Motion Aware Self-Supervision for Generic Event Boundary Detection

Ayush K. Rai, Tarun Krishna, Julia Dietlmeier, Kevin McGuinness*, Alan F. Smeaton*, Noel E. O'Connor*
Insight SFI Centre for Data Analytics, Dublin City University (DCU)
ayush.rai3@mail.dcu.ie

Abstract

The task of Generic Event Boundary Detection (GEBD) aims to detect moments in videos that are naturally perceived by humans as generic and taxonomy-free event boundaries. Modeling the dynamically evolving temporal and spatial changes in a video makes GEBD a difficult problem to solve. Existing approaches involve very complex and sophisticated pipelines in terms of architectural design choices, hence creating a need for more straightforward and simplified approaches. In this work, we address this issue by revisiting a simple and effective self-supervised method and augment it with a differentiable motion feature learning module to tackle the spatial and temporal diversities in the GEBD task. We perform extensive experiments on the challenging Kinetics-GEBD and TAPOS datasets to demonstrate the efficacy of the proposed approach compared to the other self-supervised state-of-the-art methods. We also show that this simple self-supervised approach learns motion features without any explicit motion-specific pretext task. Our results can be reproduced on github.

1. Introduction

Modeling videos using deep learning methods in order to learn effective global and local video representations is an extremely challenging task. Current state-of-the-art video models [18] are built upon a limited set of predefined action classes and usually process short clips followed by a pooling operation to generate global video-level predictions. Other mainstream computer vision tasks for video processing have been mainly focused on action anticipation [56, 1], temporal action detection [7, 22], temporal action segmentation [43, 41] and temporal action parsing [62, 67]. However, only limited attention has been given to understanding long form videos. Cognitive scientists [74] have observed that humans perceive videos by breaking them down into shorter temporal units, each carrying a semantic meaning and can also reason about them. This creates a need to investigate research problems to detect temporal boundaries in videos that is consistent with their semantic validity and interpretability from a cognitive point of view.

To this end, the GEBD task was recently introduced in [68][1] with an objective to study the long form video understanding problem through the lens of a human perception mechanism. GEBD aims at identifying changes in content, independent of changes in action, brightness, object, etc., i.e. generic event boundaries, making it different to tasks such as video localization [77]. Video events could indicate completion of goals or sub-goals, or occasions where it becomes difficult for humans to predict what will happen next. The recently released Kinetics-GEBD dataset [68] is the first dataset specific to the GEBD task. It is annotated by 5 different event boundary annotators, thereby capturing the subtlety involved in human perception and making it the dataset with the greatest number of temporal boundaries ($8\times$ EPIC-Kitchen-100 [11] and $32\times$ ActivityNet [16]). The primary challenge in the GEBD task is to effectively model generic spatial and temporal diversity as described in DDM-Net [72]. Spatial diversity is primarily the result of both low-level changes, e.g. changes in brightness or appearance, and high-level changes, e.g., changes in camera angle, or appearance and disappearance of the dominant subject. Temporal diversity, on the other hand, can be attributed to changes in action or changes by the object of interaction with different speeds and duration, depending on the subject. These spatio-temporal diversities make GEBD a difficult problem to address.

In this work, to address the biased nature of video models trained over predefined classes in a supervised setting, and the spatial diversity in GEBD, we leverage the power of self-supervised models. Self-supervised techniques like TCLR [12] and CCL [38] have achieved breakthrough results on various downstream tasks for video understanding. The representations learned using self-supervised learning (SSL) methods are not biased towards any predefined action class making SSL methods an ideal candidate for the GEBD task. In addition, in order to characterize temporal diversity in GEBD, learning motion information is essential to capture

*Equal supervision

[1] LOVEU@CVPR2021, LOVEU@CVPR2022

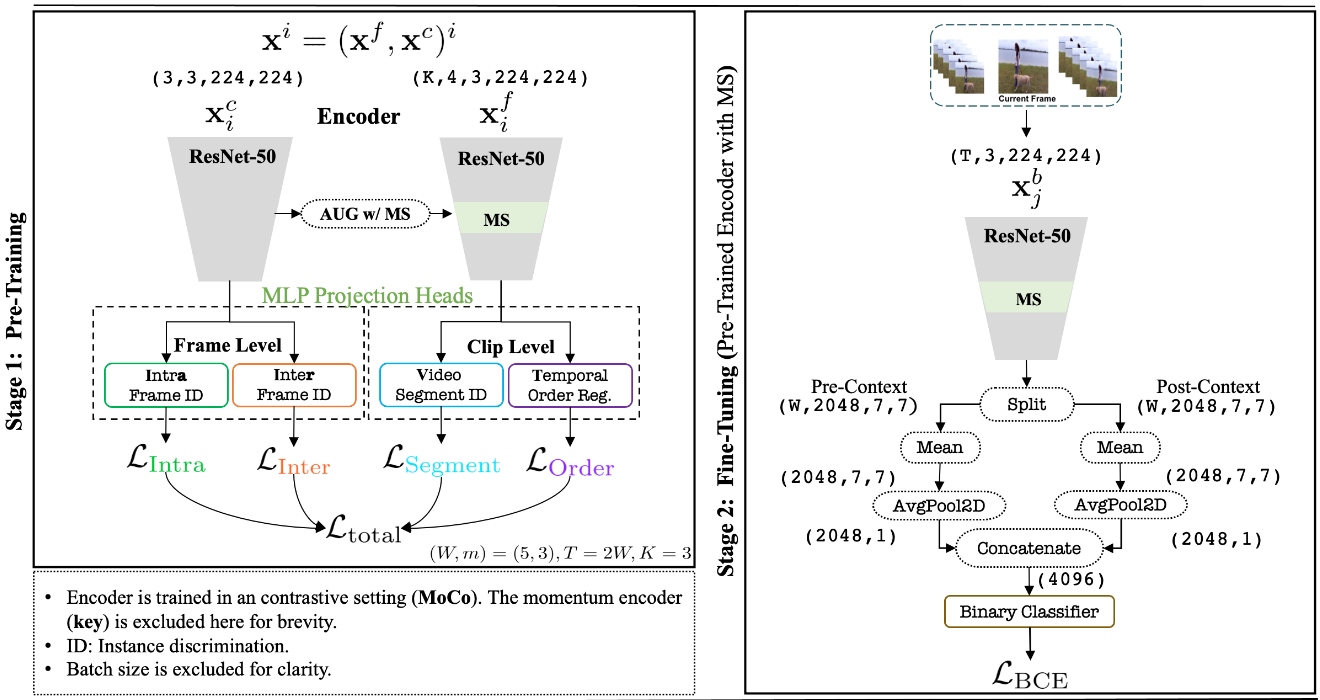

Figure 1. The overall architecture consists of two stages: a) **Stage 1** involves the pre-training of the modified ResNet50 encoder (augmented with a *MotionSqueeze* layer) with four pretext tasks using a contrastive learning based objective; b) **Stage 2** consists of fine tuning of the encoder on the downstream GEBD task. Refer to Table 1 in Supplementary material for encoder details.

the fine-grained temporal variations that occur during the change of action scenarios. Previous methods in video modeling learn temporal motion cues by pre-computing the optical flow [53, 52, 54] between consecutive frames, which is done externally and requires substantial computation. Alternatively, methods such as those described in [32, 21] estimate optical flow internally by learning visual correspondences between images. The motion features learnt on-the-fly can also be used for downstream applications such as action recognition as illustrated in [82, 42].

This presents an interesting research question: how can we develop a SSL framework for video understanding that accounts for both appearance and motion features? Do we need an explicit motion-specific training objective or can this be implicitly achieved? We answer these questions by rethinking SSL by reformulating the training objective proposed in VCLR [40] at clip-level and further integrating it with a differentiable motion estimation layers using the *MotionSqueeze* (MS) module introduced in [42] to jointly learn appearance and motion features for videos. To summarise, the main contributions of our work are as follows:

- We revisit a simple self-supervised method VCLR [40] with a noticeable change by modifying its pretext tasks by splitting them into frame-level and clip-level to learn effective video representations (cVCLR) . We further augment the encoder with a differentiable motion feature learning module for GEBD.
- We conduct exhaustive evaluation on the Kinetics-GEBD and TAPOS datasets and show that our approach achieves comparable performance to the self-supervised state-of-the-art methods without using enhancements like model ensembles, pseudo-labeling or the need for other modality features (e.g. audio).
- We show that the model can learn motion features under self-supervision even without having any explicit motion specific pretext task.

2. Related Work

2.1. Generic Event Boundary Detection.

The task of GEBD [68] is similar in nature to the Temporal Action Localization (TAL) task, where the goal is to localize the start and end points of an action occurrence along with the action category. Initial attempts to address GEBD were inspired from popular TAL solvers including the boundary matching networks (BMN) [52] and BMN-StartEnd [68], which generates proposals with precise temporal boundaries along with reliable confidence scores. Shou *et al.* [68] introduced a supervised baseline Pairwise Classifier (PC), which considers GEBD as a framewise binary classification problem (boundary or not) by having a simple linear classifier that uses concatenated average features around the neighbourhood of a candidate frame. However, since GEBD is a new task, most of the current methods are an extension of state-of-the-art video understanding tasks, which overlook the subtle differentiating characteristics of GEBD. Hence there is a necessity for GEBD specialized solutions.

DDM-Net [72] applied progressive attention on multi-level dense difference maps (DDM) to characterize motion patterns and jointly learn motion with appearance cues in a supervised setting. However, we learn generic motion features by augmenting the encoder with a MS module in a self-supervised setting. Hong *et al.* [29] used a cascaded temporal attention network for GEBD, while Rai *et al.* [64] explored the use of spatio-temporal features using two-stream networks. Li *et al.* [49] designed an end-to-end spatial-channel compressed encoder and temporal contrastive module to determine event boundaries. Recently, SC-Transformer [48] introduced a structured partition of sequences (SPoS) mechanism to learn structured context using a transformer based architecture for GEBD and augmented it with the computation of group similarity to learn distinctive features for boundary detection. One advantage of SC-Transformer is that it is independent of video length and predicts all boundaries in a single forward pass by feeding in 100 frames, however it requires substantial memory and computational resources.

Regarding unsupervised GEBD approaches, a shot detector library[2] and PredictAbility (PA) have been investigated in [68]. The authors of UBoCo [36, 35] proposed a novel supervised/unsupervised method that applies contrastive learning to a TSM[3] based intermediary representation of videos to learn discriminatory boundary features. UBoCo's recursive TSM[3] parsing algorithm exploits generic patterns and detects very precise boundaries. However, they pre-process all the videos in the dataset to have the same frames per second (fps) value of 24, which adds a computational overhead. Furthermore, like the SC-Transformer, UBoCo inputs the frames representing the whole video at once, whereas in our work we use raw video signals for pre-training and only the context around the candidate boundary as input to the GEBD task. TeG [63] proposed a generic self-supervised model for video understanding for learning persistent and more fine-grained features and evaluated it on the GEBD task. The main difference between TeG and our work is that TeG uses a 3D-ResNet-50 encoder as their backbone, which makes the training computationally expensive, whereas we use 2D-ResNet-50 model and modify it by adding temporal shift module (TSM[4]) [51] to achieve the same effect as 3D convolution while keeping the complexity of a 2D CNN.

GEBD can be used as a preliminary step in a larger downstream application, e.g. video summarization, video captioning [76], or ad cue-point detection [8]. It is, therefore, important that the GEBD model not add excessive computational overhead to the overall pipeline, unlike many of the examples of related work presented here.

[2] `https://github.com/Breakthrough/PySceneDetect`
[3] Temporal Self-Similarity Matrix
[4] Temporal Shift Module

2.2. SSL for video representation learning.

Self-supervision has become the new norm for learning representations given its ability to exploit unlabelled data [59, 23, 15, 2, 5, 81, 4, 9, 60, 39, 14]. Recent approaches devised for video understanding can be divided into two categories based on the SSL objective, namely pretext task based and contrastive learning based.

Pretext task based. The key idea here is to design a pretext task for which labels are generated in an online fashion, referred to as *pseudo labels*, without any human annotation. Examples include: predicting correct temporal order [58], Video Rot-Net [34] for video rotation prediction, clip order prediction [78], odd-one-out networks [20], sorting sequences [45], and pace prediction [75][5]. All these approaches exploit raw spatio-temporal signals from videos in different ways based on pretext tasks and consequently learn representations suitable for varied downstream tasks.

Contrastive learning based. Contrastive learning approaches bring semantically similar objects, clips, etc., close together in the embedding space while contrasting them with negative samples, using objectives based on some variant of Noise Contrastive Estimation (NCE) [24]. The Contrastive Predictive Coding (CPC) approach [60] for images was extended to videos in DPC [26] and MemDPC [27], which augments DPC with the notion of *compressed memory*. Li *et al.* [73] extends the contrasting mutli-view framework for inter-intra style video representation, while Kong *et al.* [38] combine ideas from *cycle-consistency* with contrastive learning to propose *cycle-contrast*. Likewise, Yang *et al.* [79] exploits visual tempo in a contrastive framework to learn spatio-temporal features. Similarly, [12, 3] use temporal cues with contrastive learning. VCLR [40] formulates a video-level contrastive objective to capture global context. In the work presented here, we exploit VCLR as our backbone objective. However, different to pretext tasks in VCLR, which perform computation only on frame level, we modify those pretext tasks to not only operate on frame-level but also on clip-level thereby leading to better modeling of the spatio-temporal features in videos. See [66] for a more extensive review of SSL methods for video understanding.

2.3. Motion estimation and Learning visual correspondences for video understanding.

Motion estimation. Two-stream architectures [19, 69] have exhibited promising performance on the action recognition task by using pre-computed optical flow, although

[5] leverages contrastive learning as an additional objective as well.

such approaches reduce the efficiency of video processing. Several other methods [17, 33, 47, 61, 71] have proposed architectures that learn motion internally in an end-to-end fashion. The work presented in [50, 80] introduced a motion-specific contrastive learning task to learn motion features in a self-supervised setting.

Learning visual correspondences. Many recent works have proposed to learn visual correspondences between images using neural networks [21, 25, 46, 57, 65, 70]. Regarding learning correspondences for video understanding, CPNet [55] introduced a network that learns representations of videos by mixing appearance and long-range motion features from an RGB input only. Zhao *et al.* [82] proposed a method that learns a disentangled representation of a video, namely static appearance, apparent motion and appearance change from RGB input only. *MotionSqueeze* (MS) [42] introduced an end-to-end trainable, model-agnostic and lightweight module to extract motion features that does not require any correspondence supervision for learning.

3. Method

In order to apply a contrastive learning framework to videos specifically for generic event boundary detection, we follow the framework proposed by VCLR [40] and make noticeable modifications to it. For simplicity the notations are kept similar to [40] unless otherwise explicitly stated.

3.1. SSL for Video Representation Learning

1: Contrastive encoder. Our processing backbone is a ResNet-50 based encoder equipped with four pretext tasks, as defined in VCLR [40], trained following a contrastive objective as defined in MoCo-V2 [10].

Let $\mathbf{x}_p = \mathcal{T}(\mathbf{x}_q)$ be an augmented view of an anchor image $\mathbf{x}_q$ with $\mathcal{T} \sim \mathcal{P}$ ($\mathcal{P} = \{$random scaling, color-jitter, random grayscale, random Gaussian blur, and random horizontal flip$\}$ being set of augmentations) and $\mathcal{N}^-$ negative samples. $\mathbf{x}_q$ and $\mathbf{x}_p$ are processed through query ($f_q(\mathbf{x}_q)$) and a key ($f_k(\mathbf{x}_p)$) encoder respectively. In addition, these encoders are appended with projection heads (MLP layers) to get low dimensional representations of inputs i.e. $q = g_q(f_q(\mathbf{x}_q)), p = g_k(f_k(\mathbf{x}_p))$. The overall objective can be optimized by InfoNCE loss [60]:

$$\mathcal{L}_{\text{NCE}}(q, p, \mathcal{N}^-) = -\log\left(\frac{e^{\text{sim}(q,p)}}{e^{\text{sim}(q,p)} + \sum_{j=1}^{\text{N}} e^{\text{sim}(q,n_j)}}\right), \tag{1}$$

where, $\text{sim}(\cdot,\cdot)$ is a similarity function. We note that g_q and g_k can be thought of as a task-specific projection heads with further details below[6].

2: Pre-text setup. In order to capture different generic subtle nuances (spatial variations, temporal coherency, long range dependencies) for video understanding, the pretext tasks defined in VCLR [40] are a good candidate for pre-training as they serve the purpose of capturing such semantics for powerful video representation from raw video signals. We alter the pretext task setup in VCLR to ensure that Intra and Inter instance discrimination (ID) tasks operate at frame-level while computation of the video segment ID and temporal order regularization tasks occurs at clip-level. Below we elaborate on the intuition behind this notion.

For frame-level pretext tasks, consider three randomly selected frames from a video, v_1, v_2 and v_3. v_1 undergoes different augmentations to generate v_1^a and v_1^+. $\mathcal{N}^-$ represents negative samples from other videos. v_1^a is processed through query encoder $f_q(v_1^a)$, while (v_1^+, v_2, v_3) is processed through key encoder $f_k(\cdot)$. While depending upon the pretext, the projection head varies across the tasks.

For clip-level pretext tasks, a video V is divided into K (set to 3) segments $\{S_1, S_2, ..., S_K\}$ of equal duration. Two tuples, comprising of 4 frame long clip, are randomly and independently sampled from each of these segments to form an anchor tuple and a positive tuple. For instance, let $c_k = \{u_1, u_2, u_3, u_4\}$ where c_k denote the ordered clip sampled from the k^{th} segment while $u_1.....u_4$ represent the frames in that clip. Similarly the anchor and positive tuple are given by $t^a = \{c_1^a, c_2^a, \ldots c_K^a\}$ and $t^+ = \{c_1^+, c_2^+, \ldots c_K^+\}$ respectively.

a. Intra-frame ID task. In order to model spatial diversity for the GEBD task, we adopt the intra-frame instance discrimination task proposed in VCLR [40] to model inherent spatial changes across frames. For this task only v_1^+ is considered as a positive example while v_2 and v_3 represent negative examples. MLP heads are given by g_q^r and g_k^r while anchor embedding $q_1^a = g_q^r(f_q(v_1^a))$, positive embedding as $p_1^+ = g_k^r(f_k(v_1^+))$ and the negative sample[7] embeddings $p_2 = g_k^r(f_k(v_2))$, $p_3 = g_k^r(f_k(v_3))$. The loss objective is given by:

$$\mathcal{L}_{\text{Intra}} = \mathcal{L}_{\text{NCE}}(q_1^a, p_1^+, \{p_2, p_3\}). \tag{2}$$

b. Inter-frame ID task. Detecting generic event boundaries requires encoding fine-grained temporal structures from a coherent action, which are consistent with each other. To model this, inter-frame instance discrimination task considers v_1^a as an anchor frame and (v_1^+, v_2, v_3) as positive samples while $\mathcal{N}^-$ as negative samples. g_q^e and g_k^e are MLP projection heads which output the anchor embedding $q_1^a = g_q^e(f_q(v_1^a))$ and positive embeddings as $p_1^+ = g_k^e(f_k(v_1^+))$, $p_2 = g_k^e(f_k(v_2))$, $p_3 = g_k^e(f_k(v_3))$. Let $p' \in \{p_1^+, p_2, p_3\}$, hence the inter objective becomes:

$$\mathcal{L}_{\text{Inter}} = \frac{1}{3}\sum_{p'} \mathcal{L}_{\text{NCE}}(q_1^a, p', \mathcal{N}^-). \tag{3}$$

[6]subscript q, k in f and g represents *query* and *key*

[7]*Note*: Negative samples comes from the same video i.e. two samples as shown in Eq. (2).

c. Video segment based ID task. Learning long range temporal diversity in a video is also crucial for the GEBD task. For capturing the evolving semantics in the temporal dimension we need to incorporate global video level information. The contrastive loss objective is chosen in a way that each clip in the clip anchor and clip positive tuples i.e. t^a and t^+ learn a video-level embedding through consensus operation (denoted by $\mathcal{C}$) e.g. average. Mathematically this can be represented as:

$$q_t^a = g_p^s(\mathcal{C} \mid \mathcal{C}(f_q(c_1^a)), \mathcal{C}(f_q(c_2^a)), \ldots, \mathcal{C}(f_q(c_K^a))), \quad (4)$$

$$p_t^+ = g_k^s(\mathcal{C} \mid \mathcal{C}(f_k(c_1^+)), \mathcal{C}(f_k(c_2^+)), \ldots, \mathcal{C}(f_k(c_K^+))), \quad (5)$$

$$\mathcal{L}_{\text{Segment}} = \mathcal{L}_{\text{NCE}}(q_t^a, p_t^+, \mathcal{N}^-). \quad (6)$$

Here, g_q^s and g_k^s represent the MLP heads, $\mathcal{C}(f_q(c_k^a))$ indicates the average over the encoder representation of the individual frame in the k^{th} clip while q_t^a, p_t^+ denotes the final embeddings for anchor and positive clip tuples. The video-level contrastive loss is given by $\mathcal{L}_{\text{Segment}}$.

d. Temporal order regularization task. In order to enforce inherent sequential structure on videos for signalling supervision in self-supervised video representation learning, we need a pretext task to learn the correct temporal order of the video data. This can also be attained through pretext tasks proposed in [20, 75]. However, in this work we restrict ourselves to use the temporal ordering as a regularization term (denoted by $\mathcal{L}_{\text{Order}}$) within the contrastive framework as explained in Section 3.3 in [40] though we reformulate it to include clip-level computation.

The modifications made to video segment based ID task and temporal order regularization task in VCLR to incorporate clip-level computation is referred to as `cVCLR` (clip-VCLR).

3.2. Motion Estimation

For learning motion features we use the *MotionSqueeze* (MS) module presented in [42], a learnable motion feature extractor that can be inserted into any video understanding architecture to learn motion features and replace the external computation of optical flow. The motion features are learned in three steps:

1: Correlation computation. Consider $F^{(t)}$ and $F^{(t+1)}$ represent two adjacent input feature maps of spatial resolution $H \times W$ and channel dimension C. The correlation tensor $\mathtt{S}^{(t)}$ is computed by calculating correlation score of every spatial position $\mathtt{x}$ with respect to displacement $\mathtt{p}$ following the correlation layer implementation in FlowNet [21]. The correlation for position $\mathtt{x}$ is only computed in neighborhood size $\mathtt{P} = 2l+1$ by restricting a maximum displacement $\mathtt{p} \in [-l, l]^2$ and the value of $\mathtt{P}$ is set to 15.

2: Displacement estimation. The next step involves estimating the displacement map of size $H \times W \times 2$ from the correlation tensor $\mathtt{S}^{(t)}$. To get the best matching displacement for position $\mathtt{x}$, *kernal-soft-argmax* [46] is used. In addition, a motion confidence map (of size $H \times W \times 1$) of correlation as auxiliary motion information is obtained by pooling the highest correlation on each position $\mathtt{x}$ as described in [42]. The motion confidence map helps in identifying displacement outliers and learn informative motion features. The displacement map is then concatenated with the motion confidence map to create a displacement tensor $\mathbf{D}^{(t)}$ of size $H \times W \times 3$.

3: Feature transformation. In order to convert displacement tensor $\mathbf{D}^{(t)}$ to a relevant motion feature $\mathbf{M}^{(t)}$ (with same channel dimension C as input $\mathbf{F}^{(t)}$) , $\mathbf{D}^{(t)}$ is passed through four depth-wise separable convolutions [30] similar to [42]. Contrary to [42], in our work, we apply this feature transformation in a self-supervision setting to learn displacement tensor and motion confidence map (generic motion features). Finally, the motion features $\mathbf{M}^{(t)}$ are added to the input of the next layer using an element-wise addition operation : $\mathbf{F}'^{(t)} = \mathbf{F}^{(t)} + \mathbf{M}^{(t)}$. The resulting fused feature $\mathbf{F}'^{(t)}$ is passed as input to the next layer. For more details we refer the readers to [42].

3.3. Optimisation

The overall contrastive *loss objective* is given by:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{Inter}} + \mathcal{L}_{\text{Intra}} + \mathcal{L}_{\text{Segment}} + \mathcal{L}_{\text{Order}}. \quad (7)$$

Our encoder is augmented with a MS module (introduced after `conv3_x`[8]) to jointly learn appearance and motion features. More precisely, a ResNet-50 [28] model is adopted as the CNN encoder and we insert a TSM[4] [51] for each residual block of ResNet. Each of the four losses contribute equally to $\mathcal{L}_{\text{total}}$ although weighing them appropriately might boost the performance. The overall framework is illustrated in Figure 1.

4. Experimental Setup

4.1. Implementation Details.

Stage 1: Pre-training. We closely follow VCLR [40] to train the encoder. The model was pre-trained end-to-end with the objective as defined in Eq. (7) on 2 NVIDIA GeForce RTX-2080Ti GPUs with an effective batch size ($\mathcal{B}$) of 8 distributed across the GPUs (4 each) with temperature set to 0.01 across all pretext tasks. The input to the frame level and clip level pretext task is $(\mathcal{B}, 3, 3, 224, 224)$ and $(\mathcal{B}, K, 4, 3, 224, 224)$ respectively with $K = 3$. TSM[4] and *MotionSqueeze* is only applied on clip level task. The encoder is initialised to MoCo-v2 [10] weights with negative

[8]notation as in [28]

Table 1. F1 scores on the Kinetics-GEBD validation set with Rel. Dis threshold ranging from 0.05 to 0.5 with step of 0.05. ‡: soft-labels, †: hard-labels. * is pretrained on Kinetics-400 [37] dataset.

Rel. Dis Threshold		Finetuning	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	avg
	BMN† [52]	✗	0.186	0.204	0.213	0.220	0.226	0.230	0.233	0.237	0.239	0.241	0.223
	BMN-SE [52]†	✗	0.491	0.589	0.627	0.648	0.660	0.668	0.674	0.678	0.681	0.683	0.640
	TCN-TAPOS [44]†	✓	0.464	0.560	0.602	0.628	0.645	0.659	0.669	0.676	0.682	0.687	0.627
	TCN [44]†	✗	0.588	0.657	0.679	0.691	0.698	0.703	0.706	0.708	0.710	0.712	0.685
Supervised	PC [68]†	✓	0.625	0.758	0.804	0.829	0.844	0.853	0.859	0.864	0.867	0.870	0.817
	SBoCo-Res50 [36]†	✗	0.732	-	-	-	-	-	-	-	-	-	0.866
	SBoCo-TSN [36]†,*	✗	0.787	-	-	-	-	-	-	-	-	-	0.892
	DDM-Net [72]†	✗	0.764	0.843	0.866	0.880	0.887	0.892	0.895	0.898	0.900	0.902	0.873
	Li et.al. [49]‡	✗	0.743	0.830	0.857	0.872	0.880	0.886	0.890	0.893	0.896	0.898	0.865
	SC-Transformer [48]‡	✗	**0.777**	**0.849**	**0.873**	**0.886**	**0.895**	**0.900**	**0.904**	**0.907**	**0.909**	**0.911**	**0.881**
	SceneDetect [6]	✗	0.275	0.300	0.312	0.319	0.324	0.327	0.330	0.332	0.334	0.335	0.318
	PA - Random [68]	✗	0.336	0.435	0.484	0.512	0.529	0.541	0.548	0.554	0.558	0.561	0.506
	PA [68]	✗	0.396	0.488	0.520	0.534	0.544	0.550	0.555	0.558	0.561	0.564	0.527
Un-supervised	UBoCo-Res50 [36]	✗	0.703	-	-	-	-	-	-	-	-	-	0.866
	UBoCo-TSN [36]*	✗	0.702	-	-	-	-	-	-	-	-	-	0.892
	TeG-PS [63]†	✓	0.699	-	-	-	-	-	-	-	-	-	-
(Self-supervised)	TeG-FG [63]†	✓	0.714	-	-	-	-	-	-	-	-	-	-
	Ours†	✓	0.680	0.779	0.806	0.818	0.825	0.830	0.834	0.837	0.839	0.841	0.809
	Ours‡	✓	0.711	0.777	0.791	0.795	0.798	0.799	0.801	0.802	0.802	0.803	0.788

samples $\mathcal{N}^-$ (queue size) set to 8192 and is trained with SGD for 400 epochs with a warm-start of 5 epochs following a cosine decay with base learning rate of 0.01. *Pre-training* is only performed on the Kinetics-GEBD [68] dataset.

Stage 2: Finetuning. Input to the encoder is based on the temporal window (W=5) which defines a context over a candidate frame (before and after) with a stride $m = 3$[9] resulting into a 4D tensor $(10, 3, 224, 224)$ as input. We finetune the model end-to-end with a binary cross entropy (BCE) (boundary is 0/1) as the objective augmented with Gaussian smoothing ($\sigma = 3$) for soft labeling as in [48]. The learning rate set to $7.5e^{-4}$ for Kinetics-GEBD, while for TAPOS it was set to $1e^{-4}$. Balance sampling is applied to each batch during training to avoid class imbalance. We finetune the model for 8 epochs and use early stopping to find the best model. More details in supplementary material.

To select the final boundary predictions for the video, we apply post-processing scheme on the obtained boundary proposals. *First*, proposals should be greater then a threshold of 0.5. *Second*, we aggregate all the proposals within a 1 second time window.

4.2. Results

We perform extensive quantitative and qualitative studies on the given datasets. For evaluation protocol we refer the reader to supp. material. In Tables 1 and 2, we report F1 scores for different thresholds ranging from 0.05 to 0.5 with a step of 0.05, for the Kinetics-GEBD and TAPOS datasets respectively. On the Kinetics-GEBD dataset, our model outperforms the supervised baseline PC [68] with *Rel. Dis* threshold 0.05 and is also comparable with other state-of-the-art unsupervised/self-supervised GEBD models like UBoCo [36] and TeG [63] in terms of performance. Table 1 illustrates the result on the Kinetics-GEBD dataset. On the TAPOS dataset, which consists of Olympic sport videos with 21 action classes, we have a similar observation. Our model outperforms the supervised baseline PC [68] at the *Rel. Dis* threshold of 0.05 and is comparable on other thresholds. Other state-of-the-art methods on TAPOS like DDM-Net [72] and SC-Transformer [48] fall in the supervised category and cannot be directly compared with our results. We were unable to find other state-of-the-art un/self-supervised models for GEBD to directly compare our results with, on the TAPOS dataset shown in Table 2.

We also perform a qualitative analysis of boundaries detected by our method and compare them with the supervised baseline PC [68] and the ground truth annotation in Figure 2. Figure 3 shows a visualization of the motion confidence map learned by the MS module during *pre-training*. We observe that motion confidence generalizes well to the TAPOS dataset too, which was not used for pre-training. This validates that MS module learns general motion features even in a self-supervised setting without any explicit motion specific pretext task. In addition our model's event boundary detection results on the TAPOS dataset further justifies that our model is a generic event boundary detector, as after fine-tuning it generalizes beyond Kinetics-GEBD dataset to the TAPOS benchmark. We also found that linear evaluation (freezing the encoder) on the downstream GEBD task resulted in poor performance.

4.3. Ablation Studies

1: Does *MotionSqueeze* through self-supervision helps? As shown in Figure 3, the MS module shows high confidence in regions of images that are more dynamic. The

[9] selecting one frame out of every 3 consecutive frames

Table 2. F1 scores on the TAPOS validation set with Rel. Dis threshold ranging from 0.05 to 0.5 with step of 0.05. ‡: soft-labels, †: hard-labels. (-) : Not clear.

Rel. Dis Threshold		Finetuning	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	avg
Supervised	ISBA [13]	-	0.106	0.170	0.227	0.265	0.298	0.326	0.348	0.369	0.376	0.384	0.314
	TCN [44]	✗	0.237	0.312	0.331	0.339	0.342	0.344	0.347	0.348	0.348	0.348	0.330
	CTM [31]	-	0.244	0.312	0.336	0.351	0.361	0.369	0.374	0.381	0.383	0.385	0.350
	Transparser [67]	-	0.289	0.381	0.435	0.475	0.500	0.514	0.527	0.534	0.540	0.545	0.474
	PC [68]†	✓	0.522	0.595	0.628	0.647	0.660	0.666	0.672	0.676	0.680	0.684	0.643
	DDM-Net [72]†	✗	0.604	0.681	0.715	0.735	0.747	0.753	0.757	0.760	0.763	0.767	0.728
	SC-Transformer [48]‡	✗	**0.618**	**0.694**	**0.728**	**0.749**	**0.761**	**0.767**	**0.771**	**0.774**	**0.777**	**0.780**	**0.742**
Un-supervised	SceneDetect [6]	✗	0.035	0.045	0.047	0.051	0.053	0.054	0.055	0.056	0.057	0.058	0.051
	PA - Random [68]	✗	0.158	0.233	0.273	0.310	0.331	0.347	0.357	0.369	0.376	0.384	0.314
	PA [68]	✗	0.360	0.459	0.507	0.543	0.567	0.579	0.592	0.601	0.609	0.615	0.543
(Self-supervised)	Ours†	✓	0.573	0.614	0.639	0.656	0.669	0.679	0.687	0.693	0.700	0.704	0.661
	Ours‡	✓	0.586	0.624	0.648	0.663	0.675	0.685	0.692	0.697	0.704	0.708	0.668

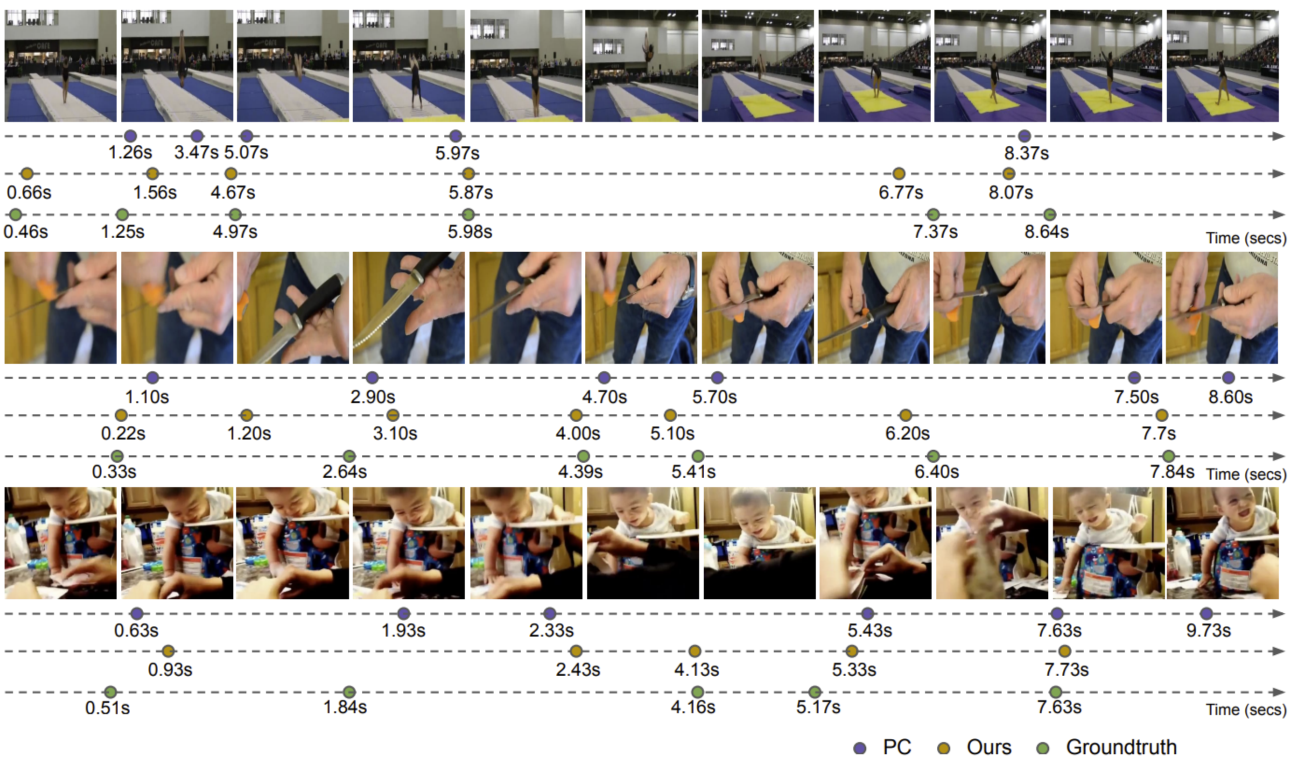

Figure 2. Qualitative Analysis I: visualization of some detected boundaries on the validation set of Kinetics-GEBD. Compared with baseline PC [68], our method produces more precise boundaries that are consistent with the ground truth.

Table 3. Ablation study on validation set of TAPOS and Kinetics-GEBD for F1 score at *Rel. Dis* threshold 0.05

Method	TAPOS	Kinetics-GEBD
Vanilla VCLR	0.496	0.596
cVCLR	0.502 (↑)	0.605 (↑)
+ *MotionSqueeze*	0.573 (↑)	0.680 (↑)
+ Soft labels	0.586 (↑)	0.711 (↑)

module learns optical flow in an online fashion even under self-supervision. Intuitively, temporal order regularization and video segment instance discrimination pre-text tasks implicitly complements the MS module to learn generic motion features. From Table 3, we observe that by incorporating the MS module, the F1@0.05 score on the GEBD task increases by 7.5% on Kinetics-GEBD and 7.1% on TAPOS, which is a significant increase.

2: Does soft labelling helps in boosting the performance? Kinetics-GEBD has 5 annotators to capture human perception differences but this introduces ambiguity. Ideally the neighbouring frames of the candidate boundary frame should also have a high value of the ground truth label. To tackle

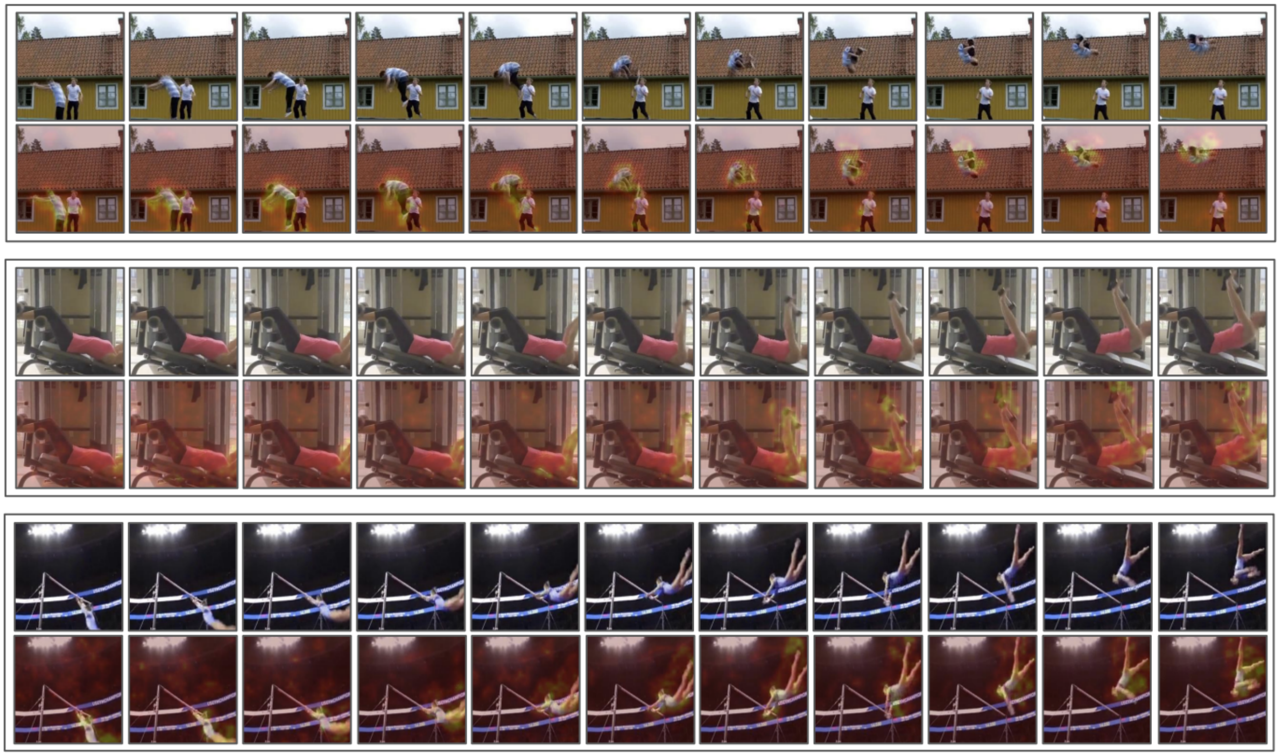

Figure 3. Qualitative Analysis II: visualization of the learned motion confidence map. The **first two** blocks (categories: jumping on trampoline and situp respectively) are taken from the Kinetics-GEBD dataset, while the **bottom** block (category: uneven bar) is derived from TAPOS. In each block, the first row shows the RGB frames while the second depicts the motion confidence map learnt by the model. **Note**: the model is only pre-trained on Kinetics-GEBD but it generalizes to the TAPOS dataset as well.

this issue, we use Gaussian smoothing ($\sigma = 3$) to create soft labels from hard labels, which ensures that model avoids making over confident predictions for the event boundary. As shown in Table 3, soft labels improves the F1@0.05 score by 3.1% on Kinetics-GEBD and 1.3% on TAPOS dataset.

Note: For GEBD challenge specific reports, we refer the reader to challenge page for 2021, 2022.

5. Conclusions and Discussion

In this work, we presented a self-supervised model that can be pre-trained for the generic event boundary detection task. The GEBD task is an ideal problem for self-supervised learning given that the task aims to learn generic boundaries and is not biased towards any predefined action categories from pre-trained state-of-the-art action recognition models. In order to learn spatial and temporal diversity we reformulate SSL objective at frame-level and clip-level to learn effective video representations (cVCLR). In addition we augment our encoder with a MS module and find this indeed compliments the overall performance on the downstream GEBD task. Furthermore, the motion features learnt are generic since the model is only pre-trained on Kinetics-GEBD but generalizes to TAPOS dataset as well. Through our extensive evaluation, we achieve comparable performance to self-supervised state-of-the-art methods on the Kinetics-GEBD as shown in Table 1.

However, there are limitations with this work. First, we have not used more powerful models, e.g. transformers as in [48], or cascaded networks as in [29]. Second, since MS module is directly applied on feature maps, it learns global motion features. However, in GEBD the boundaries are generic and every type of motion may not indicate a boundary, hence a more fine-grained motion module can boost the performance. Third, due to computational constraints, our self-supervised model is only pre-trained on the Kinetics-GEBD dataset; however, pre-training the model on Kinetics-400 could yield even better performance on the downstream GEBD task. These limitations will be addressed in our future work.

6. Acknowledgement

This work has emanated from research supported by Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289_P2, co-funded by the European Regional Development Fund and Xperi FotoNation.

References

[1] Yazan Abu Farha, Alexander Richard, and Juergen Gall. When will you do what?-anticipating temporal occurrences of activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5343–5352, 2018.

[2] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*, 2019.

[3] Yutong Bai, Haoqi Fan, Ishan Misra, Ganesh Venkatesh, Yongyi Lu, Yuyin Zhou, Qihang Yu, Vikas Chandra, and Alan Yuille. Can temporal information help with contrastive self-supervised learning? *arXiv preprint arXiv:2011.13046*, 2020.

[4] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

[6] Brandon Catellano. Pyscenedetect: an intelligent scene cut detection and video splitting tool. `https://github.com/Breakthrough/PySceneDetect`, 2014.

[7] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A. Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster R-CNN architecture for temporal action localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[8] Shixing Chen, Xiaohan Nie, David Fan, Dongqing Zhang, Vimal Bhat, and Raffay Hamid. Shot contrastive self-supervised learning for scene boundary detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9796–9805, 2021.

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[10] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[11] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 720–736, 2018.

[12] Ishan Dave, Rohit Gupta, Mamshad Nayeem Rizve, and Mubarak Shah. TCLR: Temporal contrastive learning for video representation. *Computer Vision and Image Understanding*, 219:103406, 2022.

[13] Li Ding and Chenliang Xu. Weakly-supervised action segmentation with iterative soft boundary assignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6508–6516, 2018.

[14] Yasser Abdelaziz Dahou Djilali, Tarun Krishna, Kevin McGuinness, and Noel E. O'Connor. Rethinking 360deg image visual attention modelling with unsupervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15414–15424, October 2021.

[15] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

[16] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015.

[17] Lijie Fan, Wenbing Huang, Chuang Gan, Stefano Ermon, Boqing Gong, and Junzhou Huang. End-to-end learning of motion representation for video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6016–6025, 2018.

[18] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.

[19] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1933–1941, 2016.

[20] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3636–3645, 2017.

[21] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick Van der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. *arXiv preprint arXiv:1504.06852*, 2015.

[22] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. Cascaded boundary regression for temporal action detection. *arXiv preprint arXiv:1705.01180*, 2017.

[23] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[24] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 297–304, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[25] Kai Han, Rafael S Rezende, Bumsub Ham, Kwan-Yee K Wong, Minsu Cho, Cordelia Schmid, and Jean Ponce. SCNet: Learning semantic correspondence. In *Proceedings of the IEEE international conference on computer vision*, pages 1831–1840, 2017.

[26] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[27] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. In *European conference on computer vision*, pages 312–329. Springer, 2020.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[29] Dexiang Hong, Congcong Li, Longyin Wen, Xinyao Wang, and Libo Zhang. Generic event boundary detection challenge at cvpr 2021 technical report: Cascaded temporal attention network (castanet). *arXiv preprint arXiv:2107.00239*, 2021.

[30] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[31] De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *European Conference on Computer Vision*, pages 137–153. Springer, 2016.

[32] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017.

[33] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. STM: Spatiotemporal and motion encoding for action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2000–2009, 2019.

[34] Longlong Jing, Xiaodong Yang, Jingen Liu, and Yingli Tian. Self-supervised spatiotemporal feature learning via video rotation prediction. *arXiv preprint arXiv:1811.11387*, 2018.

[35] Hyolim Kang, Jinwoo Kim, Kyungmin Kim, Taehyun Kim, and Seon Joo Kim. Winning the CVPR'2021 kinetics-gebd challenge: Contrastive learning approach. *arXiv preprint arXiv:2106.11549*, 2021.

[36] Hyolim Kang, Jinwoo Kim, Taehyun Kim, and Seon Joo Kim. UBoCo: Unsupervised boundary contrastive learning for generic event boundary detection. *arXiv preprint arXiv:2111.14799*, 2021.

[37] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[38] Quan Kong, Wenpeng Wei, Ziwei Deng, Tomoaki Yoshinaga, and Tomokazu Murakami. Cycle-contrast for self-supervised video representation learning. *Advances in Neural Information Processing Systems*, 33:8089–8100, 2020.

[39] Tarun Krishna, Kevin McGuinness, and Noel O'Connor. Evaluating contrastive models for instance-based image retrieval. In *Proceedings of the 2021 International Conference on Multimedia Retrieval*, pages 471–475, 2021.

[40] Haofei Kuang, Yi Zhu, Zhi Zhang, Xinyu Li, Joseph Tighe, Sören Schwertfeger, Cyrill Stachniss, and Mu Li. Video contrastive learning with global context. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3195–3204, 2021.

[41] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 780–787, 2014.

[42] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. Motionsqueeze: Neural motion feature learning for video understanding. In *European Conference on Computer Vision*, pages 345–362. Springer, 2020.

[43] Colin Lea, Austin Reiter, René Vidal, and Gregory D Hager. Segmental Spatiotemporal CNNs for Fine-Grained Action Segmentation. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 36–52, Cham, 2016. Springer International Publishing.

[44] Colin Lea, Austin Reiter, René Vidal, and Gregory D Hager. Segmental spatiotemporal CNNs for fine-grained action segmentation. In *European conference on computer vision*, pages 36–52. Springer, 2016.

[45] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE international conference on computer vision*, pages 667–676, 2017.

[46] Junghyup Lee, Dohyung Kim, Jean Ponce, and Bumsub Ham. SFNet: Learning object-aware semantic correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2278–2287, 2019.

[47] Myunggi Lee, Seungeui Lee, Sungjoon Son, Gyutae Park, and Nojun Kwak. Motion feature network: Fixed motion filter for action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 387–403, 2018.

[48] Congcong Li, Xinyao Wang, Dexiang Hong, Yufei Wang, Libo Zhang, Tiejian Luo, and Longyin Wen. Structured context transformer for generic event boundary detection. *arXiv preprint arXiv:2206.02985*, 2022.

[49] Congcong Li, Xinyao Wang, Longyin Wen, Dexiang Hong, Tiejian Luo, and Libo Zhang. End-to-end compressed video representation learning for generic event boundary detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13967–13976, June 2022.

[50] Rui Li, Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. Motion-focused contrastive learning of video representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2105–2114, 2021.

[51] Ji Lin, Chuang Gan, and Song Han. TSM: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7083–7093, 2019.

[52] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. BMN: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3889–3898, 2019.

[53] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. BSN: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[54] Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future frame prediction for anomaly detection–a new baseline. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6536–6545, 2018.

[55] Xingyu Liu, Joon-Young Lee, and Hailin Jin. Learning video representations from correspondence proposals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4273–4281, 2019.

[56] Antoine Miech, Ivan Laptev, Josef Sivic, Heng Wang, Lorenzo Torresani, and Du Tran. Leveraging the present to anticipate the future in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[57] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Hyperpixel flow: Semantic correspondence with multi-layer neural features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3395–3404, 2019.

[58] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016.

[59] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

[60] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[61] AJ Piergiovanni and Michael S Ryoo. Representation flow for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9945–9953, 2019.

[62] Hamed Pirsiavash and Deva Ramanan. Parsing videos of actions with segmental grammars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[63] Rui Qian, Yeqing Li, Liangzhe Yuan, Boqing Gong, Ting Liu, Matthew Brown, Serge Belongie, Ming-Hsuan Yang, Hartwig Adam, and Yin Cui. Exploring temporal granularity in self-supervised video representation learning. *arXiv preprint arXiv:2112.04480*, 2021.

[64] Ayush K Rai, Tarun Krishna, Julia Dietlmeier, Kevin McGuinness, Alan F Smeaton, and Noel E O'Connor. Discerning generic event boundaries in long-form wild videos. *arXiv preprint arXiv:2106.10090*, 2021.

[65] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6148–6157, 2017.

[66] Madeline Chantry Schiappa, Yogesh Singh Rawat, and Mubarak Shah. Self-supervised learning for videos: A survey. *ArXiv*, abs/2207.00419, 2022.

[67] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Intra- and inter-action understanding via temporal action parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[68] Mike Zheng Shou, Stan W Lei, Weiyao Wang, Deepti Ghadiyaram, and Matt Feiszli. Generic event boundary detection: A benchmark for event segmentation. *arXiv preprint arXiv:2101.10511*, 2021.

[69] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014.

[70] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018.

[71] Shuyang Sun, Zhanghui Kuang, Lu Sheng, Wanli Ouyang, and Wei Zhang. Optical flow guided feature: A fast and robust motion representation for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1390–1399, 2018.

[72] Jiaqi Tang, Zhaoyang Liu, Chen Qian, Wayne Wu, and Limin Wang. Progressive attention on multi-level dense difference maps for generic event boundary detection. *arXiv preprint arXiv:2112.04771*, 2021.

[73] Li Tao, Xueting Wang, and Toshihiko Yamasaki. Self-supervised video representation learning using inter-intra contrastive framework. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2193–2201, 2020.

[74] Barbara Tversky and Jeffrey M Zacks. Event perception. *Oxford handbook of cognitive psychology*, 1(2):3, 2013.

[75] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. In *European conference on computer vision*, pages 504–521. Springer, 2020.

[76] Yuxuan Wang, Difei Gao, Licheng Yu, Stan Weixian Lei, Matt Feiszli, and Mike Zheng Shou. Generic event boundary captioning: A benchmark for status changes understanding. *arXiv preprint arXiv:2204.00486*, 2022.

[77] Huifen Xia and Yongzhao Zhan. A survey on temporal action localization. *IEEE Access*, 8:70477–70487, 2020.

[78] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10334–10343, 2019.

[79] Ceyuan Yang, Yinghao Xu, Bo Dai, and Bolei Zhou. Video representation learning with visual tempo consistency. *arXiv preprint arXiv:2006.15489*, 2020.

[80] Xitong Yang, Xiaodong Yang, Sifei Liu, Deqing Sun, Larry Davis, and Jan Kautz. Hierarchical contrastive motion learning for video action recognition. *arXiv preprint arXiv:2007.10321*, 2020.

[81] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

[82] Yue Zhao, Yuanjun Xiong, and Dahua Lin. Recognize actions by disentangling components of dynamics. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6566–6575, 2018.