Hypercorrelation Evolution for Video Class-Incremental Learning

Sen Liang[1], **Kai Zhu**[1*], **Wei Zhai**[1*], **Zhiheng Liu**[1], **Yang Cao**[1,2]

[1]University of Science and Technology of China
[2]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
{liangsen@mail., zkzy@mail., wzhai056@, lzh990528@mail., forrest@}ustc.edu.cn

Abstract

Video class-incremental learning aims to recognize new actions while restricting the catastrophic forgetting of old ones, whose representative samples can only be saved in limited memory. Semantically variable subactions are susceptible to class confusion due to data imbalance. While existing methods address the problem by estimating and distilling the spatio-temporal knowledge, we further explores that the refinement of hierarchical correlations is crucial for the alignment of spatio-temporal features. To enhance the adaptability on evolved actions, we proposes a hierarchical aggregation strategy, in which hierarchical matching matrices are combined and jointly optimized to selectively store and retrieve relevant features from previous tasks. Meanwhile, a correlation refinement mechanism is presented to reinforce the bias on informative exemplars according to online hypercorrelation distribution. Experimental results demonstrate the effectiveness of the proposed method on three standard video class-incremental learning benchmarks, outperforming state-of-the-art methods. Code is available at: https://github.com/Lsen991031/HCE

Introduction

With the rapid advancement of deep learning research, models trained on batch data can achieve good recognition performance in known distributions. However, in real-world applications, there exists a more complex and variable semantic shift. This poses new requirements for the continuous updates of the representation and discrimination capabilities in existing recognition networks, which is called class-incremental learning (CIL).

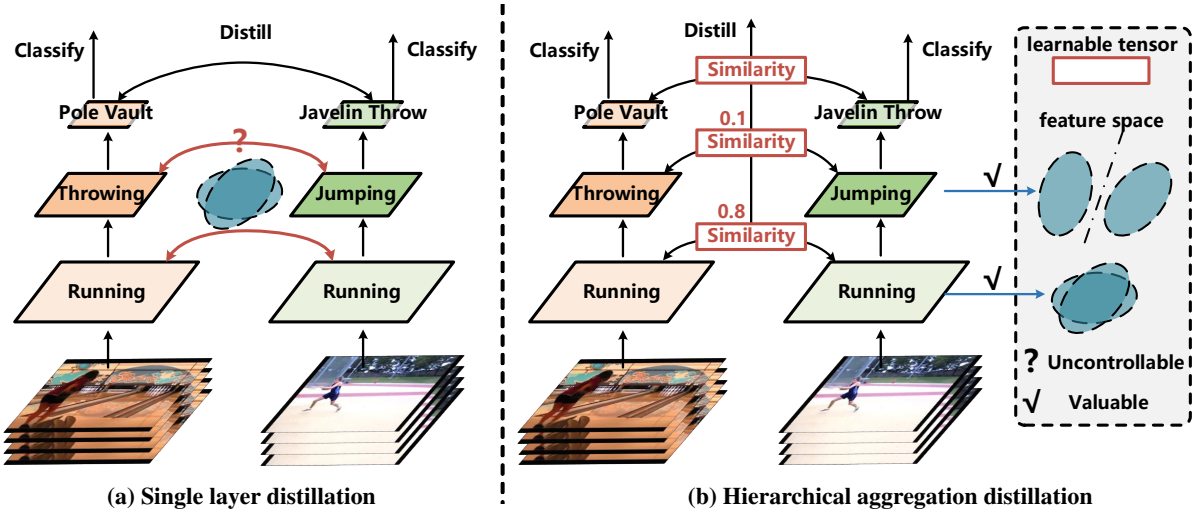
When encountering new tasks, it is often impossible to save a large amount of old class data due to equipment limitations or privacy and security concerns. In such cases, directly fine-tuning a well-trained model representation can also lead to representation and classifier biases towards the new classes due to the imbalance of new and old data, which is known as catastrophic forgetting (Dhar et al. 2019; Douillard et al. 2020). Existing class-incremental (Dhar et al. 2019; Douillard et al. 2020; Hou et al. 2019; Li and Hoiem 2017; Rebuffi et al. 2017; Yan, Xie, and He 2021; Zhai et al. 2023) methods have almost achieved the upper limit of performance in image classification tasks by distilling extracted features and expanding network structures to retain the representation ability on old classes while greatly enhancing the discriminability of new ones.

Recently, some studies (Park, Kang, and Han 2021; Pei et al. 2022; Villa et al. 2022; Douillard et al. 2020) focus on video class-incremental learning problem (VCIL) and find that directly introducing classical image class-incremental methods to action recognition is suboptimal in terms of maintaining performance on old classes due to the complexity of the multi-frame information of videos compared to images. Our further analysis shows that this is mainly attributed to multi-level feature confusion during the incremental process. As shown in Figure 1, video actions are often composed of multiple sub-actions involving different semantic levels, aligning them directly on the final layer feature will result in uncontrollable knowledge conflicts in the middle levels. In contrast, explicitly aggregating different levels of matching information can adaptively promote the alignment or discrimination of action-specific features. So the key question is how to ensure that feature combinations from different levels are reasonably involved in the distillation process, which is conducive to maintaining complex feature relationships.

Besides, the redundancy fluctuation among different video segments introduces interference in evaluating the learning imbalance between old and new class data. Many works (Douillard et al. 2020; Liu, Schiele, and Sun 2021a,b; Yan, Xie, and He 2021) in image class-incremental learning show that compensating for the optimization bias by the number ratio of old and new samples can effectively alleviate the learning imbalance caused by exemplars, especially in cases where storage space is limited. However, when applying the same strategy to VCIL, we found that it had little effect, indicating that the redundancy in video frames makes it difficult to measure the exact bias level with a simple number ratio among classes during the incremental process.

Motivated by the above analysis, we aim to improve VCIL performance by increasing the combination flexibility of the aligned features and adaptively adjusting the imbalance degree between old and new classes. Our proposed **H**yper**C**orrelation **E**volution (**HCE**) scheme is mainly manifested in two aspects. First, we present the **H**ierarchical

*Corresponding Author



(a) Single layer distillation **(b) Hierarchical aggregation distillation**

Figure 1: Motivation of the proposed method. Due to the heterogeneous distribution of different subaction semantics, the single-layer (*e.g.*, the last layer in most cases) features are valuable for classification but not for distillation in VCIL. **(a)** Constraining only the highest-level semantic (*e.g.*, pole vault vs javelin throw) alignment causes uncontrollable optimization direction of spatio-temporal features (*e.g.*, throwing vs jumping), which aggravates the possibility of confusion and forgetting between old and new actions. **(b)** Adaptively combining and aggregating hierarchical spatio-temporal features facilitates the retention of similar features (*e.g.*, running) and the discrimination of conflicting features (*e.g.*, throwing vs jumping).

Aggregation **S**trategy (**HAS**) to promote the multi-semantic knowledge retention by reorganizing and jointly optimizing the level-wise similarity matrix. Specifically, a sparse 4D convolution operator is utilized to enhance both the inter-level and intra-level interactions of the similarity relationship. Secondly, we present the **C**orrelation **R**efinement **M**echanism (**CRM**) to reinforce the knowledge imbalance by weighting the classification loss with class-specific hypercorrelation variance ratio. To demonstrate the superiority of our method, we conducted comparative experiments with both image and video class-incremental methods on three standard benchmarks UCF101, HMDB51, and Something-Something V2. We achieved the best results against the state-of-the-art methods, leading by **2%**, **2%**, and **3%**, respectively. Our main contributions are as follows:

- A hierarchical aggregation strategy is proposed for video class-incremental learning, in which the frame-efficient feature preservation is accomplished by a hierarchical distillation strategy, resulting in a plastic action representation with multi-semantic knowledge.
- A correlation refinement mechanism is presented, which calculates the variance of different hypercorrelation combination coefficients to adaptively re-weight the class bias and reinforce the knowledge imbalance.
- Extensive experiments on UCF101, HMDB51, and Something-Something V2 datasets demonstrate the superiority of our proposed method over the SOTA.

Related Works

Class-Incremental Learning

Class-Incremental Learning (CIL) involves the task of learning new classes incrementally while preserving the previously learned knowledge (Zhu et al. 2021, 2022). There are three main methods to tackle CIL: (1) Rehearsal-based methods store representative samples or network features of old classes or use GANs to generate old ones for training new tasks. iCaRL (Rebuffi et al. 2017) uses a small number of representative samples that approximate the class centroid for new task training. The feature-based playback method has the problem of feature drift. SDC (Yu et al. 2020) uses feature adaptation to solve the problem of feature drift. The GAN-based method (Ostapenko et al. 2019; Shin et al. 2017) uses generative adversarial networks (GANs) (Mirza et al. 2014; Odena, Olah, and Shlens 2017) to generate samples of old tasks. (2) Knowledge distillation (Hinton, Vinyals, and Dean 2015; Romero et al. 2014; Zagoruyko and Komodakis 2016) was first used in image classification tasks and quickly applied in other fields. Knowledge distillation guides students to realize knowledge transfer through the teacher network. How to set up a better distillation method is the focus of this kind of method improvement. LwF (Dhar et al. 2019) has created a precedent of applying knowledge distillation to incremental learning to solve catastrophic forgetting. iCarl combines knowledge distillation and rehearsal, which calculates distillation loss according to network prediction. POD-Net (Douillard et al. 2020) applies an efficient spatial distillation loss to the whole model, which significantly alleviates catastrophic forgetting. (3) The method based on model architecture continuously modifies the network structure with incremental learning. For example, add a new model structure for incremental tasks so that the old model weight can maintain the old task characteristics. At the same time, the new model structure adapts to incremental tasks to achieve the goal of adapting to both old and new tasks. DER (Chen, Zhang, and Qin 2019) improves the model performance by

expanding the model structure so that the old model structure is conducive to maintaining the performance of the original task, and the new model structure can adapt to the performance of the new incremental task.

To address the challenging VCIL task, TCD (Park, Kang, and Han 2021) involves time-channel importance maps in the knowledge distillation process. Different from their static relation calculated from the corresponding gradient information, we utilize the learnable network to explore the joint effect of hierarchical correlation.

Action Recognition

With advancements in CNNs, techniques (Carreira and Zisserman 2017; Karpathy et al. 2014; Lin, Gan, and Han 2019; Tran et al. 2015; Wang et al. 2016) utilized for video action recognition have evolved and can be broadly categorized into two methods: 2D CNN and 3D CNN.

Some methods utilize conventional 2D Convolutional Neural Networks (CNNs), while Simonyan et al. (Feichtenhofer, Pinz, and Zisserman 2016) employ a novel two-stream network that combines the characteristics of RGB and optical flow to enhance recognition accuracy. Additionally, Lin et al. (Lin, Gan, and Han 2019) propose the Temporal Shift Module (TSM), which allows for information exchange between adjacent frames by shifting part of the channels along the time dimension. This innovation is integrated into a standard 2D neural network, providing improved time modeling with zero added computational cost or extra parameters.

3D CNN has the ability to learn both spatial and temporal features simultaneously. I3D (Carreira and Zisserman 2017) inflates the weights of a pre-trained 2D model from ImageNet to the corresponding weights in a 3D model and trains it further on Kinetics400 dataset, achieving the highest classification accuracy. However, 3D CNN are computationally intensive and more prone to overfitting due to their larger parameters and more complex architecture.

To mitigate the issue mentioned above, many works use 2D CNN and 3D CNN (Wang et al. 2018; Xie et al. 2018; Zhou et al. 2018; Zolfaghari, Singh, and Brox 2018) in the network. Due to the particularity of temporal modeling in action recognition, some works have divided the 3D kernel into 2D spatial convolution and 1D temporal convolution (Qiu, Yao, and Mei 2017; Tran et al. 2018; Xie et al. 2018). More recent studies have sought to enhance temporal modeling by incorporating additional modules beyond simply using 1D temporal convolution. Of course, many other methods exist, such as applying group (Tran et al. 2019) convolution and learning 3D shift operation (Fan et al. 2020).

Method

Problem Formulation

The purpose of Class Incremental Learning is to train a model parameterized by $\mathbf{\Theta}$ step-by-step with a given set of tasks $\{\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_k, \cdots\}$. We set $\mathcal{D}_k$ to be a pre-specified dataset which not encountered in previous tasks for task $\mathcal{T}_k$, with labels belonging to a pre-defined label set λ_k where $(\lambda_1 \cup \cdots \cup \lambda_{k-1}) \cap \lambda_k = \emptyset$. To alleviate catastrophic forgetting, a representative subset of exemplars, $\mathcal{E}_k$, is preserved for future tasks. $\mathcal{E}_k$ is selected at the end of each task $\mathcal{T}_k$ and is used for training as each subsequent task progresses. In each incremental step $\mathcal{T}_k$, we use the dataset $\mathcal{D}_k^{'} = \mathcal{D}_k + \mathcal{E}_{k-1}$ for model training and evaluate the performance of the model on all seen classes.

Overview

As shown in Figure 2, the overall framework of our proposed HCE follows the standard protocol of video class-incremental methods (Park, Kang, and Han 2021; Villa et al. 2022), which is mainly based on rehearsal strategy and knowledge distillation. Specifically, We input the videos into both the new and old backbone networks to obtain the intermediate feature pairs of the new and old models. Then these intermediate feature pairs are fed into the HAS, shown in Figure 3 to obtain the hypercorrelation, which contains multi-semantic information.

In addition, we use the variance of different hypercorrelation combination coefficients generated by $\mathcal{E}_{k-1}$ and $\mathcal{D}_k$ to reinforce the knowledge imbalance. Our method offers practicality and resilience in updating models, presenting fresh perspectives for utilization in domains like video class-incremental learning.

Hierarchical Aggregation Strategy

The hierarchical distillation strategy we proposed promotes the retention of multi-semantic knowledge by reorganizing and jointly optimizing the layer-wise matching matrices.

The ResNet is comprised of fundamental components known as residual blocks, with each individual residual block referred to as a layer. So the backbone consists of L layers. Simultaneously, the residual blocks within ResNet generate feature maps e_l with varying shapes. We aggregate the residual blocks that produce feature maps of the same shape into distinct groups $\mathcal{P}_s$. Ultimately, all the residual blocks are categorized into S groups.

We feed the videos $\mathcal{D}_k^{'}$ into both the old and new backbone networks to extract all intermediate feature maps denoted by e_l^{k-1} and e_l^k, which are called layer-wise matching matrices, where l ranges from 1 to L. The layer-wise matching matrices have dimensions of $\mathbb{R}^{T \times C_l \times H_l \times W_l}$, with T representing the temporal dimension of a video, while C_l represents the number of channels in the l layer of the backbone. Then we compute the cosine similarity $\hat{\mathbf{C}}_l$ between each pair of layer-wise matching matrices, and the correlation vector is calculated as:

$$\hat{\mathbf{C}}_l = \text{ReLU}\left(\frac{e_l^k \times e_l^{k-1}}{\|e_l^k\| \times \|e_l^{k-1}\|}\right). \tag{1}$$

ReLU is used to suppress noise in the correlation. $\hat{\mathbf{C}}_l \in \mathbb{R}^{T \times H_l \times W_l \times H_l \times W_l}$ with the same spatial size are collected into $\{\hat{\mathbf{C}}_l\}_{l \in \mathcal{P}_s}$, which are then concatenated, resulting in $C_s \in \mathbb{R}^{T \times |\mathcal{P}_s| \times H_s \times W_s \times H_s \times W_s}$. After the concatenation, we can obtain the multilevel similarity $\mathcal{C} = \{C_s\}_{s=1}^{S}$.

Due to the multilevel similarity in $\mathcal{C}$ only calculates the correlation within a single level, we perform a 4D convolution operation on $\mathcal{C}$ to link the correlation of all levels, which

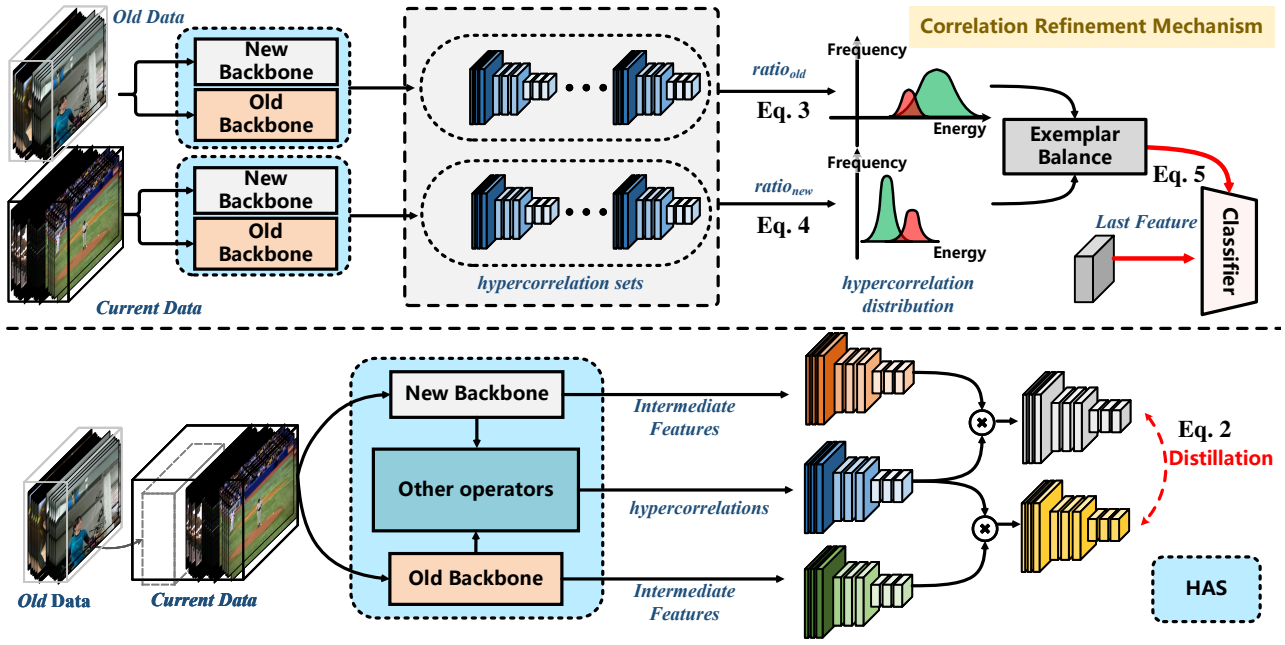


Figure 2: The overall framework of our scheme. The hypercorrelation evolution scheme combines multi-level distillation and correlation refinement mechanism to improve model performance. At each incremental step k, the correlation refinement mechanism calculates the class-specific hypercorrelation variance weights using $\mathcal{E}_{k-1}$ and $\mathcal{D}_k$ through the HAS under the current task, and then applies them to the cross-entropy loss function to help the model learn correlations better. For $\mathcal{D}_k^{'}$, we input it into the backbone network to generate new and old feature pairs and then input these feature pairs into HAS to obtain a multi-semantic knowledge hypercorrelation. Then this hypercorrelation is applied to the backbone feature layer for knowledge distillation to help the model learn more accurate features. Meanwhile, the backbone and HAS are updated simultaneously through knowledge distillation and cross-entropy loss function optimization.

reduces the last two dimensions of $\mathcal{C}$ with a large stride to a uniform size (H_e, W_e), $\mathbb{R}^{T\times|\mathcal{L}_s|\times H_s\times W_s\times H_s\times W_s} \rightarrow \mathbb{R}^{T\times 128\times H_s\times W_s\times H_e\times W_e}$, where $Hs > He$ and $Ws > We$. Then, we perform upsampling operations on the third and fourth dimensions of $\mathcal{C}$ to enlarge them to the largest (H_s, W_s) in $\mathcal{C}$, as shown in Figure 3. This enables us to aggregate all components of $\mathcal{C}$ to derive $\mathcal{C}'$. We use multiple 4D convolutions with a stride of 1 on $\mathcal{C}^{'}$ to enhance its multi-semantic knowledge while preserving its shape. Finally, we perform an avgpool operation on the last two dimensions of $\mathcal{C}^{'}$ to obtain $Z \in \mathbb{R}^{T\times 128\times H_L\times W_L}$, which contains the relevant information of all levels.

To harness the multi-level semantic information embedded in Z to aid in model training, we devise a decoder, which consists of a series of 2D convolutions, ReLU, and batch normalization as shown in Figure 3. Specifically, in the last two convolutional layers of the decoder, we employ different C_l to extract hypercorrelation $H^l \in \mathbb{R}^{T\times C_l}$ of varying scales. After the decoder, we introduce an expansion scalar as a hyperparameter to adapt more quickly to the relationships between different levels. Finally, hypercorrelation is used to optimize the backbone network through knowledge distillation, with a distillation loss of:

$$\mathcal{L}_{hc} = \sum_{l=1}^{L}\sum_{t=1}^{T}\sum_{c=1}^{C_l} H^l \| e_l^{k,t,c} - e_l^{k-1,t,c} \|_F^2, \qquad (2)$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The $e_l^{k,t,c}$ and $e_l^{k-1,t,c}$ refer to the feature maps extracted from the intermediate layer of the backbone network. H^l denotes the hypercorrelation corresponding to feature maps from l layer.

Correlation Refinement Mechanism

The existing research (Pei et al. 2022) indicates that the interference caused by redundancy fluctuation among different video segments is a significant challenge when evaluating the knowledge imbalance between old and new classes. To tackle this issue, we propose a novel correlation refinement strategy that leverages class-specific hypercorrelation variance weights to reinforce the knowledge imbalance.

During task k, we have two parts of datasets $\mathcal{E}_{k-1}$ and $\mathcal{D}_k$, and subsequently input them separately into the HAS model. This process yields two hypercorrelation sets $\{H(x)\}_{x\in\mathcal{D}_k}$ and $\{H(x)\}_{x\in\mathcal{E}_{k-1}}$, and the variances of these hypercorrelation sets are computed individually. Then, we use the variance of $\{H(x)\}_{x\in\mathcal{E}_{k-1}}$ to calculate the ratio of all classes in the old task and the variance of $\{H(x)\}_{x\in\mathcal{D}_k}$ to calculate the ratio of all classes in the new task. These ratios can balance the impact of different classes during model training and can be calculated using the following formula:

$$ratio_{old} = \log\left[\sum_{x\in\mathcal{E}_{k-1}} (H(x) - \bar{H}_{old})^2\right], \qquad (3)$$

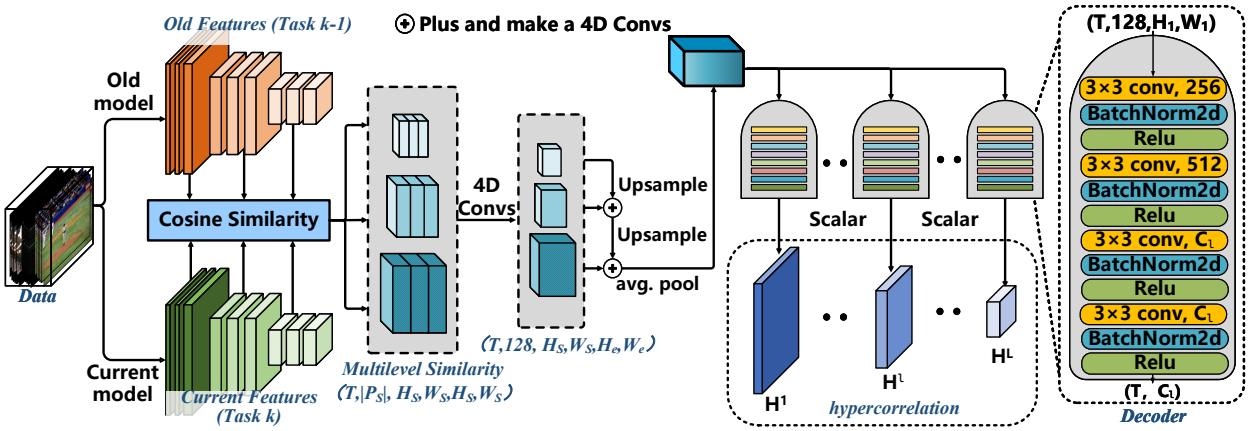


Figure 3: The overall architecture of Hierarchical Aggregation Strategy. It consists of three components: obtaining intermediate feature pairs, utilizing 4D convolution for knowledge blending, and decoding the multi-level semantic information.

$$ratio_{new} = \log\left[\sum_{y\in\mathcal{D}_k}(H(y)-\bar{H}_{new})^2\right], \tag{4}$$

where $\bar{H}_{old}$ and $\bar{H}_{new}$ are the averages of $\{H(x)\}_{x\in\mathcal{E}_{k-1}}$ and $\{H(x)\}_{x\in\mathcal{D}_k}$. Next, we set the ratios of all classes in the $\mathcal{E}_{k-1}$ dataset to $ratio_{old}$ to obtain $\mathbf{R}_{\mathcal{E}_{k-1}}$ and set the ratios of all classes in the $\mathcal{D}_k$ dataset to $ratio_{new}$ to obtain $\mathbf{R}_{\mathcal{D}_k}$. Subsequently, we concatenate two lists $\mathbf{R}_{\mathcal{E}_{k-1}}$ and $\mathbf{R}_{\mathcal{D}_k}$ to obtain the ratios of all classes $\mathbf{R}_{\mathcal{D}'_k}$. To reduce the impact of the imbalance between old and new classes, we apply this ratio to the cross-entropy loss function:

$$\mathcal{L}_{ce}^{'} = CrossEntropy(p+\mathbf{R}_{\mathcal{D}'_k}, t), \tag{5}$$

where p represents the predicted logits and t represents the data label $(\lambda_1 \cup \cdots \cup \lambda_k)$.

Training Objective

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

$$\mathcal{L}_{final}^k = \mathcal{L}_{ce}^{k\,'} + \mathcal{L}_{kd}^k + \mathcal{L}_{hc}^k, \tag{6}$$

where $\mathcal{L}_{kd}^k$ refers to the process of performing distillation on the last layer of the backbone to ensure that the model retains sufficient classification information. $\mathcal{L}_{ce}^{k\,'}$ represents the cross-entropy loss function optimized and adjusted by the correlation refinement strategy to reinforce the knowledge imbalance between old and new classes. $\mathcal{L}_{hc}^k$ refers to the backbone distillation loss applied at multiple levels in the hierarchical distillation strategy. In this way, the joint constrain with $\mathcal{L}_{ce}^{k\,'}$ and $\mathcal{L}_{hc}^k$ optimizes H towards a direction that favors both differentiation and promotes the multi-semantic knowledge retention. The comprehensive application of these techniques enables our model to perform better in handling new data, with strong generalization ability.

Experiments

Datasets

We conduct a comprehensive evaluation of the proposed HCE on three widely-used action recognition datasets: UCF101 (Soomro, Zamir, and Shah 2012), HMDB51 (Kuehne et al. 2011), and Something-Something V2 (Goyal et al. 2017). The UCF101 dataset comprises 13.3K videos belonging to 101 classes, while the HMDB51 dataset contains 6.8K videos from 51 action classes collected from various online sources. Something-Something V2 is a large-scale dataset consisting of 220K videos from 174 different action classes, which is designed to challenge the temporal reasoning capabilities of models.

Evaluation Protocol

To evaluate the performance of our VCIL method, we adopt different training strategies for each dataset. Specifically, for UCF101, we first train the model on 51 classes and then divide the remaining 50 classes into 5, 10, and 25 tasks, respectively. For HMDB51, we train the base model using videos from 26 classes and then separate the remaining 25 classes into 5 or 25 groups. For Something-Something V2, we first train on 84 classes in the initial stage and then generate groups of 10 and 5 classes.

Comparison with SOTA

To better assess the overall performance, we compare our method to the SOTA of VCIL, including LwFMC (Li and Hoiem 2017), LwM (Dhar et al. 2019), iCaRL (Rebuffi et al. 2017), UCIR (Hou et al. 2019), PODNet (Douillard et al. 2020) and TCD (Park, Kang, and Han 2021). Meanwhile, we use the same storage size for each class and model architecture to ensure the fairness of the experiments.

Table 1 and Table 2 show the overall results of our method and other baselines on the HMDB51, Something-Something V2, and UCF101 datasets. Obviously, our method consistently outperforms all related methods across all experimental settings. Specifically, we achieve significant improvements over the FrameMaker method, with growth rates of around 2%, 3%, and 2% on the HMDB51, Something-SomethingV2, and UCF101 datasets. These results demonstrate the effectiveness of HCE in retaining multi-semantic knowledge and reinforcing the knowledge imbalance.

	HMDB51				Something-Something V2			
Num. of classes	5 × 5 stages		1 × 25 stages		10 × 9 stages		5 × 18 stages	
Classifier	**CNN**	**NME**	**CNN**	**NME**	**CNN**	**NME**	**CNN**	**NME**
Fine-tuning	16.82	—	4.83	—	—	—	—	—
LwFMC (Li and Hoiem 2017; Rebuffi et al. 2017)	26.82	—	16.49	—	—	—	—	—
LwM (Dhar et al. 2019)	26.97	—	16.50	—	—	—	—	—
iCaRL (Rebuffi et al. 2017)	—	40.09	—	33.77	—	15.48	—	10.22
UCIR (Hou et al. 2019)	44.90	46.53	37.04	37.15	26.84	17.98	20.69	12.57
PODNet (Douillard et al. 2020)	44.32	48.78	38.76	46.62	34.94	27.33	26.95	17.49
TCD (Park, Kang, and Han 2021)	45.34	50.36	40.07	46.66	35.78	28.88	29.60	21.63
FrameMaker (Pei et al. 2022)	47.54	51.12	42.65	47.37	37.25	29.92	30.98	22.84
HCE (Ours)	**48.63**	**52.01**	**43.99**	**48.94**	**38.67**	**36.88**	**32.51**	**32.82**
Oracle (Upper Bound)	55.03	55.98	54.89	55.32	60.15	55.37	60.96	54.16

Table 1: Class-incremental action recognition performance on HMDB51 and Something-Something V2. HCE achieves the best performance in all experimental settings. We are unable to provide NME scores for methods that do not utilize exemplars. Additionally, iCaRL exclusively employs NME for classification purposes and does not utilize CNN. The bold-faced numbers indicate the best performance.

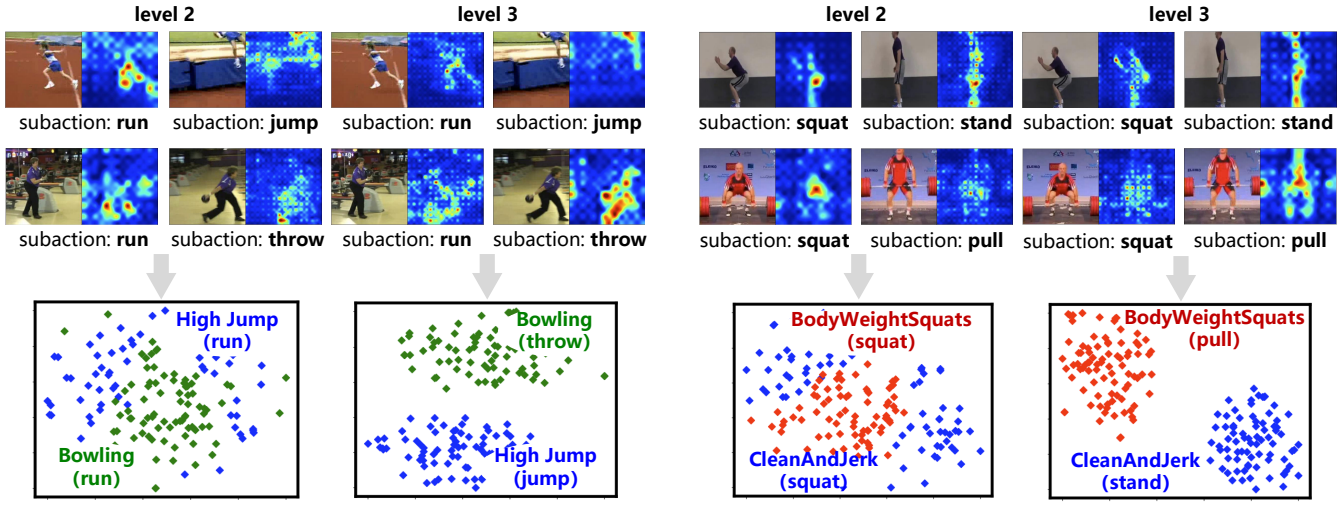


Figure 4: Analysis on Hierarchical Aggregation Strategy. Benefiting from the sensitivity of propsed HAS on local information(*e.g.*, gradcam in the upper subfigure), features related to effective sub-actions at different levels can be effectively distinguished.(*e.g.*, t-SNE in the lower)

The results of HCE and other baseline methods on the Something-SomethingV2 dataset show that HCE outperforms the other baselines on this large-scale dataset. Furthermore, we observed that the performance of TCD and FrameMaker on NME is not as good as on CNN due to the need for better representation quality. By utilizing HAS and CRM, our method preserves better representations and achieves better performance on NME.

Ablation Study and Analysis

Ablation Study To demonstrate the effectiveness of the hierarchical distillation strategy and the correlation refinement strategy on VCIL, we conduct the experiment for variant types of our objective function, $\mathcal{L}^k_{final}$. 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 $\mathcal{L}'^k_{\text{ce}} + \mathcal{L}^k_{\text{kd}} + \mathcal{L}^k_{\text{hc}}$ leads to the best performance.

Analysis on Hierarchical Aggregation Strategy To demonstrate the mechanisms of multi-level work, we conduct the experiment on the relationship between different levels and different sub-actions. Specifically, we use Grad-CAM (Selvaraju et al. 2017) to observe the attention of different sub-actions in the "bowling" and "high jump" classes for different levels of the backbone. From Figure 4, it can be seen that level 2 has greater attention to the sub-action "run" in both classes but cannot focus on the key sub-actions of "throw" in the "bowling" class and "jump" in the "high jump" class. In contrast, level 3 can focus well on the key sub-actions of "throw" in the "bowling" class and "jump" in the "high jump" class but cannot focus on "run." Furthermore, we use t-SNE (Van der Maaten and Hinton 2008) to observe the clustering effect of different levels of features for different videos. Significantly, we observe that level 3

Num. of classes	10×5 stages		5×10 stages		2×25 stages	
Classifier	**CNN**	**NME**	**CNN**	**NME**	**CNN**	**NME**
Fine-tuning	24.97	—	13.45	—	5.78	—
LwFMC	42.14	—	25.59	—	11.68	—
LwM	43.39	—	26.07	—	12.08	—
iCaRL	—	65.34	—	64.51	—	58.73
UCIR	74.31	74.09	70.42	70.50	63.22	64.00
PODNet	73.26	74.37	71.58	73.75	70.28	71.87
TCD	74.89	77.16	73.43	75.35	72.19	74.01
FrameMaker	78.13	78.64	76.38	78.14	75.77	77.49
HCE (Ours)	**79.12**	**80.01**	**77.59**	**78.81**	**75.84**	**77.62**
Oracle	84.15	83.37	83.96	83.20	83.82	83.16

Table 2: Class-incremental action recognition performance on UCF101. HCE achieves the best performance in all experimental settings.

Objective Function	**CNN**	**NME**
$\mathcal{L}_{\text{ce}}^{k} + \mathcal{L}_{\text{kd}}^{k}$	44.36	48.44
$\mathcal{L}_{\text{ce}}^{k} + \mathcal{L}_{\text{hc}}^{k}$	45.45	50.66
$\mathcal{L}_{\text{ce}}^{k} + \mathcal{L}_{\text{kd}}^{k} + \mathcal{L}_{\text{hc}}^{k}$	47.34	51.65
$\mathcal{L}_{\text{ce}}^{\prime k} + \mathcal{L}_{\text{kd}}^{k} + \mathcal{L}_{\text{hc}}^{k}$	**48.63**	**52.01**

Table 3: Ablations study on the objective function. We demonstrated the effectiveness of the hierarchical distillation strategy $\mathcal{L}_{\text{hc}}^{k}$ and correlation-related mechanism on the cross-entropy loss $\mathcal{L}_{\text{ce}}^{\prime k}$.

exhibits a strong clustering effect within these two classes, in contrast to level 2, which demonstrates a notably weaker clustering effect. This discrepancy arises from the fact that level 2 corresponds solely to the sub-action "run," whereas level 3 encompasses distinct sub-actions such as "jump" and "throw." Consequently, the inadequate clustering of level 2 can be attributed to its singular focus on "run," while the diverse sub-actions captured by level 3 contribute to its effective clustering performance. Similarly, we observed the same phenomenon in other classes with the same sub-actions, such as the "BodyWeightSquats" and "CleanAndJerk" classes shown in Figure 4. Based on the above analysis, we conclude that using only the last level or a single level of features is helpful for classification but not conducive to the preservation of video knowledge. The reason is that different levels correspond to different sub-action features, and driving only one level can easily lead to conflicts between preservation and discrimination. Therefore, we need to use a combination of multiple levels to preserve the same sub-action information and distinguish different sub-actions.

To demonstrate the advantages of multi-level integration, we compare single-level which uses distillation in the last level, multi-level (TCD) which can be regarded as an indirect way of applying information from multiple hierarchical levels, and HAS. Table 4 shows the semantic alignment method with multi-level and inter-layer interaction is superior to the single-level and non-interacting methods, demonstrating the validity of our method.

Analysis on Correlation Refinement Mechanism To demonstrate the effectiveness of CRM in compensating for optimization bias, we use CRM and the number ratio of old and new samples on the cross-entropy loss, respectively. As shown in Table 5, applying CRM to cross-entropy loss resulted in around a 2% and 1% improvement in CNN and NME, respectively, compared to using the number ratio of old and new samples. It can be seen that CRM can reinforce the knowledge imbalance by weighting the classification loss with the class-specific hypercorrelation variance ratio, thereby improving the final performance.

Classifier	**CNN**	**NME**
Single-level	44.86	48.62
Muti-level	45.34	50.36
HAS	**48.63**	**52.01**

Table 4: Analysis about different distillation methods on HMDB51 with 5 steps. The results show the superiority of the HAS with multi-level and inter-layer interaction.

Classifier	**CNN**	**NME**
Number ratio	46.88	51.08
CRM	**48.63**	**52.01**

Table 5: Analysis about the CRM on HMDB51 with 5 steps. The results indicate that applying CRM for $\mathcal{L}_{ce}^{k}{}^{\prime}$ is superior to using the number ratio of old and new samples.

Scalar	**100**	**1000**	**2000**	**5000**	**10000**
CNN	**47.88**	**48.63**	**48.43**	**48.51**	**48.47**
NME	**51.90**	**52.01**	**51.88**	**51.65**	**51.79**

Table 6: Analysis on Expanding Scalar. The results show the robustness of our algorithm to varying scalar.

Analysis on Expanding Scalar To demonstrate the robustness of our method on hyperparameter, we conduct relevant perturbation experiments on HMDB51 with 5 steps. We adjust the value of the scalar to 100, 1000, 2000, 5000, and 10000, respectively. It can be observed in Table 6 that the impact of scalar on the accuracy of the two classifiers is between 0.2% and 0.3%. Therefore, it can be concluded that our method exhibits robustness in terms of scalar.

Conclusion

To alleviate catastrophic forgetting issues that may arise in the class-incremental context of video action recognition, we proposed a hypercorrelation evolution scheme, in which a hierarchical aggregation strategy is presented to preserve semantic-effective features from previous tasks adaptively, and a correlation refinement mechanism is presented to address the knowledge imbalance interrupted by both sample number and video redundancy. In this way, our scheme achieves outstanding performance compared to existing image-specific class-incremental learning methods on multiple standard benchmarks.

Acknowledgments

This work is supported by National Key R&D Program of China under Grant 2020AAA0105701, National Natural Science Foundation of China (NSFC) under Grants 62306295 and OPPO Research Fund.

References

Carreira, J.; and Zisserman, A. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6299–6308.

Chen, X.; Zhang, Y.; and Qin, Z. 2019. Dynamic explainable recommendation based on neural attentive models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 53–60.

Dhar, P.; Singh, R. V.; Peng, K.-C.; Wu, Z.; and Chellappa, R. 2019. Learning without memorizing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5138–5146.

Douillard, A.; Cord, M.; Ollion, C.; Robert, T.; and Valle, E. 2020. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, 86–102. Springer.

Fan, L.; Buch, S.; Wang, G.; Cao, R.; Zhu, Y.; Niebles, J. C.; and Fei-Fei, L. 2020. Rubiksnet: Learnable 3d-shift for efficient video action recognition. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX*, 505–521. Springer.

Feichtenhofer, C.; Pinz, A.; and Zisserman, A. 2016. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1933–1941.

Goyal, R.; Ebrahimi Kahou, S.; Michalski, V.; Materzynska, J.; Westphal, S.; Kim, H.; Haenel, V.; Fruend, I.; Yianilos, P.; Mueller-Freitag, M.; et al. 2017. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, 5842–5850.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hou, S.; Pan, X.; Loy, C. C.; Wang, Z.; and Lin, D. 2019. Learning a Unified Classifier Incrementally via Rebalancing. In *CVPR*.

Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; and Fei-Fei, L. 2014. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 1725–1732.

Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; and Serre, T. 2011. HMDB: a large video database for human motion recognition. In *2011 International conference on computer vision*, 2556–2563. IEEE.

Li, Z.; and Hoiem, D. 2017. Learning without Forgetting. *IEEE T-PAMI*, 40(12): 2935–2947.

Lin, J.; Gan, C.; and Han, S. 2019. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, 7083–7093.

Liu, Y.; Schiele, B.; and Sun, Q. 2021a. Adaptive aggregation networks for class-incremental learning. In *CVPR*, 2544–2553.

Liu, Y.; Schiele, B.; and Sun, Q. 2021b. RMM: Reinforced Memory Management for Class-Incremental Learning. *Advances in Neural Information Processing Systems*, 34.

Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y.; Goodfellow, I. J.; and Pouget-Abadie, J. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27: 2672–2680.

Odena, A.; Olah, C.; and Shlens, J. 2017. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*, 2642–2651. PMLR.

Ostapenko, O.; Puscas, M.; Klein, T.; Jahnichen, P.; and Nabi, M. 2019. Learning to remember: A synaptic plasticity driven framework for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11321–11329.

Park, J.; Kang, M.; and Han, B. 2021. Class-incremental learning for action recognition in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 13698–13707.

Pei, Y.; Qing, Z.; Cen, J.; Wang, X.; Zhang, S.; Wang, Y.; Tang, M.; Sang, N.; and Qian, X. 2022. Learning a Condensed Frame for Memory-Efficient Video Class-Incremental Learning. *arXiv preprint arXiv:2211.00833*.

Qiu, Z.; Yao, T.; and Mei, T. 2017. Learning spatio-temporal representation with pseudo-3d residual networks. In *proceedings of the IEEE International Conference on Computer Vision*, 5533–5541.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2001–2010.

Romero, A.; Ballas, N.; Kahou, S. E.; Chassang, A.; Gatta, C.; and Bengio, Y. 2014. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.

Shin, H.; Lee, J. K.; Kim, J.; and Kim, J. 2017. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, 4489–4497.

Tran, D.; Wang, H.; Torresani, L.; and Feiszli, M. 2019. Video classification with channel-separated convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5552–5561.

Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; and Paluri, M. 2018. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 6450–6459.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Villa, A.; Alhamoud, K.; Escorcia, V.; Caba, F.; Alcázar, J. L.; and Ghanem, B. 2022. vclimb: A novel video class incremental learning benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19035–19044.

Wang, L.; Li, W.; Li, W.; and Van Gool, L. 2018. Appearance-and-relation networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1430–1439.

Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; and Van Gool, L. 2016. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In *ECCV*.

Xie, S.; Sun, C.; Huang, J.; Tu, Z.; and Murphy, K. 2018. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, 305–321.

Yan, S.; Xie, J.; and He, X. 2021. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3014–3023.

Yu, L.; Twardowski, B.; Liu, X.; Herranz, L.; Wang, K.; Cheng, Y.; Jui, S.; and Weijer, J. v. d. 2020. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6982–6991.

Zagoruyko, S.; and Komodakis, N. 2016. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*.

Zhai, W.; Cao, Y.; Zhang, J.; Xie, H.; Tao, D.; and Zha, Z.-J. 2023. On Exploring Multiplicity of Primitives and Attributes for Texture Recognition in the Wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhou, Y.; Sun, X.; Zha, Z.-J.; and Zeng, W. 2018. Mict: Mixed 3d/2d convolutional tube for human action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 449–458.

Zhu, K.; Cao, Y.; Zhai, W.; Cheng, J.; and Zha, Z.-J. 2021. Self-promoted prototype refinement for few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6801–6810.

Zhu, K.; Zhai, W.; Cao, Y.; Luo, J.; and Zha, Z.-J. 2022. Self-Sustaining Representation Expansion for Non-Exemplar Class-Incremental Learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 9296–9305.

Zolfaghari, M.; Singh, K.; and Brox, T. 2018. Eco: Efficient convolutional network for online video understanding. In *Proceedings of the European conference on computer vision (ECCV)*, 695–712.