Unsupervised Action Segmentation by Joint Representation Learning and Online Clustering

Sateesh Kumar† Sanjay Haresh† Awais Ahmed Andrey Konin
M. Zeeshan Zia Quoc-Huy Tran

Retrocausal, Inc.
Seattle, WA
www.retrocausal.ai

Abstract

We present a novel approach for unsupervised activity segmentation which uses video frame clustering as a pretext task and simultaneously performs representation learning and online clustering. This is in contrast with prior works where representation learning and clustering are often performed sequentially. We leverage temporal information in videos by employing temporal optimal transport. In particular, we incorporate a temporal regularization term which preserves the temporal order of the activity into the standard optimal transport module for computing pseudo-label cluster assignments. The temporal optimal transport module enables our approach to learn effective representations for unsupervised activity segmentation. Furthermore, previous methods require storing learned features for the entire dataset before clustering them in an offline manner, whereas our approach processes one mini-batch at a time in an online manner. Extensive evaluations on three public datasets, i.e. 50-Salads, YouTube Instructions, and Breakfast, and our dataset, i.e., Desktop Assembly, show that our approach performs on par with or better than previous methods, despite having significantly less memory constraints.

1. Introduction

With the advent of deep learning, significant progress has been made in understanding human activities in videos. However, most of the research efforts so far have been invested in action recognition [11, 76, 77, 82], where the task is to classify simple actions in short videos. Recently, a few approaches have been proposed for dealing with complex activities in long videos, e.g., temporal action localization [13, 68, 69, 88], which aims to detect video segments containing the actions of interest, and anomaly detection [27, 32, 74], whose goal is to localize video frames containing anomalous events in an untrimmed video.

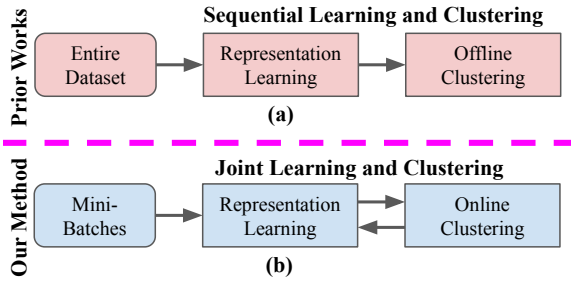

Figure 1. (a) Previous approaches [43, 52, 65, 78] often perform representation learning and clustering sequentially, while storing embedded features for the entire dataset before clustering them. (b) We unify representation learning and clustering into a single joint framework, which processes one mini-batch at a time. Our method explicitly optimizes for unsupervised activity segmentation and is much more memory efficient.

In this paper, we are interested in the problem of temporal activity segmentation, where our goal is to assign each frame of a long video capturing a complex activity to one of the action/sub-activity classes. One popular group of methods [14, 40, 41, 46, 53] on this topic require per-frame action labels for fully-supervised training. However, frame-level annotations for all training videos are generally difficult and prohibitively costly to acquire. Weakly-supervised approaches which need weak labels, e.g., the ordered action list or transcript for each video [12, 18, 35, 42, 50, 61, 62, 64], have also been proposed. Unfortunately, these weak labels are not always available a priori and can be time consuming to obtain, especially for large datasets.

To avoid the above annotation requirements, unsupervised methods [3, 43, 52, 57, 65, 66, 78] have been introduced recently. Given a collection of unlabeled videos,

† indicates joint first author.
{sateesh,sanjay,awais,andrey,zeeshan,huy}@retrocausal.ai.

they *jointly* discover the actions and segment the videos by grouping frames across all videos into clusters, with each cluster corresponding to one of the actions. Previous approaches [43, 52, 65, 78] in unsupervised activity segmentation usually separate the representation learning step from the clustering step in a sequential learning and clustering framework (see Fig. 1(a)), which prevents the feedback from the clustering step from flowing back to the representation learning step. Also, they need to store computed features for the entire dataset before clustering them in an offline manner, leading to inefficient memory usage.

In this work, we present a joint representation learning and online clustering approach for unsupervised activity segmentation (see Fig. 1(b)), which uses video frame clustering as a pretext task and hence directly optimizes for unsupervised activity segmentation. We employ temporal optimal transport to leverage temporal information in videos. Specifically, the temporal optimal transport module preserves the temporal order of the activity when computing pseudo-label cluster assignments, yielding effective representations for unsupervised activity segmentation. In addition, our approach processes one mini-batch at a time, thus having substantially lesser memory requirements.

In summary, our contributions include:

- We propose a novel method for unsupervised activity segmentation, which jointly performs representation learning and online clustering. We leverage video frame clustering as a pretext task, thus directly optimizing for unsupervised activity segmentation.
- We introduce the temporal optimal transport module to exploit temporal cues in videos by imposing temporal order-preserving constraints on computed pseudo-label cluster assignments, yielding effective representations for unsupervised activity segmentation.
- Our method performs on par with or better than the state-of-the-art in unsupervised activity segmentation on public datasets, i.e., 50-Salads, YouTube Instructions, and Breakfast, and our dataset, i.e., Desktop Assembly, while being much more memory efficient.
- We collect and label our Desktop Assembly dataset, which is available at https://bit.ly/3JKm0JP.

2. Related Work

Below we summarize related works in temporal activity segmentation and self-supervised representation learning.

Unsupervised Activity Segmentation. Early methods [3, 57, 66] in unsupervised activity segmentation explore cues from the accompanying narrations for segmenting the videos. They assume the narrations are available and well-aligned with the videos, which is not always the case and hence limits their applications. Approaches [43, 52, 65, 78] which rely purely on visual inputs have been developed recently. Sener et al. [65] propose an iterative approach which alternates between learning a discriminative appearance model and optimizing a generative temporal model of the activity, while Kukleva et al. [43] introduce a multi-step approach which includes learning a temporal embedding and performing K-means clustering on the learned features. VidalMata et al. [78] and Li and Todorovic [52] further improve the approach of [43] by learning a visual embedding and an action-level embedding respectively. The above approaches [43, 52, 65, 78] usually separate representation learning from clustering, and require storing learned features for the whole dataset before clustering them. In contrast, our approach combines representation learning and clustering into a single joint framework, while processing one mini-batch at a time, leading to better results and memory efficiency. More recently, the work by Swetha et al. [75] proposes a joint representation learning and clustering approach. However, our approach is different from theirs in several aspects. Firstly, we employ optimal transport for clustering, while they use discriminative learning. Secondly, for representation learning, we employ clustering-based loss, while they use reconstruction loss. Lastly, despite our simpler encoder, our approach has similar or superior performance than theirs on public datasets.

Weakly-Supervised Activity Segmentation. A few works focus on weak supervision for temporal activity segmentation such as the order of actions appearing in a video, i.e., transcript supervision [12, 18, 35, 42, 50, 61, 62, 64], and the set of actions occurring in a video, i.e., set supervision [21, 51, 63]. Recently, Li et al. [54] apply timestamp supervision for temporal activity segmentation, which requires annotating a single frame for each action segment. Our approach, however, does not require any action labels.

Image-Based Self-Supervised Representation Learning. Since the early work of Hinton and Zemel [34], considerable efforts [7, 22, 26, 38, 44, 45, 56, 60, 79] have been invested in designing pretext tasks with artificial image labels for training deep networks for self-supervised representation learning. These include image denoising [79], image colorization [44, 45], object counting [56, 60], solving jigsaw puzzles [7, 38], and predicting image rotations [22, 26]. Recently, a few approaches [4, 5, 8–10, 25, 36, 84, 86, 87, 90] leveraging clustering as a pretext task have been introduced. For example, in [8, 9], K-means cluster assignments are used as pseudo-labels for learning self-supervised image representations, while the pseudo-label assignments are obtained by solving the optimal transport problem in [4, 10]. In this paper, we focus on learning self-supervised video representations, which requires exploring both spatial and temporal cues in videos. In particular, we follow the clustering-based approaches of [4, 10], however, unlike them, we em-

ploy temporal optimal transport to leverage temporal cues.

Video-Based Self-Supervised Representation Learning. Over the past few decades, a variety of pretext tasks have been proposed for learning self-supervised video representations [2, 6, 15, 17, 23, 24, 28–30, 37, 47, 58, 59, 71, 80, 85, 91, 92]. A popular group of methods learn representations by predicting future frames [2, 17, 71, 80] or their encoding features [24, 30, 37]. Another group explore temporal information such as temporal order [15, 23, 47, 58, 85] and temporal coherence [6, 28, 29, 59, 91, 92]. The above approaches process a single video at a time. Recently, a few methods [19, 31, 67] which optimize over a pair of videos at once have been introduced. TCN [67] learns representations via the time-contrastive loss across different viewpoints and neighboring frames, while TCC [19] and LAV [31] perform frame matching and temporal alignment between videos respectively. Here, we learn self-supervised representations by clustering video frames, which directly optimizes for the downstream task of unsupervised activity segmentation.

3. Our Approach

We now describe our main contribution, which is an unsupervised approach for activity segmentation. In particular, we propose a joint self-supervised representation learning and online clustering approach, which uses video frame clustering as a pretext task and hence directly optimizes for unsupervised activity segmentation. We exploit temporal information in videos by using temporal optimal transport. Fig. 2 shows an overview of our approach. Below we first define some notations and then provide the details of our representation learning and online clustering modules.

Notations. We denote the embedding function as $f_{\boldsymbol{\theta}}$, i.e., a neural network with learnable parameters $\boldsymbol{\theta}$. Our approach takes as input a mini-batch $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_B\}$, where B is the number of frames in $\boldsymbol{X}$. For a frame $\boldsymbol{x}_i$ in $\boldsymbol{X}$, the embedding features of $\boldsymbol{x}_i$ are expressed as $\boldsymbol{z}_i = f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) \in \mathbb{R}^D$, with D being the dimension of the embedding features. The embedding features of $\boldsymbol{X}$ are then written as $\boldsymbol{Z} = [\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_B]^\top \in \mathbb{R}^{B\times D}$. Moreover, we denote $\boldsymbol{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \dots, \boldsymbol{c}_K]^\top \in \mathbb{R}^{K\times D}$ as the learnable prototypes of the K clusters, with $\boldsymbol{c}_j$ representing the prototype of the j-th cluster. Lastly, $\boldsymbol{P} \in \mathbb{R}_+^{B\times K}$ and $\boldsymbol{Q} \in \mathbb{R}_+^{B\times K}$ are the predicted cluster assignments (i.e., predicted "codes") and pseudo-label cluster assignments (i.e., pseudo-label "codes") respectively.

3.1. Representation Learning

To learn self-supervised representations for unsupervised activity segmentation, our proposed idea is to use video frame clustering as a pretext task. Thus, the learned features are explicitly optimized for unsupervised activity segmentation. Here, we consider a similar clustering-based self-supervised representation learning approach as [4, 10]. However, unlike their approaches which are designed for image data, we propose temporal optimal transport to make use of temporal information additionally available in video data. Below we describe our losses for learning representations for unsupervised activity segmentation.

Cross-Entropy Loss. Given the frames $\boldsymbol{X}$, we first pass them to the encoder $f_{\boldsymbol{\theta}}$ to obtain the features $\boldsymbol{Z}$. We then compute the predicted codes $\boldsymbol{P}$ with each entry written as:

$$\boldsymbol{P}_{ij} = \frac{\exp(\frac{1}{\tau}\boldsymbol{z}_i^\top \boldsymbol{c}_j)}{\sum_{j'=1}^{K}\exp(\frac{1}{\tau}\boldsymbol{z}_i^\top \boldsymbol{c}_{j'})}, \tag{1}$$

where $\boldsymbol{P}_{ij}$ is the probability that the i-th frame is assigned to the j-th cluster and τ is the temperature parameter [83]. The pseudo-label codes $\boldsymbol{Q}$ are computed by solving the temporal optimal transport problem, which we will describe in the next section. For clustering-based representation learning, we minimize the cross-entropy loss with respect to the encoder parameters $\boldsymbol{\theta}$ and the prototypes $\boldsymbol{C}$ as:

$$L_{CE} = -\frac{1}{B}\sum_{i=1}^{B}\sum_{j=1}^{K}\boldsymbol{Q}_{ij}\log \boldsymbol{P}_{ij}. \tag{2}$$

Temporal Coherence Loss. To further exploit temporal information in videos, we consider adding another self-supervised loss, i.e., the temporal coherence loss. It learns an embedding space following the temporal coherence constraints [28, 29, 59], where temporally close frames should be mapped to nearby points and temporally distant frames should be mapped to far away points. To enable fast convergence and effective representations, we employ the N-pair metric learning loss proposed by [70]. For each video, we first sample a subset of N ordered frames denoted by $\{\boldsymbol{z}_i\}$ (with $i \in \{1, 2, ..., N\}$). For each $\boldsymbol{z}_i$, we then sample a "positive" example $\boldsymbol{z}_i^+$ inside a temporal window of λ from $\boldsymbol{z}_i$. Moreover, $\boldsymbol{z}_j^+$ sampled for $\boldsymbol{z}_j$ (with $j \neq i$) is considered as a "negative" example for $\boldsymbol{z}_i$. We minimize the temporal coherence loss with respect to the encoder parameters $\boldsymbol{\theta}$ as:

$$L_{TC} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\boldsymbol{z}_i^\top \boldsymbol{z}_i^+)}{\sum_{j=1}^{N}\exp(\boldsymbol{z}_i^\top \boldsymbol{z}_j^+)}. \tag{3}$$

Final Loss. Our final loss is written as:

$$L = L_{CE} + \alpha L_{TC}. \tag{4}$$

Here, α is the weight for the temporal coherence loss. Our final loss is optimized with respect to $\boldsymbol{\theta}$ and $\boldsymbol{C}$. The cross-entropy loss and the temporal coherence loss are differentiable and can be optimized using backpropagation. Note that we do not backpropagate through $\boldsymbol{Q}$.

3.2. Online Clustering

Below we describe our online clustering module for computing the pseudo-label codes $\boldsymbol{Q}$ online. Following [4, 10], we consider the problem of computing $\boldsymbol{Q}$ as

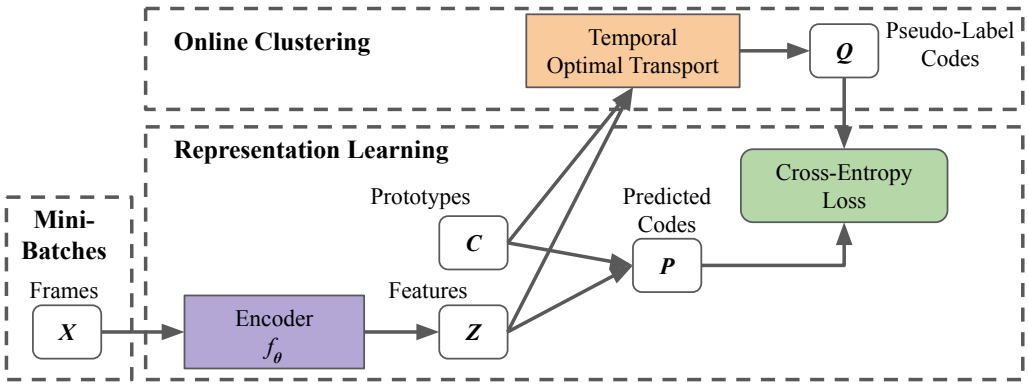

Figure 2. Given the frames $\boldsymbol{X}$, we feed them to the encoder $f_{\boldsymbol{\theta}}$ to obtain the features $\boldsymbol{Z}$, which are combined with the prototypes $\boldsymbol{C}$ to produce the predicted codes $\boldsymbol{P}$. Meanwhile, $\boldsymbol{Z}$ and $\boldsymbol{C}$ are also fed to the temporal optimal transport module to compute the pseudo-label codes $\boldsymbol{Q}$. We jointly learn $\boldsymbol{\theta}$ and $\boldsymbol{C}$ by applying the cross-entropy loss on $\boldsymbol{P}$ and $\boldsymbol{Q}$.

the optimal transport problem and solve for $\boldsymbol{Q}$ online by using a mini-batch $\boldsymbol{X}$ at a time. This is different from prior works [43, 52, 65, 78] for unsupervised activity segmentation, which require storing features for the entire dataset before clustering them in an offline fashion and hence have significantly more memory constraints.

Optimal Transport. Given the features $\boldsymbol{Z}$ extracted from the frames $\boldsymbol{X}$, our goal is to compute the pseudo-label codes $\boldsymbol{Q}$ with each entry $\boldsymbol{Q}_{ij}$ representing the probability that the features $\boldsymbol{z}_i$ are mapped to the prototype $\boldsymbol{c}_j$. Specifically, $\boldsymbol{Q}$ is computed by solving the optimal transport problem as:

$$\max_{\boldsymbol{Q}\in\mathcal{Q}} \; Tr(\boldsymbol{Q}^\top \boldsymbol{Z}\boldsymbol{C}^\top) + \epsilon H(\boldsymbol{Q}), \tag{5}$$

$$\mathcal{Q} = \left\{ \boldsymbol{Q} \in \mathbb{R}_+^{B\times K} : \quad \boldsymbol{Q}\mathbf{1}_K = \frac{1}{B}\mathbf{1}_B, \boldsymbol{Q}^\top \mathbf{1}_B = \frac{1}{K}\mathbf{1}_K \right\}. \tag{6}$$

Here, $\mathbf{1}_B$ and $\mathbf{1}_K$ denote vectors of ones in dimensions B and K respectively. In Eq. 5, the first term measures the similarity between the features $\boldsymbol{Z}$ and the prototypes $\boldsymbol{C}$, while the second term (i.e., $H(\boldsymbol{Q}) = -\sum_{i=1}^{B}\sum_{j=1}^{K} \boldsymbol{Q}_{ij} \log \boldsymbol{Q}_{ij}$) measures the entropy regularization of $\boldsymbol{Q}$, and ϵ is the weight for the entropy term. A large value of ϵ usually leads to a trivial solution where every frame has the same probability of being assigned to every cluster. Thus, we use a small value of ϵ in our experiments to avoid the above trivial solution. Furthermore, Eq. 6 represents the *equal partition* constraints, which enforce that each cluster is assigned the same number of frames in a mini-batch, thus preventing a trivial solution where all frames are assigned to a single cluster. Although the above equal partition prior does not hold for activities with various action lengths, we find that in practice it works relatively well for most activities with various action lengths (e.g., please see Fig. 5 and more discussion in the supplementary material). The solution for the above optimal transport problem can be computed by using the iterative Sinkhorn-Knopp algorithm [16] as:

$$\boldsymbol{Q}_{OT} = \text{diag}(\boldsymbol{u}) \exp\left(\frac{\boldsymbol{Z}\boldsymbol{C}^\top}{\epsilon}\right) \text{diag}(\boldsymbol{v}), \tag{7}$$

where $\boldsymbol{u} \in \mathbb{R}^B$ and $\boldsymbol{v} \in \mathbb{R}^K$ are renormalization vectors.

Temporal Optimal Transport. The above approach is originally developed for image data in [4, 10] and hence is not capable of exploiting temporal cues in video data for unsupervised activity segmentation. Thus, we propose to incorporate a temporal regularization term which preserves the temporal order of the activity into the objective in Eq. 5, yielding the temporal optimal transport.

Motivated by [73], we introduce a prior distribution for $\boldsymbol{Q}$, namely $\boldsymbol{T} \in \mathbb{R}_+^{B\times K}$, where the highest values appear on the diagonal and the values gradually decrease along the direction perpendicular to the diagonal. Specifically, $\boldsymbol{T}$ maintains a *fixed order* of the clusters, and enforces initial frames to be assigned to initial clusters and later frames to be assigned to later clusters. Mathematically, $\boldsymbol{T}$ can be represented by a 2D distribution, whose marginal distribution along any line perpendicular to the diagonal is a Gaussian distribution centered at the intersection on the diagonal, as:

$$\boldsymbol{T}_{ij} = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{d_{ij}^2}{2\sigma^2}\right), d_{ij} = \frac{|i/B - j/K|}{\sqrt{1/B^2 + 1/K^2}}, \tag{8}$$

where d_{ij} is the distance from the entry (i, j) to the diagonal line. Though the above temporal order-preserving prior does not hold for activities with permutations, we empirically observe that it performs relatively well on most datasets containing permutations (e.g., please see Tabs. 3, 4, 5, and more discussion in the supplementary material).

To encourage the distribution of values of $\boldsymbol{Q}$ to be as similar as possible to $\boldsymbol{T}$, we replace the objective in Eq. 5 with the temporal optimal transport objective:

$$\max_{\boldsymbol{Q}\in\mathcal{Q}} \; Tr(\boldsymbol{Q}^\top \boldsymbol{Z}\boldsymbol{C}^\top) - \rho KL(\boldsymbol{Q}||\boldsymbol{T}). \tag{9}$$

Here, $KL(\boldsymbol{Q}||\boldsymbol{T}) = \sum_{i=1}^{B}\sum_{j=1}^{K} \boldsymbol{Q}_{ij} \log \frac{\boldsymbol{Q}_{ij}}{\boldsymbol{T}_{ij}}$ is the Kullback-Leibler (KL) divergence between $\boldsymbol{Q}$ and $\boldsymbol{T}$, and ρ is the weight for the KL term. Note that $\mathcal{Q}$ is defined as in Eq. 6. Following [16], we can derive the solution for the above temporal optimal transport problem as:

$$\boldsymbol{Q}_{TOT} = \text{diag}(\boldsymbol{u}) \exp\left(\frac{\boldsymbol{Z}\boldsymbol{C}^\top + \rho \log \boldsymbol{T}}{\rho}\right) \text{diag}(\boldsymbol{v}), \tag{10}$$

where $\boldsymbol{u} \in \mathbb{R}^B$ and $\boldsymbol{v} \in \mathbb{R}^K$ are renormalization vectors.

In contrast to previous methods [43, 52, 65, 78] which require features of the entire dataset to be loaded into memory, our method requires only a mini-batch of features to be loaded in memory at a time. This reduces the memory requirement significantly from $O(N)$ to $O(B)$, where B is the mini-batch size, N is the total number of frames in the entire dataset, and B is much smaller than N, especially for large datasets. For example, CTE [43] requires a memory of $57795 \times 30 \times 8$ bytes for storing features on the 50 Salads dataset, whereas our method requires $512 \times 30 \times 8$ bytes for the same purpose, where $N = 57795$, $B = 512$, and 30 is the size of the final embedding.

4. Experiments

Implementation Details. We use a 2-layer MLP for learning the embedding on top of pre-computed features (see below). The MLP is followed by a dot-product operation with the prototypes which are initialized randomly and learned via backpropagation through the losses presented in Sec. 3.1. The ADAM optimizer [39] is used with a learning rate of 10^{-3} and a weight decay of 10^{-4}. For each activity, the number of prototypes is set as the number of actions in the activity. For our approach, the order of the actions is fixed as mentioned in Sec. 3.2. During inference, cluster assignment probabilities for all frames are computed. These probabilities are then passed to a Viterbi decoder for smoothing out the probabilities given the order of the actions. Note that, for a fair comparison, the above protocol is the same as in CTE [43], which is the closest work to ours. Please see more details in the supplementary material.

Datasets. We use three public datasets (all under Creative Commons License), namely 50 Salads [72], YouTube Instructions (YTI) [3], and Breakfast [40], while introducing our Desktop Assembly dataset:

- *50 Salads* consists of 50 videos of actors performing a cooking activity. The total video duration is about 4.5 hours. Following previous works, we report results at two granularity levels, i.e., *Eval* with 12 action classes and *Mid* with 19 action classes. For *Eval*, some action classes are merged into one class (e.g., "cut cucumber", "cut tomato", and "cut cheese" are all considered as "cut"). Thus, it has less number of action classes than *Mid*. We use pre-computed features by [81].

- *YouTube Instructions (YTI)* includes 150 videos belonging to 5 activities. The average video length is about 2 minutes. This dataset also has a large number of frames labeled as background. Following previous works, we use pre-computed features provided by [3].

- *Breakfast* consists of 10 activities with about 8 actions per activity. The average video length varies from few seconds to several minutes depending on the activity. Following previous works, we use pre-computed features proposed by [41] and shared by [43].

- Our *Desktop Assembly* dataset includes 76 videos of actors performing an assembly activity. The activity comprises 22 actions conducted in a fixed order. Each video is about 1.5 minutes long. We use pre-computed features from ResNet-18 [33] pre-trained on ImageNet. Please see more details in the supplementary material.

Metrics. Since no labels are provided for training, there is no direct mapping between predicted and ground truth segments. To establish this mapping, we follow [43, 65] and perform Hungarian matching. Note that the Hungarian matching is conducted at the activity level, i.e., it is computed over all frames of an activity. This is different from the Hungarian matching used in [1] which is done at the video level and generally leads to better results due to more fine-grained matching [78]. We adopt Mean Over Frames (MOF) and F1-Score as our metrics. MOF is the percentage of correct frame-wise predictions averaged over all activities. For F1-Score, to compute precision and recall, positive detections must have more than 50% overlap with ground truth segments. F1-Score is computed for each video and averaged over all videos. Please see [43] for more details.

Competing Methods. We compare against various unsupervised activity segmentation methods [3,43,52,65,75,78]. Frank-Wolfe [3] explores accompanied narrations. Mallow [65] iterates between representation learning based on discriminative learning and temporal modeling based on a generalized Mallows model. CTE [43] leverages timestamp prediction for representation learning and then K-means for clustering. VTE [78] and ASAL [52] further improve CTE [43] with visual cues (via future frame prediction) and action-level cues (via action shuffle prediction) respectively. UDE [75] uses discriminative learning for clustering and reconstruction loss for representation learning.

4.1. Ablation Study Results

We perform ablation studies on 50 Salads (i.e., *Eval* granularity) and YTI to show the effectiveness of our design choices in Sec. 3. Tabs. 1 and 2 show the ablation study results. We first begin with the standard optimal transport (OT), without any temporal prior. From Tabs. 1 and 2, OT has the worst overall performance, e.g., OT obtains 27.8 for F1-Score on 50 Salads, and 11.6 for F1-Score and 16.0% for MOF on YTI. Next, we experiment with adding temporal priors to OT, including time-stamp prediction loss in CTE [43] (yielding *OT+CTE*), temporal coherence loss in Sec. 3.1 (yielding *OT+TCL*), and temporal order-preserving prior in Sec. 3.2 (yielding *TOT*). We notice while OT+CTE, OT+TCL, and TOT all outperform OT, TOT achieves the best performance among them, e.g., TOT obtains 42.8 for F1-Score on 50 Salads, and 30.0 for F1-Score and 40.6% for MOF on YTI. The above observations are also confirmed by plotting the pseudo-label codes $\boldsymbol{Q}$ computed by different variants in Fig. 3. It can be seen that OT fails to capture any temporal structure of the activity, whereas TOT manages to capture the temporal order of the activity relatively well (i.e., initial frames should be mapped to initial prototypes and vice versa).

Finally, we consider adding more temporal priors to TOT, including time-stamp prediction loss in CTE [43] (yielding *TOT+CTE*) and temporal coherence loss in Sec. 3.1 (yielding *TOT+TCL*). We observe that TCL is often complementary to TOT, and TOT+TCL achieves the best overall performance, e.g., TOT+TCL obtains 48.2 for F1-Score on 50 Salads, and 32.9 for F1-Score and 45.3% for MOF on YTI. We notice that TOT+TCL has a lower MOF than TOT on 50 Salads, which might be because TCL optimizes for disparate representations for different actions but multiple action classes are merged into one in 50 Salads (i.e., *Eval* granularity).

4.2. Hyperparameter Setting Results

Effects of α. We study the effects of different values of α, i.e., the balancing weight between the clustering-based loss and the temporal coherence loss in Eq. 4. We measure F1-Scores on YouTube Instructions. Fig. 4(a) shows the results, where the performance peaks in the proximity of $\alpha = 1.0$.

Effects of ρ. The effects of various values of ρ, i.e., the balancing weight between the similarity term and the temporal order-preserving term in Eq. 9, are presented in Fig. 4(b). We use YouTube Instructions and measure F1-Scores. From Fig. 4(b), $\rho \in [0.07, 0.1]$ performs the best. The drop at $\rho = 0.01$ is due to numerical issues (see Fig. 6 of [73]).

Effects of η. Fig. 4(c) shows the results of varying the value of η, i.e., the number of Sinkhorn-Knopp iterations during TOT training. We measure F1-Scores on YouTube Instructions. From the results, $\eta \in [3, 5]$ performs the best. Larger values of η do not improve the performance but increase the computational cost significantly.

Effects of B. The results of increasing the value of B, i.e., the mini-batch size during TOT training, are presented in Fig. 4(d). We use 50 Salads dataset (*Eval* granularity) and measure F1-Scores. As we can see from the results, the performance improves as the mini-batch size increases.

Variants	F1-Score	MOF
OT	27.8	37.6
OT+CTE	34.3	40.4
OT+TCL	30.3	27.5
TOT	*42.8*	**47.4**
TOT+CTE	36.0	40.8
TOT+TCL	**48.2**	*44.5*

Table 1. Ablation study results on 50 Salads (i.e., *Eval* granularity). The best results are in **bold**. The second best are *underlined*.

Variants	F1-Score	MOF
OT	11.6	16.0
OT+CTE	22.0	35.2
OT+TCL	24.8	35.7
TOT	*30.0*	*40.6*
TOT+CTE	26.7	38.2
TOT+TCL	**32.9**	**45.3**

Table 2. Ablation study results on YouTube Instructions. The best results are in **bold**. The second best are *underlined*.

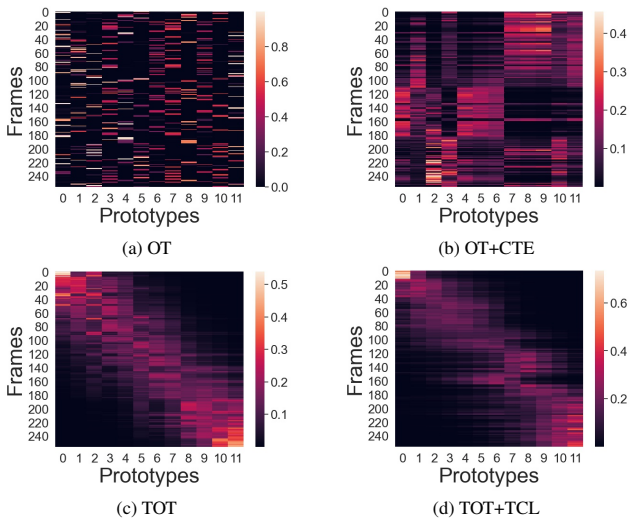

Figure 3. Pseudo-label codes $\boldsymbol{Q}$ computed by different variants for a 50 Salads video.

4.3. Results on 50 Salads Dataset

Tab. 3 presents the MOF results of different unsupervised activity segmentation methods on 50 Salads. From

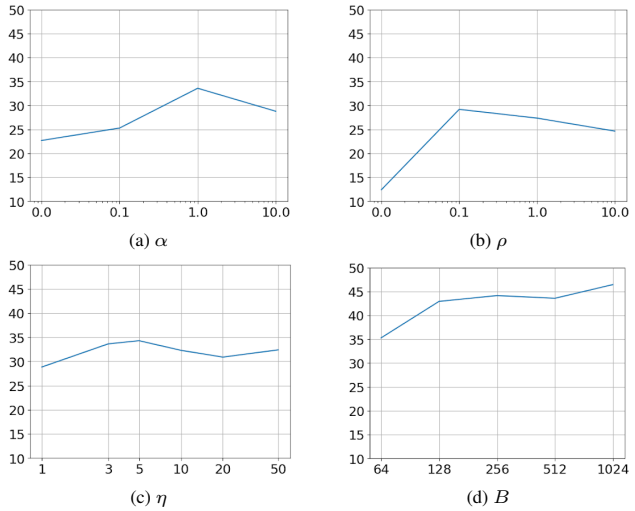

Figure 4. Hyperparameter setting results. Y axes show F1-Scores. We use YTI in (a-c) and 50 Salads (*Eval* granularity) in (d).

the results, TOT outperforms CTE [43] by 11.9% and 1.6% on the *Eval* and *Mid* granularity respectively. Similarly, TOT also outperforms VTE [78] by 16.8% and 7.6% on the *Eval* and *Mid* granularity respectively. Note that CTE, which uses a sequential representation learning and clustering framework, is our most relevant competitor. VTE further improves CTE by exploring visual information via future frame prediction, which is not utilized in TOT. The significant performance gains of TOT over both CTE and VTE show the advantages of joint representation learning and clustering. Moreover, TOT performs the best on the *Eval* granularity, outperforming the recent works of ASAL [52] and UDE [75] by 8.2% and 5.2% respectively. Finally, by combining TOT and TCL, we achieve 34.3% on the *Mid* granularity, which is very close to the best performance of 34.4% of ASAL. Also, TOT+TCL outperforms ASAL and UDE by 5.3% and 2.3% on the *Eval* granularity respectively. As mentioned previously, TOT+TCL has a lower MOF than TOT on the *Eval* granularity, which might be due to large intra-class variations in the *Eval* granularity.

4.4. Results on YouTube Instructions Dataset

Here, we compare our approach against state-of-the-art methods [3, 43, 52, 65, 75, 78] for unsupervised activity segmentation on YTI. Following all of the above works, we report the performance without considering background frames. Tab. 4 presents the results. As we can see from Tab. 4, TOT+TCL achieves the best performance on both metrics, outperforming all competing methods including the recent works of ASAL [52] and UDE [75]. In particular, TOT+TCL achieves 32.9 for F1-Score, while ASAL and UDE obtain 32.1 and 29.6 respectively. Similarly, TOT+TCL achieves 45.3% for MOF, while ASAL and UDE obtain 44.9% and 43.8% respectively. Finally, although TOT is inferior to TOT+TCL on both metrics, TOT outperforms a few competing methods. Specifically, TOT has a higher F1-Score than UDE [75], VTE [78], CTE [43], Mallow [65], and Frank-Wolfe [3], and a higher MOF than CTE [43] and Mallow [65].

Approach	Eval	Mid
CTE [43]	35.5	30.2
VTE [78]	30.6	24.2
ASAL [52]	39.2	**34.4**
UDE [75]	42.2	-
Ours (TOT)	**47.4**	31.8
Ours (TOT+TCL)	*44.5*	*34.3*

Table 3. Results on 50 Salads. The best results are in **bold**. The second best are *underlined*.

Approach	F1-Score	MOF
Frank-Wolfe [3]	24.4	-
Mallow [65]	27.0	27.8
CTE [43]	28.3	39.0
VTE [78]	29.9	-
ASAL [52]	*32.1*	*44.9*
UDE [75]	29.6	43.8
Ours (TOT)	30.0	40.6
Ours (TOT+TCL)	**32.9**	**45.3**

Table 4. Results on YouTube Instructions. The best results are in **bold**. The second best are *underlined*.

4.5. Results on Breakfast Dataset

We now discuss the performance of different methods on Breakfast. Tab. 5 shows the results. It can be seen that the recent work of ASAL [52] obtains the best performance on both metrics. ASAL [52] employs CTE [43] for initialization, and explores action-level cues for improvement, which can also be incorporated for boosting the performance of our approach. Next, TOT outperforms the sequential representation learning and clustering approach of CTE [43] by 4.6 and 5.7% on F1-Score and MOF respectively, while performing on par with VTE [78] and UDE [75], e.g., for MOF, TOT achieves 47.5% while VTE and UDE obtain 48.1% and 47.4% respectively. Also, the significant performance gains of TOT over the most relevant competitor CTE confirms the advantages of joint representation learning and clustering. Some qualitative results are shown in Fig. 5. It can be seen that our results are more closely aligned with the ground truth than those of CTE. Finally, combining TOT and TCL yields a similar F1-Score but a lower MOF than TOT, which might be due to large intra-class variations in the Breakfast dataset.

Approach	F1-Score	MOF
Mallow [65]	-	34.6
CTE [43]	26.4	41.8
VTE [78]	-	48.1
ASAL [52]	**37.9**	**52.5**
UDE [75]	*31.9*	47.4
Ours (TOT)	31.0	*47.5*
Ours (TOT+TCL)	30.3	39.0

Table 5. Results on Breakfast. The best results are in **bold**. The second best are *underlined*.

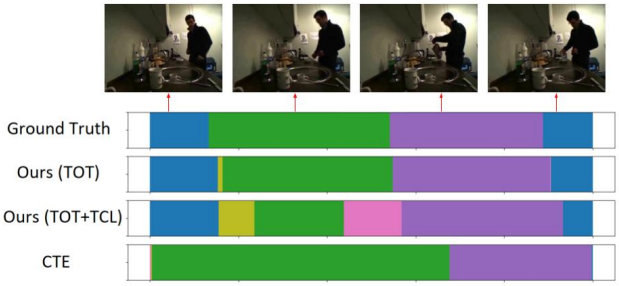

Figure 5. Segmentation results for a Breakfast video.

4.6. Results on Desktop Assembly Dataset

Prior works, e.g., CTE [43] and VTE [78], often exploit temporal information via time-stamp prediction. However, the same action might occur at various time stamps across videos in practice, e.g., different actors might perform the same action at different speeds. Our approach instead leverages temporal cues via temporal optimal transport, which preserves the temporal order of the activity. Tab. 6 shows the results of CTE and our methods (i.e., TOT and TOT+TCL) on Desktop Assembly, where the activity comprises 22 actions conducted in a fixed order. From Tab. 6, TOT+TCL performs the best on both metrics, i.e., 53.4 for F1-Score and 58.1% for MOF. Also, TOT and TOT+TCL significantly outperform CTE on both metrics, i.e., TOT and TOT+TCL obtain F1-Score gains of 6.8 and 8.5 over CTE respectively, and MOF gains of 8.7% and 10.5% over CTE respectively.

4.7. Generalization Results

So far, we have followed all previous works in unsupervised activity segmentation to use the same set of unlabelled videos for training and testing. We now explore another experiment setup to evaluate the generalization capability of our method. Specifically, we split the datasets, i.e., 50 Salads (*Eval* granularity), YouTube Instructions, Breakfast, and Desktop Assembly, into 80% for training and 20% for testing, e.g., for 50 Salads with 50 videos in total, we use 40 videos for training and 10 videos for testing. Tab. 7 presents the results of our method and CTE [43]. As expected, the results of all methods decline as compared to those reported in preceding sections. In addition, our method continues to outperform CTE in this experiment setup.

Approach	F1-Score	MOF
CTE [43]	44.9	47.6
Ours (TOT)	*51.7*	*56.3*
Ours (TOT+TCL)	**53.4**	**58.1**

Table 6. Results on Desktop Assembly. The best results are in **bold**. The second best are *underlined*.

Dataset	Approach	F1-Score	MOF
E	CTE [43]	18.4	12.2
	Ours (TOT)	*38.2*	*38.3*
	Ours (TOT+TCL)	**44.2**	**38.6**
Y	CTE [43]	16.4	17.0
	Ours (TOT)	*20.6*	*24.7*
	Ours (TOT+TCL)	**23.6**	**38.8**
B	CTE [43]	23.4	*40.6*
	Ours (TOT)	*24.5*	**45.3**
	Ours (TOT+TCL)	**25.1**	36.1
D	CTE [43]	33.8	36.0
	Ours (TOT)	*45.1*	*49.7*
	Ours (TOT+TCL)	**45.4**	**51.0**

Table 7. Generalization results. The best results are in **bold**. The second best are *underlined*. **E** denotes 50 Salads (*Eval* granularity), **Y** denotes YouTube Instructions, **B** denotes Breakfast, and **D** denotes Desktop Assembly.

5. Conclusion

We propose a novel approach for unsupervised activity segmentation, which jointly performs representation learning and online clustering. We introduce temporal optimal transport, which maintains the temporal order of the activity when computing pseudo-label cluster assignments. Our approach is online, processing one mini-batch at a time. We show comparable or superior performance against the state of the art on three public datasets, i.e., 50 Salads, YouTube Instructions, and Breakfast, and our Desktop Assembly dataset, while having substantially less memory requirements. One venue for our future work is to handle order variations and background frames such as LAVA [55]. Also, our approach can be extended to include additional self-supervised losses such as visual cues [78] and action-level cues [52]. Lastly, we can utilize deep supervision [20, 48, 49, 89] for hierarchical segmentation.

References

[1] Sathyanarayanan N Aakur and Sudeep Sarkar. A perceptual prediction framework for self supervised event segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1197–1206, 2019. 5

[2] Unaiza Ahsan, Chen Sun, and Irfan Essa. Discrimnet: Semi-supervised action recognition from videos using generative adversarial networks. *arXiv preprint arXiv:1801.07230*, 2018. 3

[3] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4575–4583, 2016. 1, 2, 5, 7

[4] YM Asano, C Rupprecht, and A Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations*, 2019. 2, 3, 4

[5] Miguel Ángel Bautista, Artsiom Sanakoyeu, Ekaterina Tikhoncheva, and Björn Ommer. Cliquecnn: Deep unsupervised exemplar learning. In *NIPS*, 2016. 2

[6] Yoshua Bengio and James S Bergstra. Slow, decorrelated features for pretraining complex cell-like networks. In *Advances in neural information processing systems*, pages 99–107, 2009. 3

[7] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019. 2

[8] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018. 2

[9] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2959–2968, 2019. 2

[10] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Neural Information Processing Systems*, 2020. 2, 3, 4

[11] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 1

[12] Chien-Yi Chang, De-An Huang, Yanan Sui, Li Fei-Fei, and Juan Carlos Niebles. 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*, pages 3546–3555, 2019. 1, 2

[13] 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*, pages 1130–1139, 2018. 1

[14] Min-Hung Chen, Baopu Li, Yingze Bao, Ghassan AlRegib, and Zsolt Kira. Action segmentation with joint self-supervised temporal domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9454–9463, 2020. 1

[15] Jinwoo Choi, Gaurav Sharma, Samuel Schulter, and Jia-Bin Huang. Shuffle and attend: Video domain adaptation. In *European Conference on Computer Vision*, pages 678–695. Springer, 2020. 3

[16] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26:2292–2300, 2013. 4, 5

[17] Ali Diba, Vivek Sharma, Luc Van Gool, and Rainer Stiefelhagen. Dynamonet: Dynamic action and motion network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6192–6201, 2019. 3

[18] 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. 1, 2

[19] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1801–1810, 2019. 3

[20] Mohammed E Fathy, Quoc-Huy Tran, M Zeeshan Zia, Paul Vernaza, and Manmohan Chandraker. Hierarchical metric learning and matching for 2d and 3d geometric correspondences. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 803–819, 2018. 8

[21] Mohsen Fayyaz and Jurgen Gall. Sct: Set constrained temporal transformer for set supervised action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 501–510, 2020. 2

[22] Zeyu Feng, Chang Xu, and Dacheng Tao. Self-supervised representation learning by rotation feature decoupling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10364–10374, 2019. 2

[23] 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. 3

[24] Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. Predicting the future: A jointly learnt model for action anticipation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5562–5571, 2019. 3

[25] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Learning representations by predicting bags of visual words. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6928–6938, 2020. 2

[26] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018. 2

[27] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1705–1714, 2019. 1

[28] Ross Goroshin, Joan Bruna, Jonathan Tompson, David Eigen, and Yann LeCun. Unsupervised learning of spatiotemporally coherent metrics. In *Proceedings of the IEEE international conference on computer vision*, pages 4086–4093, 2015. 3

[29] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 3

[30] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019. 3

[31] Sanjay Haresh, Sateesh Kumar, Huseyin Coskun, Shahram Najam Syed, Andrey Konin, Muhammad Zeeshan Zia, and Quoc-Huy Tran. Learning by aligning videos in time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 3

[32] Sanjay Haresh, Sateesh Kumar, M Zeeshan Zia, and Quoc-Huy Tran. Towards anomaly detection in dashcam videos. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1407–1414. IEEE. 1

[33] 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. 5

[34] Geoffrey E Hinton and Richard S Zemel. Autoencoders, minimum description length and helmholtz free energy. In *Advances in neural information processing systems*, pages 3–10, 1994. 2

[35] 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. 1, 2

[36] Jiabo Huang, Qi Dong, Shaogang Gong, and Xiatian Zhu. Unsupervised deep learning by neighbourhood discovery. In *International Conference on Machine Learning*, pages 2849–2858. PMLR, 2019. 2

[37] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8545–8552, 2019. 3

[38] Dahun Kim, Donghyeon Cho, Donggeun Yoo, and In So Kweon. Learning image representations by completing damaged jigsaw puzzles. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 793–802. IEEE, 2018. 2

[39] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[40] 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. 1, 5

[41] Hilde Kuehne, Juergen Gall, and Thomas Serre. An end-to-end generative framework for video segmentation and recognition. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–8. IEEE, 2016. 1, 5

[42] Hilde Kuehne, Alexander Richard, and Juergen Gall. Weakly supervised learning of actions from transcripts. *Computer Vision and Image Understanding*, 163:78–89, 2017. 1, 2

[43] Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. Unsupervised learning of action classes with continuous temporal embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12066–12074, 2019. 1, 2, 4, 5, 6, 7, 8

[44] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *European conference on computer vision*, pages 577–593. Springer, 2016. 2

[45] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6874–6883, 2017. 2

[46] 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. 1

[47] 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. 3

[48] C. Li, M. Z. Zia, Q. Tran, X. Yu, G. D. Hager, and M. Chandraker. Deep supervision with intermediate concepts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018. 8

[49] Chi Li, M Zeeshan Zia, Quoc-Huy Tran, Xiang Yu, Gregory D Hager, and Manmohan Chandraker. Deep supervision with shape concepts for occlusion-aware 3d object parsing. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 388–397. IEEE, 2017. 8

[50] Jun Li, Peng Lei, and Sinisa Todorovic. Weakly supervised energy-based learning for action segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6243–6251, 2019. 1, 2

[51] Jun Li and Sinisa Todorovic. Set-constrained viterbi for set-supervised action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10820–10829, 2020. 2

[52] Jun Li and Sinisa Todorovic. Action shuffle alternating learning for unsupervised action segmentation. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 1, 2, 4, 5, 7, 8

[53] Shi-Jie Li, Yazan AbuFarha, Yun Liu, Ming-Ming Cheng, and Juergen Gall. Ms-tcn++: Multi-stage temporal convolutional network for action segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1

[54] Zhe Li, Yazan Abu Farha, and Jurgen Gall. Temporal action segmentation from timestamp supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8365–8374, 2021. 2

[55] Weizhe Liu, Bugra Tekin, Huseyin Coskun, Vibhav Vineet, Pascal Fua, and Marc Pollefeys. Learning to align sequential actions in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 8

[56] Xialei Liu, Joost Van De Weijer, and Andrew D Bagdanov. Leveraging unlabeled data for crowd counting by learning to rank. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7661–7669, 2018. 2

[57] Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nicholas Johnston, Andrew Rabinovich, and Kevin Murphy. What's cookin'? interpreting cooking videos using text, speech and vision. In *HLT-NAACL*, 2015. 1, 2

[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. 3

[59] Hossein Mobahi, Ronan Collobert, and Jason Weston. Deep learning from temporal coherence in video. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 737–744, 2009. 3

[60] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5898–5906, 2017. 2

[61] Alexander Richard and Juergen Gall. Temporal action detection using a statistical language model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3131–3140, 2016. 1, 2

[62] Alexander Richard, Hilde Kuehne, and Juergen Gall. Weakly supervised action learning with rnn based fine-to-coarse modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 754–763, 2017. 1, 2

[63] Alexander Richard, Hilde Kuehne, and Juergen Gall. Action sets: Weakly supervised action segmentation without ordering constraints. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5987–5996, 2018. 2

[64] Alexander Richard, Hilde Kuehne, Ahsan Iqbal, and Juergen Gall. Neuralnetwork-viterbi: A framework for weakly supervised video learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7386–7395, 2018. 1, 2

[65] Fadime Sener and Angela Yao. Unsupervised learning and segmentation of complex activities from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8368–8376, 2018. 1, 2, 4, 5, 7, 8

[66] Ozan Sener, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Unsupervised semantic parsing of video collections. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4480–4488, 2015. 1, 2

[67] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1134–1141. IEEE, 2018. 3

[68] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-deconvolutional networks for precise temporal action localization in untrimmed videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5734–5743, 2017. 1

[69] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1049–1058, 2016. 1

[70] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 1857–1865, 2016. 3

[71] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852, 2015. 3

[72] Sebastian Stein and Stephen J McKenna. 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*, pages 729–738, 2013. 5

[73] Bing Su and Gang Hua. Order-preserving wasserstein distance for sequence matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1049–1057, 2017. 4, 6

[74] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6479–6488, 2018. 1

[75] Sirnam Swetha, Hilde Kuehne, Yogesh S Rawat, and Mubarak Shah. Unsupervised discriminative embedding for sub-action learning in complex activities. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2588–2592. IEEE, 2021. 2, 5, 7, 8

[76] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 1

[77] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 1

[78] Rosaura G VidalMata, Walter J Scheirer, Anna Kukleva, David Cox, and Hilde Kuehne. Joint visual-temporal em-

bedding for unsupervised learning of actions in untrimmed sequences. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1238–1247, 2021. 1, 2, 4, 5, 7, 8

[79] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008. 2

[80] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances in neural information processing systems*, pages 613–621, 2016. 3

[81] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013. 5

[82] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 1

[83] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 3

[84] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR, 2016. 2

[85] 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 Conference on Computer Vision and Pattern Recognition*, pages 10334–10343, 2019. 3

[86] Xueting Yan, Ishan Misra, Abhinav Gupta, Deepti Ghadiyaram, and Dhruv Mahajan. Clusterfit: Improving generalization of visual representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6509–6518, 2020. 2

[87] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5147–5156, 2016. 2

[88] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7094–7103, 2019. 1

[89] Bingbing Zhuang, Quoc-Huy Tran, Gim Hee Lee, Loong Fah Cheong, and Manmohan Chandraker. Degeneracy in self-calibration revisited and a deep learning solution for uncalibrated slam. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3766–3773. IEEE, 2019. 8

[90] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6002–6012, 2019. 2

[91] Will Zou, Shenghuo Zhu, Kai Yu, and Andrew Y Ng. Deep learning of invariant features via simulated fixations in video. In *Advances in neural information processing systems*, pages 3203–3211, 2012. 3

[92] Will Y Zou, Andrew Y Ng, and Kai Yu. Unsupervised learning of visual invariance with temporal coherence. In *NIPS 2011 workshop on deep learning and unsupervised feature learning*, volume 3, 2011. 3