Revisiting Foreground and Background Separation in Weakly-supervised Temporal Action Localization: A Clustering-based Approach

Qinying Liu[1] Zilei Wang[*1] Shenghai Rong [1] Junjie Li [1] Yixin Zhang[1,2]
[1] University of Science and Technology of China
[2] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
{lydyc, rongsh, hnljj@}@mail.ustc.edu.cn, {zhyx12, zlwang}@ustc.edu.cn

Abstract

Weakly-supervised temporal action localization aims to localize action instances in videos with only video-level action labels. Existing methods mainly embrace a localization-by-classification pipeline that optimizes the snippet-level prediction with a video classification loss. However, this formulation suffers from the discrepancy between classification and detection, resulting in inaccurate separation of foreground and background (F&B) snippets. To alleviate this problem, we propose to explore the underlying structure among the snippets by resorting to unsupervised snippet clustering, rather than heavily relying on the video classification loss. Specifically, we propose a novel clustering-based F&B separation algorithm. It comprises two core components: a snippet clustering component that groups the snippets into multiple latent clusters and a cluster classification component that further classifies the cluster as foreground or background. As there are no ground-truth labels to train these two components, we introduce a unified self-labeling mechanism based on optimal transport to produce high-quality pseudo-labels that match several plausible prior distributions. This ensures that the cluster assignments of the snippets can be accurately associated with their F&B labels, thereby boosting the F&B separation. We evaluate our method on three benchmarks: THUMOS14, ActivityNet v1.2 and v1.3. Our method achieves promising performance on all three benchmarks while being significantly more lightweight than previous methods. Code is available at https://github.com/Qinying-Liu/CASE

1. Introduction

Temporal action localization (TAL) [43] is a task to localize the temporal boundaries of action instances and recognize their categories in videos. In recent years, numerous works put effort into the fully supervised manner and gain great achievements. Albeit successful, these methods require extensive manual frame-level annotations, which are expensive and time-consuming. Without the requirement of frame-level annotations, weakly-supervised TAL (WTAL) has received increasing attention, as it allows us to detect the action instances with only video-level action labels.

There has been a wide spectrum of WTAL methods developed in the literature [48, 58, 37, 29]. With only video-level labels, mainstream methods employ a localization-by-classification pipeline, which formulates WTAL as a video action classification problem to learn a temporal class activation sequence (T-CAS). For this pipeline, foreground (*i.e.*, action) and background separation remains an open question since video-level labels do not provide any cue for background class. There are two types of existing approaches to solve it. The first type [48, 58] is based on the multiple instance learning (MIL), which uses the T-CAS to select the most confident snippets for each action class. The second type [37, 29] introduces an attention mechanism to learn class-agnostic foreground weights that indicate the probabilities of the snippets belonging to foreground. Despite recent progress, these methods typically rely on the video classification loss to guide the learning of the T-CAS or the attention weights. There is an inherent downside: the loss is easily minimized by the salient snippets [33] and fails to explore the distribution of the whole snippets, resulting in erroneous T-CAS or attention weights. This issue is rooted in the supervision gap between the classification and detection tasks. Recent studies [39, 31] are devoted to producing snippet-level pseudo-labels to bridge the gap. However, the pseudo-labels are still derived from the unreliable T-CAS or attention weights.

Deep clustering [6], which automatically partitions the samples into different groups, has been proven to be capable of revealing the intrinsic distribution of the samples in many label-scarce tasks [1, 4, 10, 27]. A natural issue arises: is it possible to adopt the clustering to capture the distribution of snippets? Since clustering can be conducted in a

*Corresponding author

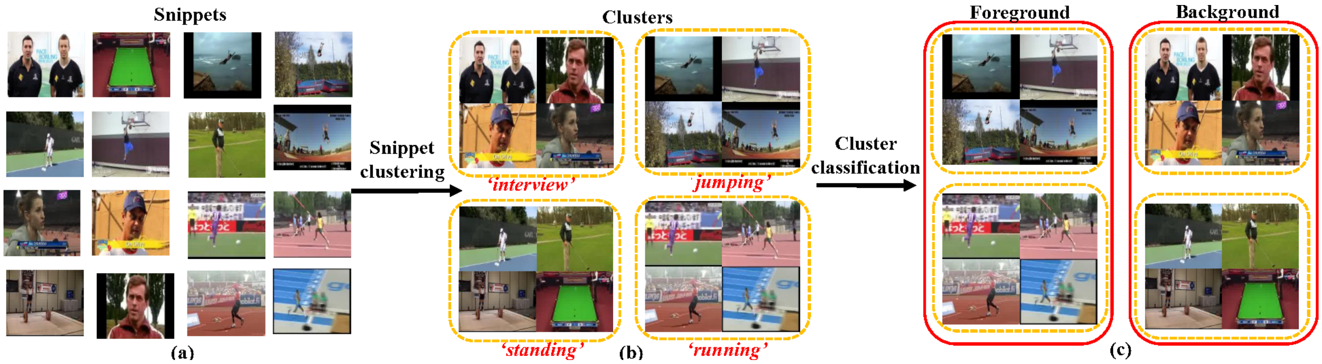

Figure 1: Conceptual illustration of our clustering-based F&B separation algorithm. In snippet clustering, we partition the snippets (or frames) into multiple clusters with explicit characteristics. In cluster classification, we classify the clusters as foreground or background. The above results are attained according to the predictions of our method.

self-supervised manner, it is immune to the video classification loss. This suggests a great potential of clustering for F&B separation in WTAL. A brute-force solution would be to group the snippets into two clusters, one for foreground and one for background. Whereas, we empirically find that it underperforms in practice (*cf.* Sec. 5.3). We argue that the reason is that snippets, regardless of foreground or background, can differ dramatically in appearance (*cf.* Fig. 1 (a)). As a result, it may be difficult for a self-supervised model to group them accurately. Fortunately, in real-world videos, there are common characteristics (*e.g.*, "interview", "running") shared by a group of snippets (*cf.* Fig. 1 (b)). Compared to learning two clusters for F&B in the complex video content, it may be easier to explore the snippet clusters with clear and distinctive characteristics. This necessitates a clustering algorithm with multiple clusters. Furthermore, it can be observed that the characteristics of clusters are sometimes indicative cues for F&B separation. For example, we can confidently classify the "running" cluster to foreground and the "interview" cluster to background according to the cluster-level characteristics. Consequently, it is promising to further leverage the cluster-level representations to assist F&B separation.

In light of the above discussion, we propose a novel **C**lustering-**A**ssisted F&B **SE**paration (CASE) network. We begin by constructing a standard WTAL baseline that provides a primary estimation of F&B snippets. We then introduce a clustering-based F&B separation algorithm (*cf.* Fig. 1) to refine the F&B separation. This algorithm is comprised of two main components: snippet clustering for dividing the snippets into multiple clusters, and cluster classification for classifying the clusters as foreground or background. Considering that no ground-truth labels are available to train the components, we propose a unified self-labeling mechanism to generate high-quality pseudo-labels for them. Specifically, we formulate the label assignment in both components as a unified optimal transport problem, which allows us to flexibly impose several customized constraints on the distribution of pseudo-labels. After training these two components, we can transform the cluster assignments of the snippets to their F&B assignments, which can be used to refine the F&B separation of the baseline.

It is demonstrated that our method yields favorable performance while being much more lightweight compared to prior approaches. In summary, our contributions are threefold. 1) We propose a clustering-based F&B separation algorithm for WTAL, which casts the problem of F&B separation as a combination of snippet clustering and cluster classification. 2) We propose a unified self-labeling mechanism based on optimal transport to guide snippet clustering and cluster classification. 3) We conduct extensive experiments that demonstrate the effectiveness and efficiency of our method compared to existing approaches.

2. Related Work

Deep clustering. Current deep clustering approaches [55, 54, 9] could be roughly divided into two categories. The first one iteratively computes the clustering assignment from the up-to-date model and supervises the network training processes by the estimated information [51, 52, 6, 3, 5, 50]. DeepCluster [51] is a typical method that iteratively groups the features and uses the subsequent assignments to update the deep network. The second one simultaneously learns feature representation and clustering assignment [12, 22, 15], which has gained popularity in recent years. Asano *et al.* [1] propose to enforce a balanced label assignment constraint to avoid degenerate solution. Caron *et al.* [4] use the algorithm in [1] to introduce a swapped mechanism that employs two random transformations of the same images to guide each other. In this work, we extend [1] from image classification to WTAL, and incorporate it with task-specific designs, *e.g.*, imposing multiple sensible constraints on the distribution of pseudo-labels.

It is worth noting that in the above methods, the number of clusters is typically set to the number of ground-truth (GT) classes so that clusters and GT classes can be mapped one-by-one during testing [46, 3]. There are attempts that utilize an extra over-clustering technique to learn a larger number of clusters than GT, which is believed to be conducive to representation learning [22, 10]. However, these methods commonly treat the technique as an auxiliary tool independent of their main task. In contrast, we are committed to building an explicit correspondence between learned clusters and F&B classes, thus unleashing the full potential of clustering in the WTAL task.

Weakly-supervised temporal action localization. Existing WTAL approaches can be categorized into four broad groups. The first group aims to improve feature discrimination ability. Various techniques, *e.g.*, deep metric learning [33, 35] and contrastive learning [58, 26], have been explored. The second group seeks to discover complete action regions. [33, 44, 60] hide some snippets to press the models in exploring more action regions, while [29, 21] use a multi-branch framework to discover complementary snippets. The third group is concerned with learning attention weights. [57, 36] design losses to regularize the values of the attention weights. [39, 31] generate pseudo-labels for them. However, the pseudo-labels are derived from the primary predictions of snippets, which are still optimized using the video classification loss. The last group is the most closely related to ours, which introduces auxiliary classes in addition to the action classes. [42] introduces a video-level context class. [30, 47, 29] mine the action units or sub-actions shared across action categories. Class-specific sub-action is explored in [18, 17]. Recently, [28] learns a set of visual concepts for fine-grained action localization. Our method is superior to these methods in three noticeable aspects. 1) These methods rely on video-level supervision to discover the auxiliary classes. Conversely, we develop the clusters in a self-supervised manner that is orthogonal to video-level supervision. 2) These methods devise multiple loss terms to regularize the auxiliary classes. In contrast, we introduce regularization into optimal transport, which can be resolved in a principled way. 3) Our method significantly outperforms these methods.

3. Preliminaries and Baseline Setup

In each training iteration, we first sample a mini-batch of B videos. For each video, we have access only to its video-level label $\boldsymbol{Y} \in \mathbb{R}^G$, where G is the number of ground-truth action classes. By convention, we first sample a sequence of T snippets from each video, and then extract snippet features with a pre-trained feature extractor for both RGB and optical-flow streams. For simplicity, only one stream is presented hereafter. As a result, we obtain a sequence of snippet features $\boldsymbol{F} \in \mathbb{R}^{T \times D}$. Here, D is the channel dimension.

For the baseline, following convention [24], we use a two-branch framework consisting of a video classification branch and an attention branch, as shown in Fig. 2(a). In the former branch, we first feed the input features $\boldsymbol{F}$ to an embedding encoder followed by an action classifier to get the temporal class activation sequence (T-CAS) $\boldsymbol{P^V} \in \mathbb{R}^{T \times G}$. In the latter branch, $\boldsymbol{F}$ is first passed through another embedding encoder to obtain the snippet embeddings, and the embeddings are then sent to an attention layer to extract one-dimension **A**ttention weights $\boldsymbol{P^A} \in \mathbb{R}^T$, which represent the foreground probabilities of the snippets.

We apply the popular multiple instance learning (MIL) to train the video classification branch. Briefly (see Supplementary for details), we first calibrate T-CAS with the attention weights to restrain background snippets. Then we select the top-k snippets for each class based on the activations to construct video-level scores $\bar{\boldsymbol{P}}^{\boldsymbol{V}} \in \mathbb{R}^G$. Finally, we optimize a video classification loss with the known video labels $\boldsymbol{Y}$:

$$\mathcal{L}_V = \mathcal{L}_{CE}(\bar{\boldsymbol{P}}^{\boldsymbol{V}}, \boldsymbol{Y}). \tag{1}$$

To train the attention branch, we adopt the pseudo-label-based scheme proposed by [32] due to its conciseness and effectiveness. Specifically, we define foreground pseudo-labels $\boldsymbol{Q^A} \in \mathbb{R}^T$ as follows: snippets appearing in the top-k activations for the ground-truth video-level classes are positive, and the other snippets are negative. To improve the robustness of the model against label noise, we use the generalized binary cross-entropy loss [59, 32]:

$$\begin{aligned}\mathcal{L}_A =& \frac{1}{N_{\text{pos}}}\sum_{t=1}^{T}\boldsymbol{Q}_t^{\boldsymbol{A}}\frac{1-(\boldsymbol{P}_t^{\boldsymbol{A}})^{\gamma}}{\gamma}+\\ & \frac{1}{N_{\text{neg}}}\sum_{t=1}^{T}(1-\boldsymbol{Q}_t^{\boldsymbol{A}})\frac{1-(1-\boldsymbol{P}_t^{\boldsymbol{A}})^{\gamma}}{\gamma},\end{aligned} \tag{2}$$

where $\gamma \in (0, 1)$ controls the noise tolerance, and N_{pos} and N_{neg} represent the number of positive and negative snippets.

4. Our Method

In this section, we present our clustering-based F&B algorithm, as depicted in Fig. 2(b), which is built upon the above baseline. We begin by providing an overview of our algorithm, which consists of two main components: snippet clustering component (SCC) and cluster classification component (CCC). Next, we introduce a unified self-labeling mechanism that we employ to provide pseudo-labels for both SCC and CCC. Lastly, we explain how SCC and CCC are used in the training and testing procedures.

4.1. Overview

Snippet Clustering Component. SCC is proposed to group snippets into latent clusters. To enable joint learning of the attention layer and snippet clustering, we append

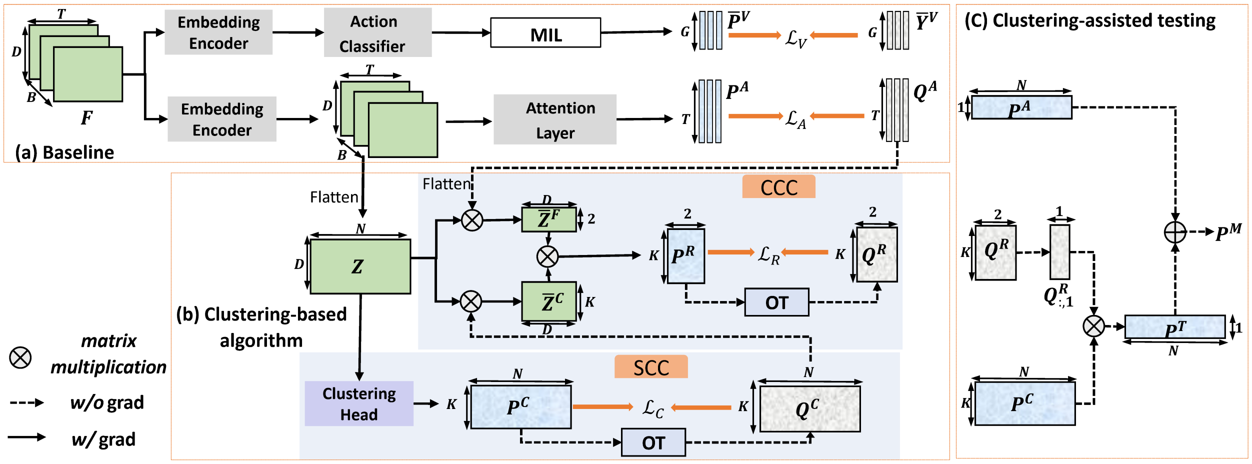

Figure 2: Framework of our CASE. (a) depicts the baseline, which includes a video classification branch and an attention branch. (b) illustrates our proposed clustering-based F&B separation algorithm, which comprises a snippet clustering component (SCC) and a cluster classification component (CCC). Both are trained using a unified self-labeling mechanism based on optimal transport (OT). (c) shows the clustering-assisted testing technique that utilizes the results of SCC and CCC to assist F&B separation during inference.

SCC over the embeddings in the attention branch, as shown in Fig. 2(b). For notation simplicity, we henceforth term the total number of snippets as $N = BT$ for a batch of B videos, and call the snippet embeddings $\boldsymbol{E} \in \mathbb{R}^{N \times D}$. We feed $\boldsymbol{E}$ into a clustering head composed of a linear classifier with K classes (clusters), producing **S**nippet-level cluster assignment probabilities dubbed $\boldsymbol{P^S} \in \mathbb{R}^{N \times K}$. Inspired by self-supervised learning [38, 41], we set K as a predefined parameter, which we find to be robust in practice. To train the clustering head, we first generate (soft) pseudo-labels $\boldsymbol{Q^S} \in \mathbb{R}^{N \times K}$ for $\boldsymbol{P^S}$, which will be described in Sec. 4.2. Then, we minimize the following loss:

$$\mathcal{L}_S = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_{CE}(\boldsymbol{Q}^{\boldsymbol{S}}_n, \boldsymbol{P}^{\boldsymbol{S}}_n). \tag{3}$$

Cluster Classification Component. CCC enforces each cluster to be classified into foreground or background by mapping the cluster prototypes to the F&B prototypes, as shown in Fig. 2(b). Specifically, based on the pseudo cluster assignments of snippets $\boldsymbol{Q^S} \in \mathbb{R}^{N \times K}$ obtained from SCC, we can compute the k-th cluster prototype over the snippet embeddings $\boldsymbol{E} \in \mathbb{R}^{N \times D}$:

$$\bar{\boldsymbol{E}}^{\boldsymbol{S}}_k = \frac{\sum_{n=1}^{N} \boldsymbol{Q}^{\boldsymbol{S}}_{n,k} \cdot \boldsymbol{E}_n}{\sum_{n=1}^{N} \boldsymbol{Q}^{\boldsymbol{S}}_{n,K}}. \tag{4}$$

where $\bar{\boldsymbol{E}}^{\boldsymbol{S}}_k \in \mathbb{R}^{D}$. In a similar vein, using the foreground pseudo-labels $\boldsymbol{Q^A} \in \mathbb{R}^{N}$ and background labels $1 - \boldsymbol{Q^A}$, we can calculate the F&B prototypes $\bar{\boldsymbol{E}}^{\boldsymbol{A}} \in \mathbb{R}^{2 \times D}$. $\bar{\boldsymbol{E}}^{\boldsymbol{A}}_1$ and $\bar{\boldsymbol{E}}^{\boldsymbol{A}}_2$ correspond to foreground and background, respectively. Then, we compute **C**luster-level classification probabilities dubbed $\boldsymbol{P^C} \in \mathbb{R}^{K \times 2}$ by measuring the similarities between cluster prototypes and F&B prototypes:

$$\boldsymbol{P}^{\boldsymbol{C}}_{k,i} = \underset{i}{Softmax}\left(\rho \cdot \cos(\bar{\boldsymbol{E}}^{\boldsymbol{S}}_k, \bar{\boldsymbol{E}}^{\boldsymbol{A}}_i)\right), \tag{5}$$

where $\cos(\cdot)$ indicates the cosine similarity function and ρ is the temperature. $\boldsymbol{P}^{\boldsymbol{C}}_{k,i}$ represents the probability that k-th cluster belongs to foreground ($i = 1$) or background ($i = 2$). To optimize the component, we generate (soft) labels $\boldsymbol{Q^C}$ for $\boldsymbol{P^C}$, which will be described in Sec. 4.2. Accordingly, we will get a loss term:

$$\mathcal{L}_C = \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}_{CE}(\boldsymbol{Q}^{\boldsymbol{C}}_k, \boldsymbol{P}^{\boldsymbol{C}}_k). \tag{6}$$

Notably, although $\boldsymbol{Q^C}$ is computed from each mini-batch of data, we observe that it will quickly converge to a stable status near the one-hot form during training. This suggests that a global and clear correspondence between clusters and F&B is established.

4.2. Self-Labeling via Unified Optimal-Transport

This section explains the self-labeling mechanism that generates the labels $\boldsymbol{Q^S}$ for $\boldsymbol{P^S}$ in SCC and labels $\boldsymbol{Q^C}$ for $\boldsymbol{P^C}$ in CCC. First, we describe a basic labeling formulation shared in both SCC and CCC. This formulation converts the label assignment to an optimal transport problem while imposing constraints on the distribution of the labels. Then, we discuss the unique adaptations required for SCC and CCC individually. For SCC, we introduce a prior distribution for $\boldsymbol{Q^S}$ to avoid the uncertain label assignment issue observed in SCC. As for CCC, we leverage snippet-level F&B labels to estimate the prior marginal distribution of $\boldsymbol{Q^C}$. These adaptations can be seamlessly integrated into the optimal-transport formulation, resulting in a unified solution that is easy to implement, as demonstrated in Alg. 1.

Basic formulation. Regarding the generation of pseudo-labels $\boldsymbol{Q}$, a straightforward solution is to search for a reasonable $\boldsymbol{Q}$ that is close to the current model predictions $\boldsymbol{P}$, *e.g.*, by applying $\arg\max$ to $\boldsymbol{P}$. However, in our unsupervised setting, this way may lead to trivial solutions, *e.g.*, all samples are assigned to only a class (*cf.* Sec. 5.3). Instead, when searching for $\boldsymbol{Q}$, we propose to impose a constraint on the proportion of elements assigned to each class. Formally, we formulate this as an optimization problem:

$$\min_{\boldsymbol{Q}\in\Omega} \mathrm{E}(\boldsymbol{P},\boldsymbol{Q}), \tag{7}$$

where $\mathrm{E}(\boldsymbol{P},\boldsymbol{Q}) = -\sum_n^N \sum_k^K \boldsymbol{Q}_{n,k} \log \boldsymbol{P}_{n,k}$ measures the distance between $\boldsymbol{Q}$ and $\boldsymbol{P}$, N is the number of samples and K is the number of classes. The constraint Ω is defined as:

$$\Omega = \{\boldsymbol{Q} \in \mathbb{R}_+^{N\times K} | \boldsymbol{Q}\mathbf{1}^K = \boldsymbol{\alpha}, \boldsymbol{Q}^\top \mathbf{1}^N = \boldsymbol{\beta}\}, \tag{8}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the marginal distributions of $\boldsymbol{Q}$ onto its rows and columns, respectively. We set $\boldsymbol{\alpha} = \mathbf{1}^N$ to ensure that $\boldsymbol{Q}$ is a probability matrix. $\boldsymbol{\beta} \in \mathbb{R}^K$ represents the proportion of elements belonging to each of the K classes. When there is no prior knowledge, equipartition [1, 4] can serve as a general inductive bias and be utilized to set $\boldsymbol{\beta}$:

$$\boldsymbol{\beta} = \frac{N}{K}\mathbf{1}^K. \tag{9}$$

This ensures that, on average, each class is assigned the same number of samples, thereby averting trivial solutions.

It is noteworthy that Eq. (7) is an optimal transport problem, which is computationally expensive to solve. Following [8], an entropy term is introduced to it:

$$\min_{\boldsymbol{Q}\in\Omega} \mathrm{E}(\boldsymbol{P},\boldsymbol{Q}) - \frac{1}{\epsilon}\mathrm{H}(\boldsymbol{Q}), \tag{10}$$

where $\epsilon > 0$ and $\mathrm{H}(\boldsymbol{Q})$ is the entropy of $\boldsymbol{Q}$. The advantage of the term is that Eq. (10) can be efficiently solved by Sinkhorn-Knopp algorithm [8].

Prior distribution for SCC. In Eq. (10), an entropy term is subtracted to make it tractable with affordable complexity. Nevertheless, maximizing the entropy can also lead to uncertain label assignments, where the samples are assigned to different classes with equal probability. In practice, the issue is pronounced in SCC but not in CCC. This may be because the former involves much more instances and classes, rendering the algorithm harder to converge (*cf.* Fig. 4).

To remedy the defect in SCC, we repurpose an early sequence-matching method [45] to introduce a prior distribution for the pseudo-labels $\boldsymbol{Q^S}$ in SCC, denoted as $\hat{\boldsymbol{Q}}^{\boldsymbol{S}} \in \mathbb{R}^{N\times K}$, which represents the probability of assigning N snippets to K clusters. A sensible prior distribution should encourage foreground snippets to have relatively high probabilities of belonging to foreground clusters, and the same is true for background. To implement this, we first sort the snippets according to their foreground probabilities $\boldsymbol{P^A}$ in ascending order, and denote the resulting ranks of the N snippets as $rank \in \mathbb{R}^N$. We then construct the prior distribution $\hat{\boldsymbol{Q}}^{\boldsymbol{S}}$, such that the snippets with high ranks (*i.e*, $rank$) are more likely to be assigned to the clusters with high foreground probabilities (*i.e*, $\boldsymbol{Q}^{\boldsymbol{C}}_{:,1}$) and vice versa. Formally, $\hat{\boldsymbol{Q}}^{\boldsymbol{S}}$ is defined as a Gaussian distribution:

$$\hat{\boldsymbol{Q}}^{\boldsymbol{S}}_{n,k} = \frac{1}{\sigma\sqrt{2\pi}} \exp\big(-\frac{\left|\frac{rank_n}{N} - \boldsymbol{Q}^{\boldsymbol{C}}_{k,1}\right|^2}{2\sigma^2}\big), \tag{11}$$

where $rank_n$ is the order of n-th snippet, and $\boldsymbol{Q}^{\boldsymbol{C}}_{k,1}$ is the foreground probability of k-th cluster. Finally, we replace Eq. (10) with the following objective function:

$$\min_{\boldsymbol{Q^S}\in\Omega^S} \mathrm{E}(\boldsymbol{P^S},\boldsymbol{Q^S}) + \frac{1}{\epsilon}\mathrm{KL}(\boldsymbol{Q^S},\hat{\boldsymbol{Q}}^{\boldsymbol{S}}), \tag{12}$$

where $\mathrm{KL}(\cdot)$ is the Kullback-Leibler divergence. By minimizing the KL term, we encourage the labels $\boldsymbol{Q^S}$ to be close to the prior distribution $\hat{\boldsymbol{Q}}^{\boldsymbol{S}}$, which helps to avoid uncertain label assignments caused by the original entropy term. Importantly, Eq. (12) can still be efficiently addressed by Sinkhorn-Knopp algorithm. For detailed derivation, we refer to Supplementary.

Prior marginal distribution for CCC. Although equipartition (*i.e.*, Eq. (9)) is a common prior in traditional clustering, it is not suitable for enforcing equipartition on the marginal distribution of the cluster-level F&B labels $\boldsymbol{Q^C}$, namely $\boldsymbol{\beta^C} \in \mathbb{R}^2$. This is due to the fact that $\boldsymbol{\beta^C}$ represents the proportions of clusters assigned to F&B, which are not always balanced. However, since SCC enforces equipartition on the snippet level, each cluster contains a similar number of snippets. Consequently, the proportions of F&B clusters are expected to be close to the proportions of F&B snippets. To this end, instead of using Eq. (9), we estimate $\boldsymbol{\beta^C}$ empirically based on the distribution of the snippet-level foreground labels $\boldsymbol{Q^A}$:

$$\boldsymbol{\beta^C} = [\frac{1}{N}\sum_{n=1}^N \boldsymbol{Q}^{\boldsymbol{A}}_n, \frac{1}{N}\sum_{n=1}^N (1 - \boldsymbol{Q}^{\boldsymbol{A}}_n)]. \tag{13}$$

Algorithm 1 Pseudo-code of labeling procedures of SCC and CCC in Pytorch-like style.

```
# L_S: logit output of clustering head (NxK)
# Q_hat_S: prior distribution in SCC (NxK)
# L_C: logit output of cluster classification (Kx2)
# Beta_C: prior marginal distribution in CCC (2)

# generating labels Q_S for SCC
Q_S = SK(L_S, Q_hat=Q_hat_S) # (NxK)
# generating labels Q_C for CCC
Q_C = SK(L_C, Beta=Beta_C) # (Kx2)

# resolve the optimal-transport problem by iterative
      Sinkhorn-Knopp (SK) algorithm
def SK(L, Q_hat=None, Beta=None, n_iter=3):
   N, K = L.size()
   # uniform distribution is the default when prior
        distribution is not given
   Q_hat = 1/K if Q_hat is None else Q_hat
   Beta = 1/K if Beta is None else Beta
   Q = exp(L / eps)
   Q = (Q * Q_hat).T
   Q /= Q.sum()
   for _ in range(n_iter):
      Q /= Q.sum(dim=1)
      Q = Q * Beta
      Q /= Q.sum(dim=0)
      Q /= N
   return Q.t() * N
```

4.3. Training and Testing

Joint training. We train all components together in an end-to-end fashion. The overall objective is written as:

$$\mathcal{L} = (\mathcal{L}_V + \mathcal{L}_A) + \lambda_S \mathcal{L}_S + \lambda_C \mathcal{L}_C, \tag{14}$$

where λ_S and λ_C represent the loss weights. As the baseline and our proposed algorithm share the same embedding encoder in the attention branch, the joint training also facilitates the training of the baseline model.

Clustering-assisted testing. In the inference period, using the cluster-level foreground probabilities $\boldsymbol{Q}^{\boldsymbol{C}}_{:,1}$, we can transform the snippet-level cluster assignments $\boldsymbol{P}^{\boldsymbol{S}}$ to snippet-level foreground probabilities dubbed $\boldsymbol{P}^{\boldsymbol{T}}$ based on *law of total probability*: $\boldsymbol{P}^{\boldsymbol{T}} = \boldsymbol{P}^{\boldsymbol{S}}\boldsymbol{Q}^{\boldsymbol{C}}_{:,1}$, as depicted in Fig. 2 (c). Considering that $\boldsymbol{Q}^{\boldsymbol{C}}$ is stable during training, we simply use the $\boldsymbol{Q}^{\boldsymbol{C}}$ from the last training iteration for inference. Moreover, as verified in Table 6, the transformed foreground probability $\boldsymbol{P}^{\boldsymbol{T}}$ is complementary to the foreground probability $\boldsymbol{P}^{\boldsymbol{A}}$ from the attention layer. Hence, we fuse $\boldsymbol{P}^{\boldsymbol{A}}$ and $\boldsymbol{P}^{\boldsymbol{T}}$ by convex combination: $\boldsymbol{P}^{\boldsymbol{M}} = 0.5\boldsymbol{P}^{\boldsymbol{A}} + 0.5\boldsymbol{P}^{\boldsymbol{T}}$. The combined probability $\boldsymbol{P}^{\boldsymbol{M}}$ is then used to help localize action instances during inference.

5. Experiments

5.1. Datasets and Evaluation Metric

THUMOS14 [23] contains videos with 20 classes. We use the 200 videos in validation set for training and the 213 videos in testing set for evaluation. **ActivityNet v1.3** [2] covers 200 action categories with 10, 024 and 4, 926 videos in the training and validation sets, respectively. **ActivityNet v1.2** is a subset of ActivityNet v1.3, and covers 100 action categories with 4, 819 and 2, 383 videos in the training and validation sets, respectively We follow the standard evaluation protocol by reporting mean Average Precision (mAP) under various temporal intersection over union (tIoU) thresholds. We refer to **Supplementary** for more experimental details about architecture, setup, baseline, *etc*.

Method	MACs	Params	mAP @ IoU 0.1	0.3	0.5	0.7	AVG 0.1:0.5	AVG
W-TALC [40]	-	-	55.2	40.1	22.8	7.6	39.8	-
BaS-Net [24]	38.60	26.26	58.2	44.6	27.0	10.4	43.6	35.3
TSCN [57]	-	-	63.4	47.8	28.7	10.2	47.0	37.8
ACM-Net [42]	9.48	12.63	68.9	55.0	34.6	10.8	53.2	42.6
CoLA [58]	9.47	12.62	66.2	51.5	32.2	13.1	50.3	40.9
UGCT [53]	-	-	69.	55.5	35.9	11.4	54.0	43.6
CO_2-Net [14]	20.88	34.13	70.1	54.5	38.3	13.4	54.4	44.6
FTCL [11]	9.45	14.73	69.6	55.2	35.6	12.2	53.8	43.6
RSKP [20]	6.90	4.20	71.3	55.8	38.2	12.5	55.6	45.1
ASM-Loc [13]	13.77	46.21	71.2	57.1	36.6	13.4	55.4	45.1
DELU [7]	20.88	34.13	71.5	56.5	40.5	15.5	56.5	**46.4**
CASE	**1.60**	**2.13**	72.3	59.2	37.7	13.7	**57.1**	46.2

Table 1: Comparison on THUMOS14 in terms of mAP (%), MACs (G), and Params (M). AVG (0.1:0.5) and AVG represent the average mAP under IoU thresholds of 0.1:0.5 and 0.1:0.7. MACs are computed from a video with 750 snippets.

5.2. Comparison with SOTA Methods

In Table 1, we compare our CASE with SOTA WTAL methods on THUMOS14. Besides mAP, we also report the model complexity in terms of multi-accumulative operations (MACs), and the number of trainable parameters (Params). We can observe that CASE achieves comparable performance to the recent SOTA method DELU [7] and evidently outperforms other approaches. Notably, CASE is significantly more lightweight than the competitors, with MACs and Params less than 1/10 of those of DELU. This is because our method does not require a heavy cross-modal consensus module [7, 14] or multi-step proposal refinement [13].

Additionally, we compare our CASE with previous methods on ActivityNet v1.2&v1.3 in Table 2. As can be seen, CASE achieves impressive improvements over the previous methods on both datasets.

5.3. Ablation Study

Contribution of core components. Our method contains two core components: snippet clustering component (SCC) and cluster classification component (CCC), which are successively stacked on the baseline model and jointly trained with it. Table 3 quantifies the contributions of each compo-

Method	mAP @ IoU			
	0.5	0.75	0.95	AVG
BaS-Net [24]	38.5	24.2	5.6	24.3
RPN [16]	37.6	23.9	5.4	23.3
EM-MIL [31]	37.4	-	-	20.3
TSCN [57]	37.6	23.7	5.7	23.6
WUM [25]	41.2	25.6	6.0	25.9
CoLA [58]	42.7	25.7	5.8	26.1
ASL [32]	40.2	-	-	25.8
D2-Net [34]	42.3	25.5	5.8	26.0
ACGNet [56]	41.8	26.0	5.9	26.1
DELU [7]	44.2	26.7	5.4	26.9
CASE	43.8	27.2	6.7	**27.9**

(a) ActivityNet v1.2

Method	mAP @ IoU			
	0.5	0.75	0.95	AVG
BaS-Net [24]	34.5	22.5	4.9	22.2
TSCN [57]	35.3	21.4	5.3	21.7
MSA [17]	36.5	22.8	6.0	22.9
ACM-Net [42]	40.1	24.2	6.2	24.6
UGCT [53]	39.1	22.4	5.8	23.8
AUMN [30]	38.3	23.5	5.2	23.5
FAC-Net [19]	37.6	24.2	6.0	24.0
FTCL [11]	40.0	24.3	6.4	24.8
RSKP [20]	40.6	24.6	5.9	25.0
ASM-Loc [13]	41.0	24.9	6.2	25.1
CASE	43.2	26.2	6.7	**26.8**

(b) ActivityNet v1.3

Table 2: Results on ActivityNet v1.2&v1.3. AVG indicates the average mAP at IoU thresholds 0.5:0.05:0.95.

#	Method	mAP @ IoU (%)			
		0.3	0.5	0.7	AVG
1	Baseline	53.8	31.9	11.9	42.1
2	+ SCC	55.3	33.7	12.4	$43.2_{+1.1}$
3	+ SCC + CCC	56.1	34.9	12.8	$43.9_{+0.7}$
4	+ SCC + CCC (T)	59.2	37.7	13.7	$46.2_{+2.3}$

Table 3: Ablation study of core components. "(T)" indicates that clustering-assisted testing is applied.

Metric	*w/o* OT		*w/* OT	
	Soft	Hard	Soft	Hard
mAP	41.2	40.8	46.2	44.7
H($\bar{\boldsymbol{P}}^{\boldsymbol{S}}$)	2.77	0.01	2.76	2.75

Table 4: Ablation study of OT-based labeling.

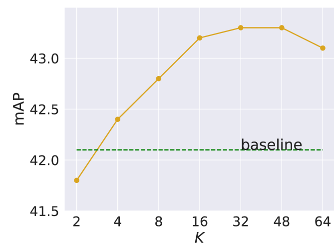

Figure 3: Ablation study of the number of clusters K.

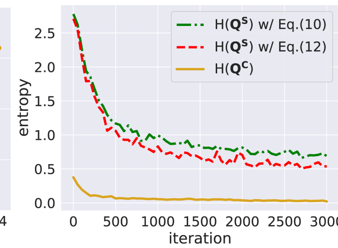

Figure 4: The evolution of $H(\boldsymbol{Q}^{\boldsymbol{C}})$ and $H(\boldsymbol{Q}^{\boldsymbol{S}})$.

nent. When compared to the baseline (line #1), both components contribute to the performance. Specifically, when the clustering-assisted testing technique is not applied, SCC and CCC yield gains of 1.1% (line #2) and 0.7% (line #3), respectively, indicating that the joint training with SCC and CCC indeed improves the performance of the baseline. The reason is that through deliberated clustering of snippets, the embedding space is shaped as well-structured, as illustrated in Fig. 7, thereby facilitating the learning of the baseline. After training SCC and CCC, we leverage the clustering-assisted testing to further make use of SCC and CCC during testing, which brings about a substantial promotion of 2.3% (line #4). In a nutshell, SCC and CCC are beneficial in both the training and testing stages.

Necessity of converting label assignment to optimal transport (OT). We propose an OT-based labeling strategy that generates labels by solving an OT problem: the objective Eq. (7) subject to a constraint Eq. (8) on the proportion of each class. To assess the necessity of our design, we use the labeling strategy in SCC as an example and compare it with alternative labeling strategies. The results are summarized in Table 4. The terms "Hard" and "Soft" indicate whether the labels are one-hot or soft, respectively. The strategy without OT, denoted as "*w/o* OT", refers to solving Eq. (7) without Eq. (8), which is no longer an OT problem and can be solved immediately. In particular, in the case of "*w/o* OT+Soft", the labels would be nearly identical to the model predictions, resulting in almost no learning. As a consequence, it shows poor performance. On the other hand, "*w/o* OT+Hard" involves applying $\arg\max$ to the model predictions, but it also performs poorly. To understand why, we calculate the entropy of the proportions of clusters, namely H($\bar{\boldsymbol{P}}^{\boldsymbol{S}}$), where $\bar{\boldsymbol{P}}^{\boldsymbol{S}} = \frac{1}{N}\sum_{n}^{N} \boldsymbol{P}_{n}^{\boldsymbol{S}}$. In Table 4, we observe that "*w/o* OT+Hard" results in a low H($\bar{\boldsymbol{P}}^{\boldsymbol{S}}$) close to zero, indicating a trivial solution where most snippets are assigned to only a few clusters. In contrast, our proposed "*w/* OT" greatly alleviates this issue and achieves much better performance. Notably, "*w/* OT+Hard" performs worse than "*w/* OT+Soft", which we attribute to the aggressive nature of obtaining hard labels. Overall, the results confirm the efficacy of our design.

Analysis of the number of clusters K. We cluster snippets into multiple clusters ($K > 2$), even though only F&B separation is required. To verify the correctness of our design, we compare the performances under different numbers of K in Fig. 3. As can be seen, a small K results in inferior performance, with $K = 2$ even causing a performance decline relative to baseline. The reason may be that the clustering results deviate too much from the true distribution of F&B snippets. Hence, clustering into multiple clusters is necessary. Besides, the performance becomes stable ($\pm 0.2\%$) and robust to K as long as enough clusters are set ($K > 16$), making it easy to tune an appropriate K in practice. These findings are in accordance with self-supervised learning [41, 38, 49].

Analysis of snippet clustering component (SCC). In SCC, we enforce pseudo-labels $\boldsymbol{Q}^{\boldsymbol{S}}$ to match the prior distribution $\hat{\boldsymbol{Q}}^{\boldsymbol{S}}$ by transitioning the objective of OT problem from the original objective Eq. (10) to the current objective Eq. (12). To evaluate the impact of this modification, in Table 5, we compare the results of our method (line #1)

#	Method	mAP
1	our CASE	46.2
2	using Eq. (10) in SCC	45.4
3	using Eq. (9) in CCC	44.0

Table 5: Analysis of unique techniques in SCC and CCC.

Method	$\boldsymbol{P^A}$	$\boldsymbol{P^T}$	$\boldsymbol{P^M}$
mAP	43.9	44.1	46.2

Table 6: Analysis of the clustering-assisted testing.

against those obtained using the original Eq. (10) (line # 2). The results show that our modified objective outperforms the original one, indicating that matching the prior distribution is beneficial. To further examine this phenomenon, in Fig. 4, we plot the evolution of the entropy of $\boldsymbol{Q^S}$, $\mathrm{H}(\boldsymbol{Q^S})$, using current Eq. (12) or original Eq. (10). A large $\mathrm{H}(\boldsymbol{Q^S})$ is typically undesired as it indicates that the assignment is uncertain. Additionally, we report the entropy of $\boldsymbol{Q^C}$ in CCC, $\mathrm{H}(\boldsymbol{Q^C})$, as an indication, though a direct comparison is not entirely fair. The results reveal that $\mathrm{H}(\boldsymbol{Q^C})$ converges rapidly to a small value, indicating that optimization can easily yield a solution close to a one-hot form in CCC. Conversely, $\mathrm{H}(\boldsymbol{Q^S})$ converges slowly and remains high, suggesting uncertain assignments in SCC. However, this issue is mitigated when using our proposed Eq. (12).

Analysis of cluster classification component (CCC). In CCC, we empirically estimate the prior marginal distribution of pseudo-labels $\boldsymbol{Q^C}$ based on Eq. (13), as opposed to using equipartition Eq. (9). To evaluate the impact of this design choice, we present the results of using equipartition Eq. (9) in line #3 of Table 5. The clear superiority of our method over the use of equipartition highlights the importance of using a data-driven approach to estimate the prior distribution of pseudo-labels in CCC.

Analysis of clustering-assisted testing. We propose to fuse the foreground probabilities $\boldsymbol{P^A}$ from the baseline model and the foreground probabilities $\boldsymbol{P^T}$ transformed from the cluster assignments, resulting in the fused one $\boldsymbol{P^M}$. In Table 6, we compare the performances of $\boldsymbol{P^A}$, $\boldsymbol{P^T}$, and $\boldsymbol{P^M}$. It can be seen that the fused one $\boldsymbol{P^M}$ outperforms both $\boldsymbol{P^A}$ and $\boldsymbol{P^T}$, which well demonstrates that $\boldsymbol{P^A}$ and $\boldsymbol{P^T}$ are complementary to each other. This is further confirmed by the qualitative results exemplified in Fig. 5. Concretely, it delivers the following inspiring observations: 1) Compared to $\boldsymbol{P^A}$, $\boldsymbol{P^T}$ tends to activate more action-related regions (*e.g.*, ②, ④) and clearer action boundaries (*e.g.*, ③, ⑤). This is in line with our motivation for using $\boldsymbol{P^T}$, which is designed to be more independent of the video classification loss, capturing a more comprehensive distribution of snippets rather than being biased by discriminative regions. 2)

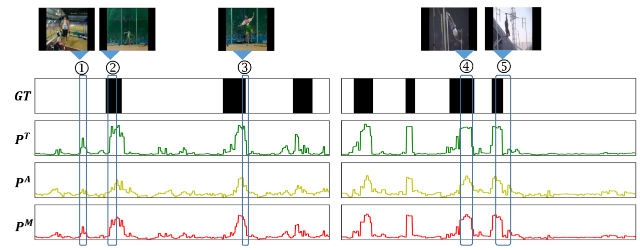

Figure 5: Qualitative results of two videos on THUMOS14. We show $\boldsymbol{P^T}$, $\boldsymbol{P^A}$ and $\boldsymbol{P^M}$, and ground-truth (GT).

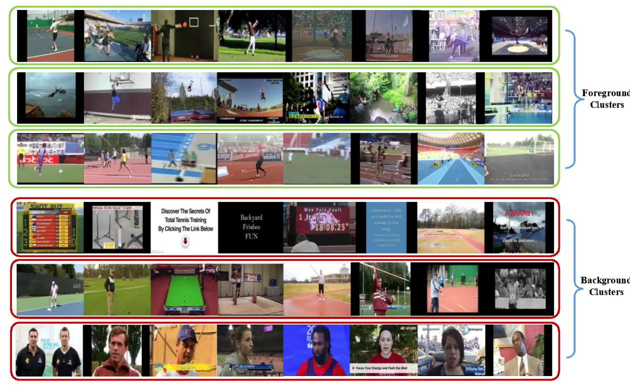

Figure 6: Qualitative results of snippet clustering and cluster classification. We show three clusters belonging to foreground at the top and there clusters belonging to background at the bottom.

$\boldsymbol{P^T}$ is not always superior to $\boldsymbol{P^A}$. When optimizing $\boldsymbol{P^T}$, CASE regards the snippets as independent samples and fails to make use of the video-level labels, resulting in irrelevant action activation (*e.g.*, ①). These observations further support the complementary nature of $\boldsymbol{P^A}$ and $\boldsymbol{P^T}$.

Visualization results. To gain more insight, we illustrate qualitative results of snippet clustering and cluster classification in Fig. 6. It can be seen that: 1) Our method effectively identifies the snippet groups with common characteristics, such as ”swinging arms” for the 1st row, ”standing” for the 5th row. 2) Our method is accurate in classifying clusters into F&B classes. For instance, some foreground frames in 1st&3rd rows and background frames in 5th row are visually similar, yet our method can still separate them into different clusters with correct F&B labels. Overall, these results vividly substantiate the efficacy of our method in effectively grouping and classifying snippets.

Furthermore, we visualize the snippet embeddings and cluster prototypes in Fig. 7 and make the following observations: 1) Each cluster prototype is surrounded by a couple of snippets, and the cluster prototypes are distinguishable between foreground and background. 2) The embedding space learned by our method exhibits a clearer boundary

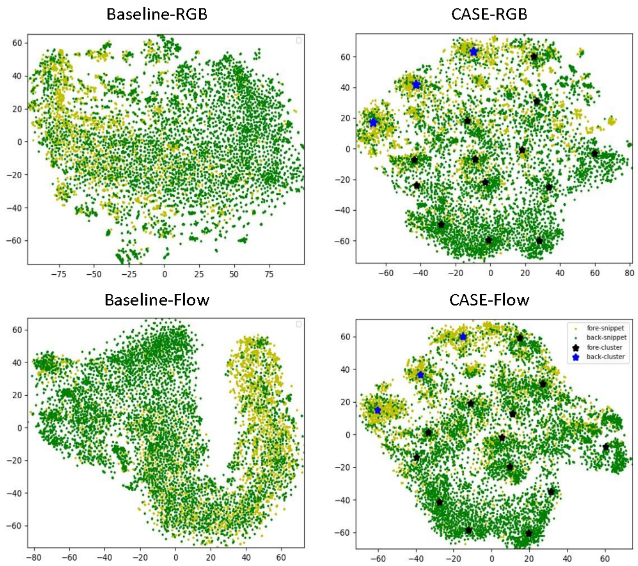

Figure 7: TSNE visualization of $10,000$ snippet embeddings and 16 cluster prototypes. ”-RGB” and ”-Flow” indicate RGB stream and optical-flow stream, respectively. The yellow, red, black, and blue marks represent foreground snippets, background snippets, foreground cluster prototypes, and background cluster prototypes, respectively.

between foreground and background classes compared to the baseline. These results clearly demonstrate that 1) the learned clusters are highly representative of the snippets; 2) the embedding space in our method is well-shaped and captures a comprehensive distribution of the snippets.

6. Conclusion and Limitation

In this work, we propose a WTAL framework named CASE, which leverages snippet clustering to improve F&B separation. Specifically, CASE comprises a snippet clustering component that partitions snippets into multiple clusters, followed by a cluster classification component that identifies the F&B clusters. To optimize these components, we employ a unified self-labeling strategy based on optimal transport. Our extensive analysis demonstrates the effectiveness and efficiency of CASE. One limitation of our method is the requirement for a WTAL baseline to provide semantic-level reference of F&B classes to classify the clusters into F&B. A more self-contained clustering-based framework is our future work.

7. Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant 62176246 and Grant 61836008. This work is also supported by Anhui Province Key Research and Development Plan (202304a05020045), Anhui Provincial Natural Science Foundation 2208085UD17 and the Fundamental Research Funds for the Central Universities (WK3490000006).

References

[1] YM Asano, C Rupprecht, and A Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2019. 1, 2, 5

[2] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 6

[3] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018. 2, 3

[4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 2020. 1, 2, 5

[5] Jianlong Chang, Yiwen Guo, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep discriminative clustering analysis. *arXiv preprint arXiv:1905.01681*, 2019. 2

[6] Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep adaptive image clustering. In *ICCV*, 2017. 1, 2

[7] Mengyuan Chen, Junyu Gao, Shicai Yang, and Changsheng Xu. Dual-evidential learning for weakly-supervised temporal action localization. In *ECCV*, 2022. 6, 7

[8] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*, 2013. 5

[9] Zhiyuan Dang, Cheng Deng, Xu Yang, and Heng Huang. Multi-scale fusion subspace clustering using similarity constraint. In *CVPR*, 2020. 2

[10] Enrico Fini, Enver Sangineto, Stéphane Lathuilière, Zhun Zhong, Moin Nabi, and Elisa Ricci. A unified objective for novel class discovery. In *ICCV*, 2021. 1, 3

[11] Junyu Gao, Mengyuan Chen, and Changsheng Xu. Fine-grained temporal contrastive learning for weakly-supervised temporal action localization. In *CVPR*, 2022. 6, 7

[12] Philip Haeusser, Johannes Plapp, Vladimir Golkov, Elie Aljalbout, and Daniel Cremers. Associative deep clustering: Training a classification network with no labels. In *GCPR*, 2018. 2

[13] Bo He, Xitong Yang, Le Kang, Zhiyu Cheng, Xin Zhou, and Abhinav Shrivastava. Asm-loc: Action-aware segment modeling for weakly-supervised temporal action localization. In *CVPR*, 2022. 6, 7

[14] Fa-Ting Hong, Jia-Chang Feng, Dan Xu, Ying Shan, and Wei-Shi Zheng. Cross-modal consensus network for weakly supervised temporal action localization. In *ACMMM*, 2021. 6

[15] Jiabo Huang, Shaogang Gong, and Xiatian Zhu. Deep semantic clustering by partition confidence maximisation. In *CVPR*, 2020. 2

[16] Linjiang Huang, Yan Huang, Wanli Ouyang, and Liang Wang. Relational prototypical network for weakly supervised temporal action localization. In *AAAI*, 2020. 7

[17] Linjiang Huang, Yan Huang, Wanli Ouyang, and Liang Wang. Modeling sub-actions for weakly supervised temporal action localization. *Transactions on Image Processing*, 2021. 3, 7

[18] Linjiang Huang, Yan Huang, Wanli Ouyang, and Liang Wang. Two-branch relational prototypical network for weakly supervised temporal action localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 3

[19] Linjiang Huang, Liang Wang, and Hongsheng Li. Foreground-action consistency network for weakly supervised temporal action localization. In *ICCV*, 2021. 7

[20] Linjiang Huang, Liang Wang, and Hongsheng Li. Weakly supervised temporal action localization via representative snippet knowledge propagation. In *CVPR*, 2022. 6, 7

[21] Ashraful Islam, Chengjiang Long, and Richard Radke. A hybrid attention mechanism for weakly-supervised temporal action localization. In *AAAI*, 2021. 3

[22] Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*, 2019. 2, 3

[23] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes, 2014. 6

[24] Pilhyeon Lee, Youngjung Uh, and Hyeran Byun. Background suppression network for weakly-supervised temporal action localization. In *AAAI*, 2020. 3, 6, 7

[25] Pilhyeon Lee, Jinglu Wang, Yan Lu, and Hyeran Byun. Weakly-supervised temporal action localization by uncertainty modeling. In *AAAI*, 2021. 7

[26] Jingjing Li, Tianyu Yang, Wei Ji, Jue Wang, and Li Cheng. Exploring denoised cross-video contrast for weakly-supervised temporal action localization. In *CVPR*, 2022. 3

[27] Ruihuang Li, Shuai Li, Chenhang He, Yabin Zhang, Xu Jia, and Lei Zhang. Class-balanced pixel-level self-labeling for domain adaptive semantic segmentation. In *CVPR*, 2022. 1

[28] Zhi Li, Lu He, and Huijuan Xu. Weakly-supervised temporal action detection for fine-grained videos with hierarchical atomic actions. In *ECCV*, 2022. 3

[29] Daochang Liu, Tingting Jiang, and Yizhou Wang. Completeness modeling and context separation for weakly supervised temporal action localization. In *CVPR*, 2019. 1, 3

[30] Wang Luo, Tianzhu Zhang, Wenfei Yang, Jingen Liu, Tao Mei, Feng Wu, and Yongdong Zhang. Action unit memory network for weakly supervised temporal action localization. In *CVPR*, 2021. 3, 7

[31] Zhekun Luo, Devin Guillory, Baifeng Shi, Wei Ke, Fang Wan, Trevor Darrell, and Huijuan Xu. Weakly-supervised action localization with expectation-maximization multi-instance learning. In *ECCV*, 2020. 1, 3, 7

[32] Junwei Ma, Satya Krishna Gorti, Maksims Volkovs, and Guangwei Yu. Weakly supervised action selection learning in video. In *CVPR*, 2021. 3, 7

[33] Kyle Min and Jason J Corso. Adversarial background-aware loss for weakly-supervised temporal activity localization. In *ECCV*. Springer, 2020. 1, 3

[34] Sanath Narayan, Hisham Cholakkal, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. D2-net: Weakly-supervised action localization via discriminative embeddings and denoised activations. In *ICCV*, 2021. 7

[35] Sanath Narayan, Hisham Cholakkal, Fahad Shabaz Khan, and Ling Shao. 3c-net: Category count and center loss for weakly-supervised action localization. In *ICCV*, 2019. 3

[36] Phuc Nguyen, Ting Liu, Gautam Prasad, and Bohyung Han. Weakly supervised action localization by sparse temporal pooling network. In *CVPR*, 2018. 3

[37] Phuc Xuan Nguyen, Deva Ramanan, and Charless C Fowlkes. Weakly-supervised action localization with background modeling. In *ICCV*, 2019. 1

[38] Bo Pang, Yifan Zhang, Yaoyi Li, Jia Cai, and Cewu Lu. Unsupervised visual representation learning by synchronous momentum grouping. In *ECCV*, 2022. 4, 7

[39] Alejandro Pardo, Humam Alwassel, Fabian Caba, Ali Thabet, and Bernard Ghanem. Refineloc: Iterative refinement for weakly-supervised action localization. In *WACV*, 2021. 1, 3

[40] Sujoy Paul, Sourya Roy, and Amit K Roy-Chowdhury. W-talc: Weakly-supervised temporal activity localization and classification. In *ECCV*, 2018. 6

[41] Qi Qian, Yuanhong Xu, Juhua Hu, Hao Li, and Rong Jin. Unsupervised visual representation learning by online constrained k-means. 2022. 4, 7

[42] Sanqing Qu, Guang Chen, Zhijun Li, Lijun Zhang, Fan Lu, and Alois Knoll. Acm-net: Action context modeling network for weakly-supervised temporal action localization. *Transactions on Image Processing*, 2021. 3, 6, 7

[43] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, 2016. 1

[44] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*, 2017. 3

[45] Bing Su and Gang Hua. Order-preserving wasserstein distance for sequence matching. In *CVPR*, 2017. 5

[46] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *ECCV*, 2020. 3

[47] Binglu Wang, Xun Zhang, and Yongqiang Zhao. Exploring sub-action granularity for weakly supervised temporal action localization. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021. 3

[48] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *CVPR*, 2017. 1

[49] Xudong Wang, Ziwei Liu, and Stella X Yu. Unsupervised feature learning by cross-level instance-group discrimination. In *CVPR*, 2021. 7

[50] Jianlong Wu, Keyu Long, Fei Wang, Chen Qian, Cheng Li, Zhouchen Lin, and Hongbin Zha. Deep comprehensive correlation mining for image clustering. In *ICCV*, 2019. 2

[51] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, 2016. 2

[52] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *CVPR*, 2016. 2

[53] Wenfei Yang, Tianzhu Zhang, Xiaoyuan Yu, Tian Qi, Yongdong Zhang, and Feng Wu. Uncertainty guided collaborative training for weakly supervised temporal action detection. In *CVPR*, 2021. 6, 7

[54] Xu Yang, Cheng Deng, Kun Wei, Junchi Yan, and Wei Liu. Adversarial learning for robust deep clustering. In *NeurIPS*, 2020. 2

[55] Xu Yang, Cheng Deng, Feng Zheng, Junchi Yan, and Wei Liu. Deep spectral clustering using dual autoencoder network. In *CVPR*, 2019. 2

[56] Zichen Yang, Jie Qin, and Di Huang. Acgnet: Action complement graph network for weakly-supervised temporal action localization. In *AAAI*, 2022. 7

[57] Yuanhao Zhai, Le Wang, Wei Tang, Qilin Zhang, Junsong Yuan, and Gang Hua. Two-stream consensus network for weakly-supervised temporal action localization. In *ECCV*, 2020. 3, 6, 7

[58] Can Zhang, Meng Cao, Dongming Yang, Jie Chen, and Yuexian Zou. Cola: Weakly-supervised temporal action localization with snippet contrastive learning. In *CVPR*, 2021. 1, 3, 6, 7

[59] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NeurIPS*, 2018. 3

[60] Jia-Xing Zhong, Nannan Li, Weijie Kong, Tao Zhang, Thomas H. Li, and Ge Li. Step-by-step erasion, one-by-one collection: A weakly supervised temporal action detector. In *ACMMM*, 2018. 3