The Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI-22)

Iterative Contrast-Classify For Semi-supervised Temporal Action Segmentation

Dipika Singhania , Rahul Rahaman , Angela Yao

National University of Singapore
dipika16@comp.nus.edu.sg, rahul.rahaman@u.nus.edu, ayao@comp.nus.edu.sg

Abstract

Temporal action segmentation classifies the action of each frame in (long) video sequences. Due to the high cost of frame-wise labeling, we propose the first semi-supervised method for temporal action segmentation. Our method hinges on unsupervised representation learning, which, for temporal action segmentation, poses unique challenges. Actions in untrimmed videos vary in length and have unknown labels and start/end times. Ordering of actions across videos may also vary. We propose a novel way to learn frame-wise representations from temporal convolutional networks (TCNs) by clustering input features with added *time-proximity condition* and *multi-resolution similarity*. By merging representation learning with conventional supervised learning, we develop an *"Iterative-Contrast-Classify (ICC)"* semi-supervised learning scheme. With more labelled data, ICC progressively improves in performance; ICC semi-supervised learning, with 40% labelled videos, performs similar to fully-supervised counterparts. Our ICC improves MoF by {+1.8, +5.6, +2.5}% on Breakfast, 50Salads and GTEA respectively for 100% labelled videos.

Introduction

Temporal action segmentation takes long untrimmed video containing multiple actions in a sequence and estimates the action labels for each video frame. There is a huge annotation cost to label each frame of all videos for action segmentation, especially as the videos are minutes long. Several works aim to reduce annotation requirements with weak supervision like transcripts (Chang et al. 2019), or few frame labels (Li, Farha, and Gall 2021). In this work, we advocate using semi-supervised learning, *i.e.* having labels only for a fraction of the videos in the training set. Specifically, we design the unsupervised representation learning step to learn the underlying distribution of all unlabelled videos, which helps achieve higher temporal action segmentation scores with very few labeled videos in the subsequent supervised training step.

For unsupervised representation learning, we are inspired by the success of contrastive learning in images (Chen et al. 2020b), short-trimmed videos (Qian et al. 2020; Lorre et al. 2020) and other areas of machine learning (Chen et al. 2021; Rahaman, Ghosh, and Thiery 2021). Works which apply contrastive learning to longer sequences bring together multiple viewpoints of a sequence (Sermanet et al. 2018) or multiple modalities, such as video and text (Miech et al. 2020) or video and audio (Alwassel et al. 2019) . These settings target multi-view or multi-modal representations and are not applicable for videos in action segmentation datasets. Also, standard contrastive technique to bring image (or video) and its augmentations near is unlikely to be effective. Action segmentation is a frame-wise (and not video-wise) classification task so a model should capture similarities across temporally disjoint but semantically similar frames, while factoring temporal continuity within every action segment. This poses significant challenges without action labels.

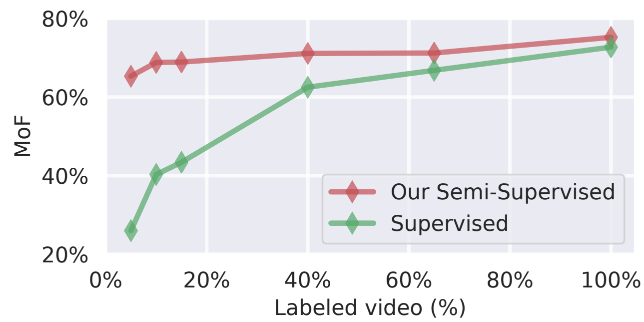

Figure 1: Frame accuracy on Breakfast dataset: Our semi-supervised approach has impressive performance with just 5% labelled videos; at 40%, we almost match the Mean over Frames (MoF) of a 100% fully-supervised setup.

As such, contrastive learning has not yet been explored for action segmentation and our work is the first. We design a novel strategy to form the positive and negative sets without labels by leveraging the discriminativeness of the pre-trained I3D (Carreira and Zisserman 2017) input features (see Fig. 2 right). As a base model, we use a *SOTA* temporal convolutional network (TCN) (Singhania, Rahaman, and Yao 2021); a key advantage is the progressive temporal upsampling in the decoder which allows us to integrate contrastive learning to multiple temporal resolutions and enforce temporal continuity by design. Our proposed multi-resolution representation for contrastive learning is is highly effective.

Equipped with our (unsupervised) learned model, we can perform semi-supervised learning with only a small fraction

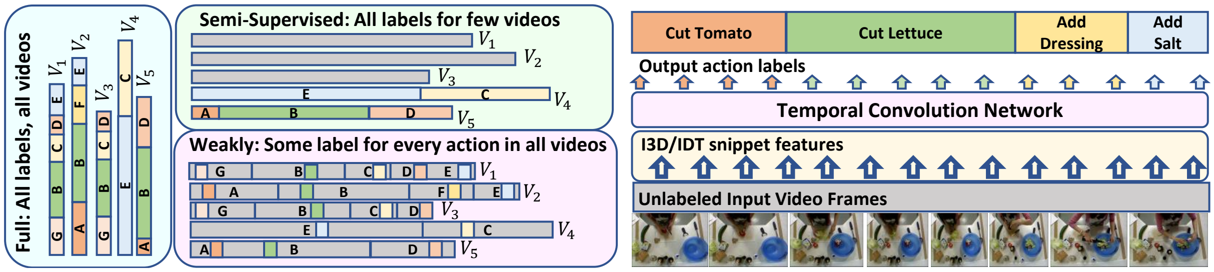

Figure 2: Left: Comparison on forms of supervision. Right: Overview of Temporal Action Segmentation task with TCN.

of labelled training videos. To fully utilize the labeled and unlabeled dataset, we propose a novel *Iterative-Contrast-Classify* algorithm that updates the representations while learning to segment sequences and assigning pseudo-labels to the unlabelled videos. We achieve noteworthy segmentation performance with just 5% labelled videos, while with 40% labels, we can almost match the full-supervision (see Fig. 1).

To the best of our knowledge, our work is the first to apply semi-supervised learning for temporal action segmentation. The closest in spirit are SSTDA (Chen et al. 2020a), TSS(Li, Farha, and Gall 2021). However, these works are weakly-supervised setups and require (weak) labels for *every* training video (see Fig. 2 left). TSS require one frame label for each action[1] While the percentage of overall labeled frames is very little (0.03%), the annotation effort should not be underestimated. Annotators must still watch *all* the videos and labelling timestamp frames gives only a 6X speedup compared to densely labelling all frames (Ma et al. 2020).

Summarizing our contributions, we:

- Proposed a novel unsupervised representation learning that leverages the discriminativeness in pre-trained input features and temporal continuity in a video sequence.
- Designed a multi-resolution representation for contrastive learning which inherently encodes sequence variations and temporal continuity.
- Formulated a new semi-supervised learning variant of temporal action segmentation and proposed an *"Iterative-Contrast-Classify (ICC)"* algorithm that iteratively fine-tunes representations and strengthens segmentation performance with few labelled videos.

Related Work

Temporal action segmentation Classifying and temporally segmenting fine grained actions in long video requires both local motion and global long-range dependencies information. It is standard to extract snippet level IDT (Wang and Schmid 2013) or I3D features to capture local temporal motion. These features are used as inputs to Temporal Convolution Networks (TCNs) which captures global action compositions, segment durations and long-range dependencies (see Fig.2 right). Fully supervised frameworks require per-frame annotations of all the video sequences in the dataset. Popular TCN frameworks include U-Net style encoder-decoders (Lea et al. 2017; Lei and Todorovic 2018; Singhania, Rahaman, and Yao 2021) or temporal resolution preserving MSTCNs (Li et al. 2020; Farha and Gall 2019).

Weakly supervised methods bypass per-frame annotations and use labels such as ordered lists of actions (Ding and Xu 2018; Richard et al. 2018; Chang et al. 2019; Li, Lei, and Todorovic 2019; Souri et al. 2019) or a small percentage of action time-stamps (Kuehne, Richard, and Gall 2018; Li, Farha, and Gall 2021; Chen et al. 2020a) for *all* videos.

Unsupervised approaches use clustering, including k-means (Kukleva et al. 2019), agglomerative (Sarfraz et al. 2021), and discriminative clustering (Sener and Yao 2018). To improve clustering performance, some works (Kukleva et al. 2019; VidalMata et al. 2021) learns representation by predicting frame-wise feature's absolute temporal positions in the video. Our unsupervised representation implicitly capture relative temporal relationships based on temporal distance rather than absolute positions.

Unsupervised Contrastive Feature Learning dates back to (Hadsell, Chopra, and LeCun 2006) but was more recently formalized in SimCLR (Chen et al. 2020b). Most works (Chen et al. 2021; Rahaman, Ghosh, and Thiery 2021; He et al. 2020; Khosla et al. 2020) hinge on well-defined data augmentations, with the goal of bringing together the original and augmented sample in the feature space.

The few direct extensions of SimCLR for video (Bai et al. 2020; Qian et al. 2020; Lorre et al. 2020) target action recognition on few seconds short clips. Others integrate contrastive learning by bringing together next-frame feature predictions with actual representations (Kong et al. 2020; Lorre et al. 2020), using path-object tracks for bringing cycle-consistency (Wang, Zhou, and Li 2020), and considering multiple viewpoints (Sermanet et al. 2018) or accompanying modalities like audio (Alwassel et al. 2019) or text (Miech et al. 2020). We are inspired by these works to develop contrastive learning for long-range segmentation. However, previous works differ fundamentally in both aim *i.e.* learning the underlying distribution of cycle-consistency in short clips, and input data *e.g.* multiple viewpoints or modalities.

[1] Our setup is analogous to semi-supervised image segmentation (Hung et al. 2018; Mittal, Tatarchenko, and Brox 2021): most training images are un-annotated, while a few are fully-annotated. The analogue of TSS is point-supervision (Bearman et al. 2016), *i.e.* labelling one pixel from each object of every training image.

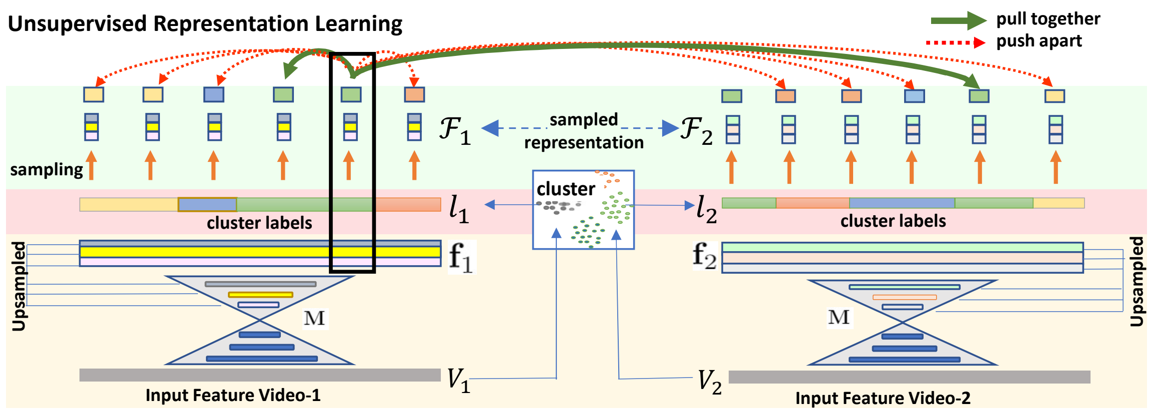

Figure 3: Unsupervised Representation Learning: Step 1 (bottom orange panel): Pass pre-trained I3D inputs V into the base TCN and generate multi-resolution representation $\mathbf{f}$. Step 2 (middle pink panel): Cluster the I3D inputs V within a training mini-batch and generate frame-wise cluster labels l. Step 3 (top green panel): Representations $\mathbf{f}$ and its corresponding cluster label l is sampled based on temporal proximity sampling strategy to form feature set $\mathcal{F}$. Step 4: Apply contrastive learning to "pull together" (green arrows) similar samples in the positive set and "push apart" (red arrows) other samples in the negative set.

Preliminaries

Definitions: We denote a video as $\mathrm{V} \in \mathbb{R}^{T\times F}$; for each temporal location $t<T$, frame $\mathrm{V}[t] \in \mathbb{R}^F$ is a F-dimensional pre-trained I3D feature. Note, input I3D feature is from model pre-trained on Kinetics dataset (Carreira and Zisserman 2017) and is not fine-tuned on our segmentation datasets.

For simplicity, unless otherwise explicitly noted, *e.g.* in Section , we treat the temporal dimension of all the videos as a normalized unit interval $t \in [0,1]$, *i.e.* $T=1$. Each frame $\mathrm{V}[t]$ has a ground truth action label $y[t] \in \{1,...,A\}$ from a pre-defined set of A action classes. Additionally, each video has a higher-level *complex activity* label $c \in \{1,\dots,C\}$. The complex activity specifies an underlying objective, *e.g.* *'making coffee'* for action sequence {*'take cup'*, *'pour coffee'*, *'add sugar'*, *'stir'*}, though the sequence ordering may differ *e.g.* *'add sugar'* may come before *'pour coffee'*.

Base Segmentation Model: We use the C2F-TCN (Singhania, Rahaman, and Yao 2021), which is a U-Net style encoder-decoder, though our method is also applicable to other base models such as the ED-TCN (Lea et al. 2017). The encoder layers $\mathbf{\Phi}$ take pre-trained video features as inputs and progressively increases the feature abstraction while reducing the temporal resolution up to some bottleneck $\mathbf{\Gamma}$. The decoder layers $\mathbf{\Psi}$ then increase the temporal resolution symmetrically with respect to the encoder. We refer to the Supplementary-S2.1 or to (Singhania, Rahaman, and Yao 2021) for more details. The overall encoder-decoder $\mathbf{M}:=(\mathbf{\Phi}:\mathbf{\Gamma}:\mathbf{\Psi})$ takes V as input and produces output frame-wise features $\mathbf{f}=\mathbf{M}(\mathrm{V})$. For each time t, $\mathbf{f}[t] \in \mathbb{R}^d$ is a d-dimensional representation. We describe in detail how $\mathbf{f}$ is formed in Section .

Learning Framework & Data Split: Our overall framework has two stages. First, we apply an unsupervised representation learning to learn a model $\mathbf{M}$ (Section). Subsequently, model $\mathbf{M}$ is trained with linear projection layers (action classifiers) with a small portion of the labelled training data to produce the semi-supervised model $(\mathbf{M}:\mathbf{G})$ (Section). For representation learning, we follow the convention of previous unsupervised works (Kukleva et al. 2019; VidalMata et al. 2021) in which actions y are unknown, but the complex activity of each video *is* known[2]. For the semi-supervised stage, we use the ground truth y for a small subset of labelled video $\mathcal{D}_L$ out of a larger training dataset $\mathcal{D}=\mathcal{D}_U \cup \mathcal{D}_L$, where $\mathcal{D}_U$ denotes the unlabelled videos.

Contrastive Learning We use contrastive learning for our unsupervised frame-wise representation learning. Following the formalism of (Chen et al. 2020b), we define a set of features $\mathcal{F}:=\{\mathbf{f}_i, i\in\mathcal{I}\}$ indexed by a set $\mathcal{I}$. Each feature $\mathbf{f}_i \in \mathcal{F}$ is associated with two disjoint sets of indices $\mathcal{P}_i \subset \mathcal{I}\setminus\{i\}$ and $\mathcal{N}_i \subset \mathcal{I}\setminus\{i\}$. The features in the positive set $\mathcal{P}_i$ should be similar to $\mathbf{f}_i$, while the features in the negative set $\mathcal{N}_i$ should be contrasted with $\mathbf{f}_i$. For each $j \in \mathcal{P}_i$, the contrastive probability p_{ij} is defined as

$$p_{ij} = \frac{e_\tau(\mathbf{f}_i,\mathbf{f}_j)}{e_\tau(\mathbf{f}_i,\mathbf{f}_j)+\sum_{k\in\mathcal{N}_i} e_\tau(\mathbf{f}_i,\mathbf{f}_k)}, \tag{1}$$

where the term $e_\tau = \exp\{\cos(\mathbf{f}_i,\mathbf{f}_j)/\tau\}$ is the exponential of the cosine similarity between $\mathbf{f}_i$ and $\mathbf{f}_j$ scaled by temperature τ. Maximizing the probability in Eq. (1) ensures that $\mathbf{f}_i, \mathbf{f}_j$ are similar while also decreasing the cosine similarity between $\mathbf{f}_i$ and any feature in the negative set. The key to effective contrastive learning is to identify the relevant positive and negative sets to perform the targeted task.

Unsupervised Representation Learning

Our representation learning applies contrastive learning at frame-level, based on input feature clustering and temporal continuity (Sec.), and at a video-level, by leveraging the complex activity labels (Sec.). We merge the two objectives into a common loss that is applied to our multi-temporal resolution feature representations (Sec.).

[2]The label is used implicitly, as the unsupervised methods are applied to videos of each complex activity individually.

Frame-Level Contrastive Formulation

Input Clustering: Our construction of positive and negative sets should respect the distinction between different action classes. But as our setting is unsupervised, there are no labels to guide the formation of these sets. Hence we propose to leverage the discriminative properties of the pre-trained input I3D features to initialize the positive and negative sets. Note that while the clusters are formed on the input features, our contrastive learning is done over the representation $\mathbf{f}$ produced by the C2F-TCN model (yellow panel of Fig. 3).

Specifically, we cluster the individual frame-wise inputs $\mathrm{V}[t]$ for all the videos within a small batch. We use k-means clustering and set the number of clusters as $2A$ (ablations in supplementary-S1), *i.e.* twice the number of actions to allow variability even within the same action. After clustering, each frame t is assigned the cluster label $l[t] \in \{1, \ldots, 2A\}$. Note that this simple clustering does not require videos of the same (or different) complex activities to appear in a mini-batch. It also does not incorporate temporal information – this differs from previous unsupervised works (Kukleva et al. 2019; VidalMata et al. 2021) that embed absolute temporal locations into the input features *before* clustering.

Representation Sampling Strategy: The videos used for action segmentation are long, *i.e.* 1-18k frames. Contrasting all the frames of every video in a batch would be too computationally expensive to consider, whereas contrastive loss of even a few representation back-propagates through the entire hierarchical TCN. To this end, we dynamically sample a fixed number of frames from each video to form the feature (representation) set $\mathcal{F}$ for each batch of videos. Note that the sampling is applied to the feature representations $\mathbf{f} = \mathbf{M}(\mathrm{V})$ and not to the inputs V; and the full input V is required to pass through the TCN to generate $\mathbf{f}$.

Let $\mathcal{I}$ denote the feature set index (as in sec) and for any feature index $(n, i) \in \mathcal{I}$, let n denote the video-id and i the sample-id within that video. For a video V_n and a fixed $K > 0$, we sample $2K$ frames $\{t_i^n : i \leq 2K\} \subset [0, 1]$ and obtain the feature set $\mathcal{F}_n := \{\mathbf{f}_n[t_i^n] : i \leq 2K\}$. To do so, we divide the unit interval $[0, 1]$ into K equal partitions and randomly choose a single frame from each partition. Another K frames are then randomly chosen ε away ($\varepsilon \ll 1/K$) from each of the first K samples. This strategy ensures diversity (the first K samples) while having nearby $\varepsilon-$distanced features (the second K samples) to either enforce temporal continuity if they are the same action, or learn boundaries if they are different actions (approximated by the cluster labels l when actions labels are unknown).

Frame-level positive and negative sets: Constructing the positive and negative set for each index $(n, i) \in \mathcal{I}$ requires a notion of similar features. The complex activity label is a strong cue, as there are either few or no shared actions across the different complex activities. For video V_n with complex activity c_n, we contrast index (m, j) with (n, i) if $c_m \neq c_n$. In datasets without meaningful complex activities (50Salads, GTEA), this condition is not applicable.

The cluster labels l of the input features already provides some separation between actions (see Table 1); we impose an additional temporal proximity condition to minimize the possibility of different action in the same cluster. Formally, we bring the representation with index (n, i) close to (m, j) if their cluster labels are same i.e $l_n[t_i^n] = l_m[t_j^m]$ *and* if they are close-by in time, i.e., $|t_i^n - t_j^m| \leq \delta$. For datasets with significant variations on the action sequence, *e.g.* 50Salads, the same action may occur at very different parts of the video so we choose higher δ, vs. smaller δ for actions that follow more regular ordering, *e.g.* Breakfast. Sampled features belonging to the same cluster but exceeding the temporal proximity, *i.e.* $l_n[t_i^n] = l_m[t_j^m]$ but $|t_i^n - t_j^m| > \delta$ are not considered for neither the positive nor the negative set.

Putting together the criteria from complex activity labels, clustering and temporal proximity, our positive and negative sets for index (n, i), *i.e.* sample i from video n is defined as

$$\begin{aligned} \mathcal{P}_{n,i} &= \{(m, j) \colon c_m = c_n, |t_i^n - t_j^m| < \delta,\, l_n[t_i^n] = l_m[t_j^m]\} \\ \mathcal{N}_{n,i} &= \{(m, j) \colon c_m \neq c_n\} \cup \\ &\qquad \{(m, j) \colon c_m = c_n, l_n[t_i^n] \neq l_m[t_j^m]\} \end{aligned} \tag{2}$$

where m, n are video indices, t_i^n is the frame-id corresponding to the i^{th} sample of video n, c_n is the complex activity of video n, and $l_n[t_i^n]$ the cluster label of frame t_i^n. For an index $(m, j) \in \mathcal{P}_{n,i}$, *i.e.* belonging to the positive set of (n, i), the contrastive probability becomes

$$p_{ij}^{nm} = \frac{e_\tau\Big(\mathbf{f}_n[t_i^n], \mathbf{f}_m[t_j^m]\Big)}{e_\tau\Big(\mathbf{f}_n[t_i^n], \mathbf{f}_m[t_j^m]\Big) + \sum\limits_{(r,k) \in \mathcal{N}_{n,i}} e_\tau\Big(\mathbf{f}_n[t_i^n], \mathbf{f}_r[t_k^r]\Big)}. \tag{3}$$

where e_τ is the τ-scaled exponential cosine similarity of Eq. (1). For feature representations $\mathbf{f}_n[t_i^n]$, Fig. 3 visualizes the positive set with pull-together green-arrows and negative set with push apart red-arrows.

Video-Level Contrastive Formulation

To further emphasize global differences between different complex activities, we construct video-level summary features $\mathbf{h}_n \in \mathbb{R}^d$ by max-pooling the frame-level features $\mathbf{f}_n \in \mathbb{R}^{T_n \times d}$ along the temporal dimension. For video V_n, we define video-level feature $\mathbf{h}_n = \max_{1 \leq t \leq T_n} \mathbf{f}_n[t]$. Intuitively, the max-pooling captures permutation-invariant features and has been effective for aggregating video segments (Sener, Singhania, and Yao 2020). With features $\mathbf{h}_n$, we formulate a video-level contrastive learning. Reusing the index set as video-ids, $\mathcal{I} = \{1, ..., |\mathcal{D}|\}$, we define a feature set $\mathcal{F} := \{\mathbf{h}_n : n \leq |\mathcal{D}|\}$, where for each video n, there is a positive set $\mathcal{P}_n := \{m \colon c_m = c_n\}$ and a negative set as $\mathcal{N}_n = \mathcal{I} \setminus \mathcal{P}_n$. For video n and another video $m \in \mathcal{P}_n$ in its positive set, the contrastive probability can be defined as

$$p_{nm} = \frac{e_\tau(\mathbf{h}_n, \mathbf{h}_m)}{e_\tau(\mathbf{h}_n, \mathbf{h}_m) + \sum_{r \in \mathcal{N}_n} e_\tau(\mathbf{h}_n, \mathbf{h}_r)}. \tag{4}$$

For our final **unsupervised representation learning** we use **contrastive loss** function $\mathcal{L}_{\text{con}}$ that sums the video-level and frame-level contrastive losses:

$$\mathcal{L}_{\text{con}} = -\frac{1}{N_1} \sum_n \sum_{m \in \mathcal{P}_n} \log p_{nm} - \frac{1}{N_2} \sum_{n,i} \sum_{m,j \in \mathcal{P}_{n,i}} \log p_{ij}^{nm}, \tag{5}$$

where $N_1 = \sum_n |\mathcal{P}_n|$, $N_2 = \sum_{n,i} |\mathcal{P}_{n,i}|$, and p_{ij}^{nm}, p_{nm} are as defined in equation (3) and (4) respectively. In practice, we compute this loss over mini-batches of videos.

Multi-Resolution Representation

We in this work show that constructing an appropriate representation can boost the performance of contrastive learning significantly. For this subsection, we switch to an absolute integer temporal index, *i.e.* for a video V the frame indices are $t \in \{1, \ldots, T\}$ where $T \geq 1$. The decoder $\mathbf{\Psi}$ has six layers; each layer producing features $\mathbf{z}_u, 1 \leq u \leq 6$ while progressively doubling the temporal resolution, *i.e.* the length of $\mathbf{z}_u$ is $\lceil T/2^{6-u} \rceil$. The temporally coarser features provide more global sequence-level information while the temporally fine-grained features contain more local information.

To leverage the full range of resolutions, we combine $\{\mathbf{z}_1, \ldots, \mathbf{z}_6\}$ into a new feature . Specifically, we upsample each decoder feature $\mathbf{z}_u$ to $\hat{\mathbf{z}}_u := \text{up}(\mathbf{z}_u, T)$ having a common length T using a temporal up-sampling function $\text{up}(\cdot, T)$ such as *'nearest'* or *'linear'* interpolation. The final frame-level representation for frame t is defined as $\mathbf{f}[t] = (\bar{\mathbf{z}}_1[t] : \bar{\mathbf{z}}_2[t] : \ldots : \bar{\mathbf{z}}_6[t])$, where $\bar{\mathbf{z}}_u[t] = \hat{\mathbf{z}}_u[t] / \|\hat{\mathbf{z}}_u[t]\|$, *i.e.* $\hat{\mathbf{z}}_u[t]$ is normalized and then concatenated along the latent dimension for each t (see Fig. 3). It immediately follows that for frames $1 \leq s, t \leq T$, the cosine similarity $\cos(\cdot)$ can be expressed as

$$\cos(\mathbf{f}[t], \mathbf{f}[s]) = \sum_{u=1}^{6} \omega_u \cdot \cos(\mathbf{z}_u[t], \mathbf{z}_u[s]). \quad (6)$$

As a result of our construction, the weights in Eq. 6 becomes $\omega_u = \frac{1}{6}$, *i.e.* each decoder layer an equal contribution in the cosine-similarity. Normalizing after concatenation, would cause Eq. (6)'s coefficients $\omega_u \propto \|\mathbf{z}_u[t]\| \cdot \|\mathbf{z}_u[s]\|$. The importance of this ordering is verified in Supplementary-S3.

Advantages: Our representation $\mathbf{f}$ encodes some degree of temporal continuity implicitly by design. For example, in the case of '*nearest*' up-sampling, it can be shown that for frames $1 \leq s, t \leq T$, if $\lfloor t/2^u \rfloor = \lfloor s/2^u \rfloor$ for some integer u, then it implies that $\text{sim}(\mathbf{f}[t], \mathbf{f}[s]) \geq 1 - u/3$ (detailed derivation in Supplementary-S3). Including temporally coarse features like $\mathbf{z}_1$ and $\mathbf{z}_2$ allows the finer-grained local features $\mathbf{z}_6$ to disagree with nearby frames without harming the temporal continuity. This makes the learned representations less prone to the common occurring problem of over-segmentation (Wang et al. 2020). This is demonstrated by the significant improvement in edit, F1 scores in the last row of Table 1.

Evaluating the Learned Representation

To evaluate the learned representations, we train a simple linear classifier $\mathbf{G}_f$ on $\mathbf{f}$ to classify frame-wise action labels. This form of evaluation is directly in line other works on unsupervised representation learning (Feichtenhofer et al. 2021; Chen et al. 2020b; Caron et al. 2021). The assumption is that if the unsupervised learned features are sufficiently strong, then a simple linear classifier is sufficient to separate the action classes. Note that while our representation learning is unsupervised, learning the classifier $\mathbf{G}_f$ is fully-supervised, using ground truth labels y with a cross-entropy loss $\mathcal{L}_{\text{ce}}$ over the standard splits of the respective datasets.

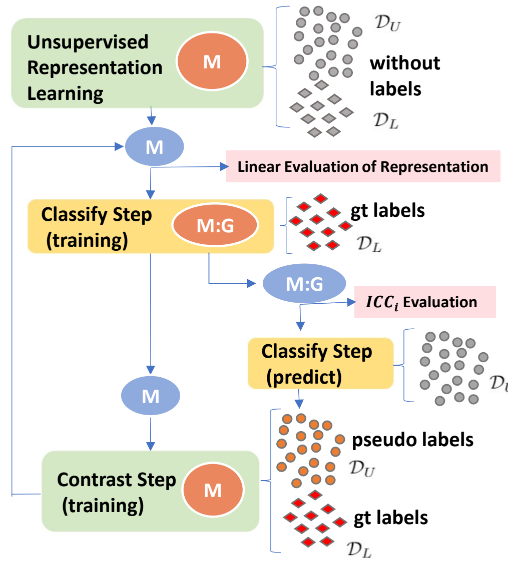

Figure 4: Depiction of Iterative-Contrast-Classify algorithm

Semi-Supervised Temporal Segmentation

After unsupervised representation learning, the model $\mathbf{M}$ cannot yet be applied for action segmentation. The decoder output must be coupled with a linear projection $\mathbf{G}$ and a soft-max to generate the actual segmentation. $\mathbf{G}$ can only be learned using labels, *i.e.* from $\mathcal{D}_L$, though the labels can be further leveraged to fine-tune $\mathbf{M}$ (Sec.). Afterwards, $\mathbf{M}$ and $\mathbf{G}$ can be applied to unlabelled data $\mathcal{D}_U$ to generate pseudo-labels. The pooled set of labels from $\mathcal{D}_L \cup \mathcal{D}_U$ can then be applied to update the representations in $\mathbf{M}$ (Sec.). By cycling between these updates, we propose an *Iterative-Contrast-Classify (ICC)* algorithm (Sec.) that performs semi-supervised action segmentation (see overview in Fig. 4).

Classify Step: Learning $\mathbf{G}, \mathbf{M}$ *with* $\mathcal{D}_L$

Similar to the supervised C2F-TCN, each decoder layer's representations $\mathbf{z}_u$ (temporal dim $\lceil T/2^{6-u} \rceil$) is projected with a linear layer $\mathbf{G}_u$ to A-dimensional vector where A is the number of action classes. This is followed by a softmax to obtain class probabilities $\mathbf{p}_u$ and a linear interpolation in time to up-sample back to the input length T. For frame t, the prediction $\mathbf{p}[t]$ is a weighted ensemble of up-sampled $\mathbf{p}_u$, *i.e.* $\mathbf{p}[t] = \sum_u \alpha_u \cdot \text{up}(\mathbf{p}_u, T)$ where α_u is the ensemble weight of decoder u with $\sum \alpha_u = 1$, and $\text{up}(\mathbf{p}_u, T)$ denotes the upsampled decoder output of length T. The sum is action-wise and the final predicted action label is $\hat{y}_t = \arg\max_{k \in \mathcal{A}} \mathbf{p}[t, k]$. Note that $\mathbf{G} := \{\mathbf{G}_u\}$ differs from the linear classifier $\mathbf{G}_f$ of Sec. . $\mathbf{G}_f$ is applied to the representation $\mathbf{f}$ used for the contrastive learning (see Sec.), whereas $\{\mathbf{G}_u\}$ are applied individually to different $\mathbf{z}_u$.

In addition to learning $\mathbf{G}$, $\mathcal{D}_L$ can be leveraged to fine-tune $\mathbf{M}$ as well. In Eq. (2), the positive and negative sets $\mathcal{P}_{n,i}$ and $\mathcal{N}_{n,i}$ can be modified for $\mathcal{D}_L$ to use the ground truth labels by replacing the unsupervised cluster labels $l_n[t_i^n]$ with ground truth action labels $y_n[t_i^n]$. Note that the learning rate used for fine-tuning the parameters of the model $\mathbf{M}$ is

	Breakfast					50Salads					GTEA				
	$F1@\{10,25,50\}$			Edit	MF	$F1@\{10,25,50\}$			Edit	MF	$F1@\{10,25,50\}$			Edit	MF
Input I3D Baseline	4.9	2.5	0.9	5.3	30.2	12.2	7.9	4.0	8.4	55.0	48.5	42.2	26.4	40.2	61.9
Our Representations	**57.0**	**51.7**	**39.1**	**51.3**	**70.5**	**40.8**	**36.2**	**28.1**	**32.4**	**62.5**	**70.8**	**65.0**	**48.0**	**65.7**	**69.1**
Improvement	52.1	49.2	38.2	46.0	40.3	28.6	28.3	24.1	24.0	7.5	22.3	22.8	21.6	25.5	7.2
Our unsupervised learning gives a large improvement in segmentation compared to input features.															
Cluster	11.7	8.0	3.9	12.2	36.1	18.5	13.7	8.5	13.6	50.8	57.3	48.6	31.6	52.4	60.5
(+) Proximity	24.4	19.2	11.5	21.3	50.0	18.6	13.5	8.0	13.5	51.6	62.9	56.6	38.0	52.6	62.2
(+) Video-Level	42.9	37.6	26.6	36.4	66.1	–	–	–	–	–	–	–	–	–	–
Contribution of clustering and time-proximity conditions and video-level constraints for contrastive learning (with $\mathbf{z}_6$).															
Last-Layer($\mathbf{z}_6$)	42.9	37.6	26.6	36.4	66.1	18.6	13.5	8.0	13.5	51.6	62.9	56.6	38.0	52.6	62.2
Multi-Resolution($\mathbf{f}$)	**57.0**	**51.7**	**39.1**	**51.3**	**70.5**	**40.8**	**36.2**	**28.1**	**32.4**	**62.5**	**70.8**	**65.0**	**48.0**	**65.7**	**69.1**
Improvement	14.1	14.1	12.5	14.9	4.4	22.2	22.7	20.1	18.9	10.9	7.9	8.4	10.0	13.1	6.9
Using Multi-Resolution($\mathbf{f}$) representation instead of final decoder $\mathbf{z}_6$ significantly improves learned representation scores.															

Table 1: Component-wise analysis of the unsupervised representation learning framework with a linear classifier.

significantly lower than linear projection layers $\mathbf{G}_u$. The loss used is $\mathcal{L} = \mathcal{L}_{\text{ce}}(\mathcal{D}_L) + \mathcal{L}'_{\text{con}}(\mathcal{D}_L)$, where $\mathcal{L}'_{\text{con}}$ is as defined in Eq. (5) but with $l_n[t_i^n]$ replaced by $y_n[t_i^n]$.

Contrast Step: Update $\mathbf{M}$ *with* $\mathcal{D}_U \cup \mathcal{D}_L$

After fine-tuning, we can use $\mathbf{M}$ and $\mathbf{G}$ to predict frame-level action labels $\hat{y}_n$ for any unlabelled videos, . *i.e.* pseudo-labels for $\mathcal{D}_U$. This affords the possibility update the representation in $\mathbf{M}$. To that end, we again modify $\mathcal{P}_{n,i}, \mathcal{N}_{n,i}$ in Eq. (2) by replacing the cluster labels $l_n[t_i^n]$ with the pseudo-labels $\hat{y}_n[t]$ and ground truth labels $y_n[t_i^n]$ for $\mathcal{D}_U$ and $\mathcal{D}_L$ respectively. $\mathbf{M}$ is then updated by applying the contrastive loss $\mathcal{L}' = \mathcal{L}'_{\text{con}}(\mathcal{D}_U \cup \mathcal{D}_L)$ where $\mathcal{L}'_{\text{con}}$ is as defined in Eq. (5).

Iterative-Contrast-Classify (ICC)

The pseudo labels for $\mathcal{D}_U$ are significantly better representative of the (unseen) action labels than the clusters obtained from the input I3D features used in the unsupervised stage. Thus we can improve our contrastive representation by using the pseduo labels (obtained after *classify*) for another *contrast* step. This refined representation in turn can help in finding better pseudo labels through another following *classify* step. By iterating between the contrast and classify in Secs. and (see Fig. 4), we can thus progressively improve the performance of the semi-supervised segmentation. The segmentation performance is evaluated at the end of the *classify* step after the training of $\mathbf{G}$. We denote the combined model of $\mathbf{M}$ and $\mathbf{G}$ for each iteration i as ICC_i. In this way, initial unsupervised representation learning can be considered the *"contrast"* step of ICC_1, where cluster labels are used instead of pseudo-labels. We found that performance saturates after 4 iterations of *contrast-classify* and refer to the performance of ICC_4 as our final semi-supervised result.

Experiments

Datasets, Evaluation, Implementation Details

We test on Breakfast Actions (Kuehne, Arslan, and Serre 2014) (1.7k videos, 10 complex activities, 48 actions), 50Salads (Stein and McKenna 2013) (50 videos, 19 actions) and GTEA (Fathi, Ren, and Rehg 2011) (28 videos, 11 actions). The standard evaluation criteria are the Mean-over-Frames (MoF), segment-wise edit distance (Edit), and $F1$-scores with IoU thresholds of 0.10, 0.25 and 0.50 ($F1@\{10,25,50\}$). We report results with the stronger I3D input features and make comparisons with IDT features in Supplementary-S4.1.

We use the specified train-test splits for each dataset and randomly select 5% or 10% of videos from the training split for labelled dataset $\mathcal{D}_L$. As GTEA and 50Salads are small, we use 3 and 5 videos as 5% and 10% respectively to incorporate all A actions,. We report mean and standard deviation of five different selections in Supplementary-S4.4. For unsupervised representation learning, we use all the unlabelled videos in the dataset which is in line with other unsupervised works (Kukleva et al. 2019; Chen et al. 2020a).

We sample frames from each video with $K = \{20, 60, 20\}$ partitions, $\varepsilon \approx \frac{1}{3K}$ for sampling, and temporal proximity $\delta = \{0.03, 0.5, 0.5\}$ for Breakfast, 50Salads, and GTEA respectively. The contrastive temperature τ in Eqs. (3) and (4) is set to 0.1. We also leverage the feature augmentations of C2F-TCN, with details and ablations in Supplementary-S2.1.

Evaluation of Representation Learning

Linear Classification Accuracy of our unsupervised representation learning (see Sec.) is shown in Table 1. In the first section, we evaluate the input I3D features with a linear classifier to serve as a baseline. Our representation has significant gains over the input I3D, verifying the ability of the base TCN to perform the task of segmentation with our designed unsupervised learning.

Frame- and Video-Level Contrastive Learning: The middle section of Table 1 breaks down the contributions from Sec. and when forming the positive and negative sets of contrastive learning from Eq. (2). The '*Cluster*' row applies cluster labels condition *i.e.* $l_n[t_i^n] = l_m[t_j^m]$ and '*(+) Proximity*' adds the condition $|t_i^n - t_j^m| < \delta$. Adding time proximity is more effective for Breakfast and GTEA likely because their videos follows a more rigid sequencing than 50Salads. Adding the *Video-Level* contrastive loss from Sec. in Breakfast gives further boosts.

Multi-Resolution Representation: The bottom section of Table 1 verifies that our multi-resolution representation $\mathbf{f}$ (see

$\%D_L$	**Method**	Breakfast $F1@\{10,25,50\}$			Edit	MoF	50Salads $F1@\{10,25,50\}$			Edit	MoF	GTEA $F1@\{10,25,50\}$			Edit	MoF
≈5	ICC_1	54.5	48.7	33.3	54.6	64.2	41.3	37.2	27.8	35.4	57.3	70.3	66.5	49.5	64.7	66.0
	ICC_2	56.9	51.9	34.8	56.5	65.4	45.7	40.9	30.7	40.9	59.5	77.0	70.6	54.1	67.8	68.0
	ICC_3	59.9	53.3	35.5	56.3	64.2	50.1	46.7	35.3	43.7	60.9	77.6	71.2	54.2	71.3	68.0
	ICC_4	60.2	53.5	35.6	56.6	65.3	52.9	49.0	36.6	45.6	61.3	77.9	71.6	54.6	71.4	68.2
	Gain	5.7	4.8	2.3	2.0	1.1	11.6	11.8	8.8	10.2	4.0	7.6	5.1	5.1	6.7	2.2
Progressive semi-supervised improvement with more iterations of ICC.																
≈5	Supervised	15.7	11.8	5.9	19.8	26.0	30.5	25.4	17.3	26.3	43.1	64.9	57.5	40.8	59.2	59.7
	Semi-Super	**60.2**	**53.5**	**35.6**	**56.6**	**65.3**	**52.9**	**49.0**	**36.6**	**45.6**	**61.3**	**77.9**	**71.6**	**54.6**	**71.4**	**68.2**
	Gain	44.5	41.7	29.7	36.8	39.3	22.4	23.6	19.3	19.3	18.2	13.0	14.1	13.8	12.2	8.5
≈10	Supervised	35.1	30.6	19.5	36.3	40.3	45.1	38.3	26.4	38.2	54.8	66.2	61.7	45.2	62.5	60.6
	Semi-Super	**64.6**	**59.0**	**42.2**	**61.9**	**68.8**	**67.3**	**64.9**	**49.2**	**56.9**	**68.6**	**83.7**	**81.9**	**66.6**	**76.4**	**73.3**
	Gain	29.5	28.4	22.7	25.6	28.5	22.2	26.6	22.8	18.7	13.8	17.5	20.2	21.4	13.9	12.7
Semi-Super (our ICC_4) significantly improves supervised counterpart using same labelled data amount. See also Fig. 1																
100	Supervised*	69.4	65.9	55.1	66.5	73.4	75.8	73.1	62.3	68.8	79.4	90.1	87.8	74.9	86.7	79.5
	Semi-Super	**72.4**	**68.5**	**55.9**	**68.6**	**75.2**	**83.8**	**82.0**	**74.3**	**76.1**	**85.0**	**91.4**	**89.1**	**80.5**	**87.8**	**82.0**
	Gain	3.0	2.6	0.8	2.1	1.8	8.0	8.9	12.0	7.3	5.6	1.3	1.3	5.6	1.1	2.5
Semi-Super (our ICC_4) helps improve fully supervised learning. * reported from C2F-TCN without test-augment-action-loss.																

Table 2: Our final all metrics evaluation of proposed ICC algorithm on 3 benchmark action segmentation datasets.

Sec.) outperforms the use of only the final decoder layer feature $\mathbf{z}_6$. Gains are especially notable for the F1-score and Edit-distance, verifying that $\mathbf{f}$ has less over-segmentation.

No Initial Representation Learning: Although model $\mathbf{M}$ is continually updated during the ICC, Supplementary-S4.2 shows that the initial unsupervised representation learning is critical. Bypassing this step in the first iteration of ICC and going straight to *classify* step results in a gap of $\geq 10\%$ F1.

Evaluation of Semi-Supervised Learning

ICC Components: The topmost section of Table 2 shows the progressive improvements as we increase the number of iterations of our proposed ICC algorithm. The gain in performance is especially noticeable for Edit and F1 scores. Segmentation result reported are after the *classify* step. Improvements gained by updating the feature representation after the *contrast* step but before *classify* of the next iteration is shown in the Supplementary-S4.3.

Additional Ablations: Due to lack of space, we refer the reader to Supplementary-S2. Notably, we demonstrate ICC results with alternative base model ED-TCN (Lea et al. 2017) and draw difference to using MS-TCN (Li et al. 2020).

Semi-Supervised vs Supervised: The middle and bottom sections of Table 2 shows our final *'Semi-Super'* results, *i.e.* ICC_4 for various percentages of labelled data. We compare with the *'Supervised'* case of training the base model C2F-TCN with the same labelled dataset $\mathcal{D}_L$; for fairness, 100% C2F-TCN results is reported without test-time augmentations and action loss. ICC significantly outperforms the supervised baseline in all the metrics (see also Fig. 1) for all amounts of training data, including 100%. In fact, with just 5% of labelled videos, we are only 8% less in MoF in Breakfast actions compared to fully-supervised (100%). Using less than 5% (3 videos for 50salads and GTEA) of training videos makes it difficult to ensure coverage of all the actions.

	Method	Breakfast	50salads	GTEA
Full	MSTCN'20	67.6	83.7	78.9
	SSTDA'20	70.2	83.2	79.8
	*C2F-TCN'21	73.4	79.4	79.5
	Ours ICC (100%)	**75.2**	**85.0**	**82.0**
Weakly	SSTDA(65%)	65.8	80.7	75.7
	TSS'21	64.1	75.6	66.4
Semi	Ours ICC (40%)	71.1	78.0	78.4
	Ours ICC (10%)	68.8	68.6	73.3
	Ours ICC (5%)	65.3	61.3	68.2

Table 3: Segmentation MoF comparison with SOTA on 3 benchmark datasets. Our ICC can improve its fully-supervised counterpart. Our semi-supervised results is competitive in MoF with different levels of supervision.

Comparison to *SOTA* and differences: Our work being the first to do semi-supervised temporal action segmentation, is not directly comparable to other works. Table 3 shows our MoF is competitive with other forms of supervision on all three datasets. TSS and SSTDA uses weak labels for *all* videos vs. our work requiring full labels for only few videos.

Conclusions

In our work we show that pre-trained input features that capture semantics and motion of short-trimmed video segments can be used to learn higher-level representations to interpret long video sequences. Our proposed multi-resolution representation formed with outputs from multiple decoder layers, implicitly bring temporal continuity and consequently large improvements in unsupervised contrastive representation learning. Our final semi-supervised learning algorithm ICC can significantly reduce the annotation efforts, with 40% labelled videos approximately achieving fully-supervised (100%) performance. Furthermore, ICC also improves performance even when used with 100% labels.

Acknowledgements

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-RP-2020-016). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

References

Alwassel, H.; Mahajan, D.; Korbar, B.; Torresani, L.; Ghanem, B.; and Tran, D. 2019. Self-supervised learning by cross-modal audio-video clustering. *arXiv preprint arXiv:1911.12667*.

Bai, Y.; Fan, H.; Misra, I.; Venkatesh, G.; Lu, Y.; Zhou, Y.; Yu, Q.; Chandra, V.; and Yuille, A. 2020. Can Temporal Information Help with Contrastive Self-Supervised Learning? *arXiv preprint arXiv:2011.13046*.

Bearman, A.; Russakovsky, O.; Ferrari, V.; and Fei-Fei, L. 2016. What's the point: Semantic segmentation with point supervision. In *European conference on computer vision*, 549–565. Springer.

Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; and Joulin, A. 2021. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. arXiv:2006.09882.

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

Chang, C.-Y.; Huang, D.-A.; Sui, Y.; Fei-Fei, L.; and Niebles, J. C. 2019. D3tw: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3546–3555.

Chen, M.-H.; Li, B.; Bao, Y.; AlRegib, G.; and Kira, Z. 2020a. Action segmentation with joint self-supervised temporal domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9454–9463.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. E. 2020b. A Simple Framework for Contrastive Learning of Visual Representations. *CoRR*, abs/2002.05709.

Chen, X.; Yao, L.; Zhou, T.; Dong, J.; and Zhang, Y. 2021. Momentum contrastive learning for few-shot COVID-19 diagnosis from chest CT images. *Pattern Recognition*, 113: 107826.

Ding, L.; and Xu, C. 2018. Weakly-supervised action segmentation with iterative soft boundary assignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6508–6516.

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.

Fathi, A.; Ren, X.; and Rehg, J. M. 2011. Learning to recognize objects in egocentric activities. In *CVPR 2011*, 3281–3288. IEEE.

Feichtenhofer, C.; Fan, H.; Xiong, B.; Girshick, R.; and He, K. 2021. A Large-Scale Study on Unsupervised Spatiotemporal Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3299–3309.

Hadsell, R.; Chopra, S.; and LeCun, Y. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, 1735–1742. IEEE.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9729–9738.

Hung, W.-C.; Tsai, Y.-H.; Liou, Y.-T.; Lin, Y.-Y.; and Yang, M.-H. 2018. Adversarial Learning for Semi-Supervised Semantic Segmentation. arXiv:1802.07934.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*.

Kong, Q.; Wei, W.; Deng, Z.; Yoshinaga, T.; and Murakami, T. 2020. Cycle-Contrast for Self-Supervised Video Representation Learning. *arXiv preprint arXiv:2010.14810*.

Kuehne, H.; Arslan, A.; and Serre, T. 2014. 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*, 780–787.

Kuehne, H.; Richard, A.; and Gall, J. 2018. A hybrid RNN-HMM approach for weakly supervised temporal action segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 42(4): 765–779.

Kukleva, A.; Kuehne, H.; Sener, F.; and Gall, J. 2019. Unsupervised learning of action classes with continuous temporal embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12066–12074.

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, J.; Lei, P.; and Todorovic, S. 2019. Weakly supervised energy-based learning for action segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6243–6251.

Li, S.-J.; AbuFarha, Y.; Liu, Y.; Cheng, M.-M.; and Gall, J. 2020. Ms-tcn++: Multi-stage temporal convolutional network for action segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Li, Z.; Farha, Y. A.; and Gall, J. 2021. Temporal Action Segmentation from Timestamp Supervision. *arXiv preprint arXiv:2103.06669*.

Lorre, G.; Rabarisoa, J.; Orcesi, A.; Ainouz, S.; and Canu, S. 2020. Temporal contrastive pretraining for video action recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 662–670.

Ma, F.; Zhu, L.; Yang, Y.; Zha, S.; Kundu, G.; Feiszli, M.; and Shou, Z. 2020. Sf-net: Single-frame supervision for temporal action localization. In *European conference on computer vision*, 420–437. Springer.

Miech, A.; Alayrac, J.-B.; Smaira, L.; Laptev, I.; Sivic, J.; and Zisserman, A. 2020. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9879–9889.

Mittal, S.; Tatarchenko, M.; and Brox, T. 2021. Semi-Supervised Semantic Segmentation With High- and Low-Level Consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(4): 1369–1379.

Qian, R.; Meng, T.; Gong, B.; Yang, M.-H.; Wang, H.; Belongie, S.; and Cui, Y. 2020. Spatiotemporal contrastive video representation learning. *arXiv preprint arXiv:2008.03800*.

Rahaman, R.; Ghosh, A.; and Thiery, A. H. 2021. Pretrained equivariant features improve unsupervised landmark discovery. *arXiv preprint arXiv:2104.02925*.

Richard, A.; Kuehne, H.; Iqbal, A.; and Gall, J. 2018. Neuralnetwork-viterbi: A framework for weakly supervised video learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7386–7395.

Sarfraz, M. S.; Murray, N.; Sharma, V.; Diba, A.; Van Gool, L.; and Stiefelhagen, R. 2021. Temporally-Weighted Hierarchical Clustering for Unsupervised Action Segmentation. *arXiv preprint arXiv:2103.11264*.

Sener, F.; Singhania, D.; and Yao, A. 2020. Temporal aggregate representations for long-range video understanding. In *European Conference on Computer Vision*, 154–171. Springer.

Sener, F.; and Yao, A. 2018. Unsupervised learning and segmentation of complex activities from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8368–8376.

Sermanet, P.; Lynch, C.; Chebotar, Y.; Hsu, J.; Jang, E.; Schaal, S.; Levine, S.; and Brain, G. 2018. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 1134–1141. IEEE.

Singhania, D.; Rahaman, R.; and Yao, A. 2021. Coarse to Fine Multi-Resolution Temporal Convolutional Network. arXiv:2105.10859.

Souri, Y.; Fayyaz, M.; Minciullo, L.; Francesca, G.; and Gall, J. 2019. Fast Weakly Supervised Action Segmentation Using Mutual Consistency. *arXiv preprint arXiv:1904.03116*.

Stein, S.; and McKenna, S. J. 2013. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, 729–738. ACM.

VidalMata, R. G.; Scheirer, W. J.; Kukleva, A.; Cox, D.; and Kuehne, H. 2021. Joint visual-temporal embedding for unsupervised learning of actions in untrimmed sequences. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1238–1247.

Wang, H.; and Schmid, C. 2013. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, 3551–3558.

Wang, N.; Zhou, W.; and Li, H. 2020. Contrastive Transformation for Self-supervised Correspondence Learning. *arXiv preprint arXiv:2012.05057*.

Wang, Z.; Gao, Z.; Wang, L.; Li, Z.; and Wu, G. 2020. Boundary-aware cascade networks for temporal action segmentation. In *European Conference on Computer Vision*, 34–51. Springer.