Collaborative Weakly Supervised Video Correlation Learning for Procedure-Aware Instructional Video Analysis

Tianyao He[1], Huabin Liu[1], Yuxi Li[1], Xiao Ma[2], Cheng Zhong[2], Yang Zhang[2], Weiyao Lin[1]*

[1]Shanghai Jiao Tong University, Shanghai, China
[2]AI Lab, Lenovo Research, Beijing, China
{hetianyao,huabinliu,lyxok1}@sjtu.edu.cn,
{maxiao3, zhongcheng3, zhangyang20}@lenovo.com, wylin@sjtu.edu.cn

Abstract

Video Correlation Learning (VCL), which aims to analyze the relationships between videos, has been widely studied and applied in various general video tasks. However, applying VCL to instructional videos is still quite challenging due to their intrinsic procedural temporal structure. Specifically, procedural knowledge is critical for accurate correlation analyses on instructional videos. Nevertheless, current procedure-learning methods heavily rely on step-level annotations, which are costly and not scalable. To address this problem, we introduce a weakly supervised framework called Collaborative Procedure Alignment (CPA) for procedure-aware correlation learning on instructional videos. Our framework comprises two core modules: collaborative step mining and frame-to-step alignment. The collaborative step mining module enables simultaneous and consistent step segmentation for paired videos, leveraging the semantic and temporal similarity between frames. Based on the identified steps, the frame-to-step alignment module performs alignment between the frames and steps across videos. The alignment result serves as a measurement of the correlation distance between two videos. We instantiate our framework in two distinct instructional video tasks: sequence verification and action quality assessment. Extensive experiments validate the effectiveness of our approach in providing accurate and interpretable correlation analyses for instructional videos.

Introduction

Video Correlation Learning (VCL) focuses on examining and quantifying the relationships between videos through a comparative paradigm. It empowers researchers to discover temporal and conceptual knowledge from the intrinsic associations between videos. Numerous previous studies have explored VCL in general videos. For example, some methods adopt VCL to grasp the similarities and differences between videos, including video contrastive learning (Qian et al. 2021; Park et al. 2022), and video retrieval (Zhou et al. 2018; Wang, Jabri, and Efros 2019). Other studies utilize VCL to analyze individual videos referring to given exemplars (or called support videos), such as video quality assessment (Mozhaeva et al. 2021; Xu et al. 2020), and few-shot action recognition (Zhu and Yang 2018; Ben-Ari et al. 2021). As depicted in Fig. 1(a), VCL in general videos primarily centers on the video and frame-level comparison.

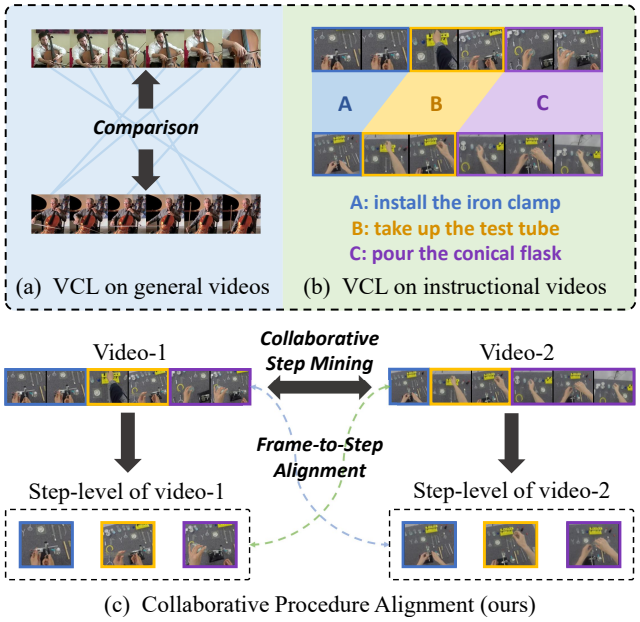

Figure 1: (a)(b) VCL on general videos and instructional videos; (c) Our Collaborative Procedure Alignment framework, which conducts a procedural alignment between frames and collaboratively mined steps.

However, applying traditional VCL approaches to *instructional* videos encounters significant challenges. Specifically, instructional videos comprise multiple fine-grained steps with varying durations and temporal locations. This results in more complex procedural structures compared to general videos (see Fig. 1(b)). Therefore, to achieve precise and interpretable correlation learning for instructional videos, the crux lies in capturing procedural knowledge.

Currently, many procedure-learning methods for instructional videos are emerging (Behrmann et al. 2022; Han, Xie, and Zisserman 2022; Xu et al. 2022). However, they heavily rely on step-level annotations. These annotations require step semantic labels and their temporal boundaries, incurring substantial costs and lacking scalability. This naturally

*Corresponding author.

raises a pivotal question: *How can we learn the intrinsic procedural knowledge of instructional videos without step-level annotations?* In this paper, we present an insight into this question for VCL: *Two instructional videos that share the same procedure often exhibit a strong internal correlation between steps, which can serve as a valuable reference for mutual procedure learning.*

Based on this insight, we present a weakly supervised collaborative procedure alignment (CPA) framework to achieve procedure-aware correlation learning for instructional videos. Here, "weakly supervised" refers to accessing only video-level classes while step-level annotations are unknown. Videos belonging to the same video-level class present identical procedures. As illustrated in Fig. 1(b), our framework harnesses the internal correlation between paired videos, allowing for the collaborative extraction of step-level information and quantifying the step-level correlation between instructional videos. Specifically, our framework consists of two core components: collaborative step mining and frame-to-step alignment. Collaborative step mining exploits both semantic and temporal similarities among video frames, enabling the simultaneous extraction of steps for paired instructional videos. It empowers us to extract a video's steps with guidance from the other, and vice versa. Built upon the step-level features, we then design a frame-to-step alignment module to quantify the procedure consistency between the two videos. The alignment is performed between the step-level features of one video and the frame-level features of the other. A higher alignment probability signifies a higher likelihood of step-level consistency. This probability serves as a distance quantifying the procedure correlation between these two instructional videos.

We validate our framework by performing video correlation learning on two instructional video tasks, including *sequence verification* and *action quality assessment*. Extensive experiments show the effectiveness of our framework in providing more precise and interpretable predictions of correlation.

Overall, our contributions can be summarized as follows:

- We propose a weakly supervised collaborative procedure alignment framework for instructional video correlation learning, which collaboratively extracts consistent steps in paired videos and then measures their distance through a procedure alignment process.
- Under this framework, we devise a collaborative step mining approach accounting for the semantic and temporal relationships between videos, which enables concurrent step segmentation in paired videos. In addition, we introduce a frame-to-step alignment module to furnish a precise measure of video distance.
- We apply our framework to two instructional video tasks, including sequence verification and action quality assessment. Extensive experiments showcase the superiority of our framework, demonstrating its capacity to deliver more accurate and explainable results over existing competitors.

Related Work

Video Correlation Learning Video correlation learning is a technology adopted by a wide range of work, which can be roughly divided into two streams. The first stream of work aims to learn the similarities and differences between two videos based on the given criteria. For example, in video retrieval, the method should find videos highly related to the query video. (Han, Xie, and Zisserman 2020; Zhang et al. 2020) solve the task through video representation learning by comparing the video-level features. (Jo et al. 2023b,a) adopt frame-level and temporal information for more accurate predictions. Another stream analyzes the query video with reference to the exemplar videos. For example, few-shot action recognition aims to classify the query video based on only a few support videos. Some studies (Cao et al. 2020; Hadji, Derpanis, and Jepson 2021) adopt temporal alignment between videos, while other works have explored more flexible alignment strategies based on attention mechanisms (Li et al. 2022; Liu et al. 2022a, 2023), and distribution distance (Wu et al. 2022; Wang et al. 2022a). Another example is video quality assessment. (Mozhaeva et al. 2021; Xu et al. 2020) assess the query video based on an exemplar whose quality score is given. Currently, the majority of the exploration on VCL focuses on general videos, while studies on instructional videos remain inadequate.

Instructional Video Learning Instructional videos are created to convey skills, knowledge, or procedural information, which find extensive usage in education, training, and demonstrations. Therefore, tasks related to instructional videos are gaining increasing attention. Related datasets including COIN (Tang et al. 2019), Diving (Li, Li, and Vasconcelos 2018), CSV (Qian et al. 2022), EPIC-KITCHENS (Damen et al. 2018), Assembly101 (Sener et al. 2022), and HiEve (Lin et al. 2023) have provided instructional videos in different scenarios. A prominent task for instructional videos is action segmentation, which aims to divide a video into successive steps. In this field, (Richard and Gall 2016; Singh et al. 2016; Lea et al. 2017; Lei and Todorovic 2018; Farha and Gall 2019) necessitates step boundary annotations for segmentation, while (Aakur and Sarkar 2019; Sarfraz et al. 2021; Du et al. 2022; Wang et al. 2022b) achieves unsupervised segmentation based on the semantic similarity and temporal continuity of frames within a step. Recently, novel instructional video tasks have been developed. (Qian et al. 2022) proposed the sequence verification task, aiming at verifying whether two instructional videos have the same procedure. Additionally, (Xu et al. 2022) proposed the procedure-aware action quality assessment task to score the diving sports videos based on a standard exemplar video. This paper focuses on correlation learning for instructional videos without relying on step-level annotations, which can be applied to various specific tasks.

Method

Overview

The overall pipeline of our CPA framework is illustrated in Fig. 2. The input paired sample is $\{X_1, X_2; Y_1, Y_2\}$, where

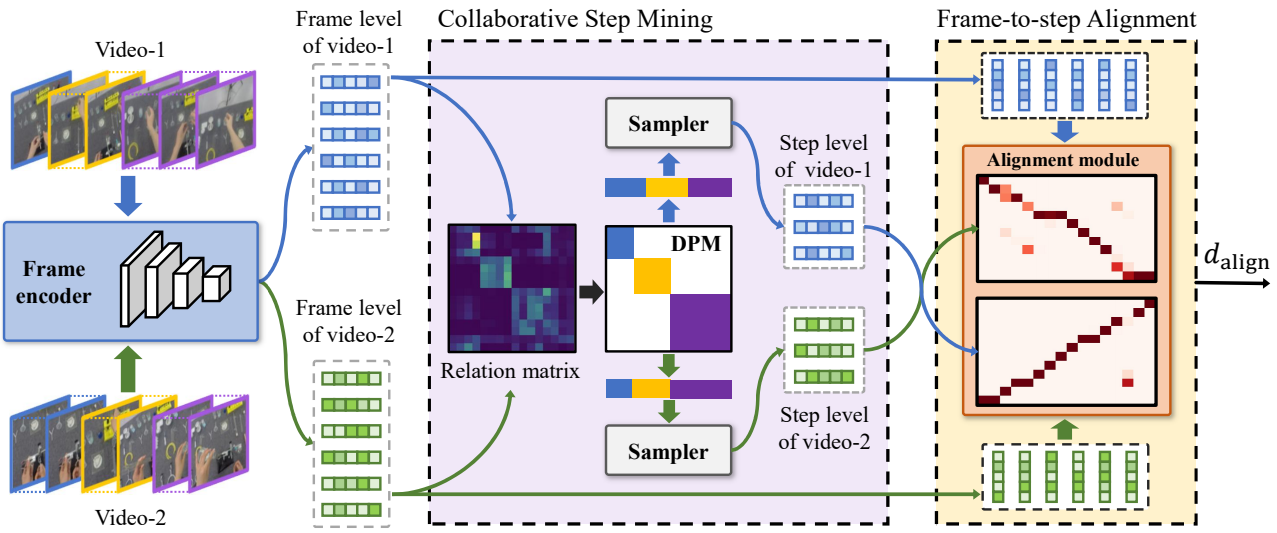

Figure 2: The pipeline of CPA. Based on the frame-level features, we utilize the collaborative step mining module (CSM) to produce the step segmentation of two videos simultaneously. Then, we can sample step-level features according to the step boundaries. Finally, we design a frame-to-step alignment (FSA) between two videos to get their correlation distance.

X_1, X_2 are two videos' frame/clip sets, and Y_1, Y_2 are their video-level labels. It is important to note that *we do not use any step-level annotation under this setting.* To begin with, the video data are fed into the frame encoder $\Psi(\cdot)$ to generate frame-level features: $F_1 = \Psi(X_1) = \{f_1^1, f_2^1, \ldots, f_T^1\}$ and $F_2 = \Psi(X_2) = \{f_1^2, f_2^2, \ldots, f_T^2\}$. Based on the frame-level features, we initially employ our collaborative step mining module to obtain a coherent step segmentation of the paired videos. For each step segment, we can sample corresponding step-level representation from frame features. Then, we apply the frame-to-step alignment between one video's frame-level features and another video's step-level features to produce the video distance.

Collaborative Step Mining

In instructional videos, we have the observation (Sarfraz et al. 2021; Du et al. 2022) that frames within the same step should have: (1) high semantic similarity and (2) continuous temporal order. Therefore, for two videos sharing the same procedure, their corresponding steps should also exhibit high semantic similarity and temporal continuity. As shown in Fig. 3(a), we can observe block-diagonal structures in the relational matrix of two consistent instructional videos, where each block represents a coherent step segment. Consequently, we can achieve a consistent step segmentation for paired videos based on the block-diagonal structure. We propose a dynamic programming-based Collaborative Step Mining (CSM) module to extract the block-diagonal structure from the relational matrix so that we can produce step segments for paired videos simultaneously. Collaboratively extracting steps from paired videos can ensure the consistency of the step-level information from two videos, which takes advantage of their internal correlation. In this section, we will elaborate on how it works.

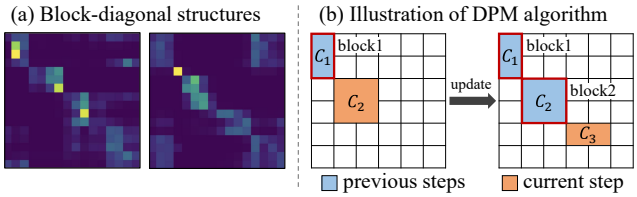

Figure 3: The illustration of: (a) the block-diagonal structure; (b) the DPM algorithm.

Relational Matrix Calculation First, we calculate two videos' relational matrix $\mathcal{M}$ by their frame-wise similarity: $\mathcal{M} = \text{Softmax}\left[(F_1 \cdot F_2^{\text{T}})/\sqrt{d}\right]$, where d denotes their feature dimension number. $\mathcal{M}_{ij}$ means the similarity between the i_{th} frames of video-1 and the j_{th} frame of video-2.

Dynamic Procedure Matching Then, we design the dynamic procedure matching (DPM) to seek the best step segmentation between two videos by adopting the idea of dynamic programming. Since it remains uncertain how many steps (blocks) should be divided, we set the step number to K, which indicates we expect to partition the relational matrix into K blocks (i.e., K steps). We define the K blocks as: $\mathcal{B} = \{\text{block}_1, \text{block}_2, \ldots, \text{block}_K\}$. Here, $\text{block}_i = (a_i, b_i, x_i, y_i)$ where (a_i, b_i) and (x_i, y_i) are the top-left and bottom-right coordinates of block_i on the relational matrix. Consequently, we can calculate a consistency score for each block as the average of all similarity values within it:

$$\mathcal{C}_i = \mathcal{C}(a_i, b_i, x_i, y_i) = \frac{\sum_{m=a_i}^{x_i} \sum_{n=b_i}^{y_i} \mathcal{M}_{mn}}{(x_i - a_i)(y_i - b_i)}. \quad (1)$$

Eq. 1 measures the step consistency between two videos within the period covered by the i-th block. A higher $\mathcal{C}_i$ indicates that these two videos are more step-wise consistent.

Algorithm 1: Step segmentation backtracing

Input: The dynamic index table D_{id}; The required step number K.

Output: Two videos' step boundaries B_1 and B_2.

Initialization: $B_1 \leftarrow \{\}$, $B_2 \leftarrow \{\}$, $k \leftarrow K$, $i \leftarrow T$, $j \leftarrow T$
repeat
  $a, b \leftarrow D_{\text{id}}[i, j, k]$
  append a to B_1
  append b to B_2
  $k \leftarrow k - 1$; $i \leftarrow a$; $j \leftarrow b$
until x==0

Given a relational matrix $\mathcal{M}$, our objective is to find a K-block partition of the matrix that maximizes the cumulative sum of the step consistency scores. This way, the partition is regarded as the optimal step segmentation of these two videos. Specifically, the objective can be expressed as:

$$\underset{\mathcal{B}}{\text{maximize}} \sum_{i=1}^{K} \mathcal{C}(a_i, b_i, x_i, y_i). \tag{2}$$

Eq. 2 involves many variables to be optimized, making it difficult to be solved directly. Fortunately, resorting to dynamic programming allows us to address it efficiently. Specifically, we devise a three-dimensional dynamic step mining algorithm equipped with a dynamic table D_{csm} to find an optimal solution for Eq. 2. $D_{\text{csm}}(i, j, k)$ represents the sum of k consistency scores composed of the first i frames $f^1_{1:i}$ of video-1 and the first j frames $f^2_{1:j}$ of video-2.

For the boundary condition $k = 1$, we can easily get the dynamic table's value by:

$$D_{\text{csm}}(i, j, 1) = \mathcal{C}(1, 1, i, j). \tag{3}$$

The cumulative sum of consistency scores at step k can be calculated by adding the consistency score in k_{th} step to the cumulative sum of $\mathcal{C}$ at step $k - 1$. Therefore, we can calculate $D_{\text{csm}}(\cdot, \cdot, k)$ from $D_{\text{csm}}(\cdot, \cdot, k - 1)$, as illustrated in Fig. 3(b). The update function of $D_{\text{csm}}(i, j, k)$ is:

$$D_{\text{csm}}(i, j, k) = \max_{a \le i, b \le j} \left[D_{\text{csm}}(a, b, k - 1) + \mathcal{C}(a, b, i, j) \right] \tag{4}$$

The value $D_{\text{csm}}(T, T, K)$ means the maximal consistency score of partitioning video-1 and video-2 into K steps.

Step Segmentation Backtracing Upon obtaining the optimal score, we need to retrace the alignment decisions to achieve a coherent step segmentation. Therefore, during the forward phase, we also need to record the selected indices of each update in the dynamic index table D_{id}:

$$D_{\text{id}}(i, j, k) = \underset{a, b;\ a \le i, b \le j}{\arg\max} \left[D_{\text{csm}}(a, b, k - 1) + \mathcal{C}(a, b, i, j) \right] \tag{5}$$

Then, starting from the end $D_{\text{id}}(T, T, K)$, we trace back to get each step's boundary, which is presented in Algorithm 1.

Through backtracing, we can get two videos' step boundaries. Within each step, we randomly sample one frame-level feature to act as the representation of the step. Then we can get the step-level features $S_1 = \{s^1_1, s^1_2, \ldots, s^1_K\}$ and $S_2 = \{s^2_1, s^2_2, \ldots, s^2_K\}$, which is used in the following frame-to-step alignment stage.

Frame-to-Step Alignment

Based on the collaboratively mined steps of paired videos, we further propose a frame-to-step alignment (FSA) module. It calculates the probability of aligning the step-level features of one video with the frame-level features of another. A larger alignment probability indicates they are more likely to be step-wise consistent. The motivation behind this *cross-verification* design is: *if two videos are step-level consistent, then we can achieve a good alignment between the frames of video-1 and the steps of video-2, as video-2's step is extracted under the guidance of video-1.* Experiments in Sec. also prove this argument.

Given the frame-level features $F = \{f_1, \ldots f_T\}$ of one video and the step-level features $S = \{s_1, \ldots s_K\}$ of another video, the alignment between them is a dense-to-sparse ($T > K$) mapping that includes many possibilities. We denote one possible alignment as $\pi = (\pi_1, \pi_2, \ldots, \pi_T)$, which represents the frame-to-step assignments. The probability of alignment π is calculated as:

$$p(\pi|F) = \prod_{t=1}^{T} p^t_{\pi_t},\ \pi_t \in \{1, 2, \ldots, K\} \tag{6}$$

where $p^t_{\pi_t}$ means the probability of assigning frame t to step π_t. Here π_t is one of the K steps. For a given step-level features S, we can calculate the cumulative probability of all possible alignments with:

$$P(S|F) = \sum_{\pi \in \Omega(F,S)} p(\pi|F) \tag{7}$$

where $\Omega(F, S)$ is the set of all possible frame-to-step alignments given F and S. It is hard to compute all possible alignments, but we can also resort to dynamic programming to make the computation tractable. We first compute the frame-to-step probability matrix $\mathcal{W} = \text{Softmax}\left[(F \cdot S^{\text{T}})/\sqrt{d}\right]$. From $\mathcal{W}$, we can get the frame-to-step assignment probability $p^t_{L_k} = \mathcal{W}_{tk}$. Then, we design a two-dimensional dynamic table D_{fsa} with shape $T \times K$ to record the alignment probabilities. $D_{\text{fsa}}(t, k)$ means the probability of aligning frames $F_{1:t}$ to steps $S_{1:k}$.

First, we define the boundary conditions: $D_{\text{fsa}}(t, 1) = 1$. Then, we need to define the update function for $D_{\text{fsa}}(t, k)$. At each timestamp t, we have two choices. If frame t does not switch steps, then the probability comes from $D_{\text{fsa}}(t - 1, k)$. If frame t switch step compared with the previous step, then the probability comes from $D_{\text{fsa}}(t - 1, k - 1)$. Therefore, we can update the dynamic table with Eq. 8:

$$D_{\text{fsa}}(t, k) = \mathcal{W}_{tk} \left[D_{\text{fsa}}(t - 1, k) + D_{\text{fsa}}(t - 1, k - 1) \right] \tag{8}$$

Our goal is to get the probability of aligning frame features F to step features S, which is:

$$p(S|F) = D_{\text{fsa}}(T, K) \tag{9}$$

where $D_{\text{fsa}}(T, K)$ represents the complete frame-to-step alignment probability.

We use the negative log-likelihood value as the procedure correlation distance. For two videos' frame-level features F_1, F_2 and step-level features S_1, S_2, we compute the alignment scores from two directions to maintain symmetry. The final distance d_{align} is:

$$d_{\text{align}} = -\frac{1}{2}\left[\log P(S_2|F_1) + \log P(S_1|F_2)\right] \tag{10}$$

Optimization

During training, we adopt three loss functions for optimization. First, from the CSM, we can get the cumulative sum of consistency score: $D_{\text{csm}}(T, T, K)$. We design a step-enhancing loss $\mathcal{L}_{\text{step}}$ to maximize the consistency scores of positive pairs to enhance the frame similarity within the same step. It can be formulated as:

$$\mathcal{L}_{\text{step}} = -D_{\text{csm}}(T, T, K) \tag{11}$$

Second, from the FSA, we can get the procedure correlation distance d_{align} between two videos. We design a aligning loss $\mathcal{L}_{\text{align}}$ to minimize the d_{align} of positive pairs:

$$\mathcal{L}_{\text{align}} = d_{\text{align}} \tag{12}$$

The third loss $\mathcal{L}_{\text{task}}$ changes with the tasks. For sequence verification, we follow (Qian et al. 2022) to adopt procedure classification as an auxiliary task and its loss is:

$$\mathcal{L}_{\text{task}} = \text{Cross-Entropy}\left(pred, Y\right) \tag{13}$$

where $pred$ and Y are the procedure predictions and labels. For action quality assessment, we follow (Xu et al. 2022) to optimize the mean squared error between the ground truth y_X and predicted action score $\hat{y}_X$:

$$\mathcal{L}_{\text{task}} = \|\hat{y}_X - y_X\|^2 \tag{14}$$

Hence, the overall loss $\mathcal{L}$ for optimization is:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \mathcal{L}_{\text{step}} + \mathcal{L}_{\text{align}} \tag{15}$$

Experiments

Implementation Details

We implement our method on two instructional video tasks, including sequence verification and action quality assessment. For sequence verification, our implementation adheres to (Qian et al. 2022) for a fair comparison. Besides ResNet-50, we additionally utilize X3D-m pretrained on Kinetics-400 (Kay et al. 2017) as our backbone for experiments. For action quality assessment, our implementation sticks to the method described in (Xu et al. 2022) for a fair comparison. For all tasks, we trained our model on 2 NVIDIA TITAN RTX GPUs with batch size 8.

Sequence Verification

Goal Sequence verification (SV) aims to verify whether two instructional videos have identical procedures. Two videos executing the same steps in the same order form a positive pair, otherwise negative. The method should give a verification distance between each video pair and give the prediction by thresholding the distance. We conduct experiments on three sequence verification datasets (CSV, Diving-SV, and COIN-SV) proposed by (Qian et al. 2022). We adopt Area Under ROC Curve (AUC) for evaluation. A higher AUC indicates better performance.

Method	Text anno.	AUC		
		CSV	DivingSV	COINSV
TRN	×	80.32	80.69	57.19
Video-Swin	×	54.06	73.10	43.70
CAT	×	83.02	83.11	51.13
OTAM	×	69.03	77.86	50.55
TAP	×	73.29	75.47	47.45
Drop-DTW	×	84.86	74.12	53.33
CLIP+TE+MLP	✓	79.38	83.48	48.50
WeakSVR	✓	86.92	86.09	**59.57**
CPA+R50 (ours)	×	**88.14**	84.29	57.57
CPA+X3D (ours)	×	86.06	**88.11**	57.55

Table 1: Results of sequence verification.

Competitors We compare our method with various approaches, including (1) video methods: TRN (Zhou et al. 2018), TSM (Lin, Gan, and Han 2019), Video-swin (Liu et al. 2022b), CAT (Qian et al. 2022); (2) sequence alignment methods: OTAM (Cao et al. 2020), TAP (Pan et al. 2021), Drop-DTW (Dvornik et al. 2021); (3) visual-language methods: CLIP+TE+MLP (Radford et al. 2021), WeakSVR (Dong et al. 2023). Visual-language methods are pretrained on CLIP (Radford et al. 2021), while other methods are pretrained on Kinetics-400 (K-400) (Kay et al. 2017). For video and visual-language methods, we follow (Dong et al. 2023)'s setting to calculate the normalized L2 distance between two videos' representations as their verification distance. For sequence alignment methods, we use their alignment distance d_{align} as the verification distance, which is the same as our method.

Results The results are presented in Tab. 1. For CSV, our method achieves the best performance (88.14%). For Diving-SV, our method gets the best performance among methods with a 2D backbone. Furthermore, applying the 3D backbone X3D further boosts our performance to the new state-of-the-art result (88.11%). For COIN-SV, our approach achieves the best among visual-based methods. It is worth noticing that our method remains competitive even among visual-language methods, despite these methods being equipped with extra text narrations on procedures.

Action Quality Assessment

Goal In action quality assessment (AQA), we adopt (Xu et al. 2022)'s setting, where an exemplar video and its score are given. Then for a query video with the same procedure, the method should predict its score based on the exemplar. We conduct experiments on the FineDiving dataset. We use Spearman's rank correlation (ρ) and relative $\ell 2$-distance (R-$\ell 2$) for evaluation. A Higher ρ and lower R-$\ell 2$ indicate better performance. Average Intersection over Union (AIoU) is adopted to evaluate the procedure segmentation.

Method	Step anno.	AIoU@ 0.5	AIoU@ 0.75	ρ	R-ℓ2 ($\times$100)
USDL	×	/	/	0.8913	0.3822
MUSDL	×	/	/	0.8978	0.3704
CoRe	×	/	/	0.9061	0.3615
Lian et al.	×	/	/	0.9222	0.3304
PECoP	✓	-	-	0.9315	-
TSA	✓	82.51	34.31	0.9203	0.3420
TSA*	✓	93.23	**53.39**	0.9302	0.3154
CPA (ours)	×	**94.28**	21.14	**0.9364**	**0.2909**

Table 2: Results of action quality assessment on FineDiving. / indicates "without procedure segmentation". * means our implementation.

Competitors We compare our approach with various advanced AQA methods, including (1) non-procedure methods: USDL, MUSDL (Tang et al. 2020), CoRe (Yu et al. 2021), Lian et al. (Lian and Shao 2023); (2) procedure-aware method: TSA (Xu et al. 2022), PECoP (Dadashzadeh et al. 2023). Vanilla TSA's procedure segmentation module is supervised by step-level annotations. Here we replace TSA's procedure segmentation module with our CPA *to achieve segmentation without the help of step-level annotations.*

Results The results are presented in Tab. 2. Our method achieves new state-of-the-art results. Note that our method outperforms TSA without step-level annotations. The reason is that TSA strictly divides the procedure into three steps: *take-off*, *flight*, and *entry*, which might not be the most suitable division for assessment. In contrast, our CPA can flexibly adjust the procedure segmentation through optimization to get better assessments. Furthermore, by observing the AIoU, our CPA can get satisfying coarse-grain procedure segmentation (AIoU@0.5=94.28). Note that we uniformly divide CPA's segments into three steps for calculating AIoU. Finally, we can find that procedure-aware methods generally outperform non-procedure methods, which further emphasizes the importance of procedural knowledge for accurate predictions on instructional videos.

In-Depth Analysis

Ablations on Main Components We conduct an ablation analysis of the proposed modules on SV and AQA. Results summarized in Tab. 3 demonstrate the effectiveness of each module. Note that using FSA individually means we just uniformly sample K frame features as the step-level features. In the first row of SV, we use the video-level features to calculate the verification distance. In the first two rows of AQA, we uniformly divide the video into K segments. According to the results, using FSA enable the method to learn more distinctive frame-level features and improve performance on both tasks. Moreover, introducing CSM contributes significantly to improvements, highlighting the importance of step information in this task and the effectiveness of our CSM on step mining.

Analysis on Step Number In real-world videos, the definition of a step is flexible and ambiguous, where several adjacent steps can be reorganized as one step at a coarser level. Our CSM can provide step-mining results across different step granularity. Fig. 7 illustrates the results of CSM under different step numbers K, where each block signifies a step. Our method degrades to frame-to-frame comparison when $K = T$. We further conduct sensitivity analysis on SV and AQA by adjusting K. From Tab. 4, we can observe that the performances first rise and then drop as the step number increases. The performances peak at $K = 13$ for CSV and $K = 6$ for FineDiving, whose trend corresponds with the maximum step number in CSV (16 steps) and FineDiving (5 steps). Therefore, choosing the proper step number depending on the dataset can lead to better performance.

CSM	FSA	SV on CSV	AQA on FineDiving	
		AUC	ρ	R-ℓ2
		81.65	0.9221	0.3456
	✓	86.23	0.9275	0.3288
✓	✓	**88.14**	**0.9364**	**0.2909**

Table 3: Ablation study of the proposed modules. CSM: Collaborative Step Mining; FSA: Frame-to-Step Alignment.

Sequence Verification					
Metric	Step number K				
	11	12	13	14	15
AUC	84.91	87.00	**88.14**	87.94	84.67
Action Quality Assessment					
Metric	Step number K				
	3	4	5	6	7
ρ	0.9302	0.9309	0.9316	**0.9364**	0.9340
R-ℓ2	0.3098	0.3202	0.3119	**0.2909**	0.2943

Table 4: Sensitivity analysis on step number K.

Collaborative Step Mining Visualization We visualize step segmentation produced by CSM in Fig. 4(a) on "diving" and "changing watch battery". We set $T = 16$, $K = 6$, and manually name the text descriptions of steps for better visualization. By observation, some steps (e.g., "35som" and "install back cover") are further divided into more steps in our six-step segmentation. The reason is that step segmentation can sometimes be flexible and multi-grained. CSM can well identify consistent steps, thereby providing dependable guidance for subsequent frame-to-step alignment.

Frame-to-Step Alignment Visualization As depicted in Fig. 6, we show the frame-to-step assignment probabilities for both positive and negative pairs on SV. By observation, the positive pairs can form continuous probability paths from the top-left corner to the bottom-right corner, while negative pairs will exhibit obvious discontinuities. Consequently, paired videos adhering to a chronological procedural alignment can achieve minimal distance using our proposed method, which validates our motivation for the cross-verification design in frame-to-step alignment.

Multi-Level Feature Visualization We demonstrate our method's feature learning ability by using t-SNE to visu-

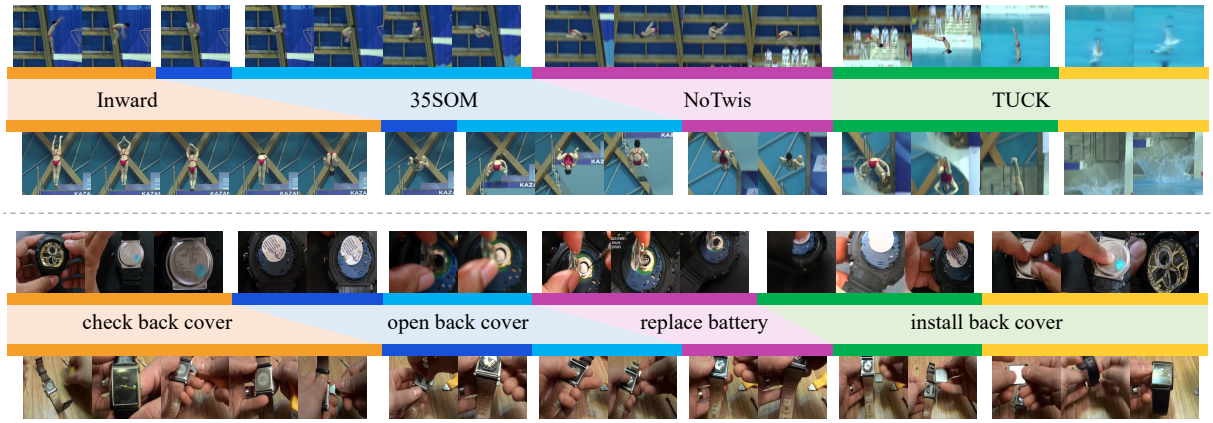

Figure 4: Visualization of collaborative step segmentation with $T = 16$ and $K = 6$.

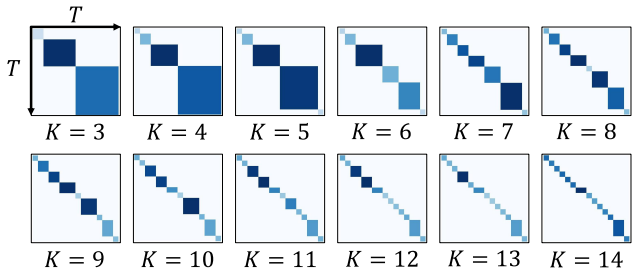

Figure 5: The block-diagonal structures representing the multi-grained step segmentation, experimented on CSV.

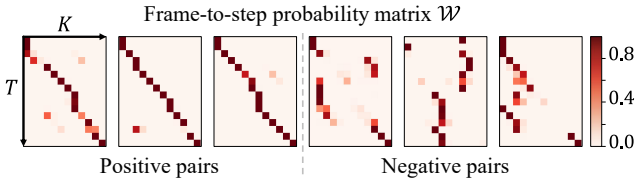

Figure 6: The frame-to-step aligning probability on CSV, with three positive and negative pairs. The vertical and horizontal axes represent T frames and K steps respectively.

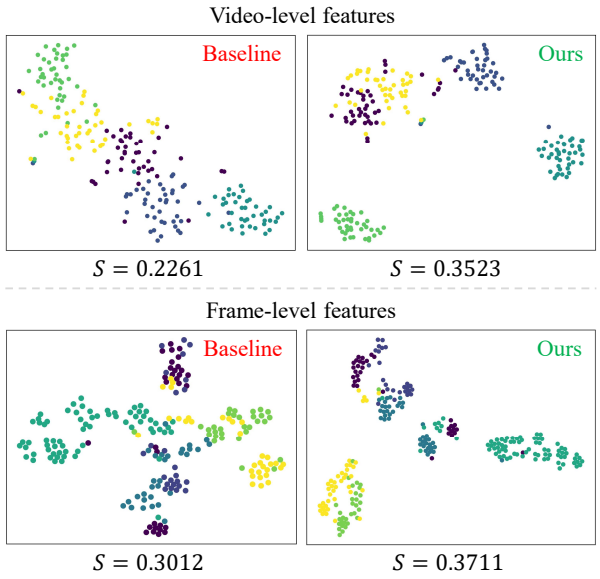

Figure 7: The t-SNE visualization of multi-level features and their respective Silhouette scores S. top: the video-level features with colors indicating video classes. Bottom: the frame-level features with colors indicating step classes.

alize both the video-level and frame-level features trained on CSV, which is shown in Fig. 4(b). For video-level features, the color represents its video-level class. For frame-level features, the color represents its step, which is manually annotated for visualization. Besides qualitative visualization, we also adopt the Silhouette score S to quantify the clustering effect. A higher Silhouette score indicates a better clustering outcome. From Fig. 4(b), compared with baseline, our method improves the feature clustering effects on both the video-level features ($0.2261 \rightarrow 0.3523$) and the frame-level features ($0.3012 \rightarrow 0.3711$). This result indicates that our method can learn more distinctive multi-level features, which is beneficial for instructional video analysis.

Conclusion

In this paper, we propose a weakly supervised framework for procedure-aware correlation learning on instructional videos, named the Collaborative Procedure Alignment (CPA). Under this framework, we first design the collaborative step mining (CSM) module to simultaneously produce step segmentation for paired videos and get the representative step-level features. Furthermore, we propose the frame-to-step alignment (FSA) module to calculate the correlation distance between videos. Extensive and in-depth experiments on two instructional video tasks showcase the superiority of our framework, demonstrating its capacity to deliver more explainable and accurate understandings on complex instructional videos.

Acknowledgments

The paper is supported in part by the National Natural Science Foundation of China (No. 62325109, U21B2013, 61971277) and the Lenovo Academic Collaboration Project.

References

Aakur, S. N.; and Sarkar, S. 2019. A perceptual prediction framework for self supervised event segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1197–1206.

Behrmann, N.; Golestaneh, S. A.; Kolter, Z.; Gall, J.; and Noroozi, M. 2022. Unified fully and timestamp supervised temporal action segmentation via sequence to sequence translation. In *European Conference on Computer Vision*, 52–68. Springer.

Ben-Ari, R.; Nacson, M. S.; Azulai, O.; Barzelay, U.; and Rotman, D. 2021. TAEN: temporal aware embedding network for few-shot action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2786–2794.

Cao, K.; Ji, J.; Cao, Z.; Chang, C.-Y.; and Niebles, J. C. 2020. Few-shot video classification via temporal alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10618–10627.

Dadashzadeh, A.; Duan, S.; Whone, A.; and Mirmehdi, M. 2023. PECoP: Parameter Efficient Continual Pretraining for Action Quality Assessment. *arXiv preprint arXiv:2311.07603*.

Damen, D.; Doughty, H.; Farinella, G. M.; Fidler, S.; Furnari, A.; Kazakos, E.; Moltisanti, D.; Munro, J.; Perrett, T.; Price, W.; et al. 2018. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European conference on computer vision (ECCV)*, 720–736.

Dong, S.; Hu, H.; Lian, D.; Luo, W.; Qian, Y.; and Gao, S. 2023. Weakly Supervised Video Representation Learning with Unaligned Text for Sequential Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2437–2447.

Du, Z.; Wang, X.; Zhou, G.; and Wang, Q. 2022. Fast and unsupervised action boundary detection for action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3323–3332.

Dvornik, M.; Hadji, I.; Derpanis, K. G.; Garg, A.; and Jepson, A. 2021. Drop-dtw: Aligning common signal between sequences while dropping outliers. *Advances in Neural Information Processing Systems*, 13782–13793.

Farha, Y. A.; and Gall, J. 2019. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3575–3584.

Hadji, I.; Derpanis, K. G.; and Jepson, A. D. 2021. Representation learning via global temporal alignment and cycle-consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11068–11077.

Han, T.; Xie, W.; and Zisserman, A. 2020. Self-supervised co-training for video representation learning. *Advances in Neural Information Processing Systems*, 33: 5679–5690.

Han, T.; Xie, W.; and Zisserman, A. 2022. Temporal alignment networks for long-term video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2906–2916.

Jo, W.; Lim, G.; Hwang, Y.; Lee, G.; Kim, J.; Yun, J.; Jung, J.; and Choi, Y. 2023a. Simultaneous Video Retrieval and Alignment. *IEEE Access*, 11: 28466–28478.

Jo, W.; Lim, G.; Lee, G.; Kim, H.; Ko, B.; and Choi, Y. 2023b. VVS: Video-to-Video Retrieval with Irrelevant Frame Suppression. *arXiv preprint arXiv:2303.08906*.

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

Lea, C.; Flynn, M. D.; Vidal, R.; Reiter, A.; and Hager, G. D. 2017. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 156–165.

Lei, P.; and Todorovic, S. 2018. Temporal deformable residual networks for action segmentation in videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6742–6751.

Li, S.; Liu, H.; Qian, R.; Li, Y.; See, J.; Fei, M.; Yu, X.; and Lin, W. 2022. TA2N: Two-stage action alignment network for few-shot action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 1404–1411.

Li, Y.; Li, Y.; and Vasconcelos, N. 2018. Resound: Towards action recognition without representation bias. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 513–528.

Lian, P.-X.; and Shao, Z.-G. 2023. Improving action quality assessment with across-staged temporal reasoning on imbalanced data. *Applied Intelligence*, 1–12.

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.

Lin, W.; Liu, H.; Liu, S.; Li, Y.; Xiong, H.; Qi, G.; and Sebe, N. 2023. HiEve: A Large-Scale Benchmark for Human-Centric Video Analysis in Complex Events. *International Journal of Computer Vision*, 131(11): 2994–3018.

Liu, H.; Lin, W.; Chen, T.; Li, Y.; Li, S.; and See, J. 2023. Few-shot Action Recognition via Intra-and Inter-Video Information Maximization. *arXiv preprint arXiv:2305.06114*.

Liu, H.; Lv, W.; See, J.; and Lin, W. 2022a. Task-adaptive Spatial-Temporal Video Sampler for Few-shot Action Recognition. In *Proceedings of the 30th ACM International Conference on Multimedia*, 6230–6240.

Liu, Z.; Ning, J.; Cao, Y.; Wei, Y.; Zhang, Z.; Lin, S.; and Hu, H. 2022b. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3202–3211.

Mozhaeva, A.; Streeter, L.; Vlasuyk, I.; and Potashnikov, A. 2021. Full reference video quality assessment metric on base human visual system consistent with PSNR. In *2021 28th Conference of Open Innovations Association (FRUCT)*, 309–315. IEEE.

Pan, F.; Xu, C.; Guo, J.; and Guo, Y. 2021. Temporal Alignment Prediction for Few-Shot Video Classification. *arXiv preprint arXiv:2107.11960*.

Park, J.; Lee, J.; Kim, I.-J.; and Sohn, K. 2022. Probabilistic representations for video contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14711–14721.

Qian, R.; Meng, T.; Gong, B.; Yang, M.-H.; Wang, H.; Belongie, S.; and Cui, Y. 2021. Spatiotemporal contrastive video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6964–6974.

Qian, Y.; Luo, W.; Lian, D.; Tang, X.; Zhao, P.; and Gao, S. 2022. SVIP: Sequence VerIfication for Procedures in Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19890–19902.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Richard, A.; and Gall, J. 2016. Temporal action detection using a statistical language model. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3131–3140.

Sarfraz, S.; Murray, N.; Sharma, V.; Diba, A.; Van Gool, L.; and Stiefelhagen, R. 2021. Temporally-weighted hierarchical clustering for unsupervised action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11225–11234.

Sener, F.; Chatterjee, D.; Shelepov, D.; He, K.; Singhania, D.; Wang, R.; and Yao, A. 2022. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21096–21106.

Singh, B.; Marks, T. K.; Jones, M.; Tuzel, O.; and Shao, M. 2016. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1961–1970.

Tang, Y.; Ding, D.; Rao, Y.; Zheng, Y.; Zhang, D.; Zhao, L.; Lu, J.; and Zhou, J. 2019. Coin: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1207–1216.

Tang, Y.; Ni, Z.; Zhou, J.; Zhang, D.; Lu, J.; Wu, Y.; and Zhou, J. 2020. Uncertainty-aware score distribution learning for action quality assessment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9839–9848.

Wang, X.; Jabri, A.; and Efros, A. A. 2019. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2566–2576.

Wang, X.; Zhang, S.; Qing, Z.; Tang, M.; Zuo, Z.; Gao, C.; Jin, R.; and Sang, N. 2022a. Hybrid relation guided set matching for few-shot action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19948–19957.

Wang, Z.; Chen, H.; Li, X.; Liu, C.; Xiong, Y.; Tighe, J.; and Fowlkes, C. 2022b. Sscap: Self-supervised co-occurrence action parsing for unsupervised temporal action segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1819–1828.

Wu, J.; Zhang, T.; Zhang, Z.; Wu, F.; and Zhang, Y. 2022. Motion-modulated temporal fragment alignment network for few-shot action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9151–9160.

Xu, J.; Rao, Y.; Yu, X.; Chen, G.; Zhou, J.; and Lu, J. 2022. Finediving: A fine-grained dataset for procedure-aware action quality assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2949–2958.

Xu, M.; Chen, J.; Wang, H.; Liu, S.; Li, G.; and Bai, Z. 2020. C3DVQA: Full-reference video quality assessment with 3D convolutional neural network. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4447–4451. IEEE.

Yu, X.; Rao, Y.; Zhao, W.; Lu, J.; and Zhou, J. 2021. Group-aware contrastive regression for action quality assessment. In *Proceedings of the IEEE/CVF international conference on computer vision*, 7919–7928.

Zhang, C.; Hu, B.; Suo, Y.; Zou, Z.; and Ji, Y. 2020. Large-scale video retrieval via deep local convolutional features. *Advances in Multimedia*, 2020: 1–8.

Zhou, B.; Andonian, A.; Oliva, A.; and Torralba, A. 2018. Temporal relational reasoning in videos. In *Proceedings of the European conference on computer vision (ECCV)*, 803–818.

Zhu, L.; and Yang, Y. 2018. Compound memory networks for few-shot video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 751–766.