Rep Deep & Machine Learning: Exemplar-Free Continual Video Action Recognition via Slow-Fast Collaborative Learning

Xueyi Zhang[1,2,3], Chengwei Zhang[4], Zheng Li[1], Xiyu Wang[3], Siqi Cai[2, 5*], Mingrui Lao[1*], Yanming Guo[1], Huiping Zhuang[6]

[1]College of Systems Engineering, National University of Defense Technology
[2] Shenzhen Loop Area Institute
[3] School of Artificial Intelligence, The Chinese University of Hong Kong, Shenzhen
[4]School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences
[5]School of Intelligence Science and Engineering, Harbin Institute of Technology, Shenzhen
[6]Shien-Ming Wu School of Intelligent Engineering, South China University of Technology

Abstract

In real-world applications, video action recognition models must continuously learn new action categories while retaining previously acquired knowledge. However, most existing approaches rely on storing historical data for replay, which introduces storage burdens and raises data privacy concerns. To address these challenges, we investigate the problem of Exemplar-Free Continual Video Action Recognition (EF-CVAR) and propose a novel framework named Slow-Fast Collaborative Learning (SFCL). SFCL integrates two complementary learning paradigms: a slow branch based on gradient-driven deep learning, which provides strong adaptability to new tasks, and a fast branch based on analytic learning (e.g., Recursive Least Squares), which efficiently preserves old knowledge without requiring access to past samples. To enable effective collaboration between the two branches, we design the Slow-Fast Dynamic Reparameterization (SFDR) mechanism for adaptive fusion, and the Knowledge Reflection Mechanism (KRM), which mitigates forgetting and task-recency bias via pseudo-feature generation and dual-level knowledge distillation. Extensive experiments on UCF101, HMDB51, and Something-Something V2 demonstrate that SFCL achieves superior performance compared to existing replay-based methods, despite being exemplar-free. Notably, in long-duration continual learning scenarios, SFCL exhibits remarkable robustness, achieving up to a 30.39% improvement in accuracy over baselines while maintaining a low forgetting rate, highlighting its scalability and effectiveness in real-world video recognition tasks.

1 Introduction

Video action recognition is a vital task in computer vision, with widespread applications in human-computer interaction, security, healthcare, social media, and entertainment (Zhang et al. 2022; Leng et al. 2024; Lin, Gan, and Han 2019; Zhang, Sheng, and Liu 2021; Lu and Elhamifar 2024; Qu, Cai, and Liu 2024; Huang and Zhang 2022; Zhu et al. 2024; Leng et al. 2024, 2025; Wu et al. 2022; Zhang et al. 2024, 2025a). As novel actions continually emerge in the digital era, models must learn to recognize them without forgetting previously acquired knowledge. Continual learning (Feng et al. 2025; Lu et al. 2025; Bian et al. 2024; Liu et al. 2026; Feng, Wang, and Yuan 2022) addresses this challenge by enabling models to acquire new knowledge while retaining prior knowledge. Traditionally, continual action recognition(Yue et al. 2024; Zhang et al. 2023; Li et al. 2025; Zhang et al. 2025c) has relied on storing previously seen action videos to mitigate catastrophic forgetting (Rebuffi et al. 2017; Hou et al. 2019; Douillard et al. 2020; Park, Kang, and Han 2021; Jiao et al. 2024; Pei et al. 2022; Liang et al. 2024). However, this strategy is often impractical due to significant storage requirements and privacy concerns (Pei et al. 2023).

Therefore, we focus on a more challenging and realistic setting: Exemplar-Free Continual Video Action Recognition (EF-CVAR), where the model must learn new action classes without storing or replaying samples from previous tasks. In the absence of historical data, traditional gradient-based learning methods are particularly vulnerable to task-recency bias (Rypeść et al. 2024), where the model disproportionately favors recently learned classes while forgetting earlier ones. To overcome this limitation, we introduce analytic learning (Yue et al. 2025; Zhang et al. 2025b), which employs pseudoinverse techniques, is effective in mitigating catastrophic forgetting and Task-Recency Bias by maintaining stable representations of past knowledge without storing past data. However, analytic learning uses a fixed feature extractor and updates the model through a single forward pass without gradient-based optimization, which limits its ability to acquire new knowledge compared to iterative, backpropagation-based methods.

This work aims to integrate the complementary advantages of analytic and gradient-based learning—preserving prior knowledge without exemplars while improving adaptability to new tasks—by designing a unified framework that overcomes the limitations of each.

In this context, we propose a novel approach called Slow-Fast Collaborative Learning (SFCL), designed for Exemplar-Free Continual Video Action Recognition (EF-CVAR). The SFCL framework is centered on two key innovations: the Slow-Fast Dynamic Re-parameterization (SFDR) and the Knowledge Reflection Mechanism (KRM).

*Corresponding authors.

Specifically, we propose a novel framework called Slow-Fast Collaborative Learning, which jointly exploits the stability of analytic learning and the flexibility of deep learning. To this end, SFCL is built upon two core components:

Slow-Fast Dynamic Re-parameterization (SFDR) introduces a bi-branch architecture, where the slow branch incrementally learns new action classes via backpropagation, while the fast branch utilizes Recursive Least Squares (RLS) to consolidate past knowledge in a closed-form manner. Critically, unlike traditional analytic learning that employs a fixed feature extractor, our approach allows analytic components to be re-parameterized, enabling them to evolve alongside deep learning representations. A streaming discriminator is introduced to dynamically re-weight the contribution from each branch based on input features, effectively balancing the trade-off between plasticity and stability.

Knowledge Reflection Mechanism (KRM) further enhances coordination between the two branches by synthesizing pseudo-features that approximate historical data distributions. This enables dual-level knowledge distillation—at the feature and prediction levels—ensuring that the re-parameterized model not only adapts to new tasks but also retains alignment with previously acquired knowledge.

We validate SFCL through extensive experiments on UCF101, HMDB51, and Something-Something V2. Despite being exemplar-free, SFCL consistently outperforms competitive replay-based methods, achieving up to 30.39% accuracy improvement in long-term continual learning scenarios, while maintaining low forgetting rates, thus offering a practical, scalable, and privacy-preserving solution for evolving video understanding tasks.

Our contributions are summarized as:

- We introduce the **Slow-Fast Dynamic Re-parameterization (SFDR)** mechanism, which enables collaborative learning through a bi-branch structure. A streaming discriminator adaptively fuses the gradient-based and analytic branches, while re-parameterizing analytic components to improve adaptability.
- We design a **Knowledge Reflection Mechanism (KRM)** that synthesizes pseudo-features via Gaussian memory modeling and performs dual-level knowledge distillation, mitigating catastrophic forgetting and task-recency bias without relying on exemplars.
- Extensive experiments on three benchmark datasets—UCF101, HMDB51, and Something-Something V2—demonstrate that SFCL achieves state-of-the-art performance among exemplar-free methods and even surpasses exemplar-based approaches, with up to **30.39% accuracy gain** in long-term continual learning settings.

2 Related Work

Deep learning has significantly advanced video action recognition, yet the demand for recognizing activities in dynamic environments has spurred research into class-incremental action recognition. Park et al. (Park, Kang, and Han 2021) employed time-channel information for weighted knowledge distillation to capture temporal dynamics. Maraghi et al. (Maraghi and Faez 2022) used network sharing and multi-level knowledge distillation, while Villa et al. (Villa et al. 2023) developed the PIVOT model to enhance temporal modeling with spatial prompts and memory replay. Pei et al. (Pei et al. 2022) proposed a memory-efficient method using compact frame representations. Jiao et al. (Jiao et al. 2024) improved replay efficiency with sparse sampling and interpolation alignment. Liang et al. (Liang et al. 2024) introduced a hierarchical distillation method for spatiotemporal alignment. *Despite these advancements, the reliance on old data for memory replay raises privacy concerns and limits practicality under strict storage constraints. In contrast, our work addresses this gap by advancing exemplar-free class-incremental video action recognition without sacrificing performance.*

3 Methodology

We outline the fundamental notation and initialization for class incremental learning (CIL) in video action recognition (VAR) in Section 3.1. Section 3.2 introduces the Slow-Fast Dynamic Re-parameterization (SFDR). Section 3.3 presents the Knowledge Reflection Mechanism (KRM).

3.1 Preliminaries

Problem Formulation. Class incremental learning for video action recognition extends the static classification task (Zhu et al. 2020; Wang, Xing, and Liu 2021; Feichtenhofer, Pinz, and Wildes 2017; Wu et al. 2018; Yang et al. 2022; Kong and Fu 2022; Chen and Ho 2022; Wang, She, and Smolic 2021; Stroud et al. 2020) by allowing the model to identify new actions while preserving its memory of previously learned actions. We consider the general setting for class incremental learning, given as a sequence of tasks $T = \{T_0, T_1, T_2, \ldots, T_K\}$, where K represents the total number of tasks. Each task $T_k = \{X_k^{\text{train}}, Y_k^{\text{train}}, X_k^{\text{test}}, Y_k^{\text{test}}\}$ consists of all videos X_k^{part} with corresponding action labels Y_k^{part} belonging to the action class set O_k. N_k^{part} denotes the number of task samples, NC_i^{part} represents the sample count of each action, where $part \in \{\text{train}, \text{test}\}$ is the dataset splits, and $E_k = \text{len}(O_k)$ represents the number of actions for task T_k. The action categories of any two tasks T_{k_1} and T_{k_2} are disjoint, expressed as $O_{k_1} \cap O_{k_2} = \emptyset$. Task T_0 is defined as the base task, typically consisting of a larger set of action categories compared to the incremental tasks $T_{1:K}$, to help the model build a basic understanding of video action patterns.

Initialization of Model Parameters. In video action recognition CIL methods, the backbone network is typically pre-trained on the base task T_0, as shown in the left column of Fig. 1. Analytic learning approaches excel by not requiring additional exemplars to store knowledge of previous classes, pre-training most parameters using base task data, and avoiding fine-tuning in subsequent tasks. Building on this typical CIL network design, we incorporate a Temporal Shift Module (TSM) to enhance temporal modeling capabilities and a linear bottleneck layer between the

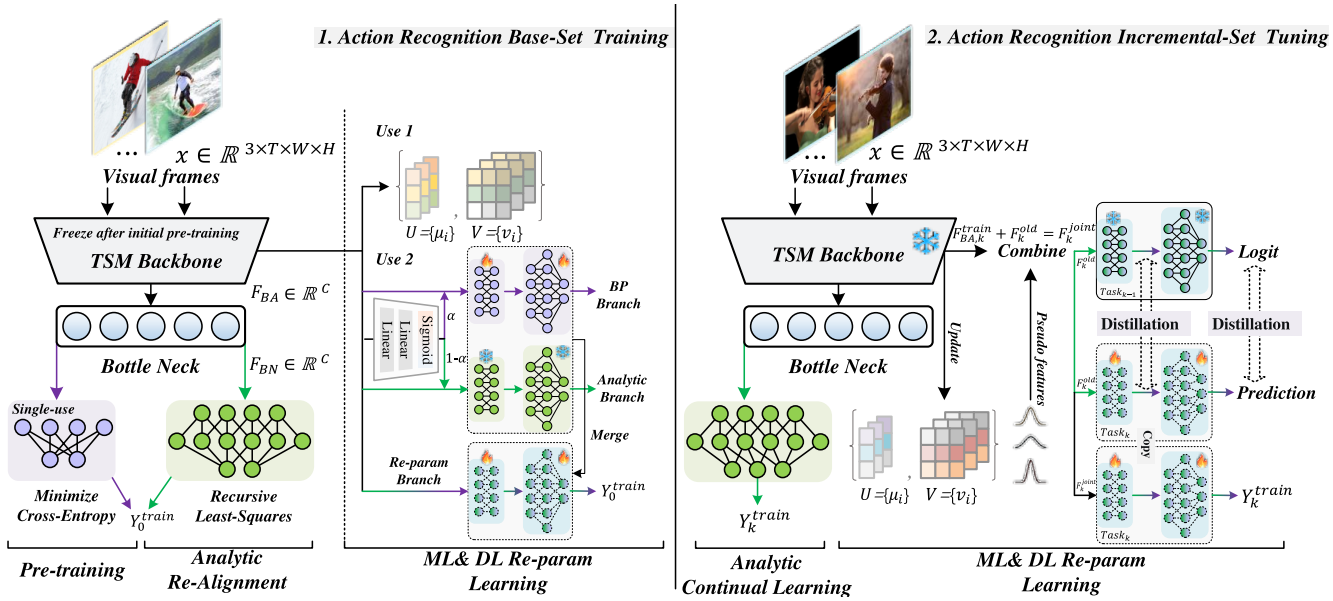

Figure 1: Overall structure of the proposed method.

backbone and classifier, which supports our proposed re-parameterization (Rep) strategy, detailed in Section 3.3. Let $X \in \mathbb{R}^{3\times L\times W\times H}$ denote a video sample, where L, W, and H represent the number of frames and spatial resolution. The data flow of backbone pre-training in the base task is formulated as:

$$\begin{aligned} F_{BA} &= Backbone(\theta_{BA}, X), \\ F_{BN} &= BottleNeck(\theta_{BN}, F_{BA}), \\ \hat{Y} &= Head(\theta_H, F_{BN}), \end{aligned} \tag{1}$$

where $F_{BA} \in \mathbb{R}^C$ is the global feature and C is the feature channel. $F_{BN} \in \mathbb{R}^C$ is the output of the bottleneck layer, $\hat{Y}$ denotes the predicted classification probabilities. θ_{BA}, θ_{BN} and θ_H are the parameters of the backbone network, bottleneck layer and linear classification head, respectively.

We first train and evaluate the above network with the base task data T_0 by minimizing the cross-entropy loss to obtain the optimal parameters θ_{BA}, θ_{BN}, and θ_H where θ_H is not used for the following steps.

3.2 Slow-Fast Dynamic Re-parameterization

In this subsection, we detail the proposed slow-fast dynamic re-parameterize strategy. This strategy combines a slow branch for deep learning new actions through back-propagation, and a fast branch for quickly adapting to past actions via analytic learning without exemplars, which is achieved by a novel dynamical re-parameterization strategy.

VAR Analytic Learning Branch. Base Task Analytic Re-alignment: After completing the base task cross-entropy training, VAR analytic re-alignment, as in the second column in Fig. 1, retain the parameters θ_{BA} and θ_{BN} of the backbone network and bottleneck layer. We perform inference on all base task training samples X_0^{train} to obtain the backbone feature set $F_{BA,0}^{\text{train}} \in \mathbb{R}^{N_0^{\text{train}}\times C}$ and the bottleneck feature set $F_{BN,0}^{\text{train}} \in \mathbb{R}^{N_0^{\text{train}}\times C}$. Then, a randomly initialized 2-layer MLP analytic head with hidden dimension C^H, where $C^H > C$, is used to replace the classification head. The bottleneck features are first upsampled to the hidden dimension:

$$F_{AU,0}^{\text{train}} = ReLU(Linear(\theta_{AU,0}^{AN}, F_{BN,0}^{\text{train}})), \tag{2}$$

where $F_{AU,0}^{\text{train}} \in \mathbb{R}^{N_0^{\text{train}}\times C^H}$ represents the upsampled features, $\theta_{AU,0}^{AN}$ denotes the weights of the upsampling layer, and $ReLU(\cdot)$ is activation function. Thereupon, we can obtain the analytic classification results by optimizing the parameters of the downsampling linear layer as follows:

$$\underset{\hat{\theta}_{AD,0}^{AN}}{\text{argmin}} = ||Y_0^{\text{train}} - F_{AU,0}^{\text{train}}\theta_{AD,0}^{AN}||_F^2 + \eta||\theta_{AD,0}^{AN}||_F^2, \tag{3}$$

where $\theta_{AD,0}^{AN}$ represents the weights of the downsampling layer, and $||\cdot||_F$ is the Forbenius form and η is the regularization term. The optimal weights is given by:

$$\hat{\theta}_{AD,0}^{AN} = \left((F_{AU,0}^{\text{train}})^T F_{AU,0}^{\text{train}} + \eta I\right)^{-1} (F_{AU,0}^{\text{train}})^T Y_0^{\text{train}}, \tag{4}$$

where $\hat{\theta}_{AD,0}^{AN}$ is the optimal parameters.

Incremental Task Analytic Learning: Our objective is to achieve exemplar-free CIL. We use the frozen backbone and bottleneck to extract features and then upsample it to obtain $F_{AU,k}^{\text{train}} \in \mathbb{R}^{N_k\times C^H}$ for fast learning of new task data. The optimal weight using all seen data is given by:

$$\underset{\hat{\theta}_{AD,k}^{AN}}{\text{argmin}} = ||Y_{0:k}^{\text{train}} - F_{AU,0:k}^{\text{train}}\theta_{AD,k}^{AN}||_F^2 + \eta||\theta_{AD,k}^{AN}||_F^2, \tag{5}$$

where the solution of optimal weight is calculated as:

$$\hat{\theta}_{AD,k}^{AN} = \left((F_{AU,0:k}^{\text{train}})^T F_{AU,0:k}^{\text{train}} + \eta I\right)^{-1} (F_{AU,0:k}^{\text{train}})^T Y_{0:k}^{\text{train}}, \tag{6}$$

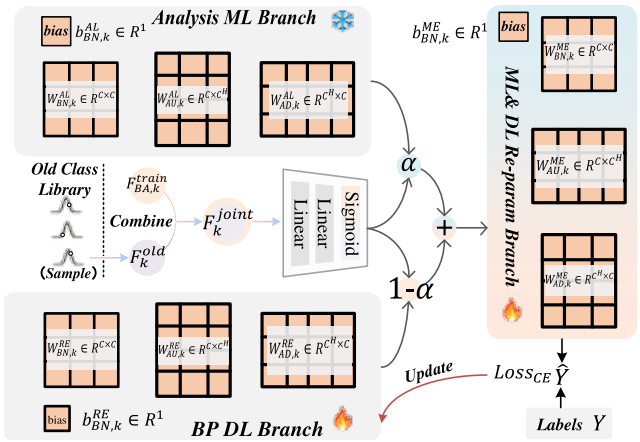

Figure 2: Details of the re-parametrization process. The backbone feature F_{BA} is firstly feed to router network to get a re-parametrization weight a to fuse the analytic branch and back propagation branch. Then the feature is feed to the fused branch to get loss to update the parameter of back propagation branch.

but the above equation still relies on data from previous classes. To get rid of the dependence on old class data, let:

$$R_k = \left((F^{\text{train}}_{AU,0:k})^T F^{\text{train}}_{AU,0:k} + \eta I\right)^{-1}, \tag{7}$$

and Eq. 6 can be formed as:

$$\hat{\theta}^{AN}_{AD,k} = \theta^{AN}_{AD,k-1} - R_k(F^{\text{train}}_{AU,k})^T F^{\text{train}}_{AU,k}\theta^{AN}_{AD,k-1} + R_k(F^{\text{train}}_{AU,k})^T Y^{\text{train}}_k, \tag{8}$$

where R_k can be only updated by the current task data as:

$$R_k = R_{k-1} - R_{k-1}(F^{\text{train}}_{U,k})^T \left(F^{\text{train}}_{U,k} R_{k-1}(F^{\text{train}}_{U,k})^T + I\right)^{-1} F^{\text{train}}_{U,k} R_{k-1}. \tag{9}$$

The process, proved as in Appendix Theorem 1, shows that optimal weights for each task can be obtained using only its training data, eliminating the need to restore historical data and enabling an exemplar-free approach.

Dynamic Re-parameterization. Traditional analytic learning struggles with adaptability and integrating new information. To tackle this, we present a dynamic streaming discriminator re-parameterization method, illustrated in Fig. 2. During initialization, the weights of the analytic branch are duplicated and set to be trainable, yielding $\theta^{RE}_{BN,0} = \{W^{RE}_{BN,0}, b^{RE}_{BN,0}\}$, $\theta^{RE}_{AU,0} = \{W^{RE}_{AU,0}\}$, and $\theta^{RE}_{AD,0} = \{W^{RE}_{AD,0}\}$, which serve as slow BP branches to provide more adaptability. An additional streaming discriminator is used to dynamically control the fusion weights of the analytic branch and BP branch. The forward propagation process is as follows:

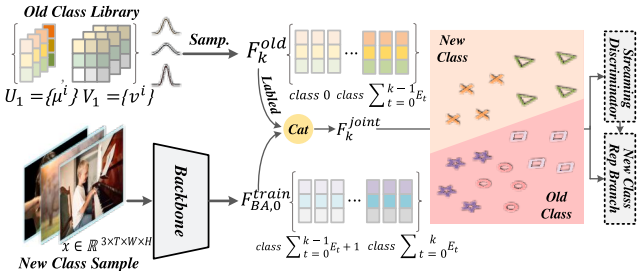

Figure 3: Details of the GMS process. We use the stored old class feature mean μ_i and variance matrix v_i to construct Multivariate Gaussian Distribution and sample feature. The sampled old class feature is concatenated with extracted new class feature.

$$\begin{aligned} a &= \mathit{Sigmoid}(\mathit{Linear}(\theta_{SD}, F_{BA})), \\ F^M_{BN} &= (a \cdot W^{AL}_{BN,0} + (1-a) \cdot W^{RE}_{BN,0})F_{BA} \\ &\quad + a \cdot b^{AL}_{BN,0} + (1-a) \cdot b^{RE}_{BN,0}, \\ F^M_{AU} &= \text{ReLU}((a \cdot W^{AL}_{AU,0} + (1-a) \cdot W^{RE}_{AU,0})F^M_{BN}), \\ \hat{Y}^M_0 &= (a \cdot W^{AL}_{AD,0} + (1-a) \cdot W^{RE}_{AD,0})F^M_{AU}, \end{aligned} \tag{10}$$

where a is the fusion weight ranging from 0 to 1, and θ_{SD} represents the weight of the streaming discriminator. We use the backbone feature set $F^{\text{train}}_{BA,0}$ for training and minimize the cross-entropy loss between the re-parameterized network output $\hat{Y}^M_0$ and the labels Y^{train}_0 to obtain initialized streaming discriminator and BP branch.

Discussion: *Discrepancies between deep learning and analytic learning can lead to inconsistent knowledge updates, limiting overall system performance. These challenges highlight the need for a mechanism to harmonize both learning strategies and mitigate Task-Recency Bias.*

3.3 Knowledge Reflection Mechanism

Re-parameterization training can create discrepancies between model components, leading to instability in knowledge integration. To address this, we use Gaussian Memory Synthesis to generate pseudo-features for balanced data representation, followed by knowledge distillation to align outputs for cohesive learning.

Gaussian Memory Synthesis. The construction of i-th class prototype knowledge is as:

$$\begin{aligned} \mu_i &= \frac{1}{C}\sum_{k=1}^{C} F_{\text{lib},i}, \\ v_i &= \frac{1}{C-1}(F_{\text{lib},i} - \mu_i)(F_{\text{lib},i} - \mu_i)^T, \end{aligned} \tag{11}$$

where $F_{\text{lib},i} = \text{Index}(F^{\text{train}}_{BA,0}, i)$ is the feature of i-th class, $\mu_i \in \mathbb{R}^C$ and $\Sigma_i \in \mathbb{R}^{C\times C}$ represent the mean feature and covariance for class i.

For continuously-arrived tasks t, the parameter of downsampling layer $\theta_{AD,t}$ is expanded and updated according

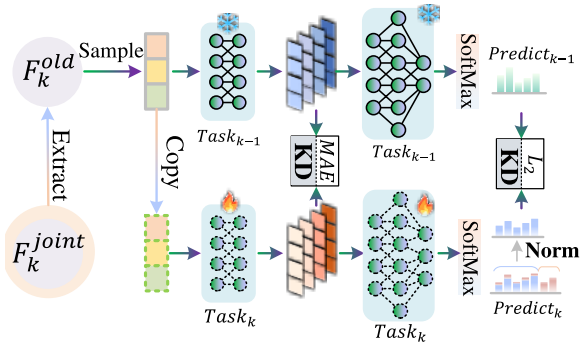

Figure 4: Details of Dual Knowledge Distillation. The distillation is only performed for the old classes. The new classes classification probabilities is normed to old classes to align the output shape.

to the method in Section 3.2 to classify new classes. The BP branch parameters are inherited from the previous task. The constructed prototype knowledge library of the old class knowledge based on seen data is as:

$$\begin{aligned} U_k &= \{\mu_0, \mu_1, \ldots, \mu_{\sum_{t=0}^{k} E_t}\}, \\ V_k &= \{\mathrm{v}_0, \mathrm{v}_1, \ldots, \mathrm{v}_{\sum_{t=0}^{k} E_t}\}. \end{aligned} \tag{12}$$

Dual Knowledge Distillation. We sample a certain number of old class features from the prototype feature library and combine them with the features of the new classes to form a training set for the new task, as follows:

$$\begin{aligned} F_k^{\text{old}} &= \{F_{0,j}^s, F_{1,j}^s, \ldots, F_{\sum_{t=0}^{k} E_t, j}^s\}, \\ F_k^{\text{joint}} &= \text{cat}(F_k^{\text{old}}, F_{BA,k}^{\text{train}}), \end{aligned} \tag{13}$$

where $F_{i,j}^s \sim \mathcal{N}(\mu_i, \mathrm{v}_i)$, i is old class index, $j = 1, \ldots, N_k^{\text{train}}/E_k$ represents the average number of samples per class for the new task, and F_k^{joint} represents the mixed training set containing both sampled old class pseudo-features and extracted new class features. Our main training loss is the cross-entropy between the predicted class probabilities of the re-parameterized branch and the labels:

$$Loss_{Main} = Cross\text{-}Entropy(\hat{Y}, Y). \tag{14}$$

We introduce a knowledge distillation strategy across tasks, freezing the BP branch of previous tasks as a teacher to guide the learnable re-parameterized branch of the new task in retaining old class knowledge by comparing differences in bottleneck features and classification probabilities. The feature-level distillation loss is as follows:

$$Loss_{Feat} = MSE(BottleNeck(\theta_{BN,k-1}^{RE}, F^r), BottleNeck(\theta_{BN,k}^{ME}, F^r)), \tag{15}$$

where $\theta_{BN,k}^{ME} = w\theta_{BN,k}^{AL} + (1-w)\theta_{BN,k}^{RE}$, $MSE(\cdot,\cdot)$ denotes mean squared error, and F^r is a sampled feature from F_k^{joint}. The output of the teacher network includes only the old class probabilities, which do not match the output shape of the student network that predicts both old and new classes. To address this, we normalize the probabilities of the student network's new classes to align with those of the old classes, as follows:

$$\begin{aligned} Prob_k^{norm} &= Prob_k[:\sum_{t=0}^{k-1} E_t] + \frac{1}{\sum_{t=0}^{k-1} E_t} \sum (Prob_k[\sum_{t=0}^{k-1} E_t :]), \\ Loss_{Prob} &= KL(Prob_{k-1}, Prob_k^{norm}), \end{aligned} \tag{16}$$

where KL denotes the KL divergence, and $[\cdot : \cdot]$ denotes list subset. Therefore, the final loss is:

$$Loss = Loss_{Main} + \alpha Loss_{Feat} + \beta Loss_{Prob}. \tag{17}$$

4 Experiments

In this section, we first elabrate the incremental settings of video action recognition task with implementation details of our methods. Then, we compare SFCL with state-of-the-art approaches with explicit ablation studies.

Evaluation Protocol and Metrics. We follow the evaluation protocol from previous work (Park, Kang, and Han 2021), conducting experiments on three video action recognition datasets: UCF101 (Soomro 2012), HMDB51 (Kuehne et al. 2011), and Something-Something V2 (Goyal et al. 2017). All categories are shuffled using fixed random seeds (three for UCF101 and HMDB51, one for Something-Something V2), then split the class list into base and incremental classes. Base classes initialize the backbone and branch, while incremental classes evaluate incremental learning performance.

In the UCF101 dataset, we select 51 classes as base classes and divide the remaining 50 classes into incremental tasks with 10, 5, and 2 classes per task. In the HMDB51 dataset, we use 26 classes as base classes and divide the remaining 25 classes into incremental tasks with 5 and 1 classes per task. For the Something-Something V2 dataset, we use 84 classes as base classes and divide the remaining 90 classes into incremental tasks with 9, 3, and 1 classes per task, which is a longer incremental setting than UCF101 and HMDB51, providing a challenge for learning new classes while retaining memory of old classes.

To evaluate the performance of our EF-CVAR method, we use three metrics: *average incremental accuracy (accuracy)* (Douillard et al. 2020), *forgetting rate* (Liu et al. 2020), and *performance dropping rate (PD)* (Zhang et al. 2021). We compute *average incremental accuracy* across all tasks as the model's accuracy on test samples from tasks 0 to K at the end of each task t. The *forgetting rate* is the difference in accuracy on base classes between base task and current task. *Performance dropping rate (PD)* is defined as $PD = Acc_1 - Acc$, where Acc_1 is the accuracy on task 1, and Acc is the accuracy across all tasks. Previous methods use exemplars for memory retention and nearest mean of exemplars (NME) accuracy during testing. Our method, which does not require exemplars, reports only the CNN accuracy across all tasks.

Implementation Details. We use the typical and widely-used implementation of TSM (Lin, Gan, and Han 2019) as our backbone network for feature extraction and context modeling. To accelerate network convergence without introducing prior knowledge of action recognition, we initialize

Dataset	UCF101						HMDB51			
Num. of Classes	10×5 tasks		5×10 tasks		2×25 tasks		5×5 tasks		1×25 tasks	
Classifier	CNN	NME	CNN	NME	CNN	NME	CNN	NME	CNN	NME
iCaRL	-	65.34%	-	64.51%	-	58.73%	-	40.09%	-	33.77%
UCIR	74.31%	74.09%	70.42%	70.50%	63.22%	64.00%	44.90%	46.53%	37.04%	37.15%
PODNet	73.26%	74.37%	71.58%	73.75%	70.28%	71.87%	44.32%	48.78%	38.76%	46.62%
TCD	74.89%	77.16%	73.43%	75.35%	72.19%	74.01%	45.34%	50.36%	40.47%	46.66%
SNRO	78.96%	77.76%	77.60%	76.95%	76.84%	76.21%	48.65%	2.10%	46.40%	9.38%
FrameMaker	78.13%	78.64%	76.38%	78.14%	75.77%	77.49%	47.54%	51.12%	42.65%	47.37%
HCE	79.12%	0.01%	77.59%	8.81%	75.84%	7.62%	48.63%	52.01%	43.99%	48.94%
Exemplar-Free										
Finetuning	24.97%	-	13.45%	-	5.78%	-	16.82%	-	4.83%	-
LwFMC	42.14%	-	25.59%	-	11.68%	-	26.82%	-	16.49%	-
LwM	43.39%	-	26.07%	-	12.08%	-	26.97%	-	16.50%	-
DBK	77.05%	-	74.12%	-	72.07%	-	50.75%	-	-	-
SFCL(Ours)	**81.72%**	**-**	**81.79%**	**-**	**81.58%**	**-**	**53.02%**	**-**	**52.61%**	**-**
Oracle	84.15%	83.37%	83.96%	83.20%	83.82%	83.16%	55.03%	55.98%	54.89%	55.32%

Table 1: Comparison with state-of-the-art methods on UCF101 and HMDB51 under the TCD protocol using TSM [1].

the backbone parameters with ResNet weights pre-trained on ImageNet (Deng et al. 2009), which better reflects the performance of incremental methods on new actions. For UCF101, we use ResNet34 with a batch size of 32, and for HMDB51 and Something-Something V2, we use ResNet50 with a batch size of 64. The learning rate for both parameter pre-training and re-parameterization training is set to 1×10^{-3}, optimized with SGD with a momentum of 0.9 and weight decay of 5×10^{-4}. Base training is conducted for 50 epochs, with re-parameterization training for 10 epochs. The α and β is both set to 1.

Comparisons with the state-of-the-art. As illustrated in Tab. 1 and Tab. 2, we divided competitive methods into exemplar-based and exemplar-free approaches, listed in the first and second part 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), TCD (Park, Kang, and Han 2021), SNRO (Jiao et al. 2024), FrameMaker (Pei et al. 2022), HCE (Liang et al. 2024) and DBK (Maraghi and Faez 2022). We also provide the lower bound accuracy by training only with new class data (finetuning) and using all previous data (Oracle).

According to the results in Tables 1 and 2, CNN and NME accuracies vary across datasets, with NME performing better on HMDB51. Notably, as a data-free method, our SFDR achieves state-of-the-art performance in all settings, demonstrating its ability to learn new classes and retain old knowledge without relying on exemplars. In more challenging long-term incremental tasks, the improvements are even more significant, highlighting the strong capability of SFDR and KRM in preserving old class knowledge while learning new ones.

5 Analysis and Ablation Studys

In this section, we first analyze the strengths and weaknesses of Recursive Least Squares and Gradient Backpropagation in continual action recognition. Subsequently, we conduct an ablation study on the proposed Knowledge Reflection Mechanism. We then investigate the robustness of our method in tackling more challenging long-duration continual tasks for action recognition across various datasets. Finally, we discuss the Task-Recency Bias in continual action recognition.

[1]https://github.com/bellos1203/TCD

Dataset	Something-Something V2			
Num. of Classes	10×9 tasks		5×18 tasks	
Classifier	CNN	NME	CNN	NME
iCaRL	-	15.48%	-	10.22%
UCIR	26.84%	17.98%	20.69%	12.57%
PODNet	34.94%	27.33%	26.95%	17.49%
TCD	35.78%	28.88%	29.60%	21.63%
FrameMaker	37.25%	29.92%	30.98%	22.84%
HCE	8.67%	36.88%	32.51%	2.82%
SFCL (Ours)	**44.80%**	-	**44.34%**	-
Oracle	60.15%	55.37%	60.96%	54.16%

Table 2: Comparisons with the state-of-the-art methods on the Something-Something V2 dataset.

Analytic vs. BP: Insights. In experiments on the HMDB51 dataset with a 5×5 task division, we evaluated performance on both new and old action data streams. As shown in Tab. 3, Analytic Class Incremental Learning (ACIL) outperformed on old classes by 13.23%, thanks to its block-wise recursive Moore-Penrose learning (BRMP), which preserves memory of old tasks but limits adaptability to new actions. Conversely, backpropagation shows a 16.73% advantage over analytic methods on new classes; however, it results in a forgetting rate of 35.64% on old classes, compared to only 4.62% for ACIL. Our method integrates the strengths of ACIL and backpropagation through re-parameterization, using a Streaming Discriminator and Knowledge Reflection Mechanism (KRM) to mitigate their weaknesses. As a result, our approach achieves superior performance on both new and old classes compared to ACIL, and with an overall performance improvement of 10.33% over backpropagation.

Task	ACIL				Backpropagation				SFCL (Ours)			
	Accuracy ↑			Forget Rate ↓	Accuracy ↑			Forget Rate ↓	Accuracy ↑			Forget Rate ↓
	Total	*Old*	*New*		*Total*	*Old*	*New*		*Total*	*Old*	*New*	
Base	61.28%	61.28%	61.28%	0.00%	**62.31%**	**62.31%**	**62.31%**	0.00%	2.05%	2.05%	2.05%	0.00%
Task 1	5.59%	**59.74%**	41.33%	1.15%	55.48%	56.92%	2.00%	5.38%	**58.06%**	9.62%	**48.00%**	2.05%
Task 2	1.67%	5.70%	31.33%	2.82%	50.37%	50.65%	**48.00%**	7.56%	**53.06%**	**56.67%**	5.33%	4.10%
Task 3	9.02%	0.00%	36.67%	3.08%	37.15%	34.63%	**59.33%**	19.23%	**51.54%**	**51.76%**	8.00%	4.49%
Task 4	4.86%	8.21%	22.67%	3.21%	25.65%	22.52%	**55.33%**	30.64%	**47.10%**	**48.70%**	4.00%	5.38%
Task 5	4.18%	5.07%	47.33%	4.62%	25.16%	19.28%	**74.00%**	35.64%	**46.27%**	**45.87%**	0.00%	6.79%
Avg	1.10%	3.33%	40.10%	-	42.69%	41.05%	**56.83%**	-	**53.02%**	**54.11%**	2.90%	-

Table 3: Comparison of analytic and back-propagation learning manners on the HMDB51 dataset under the 5×5 task setting.

GMS	Feature	Distribution	HMDB51	
			5×5 tasks *Acc. (%)* ↑	1×25 tasks *Acc. (%)* ↑
			51.95	51.38
✓		✓	52.51	51.79
✓	✓		52.73	51.92

Table 4: Ablation study on the knowledge reflection mechanism.

Dataset	UCF101		SS V2			
	1×50		3×30		1×90	
Method	CNN	NME	CNN	NME	CNN	NME
TCD	68.19%	8.81%	17.44%	3.81%	12.88%	3.17%
SFCL (Ours)	**81.34%**	–	**44.88%**	–	**43.56%**	–

Table 5: Results under long-task settings, highlighting the robustness of our method in retaining old classes.

Knowledge Reflection Mechanism. Our ablation study (Table 4) shows that Gaussian Memory Synthesis (GMS) provides the largest improvement by generating pseudo-features for past classes. Feature- and distribution-level distillation further help preserve old knowledge, confirming that all KRM components are complementary.

Robustness in Lengthy Training Scenarios. We evaluate the robustness of SFCL in extended continual learning settings using 90 and 30 incremental tasks on Something-Something V2, and 50 tasks on UCF101. As shown in Tab.5 and Fig.5, SFCL consistently outperforms baselines, achieving up to **30.68% accuracy gain** and significantly lower forgetting rates. These results demonstrate that our method remains effective over long task sequences, successfully learning new actions while preserving knowledge of earlier ones.

Discussion on Task-Recency Bias. To examine Task-Recency Bias, we visualize predicted logits for a sample during Task 5 on HMDB51 in Fig. 6. Finetuning exhibits strong bias toward recent classes, while TCD retains some balance through memory replay but still favors new tasks. In contrast, SFCL maintains a more uniform logit distribution across old and new classes, effectively mitigating Task-Recency Bias. This balanced output aligns with SFCL's low forgetting rate, confirming its fairness and stability in continual learning.

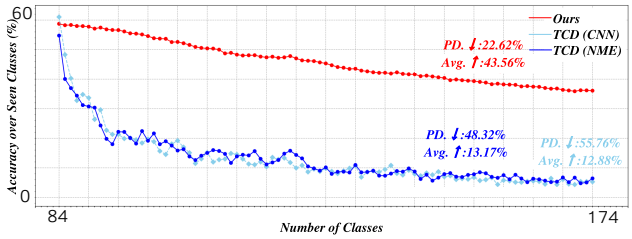

Figure 5: Accuracy visualization under longer incremental tasks.

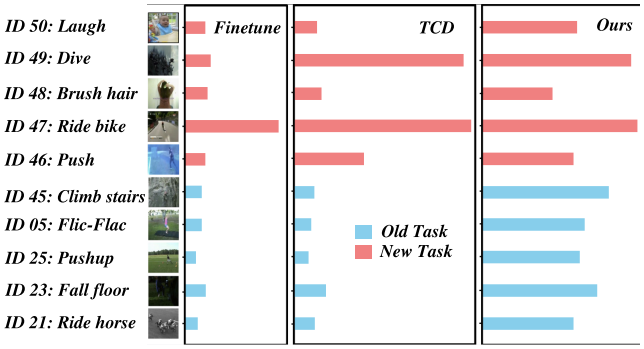

Figure 6: Visualization of the logit distributions for the same sample under fine-tuning, TCD and SFCL(Ours).

6 Conclusion

We introduced Slow-Fast Collaborative Learning (SFCL), a novel framework for exemplar-free continual video action recognition that effectively integrates the adaptability of gradient-based deep learning with the stability of analytic learning. Through the proposed Slow-Fast Dynamic Reparameterization (SFDR) and Knowledge Reflection Mechanism (KRM), our approach enables the model to learn new action classes incrementally while mitigating catastrophic forgetting—without storing any historical data. Extensive evaluations on UCF101, HMDB51, and Something-Something V2 demonstrate that SFCL not only surpasses state-of-the-art exemplar-free methods but also outperforms many exemplar-based baselines. Particularly in long-term incremental scenarios, SFCL achieves up to 30.68% accuracy improvement, affirming its robustness, scalability, and practical value for real-world, privacy-sensitive video understanding systems.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (62306117, 62502530), the Guangzhou Basic and Applied Basic Research Foundation (2024A04J3681), the Shenzhen Science and Technology Program (Shenzhen Key Laboratory, ZDSYS20230626091302006), the Shenzhen Science and Technology Research Fund (JCYJ20220818103001002), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (2023ZT10X044), the Hunan Province Key Research and Development Program (2025QK3004), and the NUDT Foundation (ZZCX-JDZ-39, ZK24-27).

References

Bian, A.; Li, W.; Yuan, H.; Wang, M.; Zhao, Z.; Lu, A.; Ji, P.; Feng, T.; et al. 2024. Make Continual Learning Stronger via C-Flat. *NeurIPS*.

Chen, J.; and Ho, C. M. 2022. Mm-vit: Multi-modal video transformer for compressed video action recognition. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 1910–1921.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

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.

Feichtenhofer, C.; Pinz, A.; and Wildes, R. P. 2017. Spatiotemporal multiplier networks for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4768–4777.

Feng, T.; Li, W.; Zhu, D.; Yuan, H.; Zheng, W.; Zhang, D.; and Tang, J. 2025. ZeroFlow: Overcoming Catastrophic Forgetting is Easier than You Think. *ICML*.

Feng, T.; Wang, M.; and Yuan, H. 2022. Overcoming Catastrophic Forgetting in Incremental Object Detection via Elastic Response Distillation. In *CVPR*.

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.

Hou, S.; Pan, X.; Loy, C. C.; Wang, Z.; and Lin, D. 2019. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 831–839.

Huang, A.; and Zhang, X. 2022. Dual-flow spatio-temporal separation network for lip reading. In *Journal of Physics: Conference Series*, volume 2400, 012028. IOP Publishing.

Jiao, J.; Dai, Y.; Mei, H.; Qiu, H.; Gong, C.; Tang, S.; Hao, X.; and Li, H. 2024. Slightly Shift New Classes to Remember Old Classes for Video Class-Incremental Learning. *arXiv preprint arXiv:2404.00901*.

Kong, Y.; and Fu, Y. 2022. Human action recognition and prediction: A survey. *International Journal of Computer Vision*, 130(5): 1366–1401.

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.

Leng, J.; Kuang, C.; Li, S.; Gan, J.; Chen, H.; and Gao, X. 2025. Dual-Space Video Person Re-identification. *International Journal of Computer Vision*, 133(6): 3667–3688.

Leng, J.; Wu, Z.; Tan, M.; Liu, Y.; Gan, J.; Chen, H.; and Gao, X. 2024. Beyond euclidean: Dual-space representation learning for weakly supervised video violence detection. *Advances in Neural Information Processing Systems*, 37: 17373–17397.

Li, Z.; and Hoiem, D. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12): 2935–2947.

Li, Z.; Zhang, X.; Guo, Y.; Cai, S.; and Lao, M. 2025. PENCIL: Prototype-ENhanced ComposItional Learning for Class-Incremental Hand Gesture Recognition. *IEEE Transactions on Consumer Electronics*.

Liang, S.; Zhu, K.; Zhai, W.; Liu, Z.; and Cao, Y. 2024. Hypercorrelation Evolution for Video Class-Incremental Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 3315–3323.

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.; Su, Y.; Liu, A.-A.; Schiele, B.; and Sun, Q. 2020. Mnemonics training: Multi-class incremental learning without forgetting. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 12245–12254.

Liu, Z.; Kang, B.; Li, W.; Yuan, H.; Yang, Y.; Li, W.; Luo, J.; Zhu, Y.; and Feng, T. 2026. Branch, or Layer? Zeroth-Order Optimization for Continual Learning of Vision-Language Models.

Lu, A.; Yuan, H.; Feng, T.; and Sun, Y. 2025. Rethinking the stability-plasticity trade-off in continual learning from an architectural perspective. *ICML*.

Lu, Z.; and Elhamifar, E. 2024. Fact: Frame-action cross-attention temporal modeling for efficient action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18175–18185.

Maraghi, V. O.; and Faez, K. 2022. Class-Incremental Learning on Video-Based Action Recognition by Distillation of Various Knowledge. *Computational Intelligence and Neuroscience*, 2022(1): 4879942.

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. *Advances in Neural Information Processing Systems*, 35: 31002–31016.

Pei, Y.; Qing, Z.; Zhang, S.; Wang, X.; Zhang, Y.; Zhao, D.; and Qian, X. 2023. Space-time prompting for video class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11932–11942.

Qu, H.; Cai, Y.; and Liu, J. 2024. Llms are good action recognizers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18395–18406.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. iCaRL: Incremental Classifier and Representation Learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Rypeść, G.; Cygert, S.; Trzciński, T.; and Twardowski, B. 2024. Task-recency bias strikes back: Adapting covariances in Exemplar-Free Class Incremental Learning. *arXiv preprint arXiv:2409.18265*.

Soomro, K. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Stroud, J.; Ross, D.; Sun, C.; Deng, J.; and Sukthankar, R. 2020. D3d: Distilled 3d networks for video action recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 625–634.

Villa, A.; Alcázar, J. L.; Alfarra, M.; Alhamoud, K.; Hurtado, J.; Heilbron, F. C.; Soto, A.; and Ghanem, B. 2023. Pivot: Prompting for video continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 24214–24223.

Wang, M.; Xing, J.; and Liu, Y. 2021. Actionclip: A new paradigm for video action recognition. *arXiv preprint arXiv:2109.08472*.

Wang, Z.; She, Q.; and Smolic, A. 2021. Action-net: Multipath excitation for action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13214–13223.

Wu, C.-Y.; Zaheer, M.; Hu, H.; Manmatha, R.; Smola, A. J.; and Krähenbühl, P. 2018. Compressed video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6026–6035.

Wu, X.; Zhang, X.; Feng, X.; Lopez, M. B.; and Liu, L. 2022. Audio-visual kinship verification: a new dataset and a unified adaptive adversarial multimodal learning approach. *IEEE Transactions on Cybernetics*, 54(3): 1523–1536.

Yang, J.; Dong, X.; Liu, L.; Zhang, C.; Shen, J.; and Yu, D. 2022. Recurring the transformer for video action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14063–14073.

Yue, X.; Chen, Y.; Zhang, X.; Gao, X.; Feng, M.; Lao, M.; Zhuang, H.; and Li, H. 2025. PAL: Prompting Analytic Learning with Missing Modality for Multi-Modal Class-Incremental Learning. *arXiv preprint arXiv:2501.09352*.

Yue, X.; Zhang, X.; Chen, Y.; Zhang, C.; Lao, M.; Zhuang, H.; Qian, X.; and Li, H. 2024. Mmal: Multi-modal analytic learning for exemplar-free audio-visual class incremental tasks. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 2428–2437.

Zhang, C.; Song, N.; Lin, G.; Zheng, Y.; Pan, P.; and Xu, Y. 2021. Few-shot incremental learning with continually evolved classifiers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12455–12464.

Zhang, X.; Lao, M.; Zhao, P.; Tang, J.; Guo, Y.; Cai, S.; Yue, X.; and Li, H. 2024. Language without borders: A dataset and benchmark for code-switching lip reading. *Advances in Neural Information Processing Systems*, 37: 30727–30739.

Zhang, X.; Sheng, C.; and Liu, L. 2021. Lip motion magnification network for lip reading. In *2021 7th International Conference on Big Data and Information Analytics (BigDIA)*, 274–279. IEEE.

Zhang, X.; Sun, J.; Zhang, C.; Yue, X.; Xiao, T.; Cai, S.; Lao, M.; and Li, H. 2025a. EventLip: Enhancing Event-Based Lip Reading via Frequency-Aware Spatiotemporal Hypergraph Modeling. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 8263–8272.

Zhang, X.; Zhang, C.; Sui, J.; Sheng, C.; Deng, W.; and Liu, L. 2022. Boosting lip reading with a multi-view fusion network. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, 1–6. IEEE.

Zhang, X.; Zhang, C.; Wang, T.; Tang, J.; Lao, S.; and Li, H. 2023. Slow-fast time parameter aggregation network for class-incremental lip reading. In *Proceedings of the 31st ACM International Conference on Multimedia*, 747–756.

Zhang, X.; Zhu, P.; Sui, J.; Yang, X.; Tian, J.; Lao, M.; Cai, S.; Guo, Y.; and Tang, J. 2025b. Choose Your Expert: Uncertainty-Guided Expert Selection for Continual Deepfake Detection. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 11502–11511.

Zhang, X.; Zhu, P.; Zhang, C.; Yan, Z.; Cheng, J.; Lao, M.; Cai, S.; and Guo, Y. 2025c. Generalization-Preserved Learning: Closing the Backdoor to Catastrophic Forgetting in Continual Deepfake Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3798–3808.

Zhu, A.; Ke, Q.; Gong, M.; and Bailey, J. 2024. Part-aware Unified Representation of Language and Skeleton for Zero-shot Action Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18761–18770.

Zhu, Y.; Li, X.; Liu, C.; Zolfaghari, M.; Xiong, Y.; Wu, C.; Zhang, Z.; Tighe, J.; Manmatha, R.; and Li, M. 2020. A comprehensive study of deep video action recognition. *arXiv preprint arXiv:2012.06567*.