. 1)
$L_{POD\text{-}flat}$ (Eq 2)
Flatten
embedding
Block 1
Block 2
Block 3
Block 4
Teacher model Θ_{t-1}

Figure 5

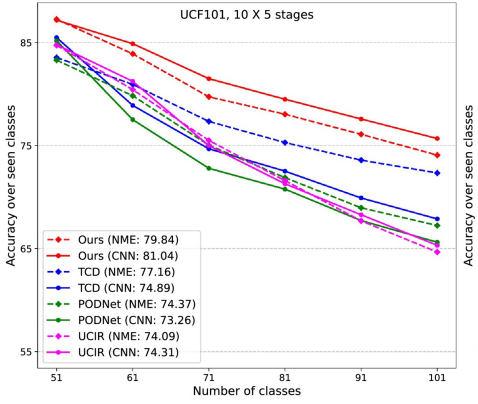
(a) *actor-and-scene* background.



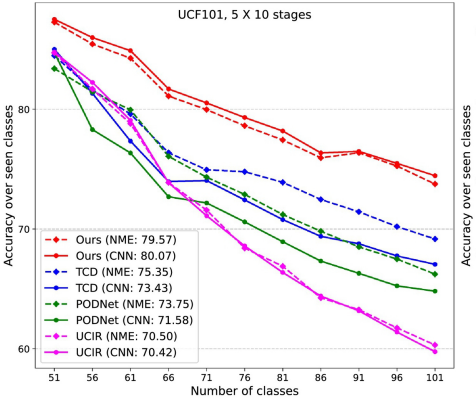
(b) *scene-only* background.

(c) *high-motion* background.

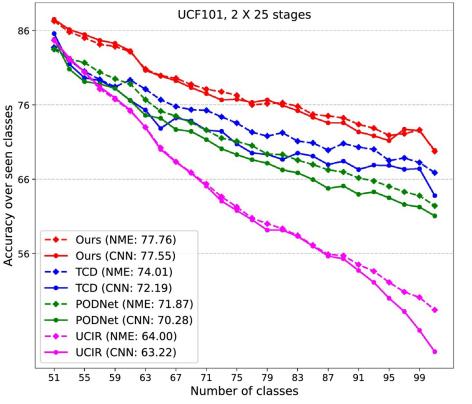
Figure 6



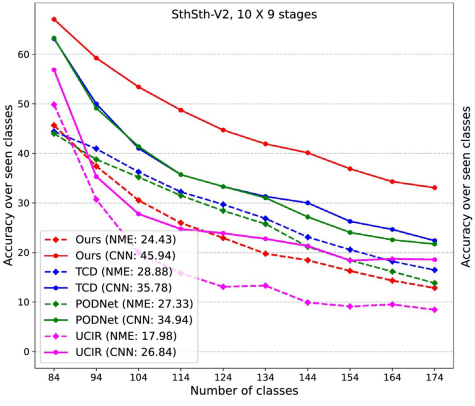
(a) UCF-101 (10 × 5)



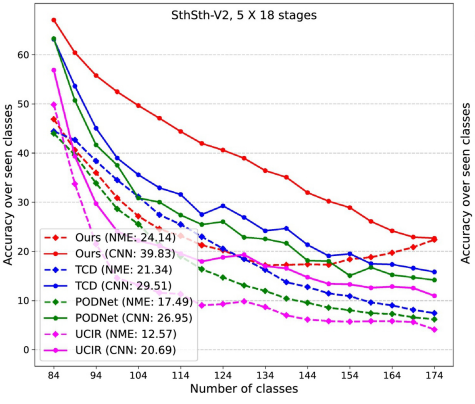
(b) UCF-101 (5 × 10)



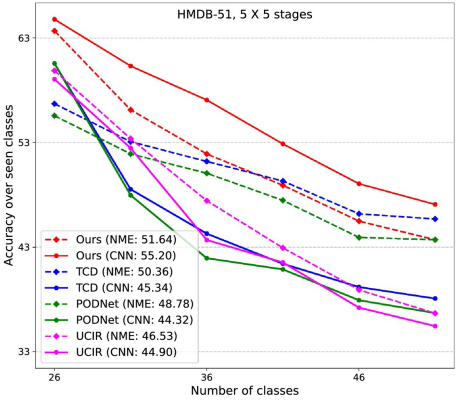
(c) UCF-101 (2 × 25)



(d) Something-Something-v2 (10 × 9)



(e) Something-Something-v2 (5 × 18)



(f) HMDB-51 (5 × 5)

Figure 7

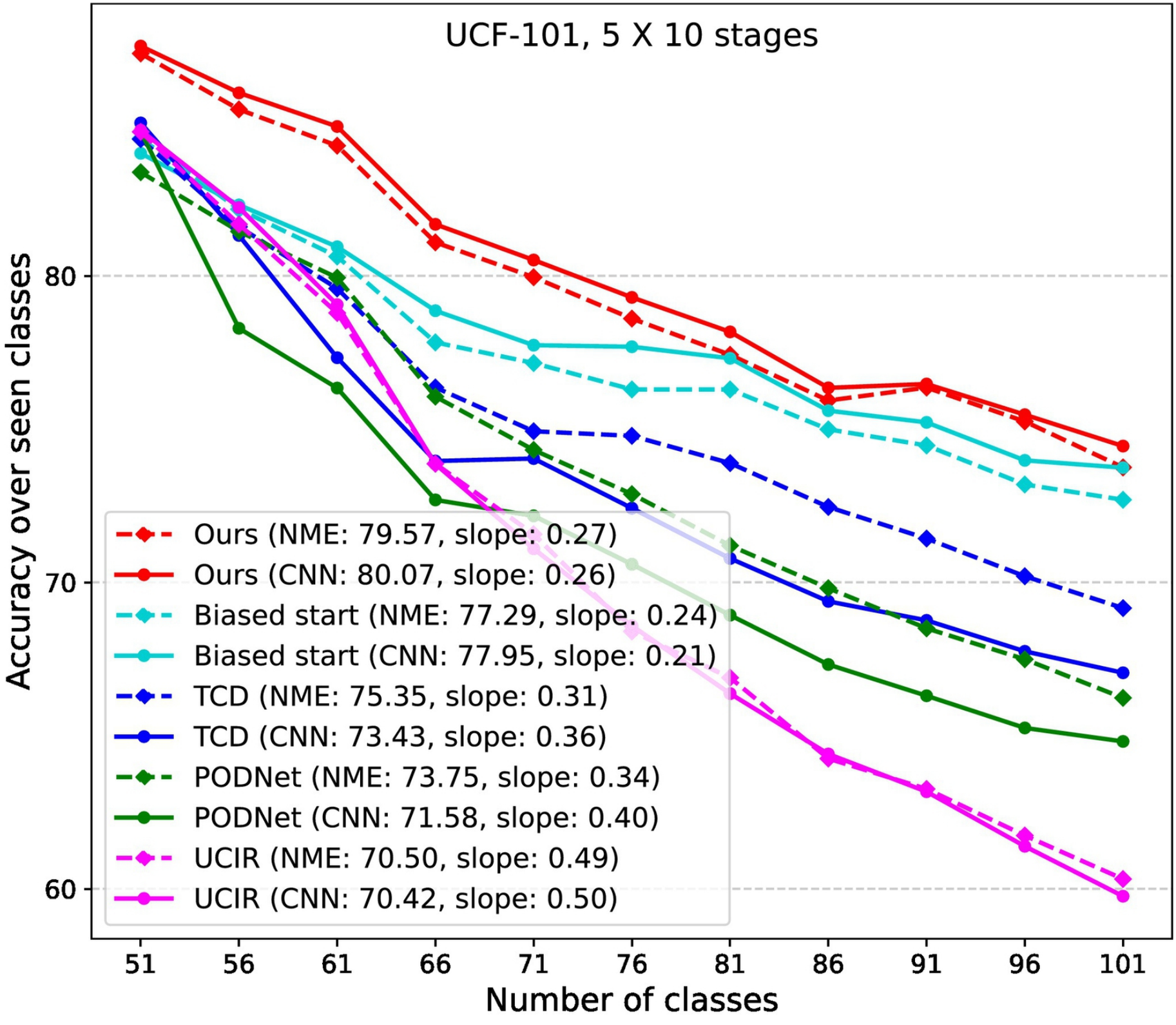
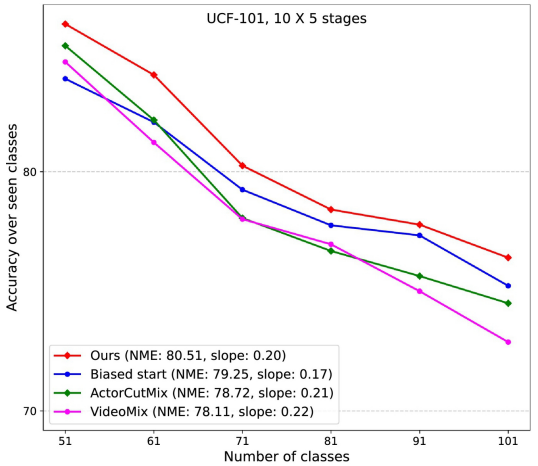
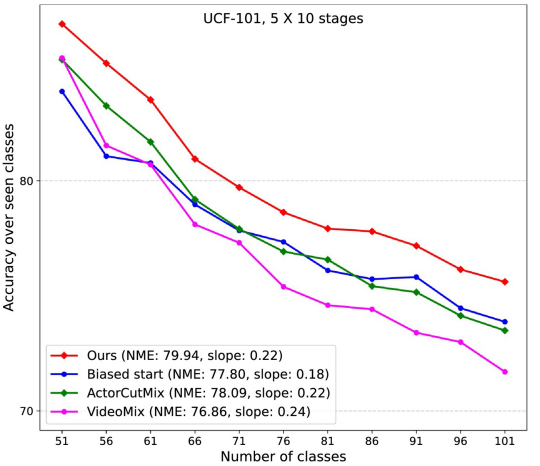

UCF-101, 5 X 10 stages
Accuracy over seen classes
80
70
60
51
56
61
66
71
76
81
86
91
96
101
Number of classes
Ours (NME: 79.57, slope: 0.27)
Ours (CNN: 80.07, slope: 0.26)
Biased start (NME: 77.29, slope: 0.24)
Biased start (CNN: 77.95, slope: 0.21)
TCD (NME: 75.35, slope: 0.31)
TCD (CNN: 73.43, slope: 0.36)
PODNet (NME: 73.75, slope: 0.34)
PODNet (CNN: 71.58, slope: 0.40)
UCIR (NME: 70.50, slope: 0.49)
UCIR (CNN: 70.42, slope: 0.50)

Figure 8

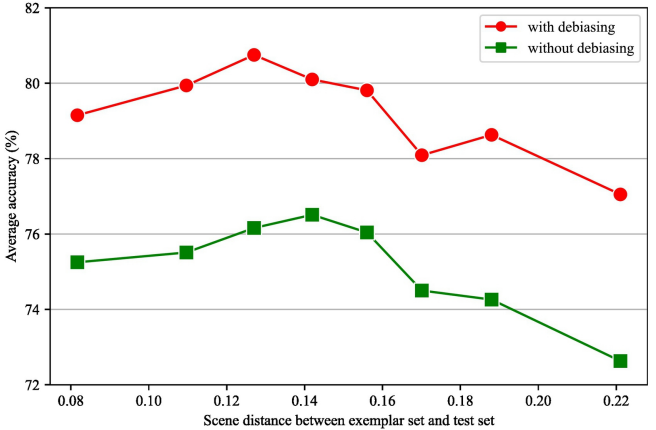


(a) 10×5 stages



(b) 5×10 stages

Figure 9


with debiasing
without debiasing
Average accuracy (%)
Scene distance between exemplar set and test set
82
80
78
76
74
72
0.08
0.10
0.12
0.14
0.16
0.18
0.20
0.22

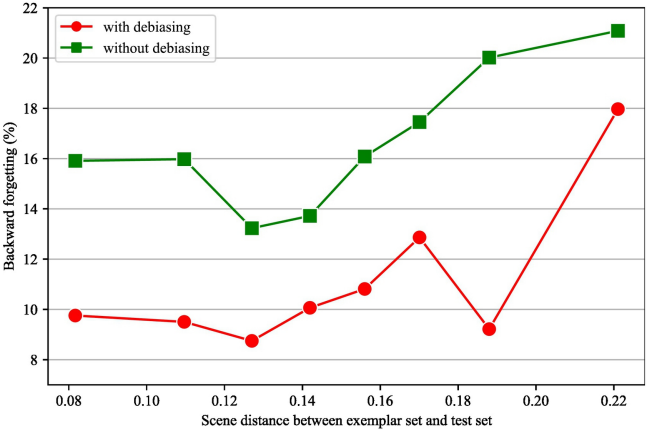

with debiasing
without debiasing
Backward forgetting (%)
Scene distance between exemplar set and test set
22
20
18
16
14
12
10
8
0.08
0.10
0.12
0.14
0.16
0.18
0.20
0.22

Figure 10

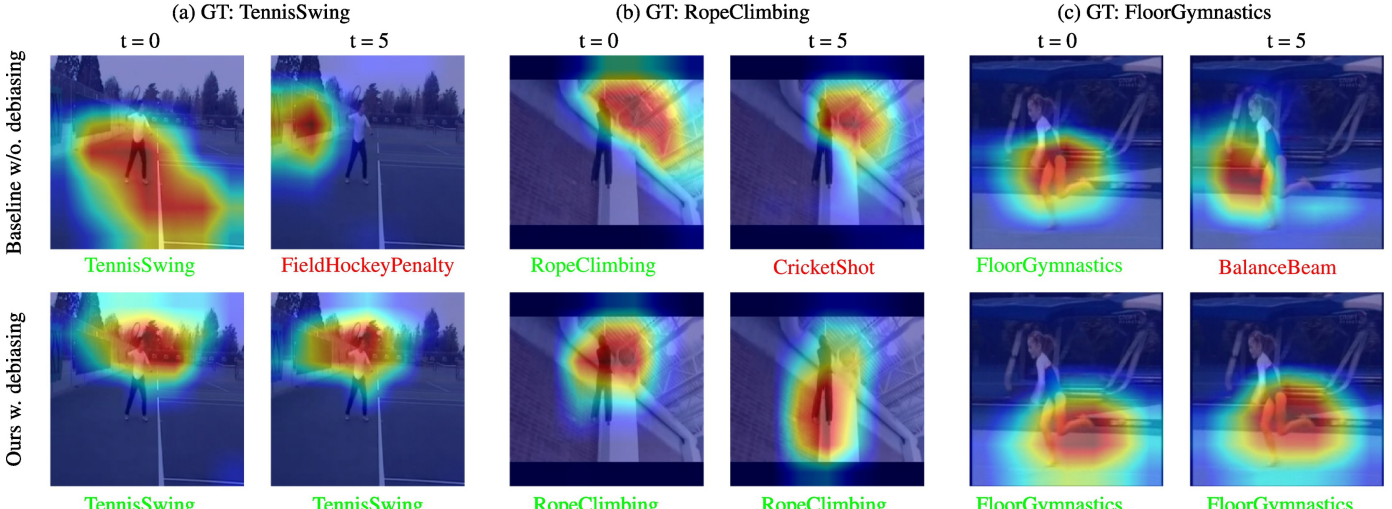

(a) GT: TennisSwing
(b) GT: RopeClimbing
(c) GT: FloorGymnastics
t = 0
t = 5
t = 0
t = 5
t = 0
t = 5
Baseline w/o. debiasing
TennisSwing
FieldHockeyPenalty
RopeClimbing
CricketShot
FloorGymnastics
BalanceBeam
Ours w. debiasing
TennisSwing
TennisSwing
RopeClimbing
RopeClimbing
FloorGymnastics
FloorGymnastics

Figure 11