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

Qinying Liu1, Zilei Wang1‡, Shenghai Rong1, Junjie Li1, Yixin Zhang1,2
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 clusters 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/Qinyying-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 simply 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
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 Clustering-Assisted F&B SEparation (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 solutions. 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 \( Y \in \mathbb{R}^G \), 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 \( F \in \mathbb{R}^{T \times D} \). 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 \( F \) to an embedding encoder followed by an action classifier to get the temporal class activation sequence \((T-CAS) P^V \in \mathbb{R}^{T \times G}\). In the latter branch, \( F \) 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 Attention weights \( P^A \in \mathbb{R}^G \), 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 \( \bar{P}^V \in \mathbb{R}^G \). Finally, we optimize a video classification loss with the known video labels \( Y \):

\[
\mathcal{L}_V = \mathcal{L}_{CE}(\bar{P}^V, Y).
\]  

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 \( Q^A \in \mathbb{R}^G \) 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]:

\[
\begin{align*}
\mathcal{L}_A &= \frac{1}{N_{pos}} \sum_{t=1}^{T} Q^A_t \frac{1 - (P^A_t)^\gamma}{\gamma} + \\
&\quad \frac{1}{N_{neg}} \sum_{t=1}^{T} (1 - Q^A_t) \frac{1 - (1 - P^A_t)^\gamma}{\gamma},
\end{align*}
\]  

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 $E \in \mathbb{R}^{N \times D}$. We feed $E$ into a clustering head composed of a linear classifier with $K$ classes (clusters), producing snippet-level cluster assignment probabilities dubbed $P^S \in \mathbb{R}^{N \times K}$. 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 $Q^S \in \mathbb{R}^{N \times K}$ for $P^S$, which will be described in Sec. 4.2. Then, we minimize the following loss:

$$L_S = \frac{1}{N} \sum_{n=1}^{N} L_{CE}(Q^S_n, P^S_n). \quad (3)$$

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 $Q^S \in \mathbb{R}^{N \times K}$ obtained from SCC, we can compute the $k$-th cluster prototype over the snippet embeddings $E \in \mathbb{R}^{N \times D}$:

$$\bar{E}_k^S = \frac{\sum_{n=1}^{N} Q^S_{n,k} \cdot E_n}{\sum_{n=1}^{N} Q^S_{n,K}}. \quad (4)$$

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

$$P^C_{k,i} = Softmax_i \left( \rho \cdot \cos(\bar{E}_k^S, \bar{E}_i^A) \right), \quad (5)$$

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

$$L_C = \frac{1}{K} \sum_{k=1}^{K} L_{CE}(Q^C_k, P^C_k). \quad (6)$$

Notably, although $Q^C$ is computed from each minibatch 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 $Q^S$ for $P^S$ in SCC and labels $Q^C$ for $P^C$ 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 \( Q^S \) 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 \( Q^C \). 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 \( Q \), a straightforward solution is to search for a reasonable \( Q \) that is close to the current model predictions \( P \), e.g., by applying arg max to \( P \). 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 \( Q \), we propose to impose a constraint on the proportion of elements assigned to each class. Formally, we formulate this as an optimization problem:

\[
\min_{Q \in \Omega} E(P, Q),
\]

where \( E(P, Q) = -\sum_{n}^{N} \sum_{k}^{K} Q_{n,k} \log P_{n,k} \) measures the distance between \( Q \) and \( P \), \( N \) is the number of samples and \( K \) is the number of classes. The constraint \( \Omega \) is defined as:

\[
\Omega = \{ Q \in \mathbb{R}^{N \times K} | Q1^{K} = \alpha, Q^{T}1^{N} = \beta \},
\]

where \( \alpha \) and \( \beta \) are the marginal distributions of \( Q \) onto its rows and columns, respectively. We set \( \alpha = 1^{N} \) to ensure that \( Q \) is a probability matrix. \( \beta \in \mathbb{R}^{K} \) 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 \( \beta \):

\[
\beta = \frac{N}{K}1^{K}.
\]

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:

\[
\min_{Q \in \Omega} E(P, Q) - \frac{1}{\epsilon} H(Q),
\]

where \( \epsilon > 0 \) and \( H(Q) \) is the entropy of \( Q \). 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 \( Q^S \) in SCC, denoted as \( \hat{Q}^S \in \mathbb{R}^{N \times K} \), 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 \( P^{A} \) in ascending order, and denote the resulting ranks of the \( N \) snippets as \( rank \in \mathbb{R}^{N} \). We then construct the prior distribution \( \hat{Q}^S \), 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., \( Q^{C}_{k,1} \)) and vice versa. Formally, \( \hat{Q}^S \) is defined as a Gaussian distribution:

\[
\hat{Q}^S_{n,k} = \frac{1}{\sigma \sqrt{2\pi}} \exp \left( -\frac{(rank_{n} - Q^{C}_{k,1})^2}{2\sigma^2} \right),
\]

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

\[
\min_{Q^S \in \Omega} E(P^S, Q^S) - \frac{1}{\epsilon} KL(Q^S, \hat{Q}^S),
\]

where \( KL(\cdot) \) is the Kullback-Leibler divergence. By minimizing the KL term, we encourage the labels \( Q^S \) to be close to the prior distribution \( \hat{Q}^S \), 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 \( Q^C \), namely \( \beta^C \in \mathbb{R}^{2} \). This is due to the fact that \( \beta^C \) 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 \( \beta^C \) empirically based on the distribution of the snippet-level foreground labels \( Q^A \):

\[
\beta^C = \left[ \frac{1}{N} \sum_{n=1}^{N} Q^A_n, \frac{1}{N} \sum_{n=1}^{N} (1 - Q^A_n) \right].
\]
\begin{algorithm}
\begin{algorithmic}[1]
\STATE # L: S: logit output of clustering head (NxK)
\STATE # Q_h: prior distribution in SCC (N x K)
\STATE # L_C: logit output of cluster classification (Kx2)
\STATE # Beta: prior marginal distribution in CCC (K)
\STATE # generating labels Q_S for SCC
\STATE \textbf{Q}_S = \text{SK}(L, Q\_hat=Q\_hat_S) \# (N x K)
\STATE # generating labels Q_C for CCC
\STATE \textbf{Q}_C = \text{SK}(L_C, Beta= Beta_C) \# (K x 2)
\STATE # uniform distribution is the default when prior distribution is not given
\STATE \textbf{Q}\_hat = \frac{1}{K} \text{ if } Q\_hat = \text{None } \text{ else } Q\_hat
\STATE Q = \exp(L / \text{ eps})
\STATE Q = \text{Q} + \text{Q}\_hat \cdot T
\STATE Q /= \text{Q. sum(dim=1)}
\STATE Q = \text{Q} \cdot \text{Beta}
\STATE Q /= \text{Q. sum(dim=0)}
\STATE Q /= N
\STATE # resolve the optimal-transport problem by iterative Sinkhorn-Knopp (SK) algorithm
\STATE def SK(L, Q\_hat=None, Beta=None, n\_iter=3):
\STATE \textbf{Q} = \text{L. size()}
\STATE uniform distribution is the default when prior distribution is not given
\STATE \textbf{Q}\_hat = \frac{1}{K} \text{ if } Q\_hat = \text{None } \text{ else } Q\_hat
\STATE \textbf{Beta} = \frac{1}{K} \text{ if } \text{Beta} = \text{None } \text{ else } \text{Beta}
\STATE Q = \exp(L / \text{ eps})
\STATE Q = Q + Q\_hat \cdot T
\STATE Q /= \text{Q. sum(dim=1)}
\STATE Q = Q \cdot Beta
\STATE Q /= \text{Q. sum(dim=0)}
\STATE Q /= N
\STATE return Q.t() * N
\end{algorithmic}
\end{algorithm}

4.3. Training and Testing

Joint training. We train all components together in an end-to-end fashion. The overall objective is written as:
\[
L = (L_V + L_A) + \lambda_S L_S + \lambda_C L_C,
\]
where $\lambda_S$ and $\lambda_C$ 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 $Q_C$, we can transform the snippet-level cluster assignments $P_S^C$ to snippet-level foreground probabilities dubbed $P^T$ based on law of total probability: $P^T = P^S Q_C$, as depicted in Fig. 2 (c). Considering that $Q_C$ is stable during training, we simply use the $Q_C$ from the last training iteration for inference. Moreover, as verified in Table 6, the transformed foreground probability $P^T$ is complementary to the foreground probability $P^A$ from the attention layer. Hence, we fuse $P^A$ and $P^T$ by convex combination: $P^M = 0.5 P^A + 0.5 P^T$. The combined probability $P^M$ 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 (IoU) thresholds. We refer to Supplementary for more experimental details about architecture, setup, baseline, etc.

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

<table>
<thead>
<tr>
<th>Method</th>
<th>mAP @ IoU (%)</th>
<th>0.3</th>
<th>0.5</th>
<th>0.7</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>BasNet [24]</td>
<td>34.5</td>
<td>22.5</td>
<td>4.9</td>
<td></td>
<td>22.2</td>
</tr>
<tr>
<td>RPN [16]</td>
<td>35.3</td>
<td>21.4</td>
<td>5.3</td>
<td></td>
<td>21.7</td>
</tr>
<tr>
<td>EM-MIL [31]</td>
<td>36.5</td>
<td>22.8</td>
<td>6.0</td>
<td></td>
<td>22.9</td>
</tr>
<tr>
<td>TSCN [57]</td>
<td>35.3</td>
<td>20.8</td>
<td>6.0</td>
<td></td>
<td>20.7</td>
</tr>
<tr>
<td>WUM [25]</td>
<td>35.8</td>
<td>20.9</td>
<td>6.0</td>
<td></td>
<td>20.9</td>
</tr>
<tr>
<td>CoLA [58]</td>
<td>36.7</td>
<td>21.8</td>
<td>6.0</td>
<td></td>
<td>21.8</td>
</tr>
<tr>
<td>ASL [32]</td>
<td>37.7</td>
<td>22.9</td>
<td>6.0</td>
<td></td>
<td>22.9</td>
</tr>
<tr>
<td>D2-Net [34]</td>
<td>38.2</td>
<td>23.4</td>
<td>6.0</td>
<td></td>
<td>23.4</td>
</tr>
<tr>
<td>AGCNN [35]</td>
<td>38.5</td>
<td>23.7</td>
<td>6.0</td>
<td></td>
<td>23.7</td>
</tr>
<tr>
<td>DELU [7]</td>
<td>39.0</td>
<td>24.1</td>
<td>6.0</td>
<td></td>
<td>24.1</td>
</tr>
<tr>
<td>CASE</td>
<td>40.8</td>
<td>25.5</td>
<td>6.0</td>
<td></td>
<td>25.5</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>#</th>
<th>Method</th>
<th>mAP @ IoU (%)</th>
<th>0.3</th>
<th>0.5</th>
<th>0.7</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Baseline</td>
<td>53.8</td>
<td>31.9</td>
<td>11.9</td>
<td>42.1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>+ SCC</td>
<td>55.3</td>
<td>33.7</td>
<td>12.4</td>
<td>43.2±0.1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>+ SCC + CCC</td>
<td>56.1</td>
<td>34.9</td>
<td>12.8</td>
<td>43.9±0.7</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>+ SCC + CCC (T)</td>
<td>59.2</td>
<td>37.7</td>
<td>13.7</td>
<td>46.2±0.3</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Ablation study of OT-based labeling.

<table>
<thead>
<tr>
<th>Metric</th>
<th>w/o OT</th>
<th>w/ OT</th>
</tr>
</thead>
<tbody>
<tr>
<td>mAP</td>
<td>Soft</td>
<td>Hard</td>
</tr>
<tr>
<td>H(P_S)</td>
<td>41.2</td>
<td>40.8</td>
</tr>
</tbody>
</table>

Figure 3: Ablation study of the number of clusters K. Figure 4: The evolution of H(Q^C) and H(Q^S).

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 proportion of clusters, namely H(P_S), where P_S = \frac{1}{N} \sum_{i=1}^{N} P_i. In Table 4, we observe that "w/o OT+Hard" results in a low H(P_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 (±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 [31, 38, 49].

Analysis of snippet clustering component (SCC). In SCC, we enforce pseudo-labels Q^S to match the prior distribution Q^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)
Table 5: Analysis of unique techniques in SCC and CCC.

<table>
<thead>
<tr>
<th>#</th>
<th>Method</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>our CASE</td>
<td>46.2</td>
</tr>
<tr>
<td>2</td>
<td>using Eq. (10) in SCC</td>
<td>45.4</td>
</tr>
<tr>
<td>3</td>
<td>using Eq. (9) in CCC</td>
<td>44.0</td>
</tr>
</tbody>
</table>

Table 6: Analysis of the clustering-assisted testing.

<table>
<thead>
<tr>
<th>Method</th>
<th>$P^A$</th>
<th>$P^T$</th>
<th>$P^M$</th>
</tr>
</thead>
<tbody>
<tr>
<td>mAP</td>
<td>43.9</td>
<td>44.1</td>
<td>46.2</td>
</tr>
</tbody>
</table>

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 $Q^S$, $H(Q^S)$, using current Eq. (12) or original Eq. (10). A large $H(Q^S)$ is typically undesired as it indicates that the assignment is uncertain. Additionally, we report the entropy of $Q^C$ in CCC, $H(Q^C)$, as an indication, though a direct comparison is not entirely fair. The results reveal that $H(Q^C)$ converges rapidly to a small value, indicating that optimization can easily yield a solution close to a one-hot form in CCC. Conversely, $H(Q^S)$ 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 $Q^C$ 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 $P^A$ from the baseline model and the foreground probabilities $P^T$ transformed from the cluster assignments, resulting in the fused one $P^M$. In Table 6, we compare the performances of $P^A$, $P^T$, and $P^M$. It can be seen that the fused one $P^M$ outperforms both $P^A$ and $P^T$, which well demonstrates that $P^A$ and $P^T$ 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 $P^A$, $P^T$ 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 $P^T$, 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) $P^T$ is not always superior to $P^A$. When optimizing $P^T$, 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 $P^A$ and $P^T$.

**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.
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