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

Contents

A. Notation

In Table 1, we summarize the key notations used in our paper for easy reference.

B. Architecture

Table 2 presents the details of the architecture of our proposed CASE. As can be seen, the architecture is simple and lightweight, consisting of only a few 1D convolutional layers with a kernel size of 1 and linear layers, leading to the high efficiency of our method.

	Notation	Shape	Description
Baseline	$\boldsymbol{P}^{\boldsymbol{A}}$	N	foreground probability
	$\boldsymbol{Q}^{\boldsymbol{A}}$	N	pseudo-labels of $\boldsymbol{P}^{\boldsymbol{A}}$
	$\boldsymbol{Z}$	$N \times D$	snippet embedding
SCC	$\boldsymbol{P}^{\boldsymbol{S}}$	$N \times K$	cluster assignment probability
	$\boldsymbol{Q}^{\boldsymbol{S}}$	$N \times K$	pseudo-labels of $\boldsymbol{P}^{\boldsymbol{S}}$
	$\hat{\boldsymbol{Q}}^{\boldsymbol{S}}$	$N \times K$	prior distribution of $\boldsymbol{Q}^{\boldsymbol{S}}$
CCC	$\boldsymbol{P}^{\boldsymbol{C}}$	$K \times 2$	cluster classification probability
	$\boldsymbol{Q}^{\boldsymbol{C}}$	$K \times 2$	pseudo-labels of $\boldsymbol{P}^{\boldsymbol{C}}$
	$\boldsymbol{\beta}^{\boldsymbol{C}}$	2	prior marginal distribution of $\boldsymbol{Q}^{\boldsymbol{C}}$
Testing	$\boldsymbol{P}^{\boldsymbol{T}}$	$N \times K$	transformed foreground probability
	$\boldsymbol{P}^{\boldsymbol{M}}$	$N \times K$	fused foreground probability

Table 1: Key notations in this paper.

component	layer	kernel	stride	dim	act	output size
Baseline	Embedding Encoder					
	Conv1d	1	1	512	Relu	512×T
	Action Classifier					
	Conv1d	1	1	G	Softmax	$G \times T$
	Embedding Encoder					
	Conv1d	1	1	512	Relu	512×T
	Attention layer					
	Conv1d	1	1	1	Sigmoid	$1 \times T$
Our Algorithm	Clustering Head					
	Linear	1	1	K	Softmax	$K \times T$

Table 2: The detailed architecture of CASE, where the RGB stream and optical flow stream share the same structure.

C. Additional Ablation Experiments

C.1. Ablation on multiple datasets

To show the effectiveness of our method in various scenarios, we perform a component-wise ablation study for the snippet clustering component (SCC) and the cluster classification component (CCC) on THUMOS14, ActivityNet v1.2 and v1.3. The corresponding results are provided in Table 3. We observe consistent trends across all datasets, indicating the robustness and effectiveness of our approach.

*Corresponding author

	THUMOS14	ActivityNet v1.2	ActivityNet v1.3
Baseline	42.1	25.6	24.7
+ SCC	43.2	26.5	25.4
+ SCC + CCC	43.9	27.0	25.7
+ SCC + CCC (T)	46.2	27.9	26.8

Table 3: Component-wise ablation study on THUMOS14, ActivityNet v1.2 and v1.3. ”(T)” indicates that the clustering-assisted testing technique is appiled.

VTB	ATB	GBCE	mAP
✓			32.0
✓	✓		41.7
✓	✓	✓	42.1

Table 4: Ablation study on the baseline. VTB, ATB, and GBCE indicate video classification branch, attention branch, and generalized binary cross-entropy loss, respectively. Notably, if GBCE is not used, we use the traditional binary cross-entropy loss to train the ATB.

C.2. Analysis of baseline model

We carry out several ablation experiments to analyze the components of the baseline. The results are illustrated in Table 4. It can be seen that the attention branch largely increases the performance, demonstrating the significance of class-agnostic F&B separation. Additionally, we find that the use of the generalized binary cross-entropy loss yields better results than the traditional binary cross-entropy loss, proving that enhancing the label noise tolerance is advantageous.

C.3. Analysis of ranking indices $rank$

In SCC, we use the distance between the normalized ranking indices of the snippets $rank/N$ and the cluster-level pseudo-labels $\boldsymbol{Q^C}$ to compute a 2D gaussian distribution. In principle, $rank/N$ can be replaced by $\boldsymbol{P^A}$. However, we experimentally find that the performance of using $\boldsymbol{P^A}$ is inferior to that of using $rank/N$ (*i.e.*, 45.1 for $\boldsymbol{P^A}$ *vs.* 46.2 for $rank/N$ on average mAP). To explain it, we show the statistics (*i.e.*, maximum, average and minimum) of $\boldsymbol{P^A}$ and $\boldsymbol{Q^C}$ in Fig. 1. The statistics are computed over each batch (*i.e.*, iteration). Notably, the maximum, average, and minimum of $rank/N$ are always $\frac{1}{N} \simeq 0$, $0.5 + 0.5\frac{1}{N} \simeq 0.5$ and 1, respectively. As can be seen, compared with $\boldsymbol{P^A}$, $rank/N$ is more comparable to $\boldsymbol{Q^C}$. For example, both the average of $\boldsymbol{Q^C}$ and the average of $rank/N$ are around 0.5 and are evidently larger than the average of $\boldsymbol{P^A}$. This observation confirms the validity of our approach.

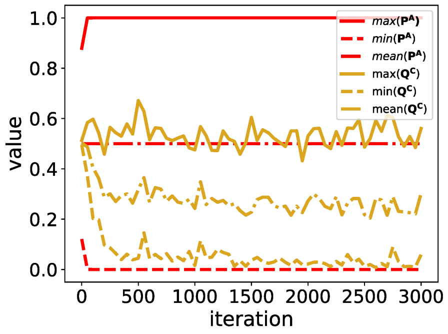

Figure 1: The maximum, average, and minimum values of $\boldsymbol{P^A}$ and $\boldsymbol{Q^C}$ of each iteration during training.

D. Additional Visualizations

D.1. Comparison to baseline

In Fig. 2, four visualized examples are provided to illustrate the differences between the F&B separation results of CASE and that of baseline. It can be observed that: 1) CASE is advantageous to capture fine-grained patterns of snippets that are helpful to distinguish different snippets (see the solid boxes). For instance, in the region of ’4’, which is near the boundary of a ’diving’ action instance, the foreground snippets and the background snippets are visually similar. However, CASE can accurately classify these snippets into correct F&B classes, whereas the baseline cannot, showing that CASE can capture the underlying fine-grained structure of the snippets. 2) CASE performs worse than the baseline in some ’suspicious’ regions (see the dashed boxes). To name a few, in the region of ’8’, an athlete raises her leg, causing CASE to mistake the region for an action instance. This mistake may be avoided by the baseline model because the video-level labels used to train the baseline can offer instructive information for the potential action types within the videos.

D.2. Failure cases

We showcase some examples of failure cases of our method in Fig. 3. From the figure, we conjecture the reason that accounts for the failure cases are: 1) low quality of images, *e.g.*, ’1’ and ’8’; 2) indistinguishable body motions, *e.g.*, ’3’ and ’7’; 3) small objects, *e.g.*, ’2’ and ’4’; 4) incorrect annotation, *e.g.*, ’5’ and ’6’. These challenging cases represent future directions for our work.

E. Additional Discussion on Related Work

In our main paper, we extensively discussed the differences between our method and previous deep clustering and WTAL methods in the Related Work section. In this section, we would like to provide additional insights on other related

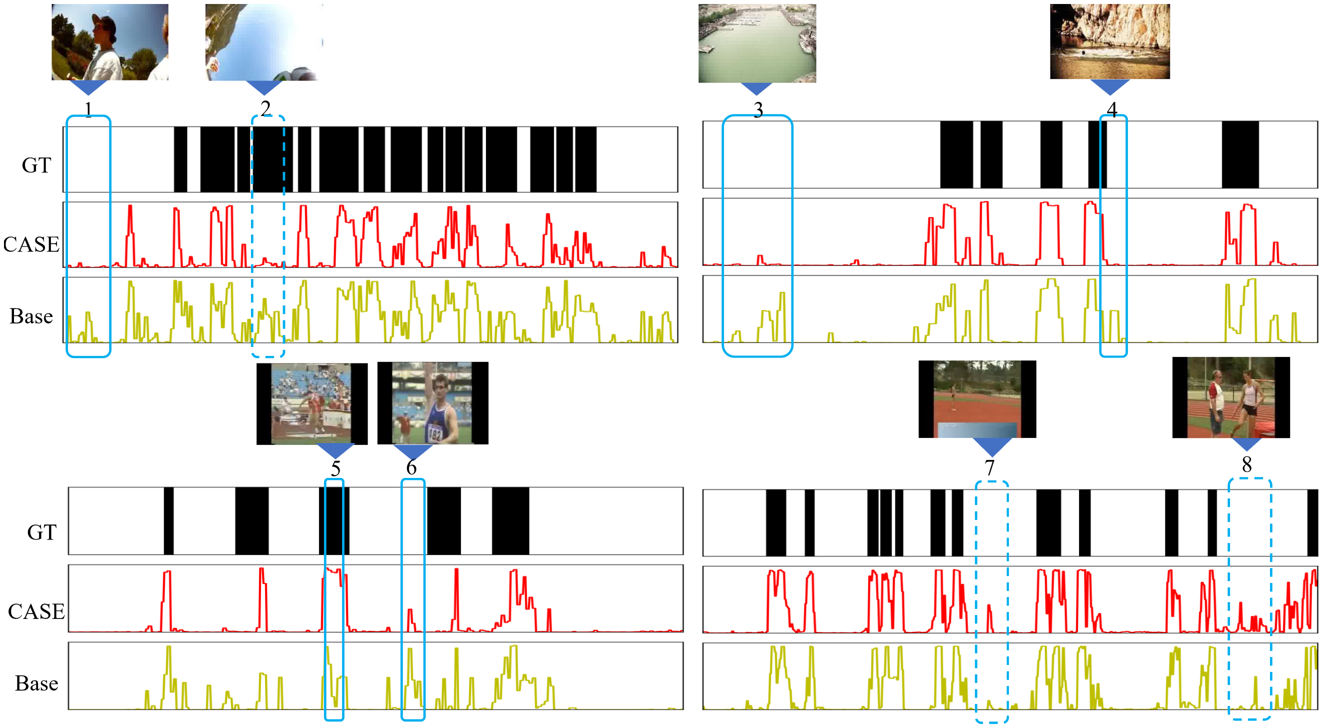

Figure 2: Comparison between our CASE and the baseline. The solid and dashed boxes represent the regions where CASE outperforms and underperforms the baseline, respectively.

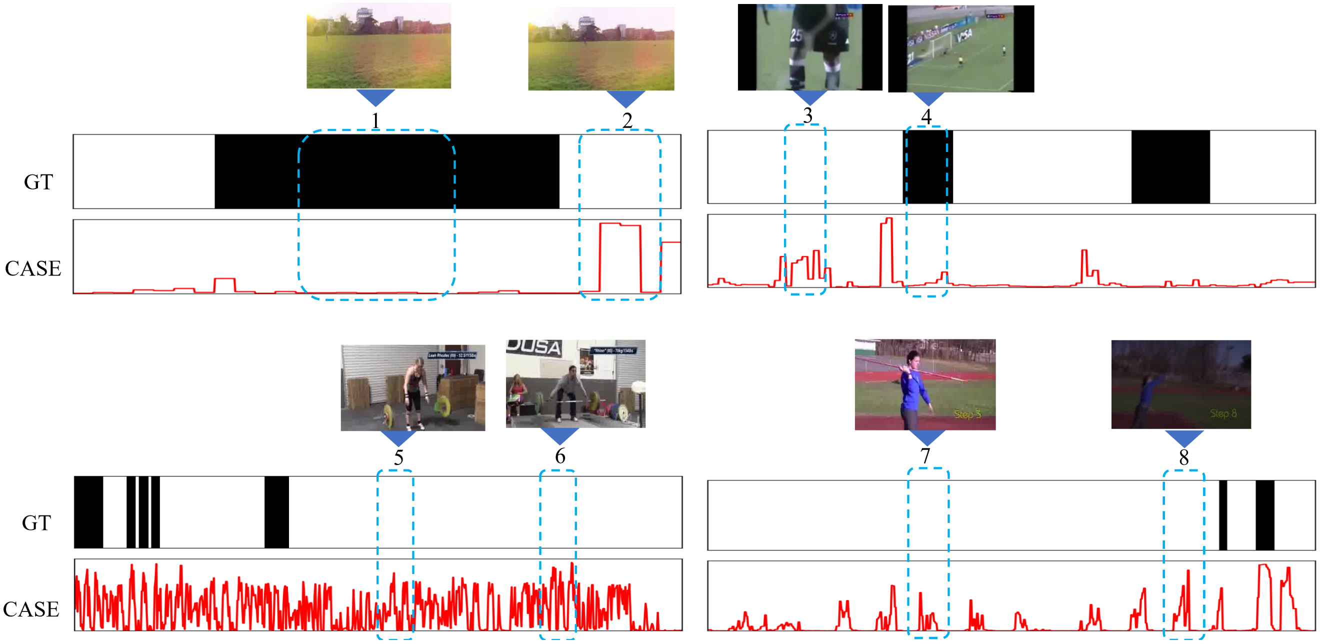

Figure 3: Samples of failure cases. The dashed boxes represent the regions with wrong predictions.

methods.

In the snippet clustering component (SCC), we draw inspiration from the early sequence-matching method [13] to construct a prior distribution for the pseudo-labels of cluster assignments of snippets. However, our method differs significantly from [13] in both purpose and solution.

Specifically, [13] aims to measure the distance between two sequences by matching the frames of one sequence with the frames of another sequence with similar temporal positions. Consequently, it constructs a prior distribution for the mapping between the frames in different sequences based on their temporal locations. In contrast, we aim to disambiguate the assignments between snippets and clusters by enforcing the snippets with high foreground/background probabilities to be assigned to the clusters with high foreground/background probabilities. To achieve this, we construct the prior distribution for the cluster assignments of the snippets based on the distance between the foreground probabilities of snippets and the foreground probabilities of clusters. Besides, to better suit our approach, we rank the snippets according to their foreground probabilities, resulting in ranking indices that are more comparable with the foreground probabilities of the clusters. This approach allows us to better match the snippets to the appropriate clusters, as demonstrated in Appendix C.3.

Furthermore, our method is somewhat related to context-based methods [10, 8]. Previous context-based methods [10, 8] typically regard the context as a special type of background. That is, they divide foreground and background snippets into *three latent groups*: action, context, and normal background. This approach provides a more detailed description of the background distribution. Our method extends this approach by dividing snippets into *multiple latent groups*, which allows for a more comprehensive description of both the foreground and background distributions. The visualized results (*e.g.*, 5th row of Fig.6 in the main paper) reveal that some of the learned clusters are very close to the concept of context. From this view, our model already has some contextual modeling capabilities.

F. Implementation Details

F.1. Baseline model

Here we present more details about the multiple instance learning (MIL) used in the baseline. Specifically, we first calibrate T-CAS $\boldsymbol{P}^{\boldsymbol{V}} \in \mathbb{R}^{T\times G}$ with the attention weights $\boldsymbol{P}^{\boldsymbol{A}} \in \mathbb{R}^{T}$ to highlight foreground snippets and suppress background snippets, resulting in the calibrated T-CAS (dubbed $\hat{\boldsymbol{P}}^{\boldsymbol{V}} \in \mathbb{R}^{T\times G}$). It can be implemented in multiple ways. Here following [9, 10], we fuse the scores by weighted summation, $\hat{\boldsymbol{P}}^{\boldsymbol{V}} = \omega \boldsymbol{P}^{\boldsymbol{V}} + (1-\omega)\boldsymbol{P}^{\boldsymbol{A}}$. ω is a predefined weight. Thereafter, we select K snippets from each video for each class based on $\hat{\boldsymbol{P}}^{\boldsymbol{V}}$:

$$\Gamma_c = \arg \max_{\substack{\Gamma\subset\{1,..,T\}\\ |\Gamma|=K}} \sum_{\tau\in\Gamma} \hat{\boldsymbol{P}}^{\boldsymbol{V}}_{\tau,c}, \tag{1}$$

where K is a hyper-parameter. Temporal pooling is applied to the selected snippets in Γ_c to build video-level class prediction $\bar{\boldsymbol{P}} \in \mathbb{R}^G$:

$$\bar{\boldsymbol{P}}_c = \underset{c}{Softmax}(\frac{1}{K}\sum_{\tau\in\Gamma_c} \boldsymbol{P}^{\boldsymbol{V}}_{\tau,c}). \tag{2}$$

Finally, $\bar{\boldsymbol{P}}$ is used to compute a video classification loss, as shown in the main paper.

F.2. Co-labeling

In our framework, there are several procedures of pseudo-labeling that can be summarized with a unified formulation as $\boldsymbol{Q} = \Psi(\boldsymbol{P})$. Here $\boldsymbol{P}$ is the prediction of the model, Ψ is the function of generating pseudo-labels, $\boldsymbol{Q}$ is the pseudo-labels. To improve the quality of the pseudo-labels, following [15], we propose to apply the two-stream co-labeling (TSCL) strategy, which is model-agnostic and naturally compatible with our method. That is, we aggregate the predictions of RGB and optical-flow streams to generate the modality-sharing pseudo-labels, *i.e.*, $\boldsymbol{Q} = \Psi(0.5\boldsymbol{P}^{\text{RGB}} + 0.5\boldsymbol{P}^{\text{Flow}})$. To be specific, for $\boldsymbol{Q}^{\boldsymbol{C}}$, we fuse the cluster assignments of RGB stream (dubbed $\boldsymbol{P}^{\boldsymbol{C},\text{RGB}}$) and that of Flow stream (dubbed $\boldsymbol{P}^{\boldsymbol{C},\text{Flow}}$) by:

$$\boldsymbol{P}^{\boldsymbol{C}} = 0.5\boldsymbol{P}^{\boldsymbol{C},\text{RGB}} + 0.5\boldsymbol{P}^{\boldsymbol{C},\text{Flow}}. \tag{3}$$

Then the pseudo-labels $\boldsymbol{Q}^{\boldsymbol{C}}$ is generated by:

$$\min_{\boldsymbol{Q}^{\boldsymbol{C}}\in\Omega^C} \langle \boldsymbol{Q}^{\boldsymbol{C}}, -\log \boldsymbol{P}^{\boldsymbol{C}}\rangle. \tag{4}$$

As for $\boldsymbol{Q}^{\boldsymbol{R}}$, the prediction of cluster classifier of RGB stream (dubbed $\boldsymbol{P}^{\boldsymbol{R},\text{RGB}}$) and that of Flow stream (dubbed $\boldsymbol{P}^{\boldsymbol{R},\text{Flow}}$) are combined as follows

$$\boldsymbol{P}^{\boldsymbol{R}} = 0.5\boldsymbol{P}^{\boldsymbol{R},\text{RGB}} + 0.5\boldsymbol{P}^{\boldsymbol{R},\text{Flow}}. \tag{5}$$

Then the pseudo-labels $\boldsymbol{Q}^{\boldsymbol{R}}$ is generated by:

$$\min_{\boldsymbol{Q}^{\boldsymbol{R}}\in\Omega^R} \langle \boldsymbol{Q}^{\boldsymbol{R}}, -\log \boldsymbol{P}^{\boldsymbol{R}}\rangle. \tag{6}$$

Moreover, the top-K selection used in Eq. (1) can be regarded as a procedure of defining the F&B snippets. Hence, we utilize the TSCL to improve the quality of the top-K selection. Specifically, we fuse the calibrated T-CAS of RGB stream (dubbed $\hat{\boldsymbol{P}}^{\boldsymbol{V},\text{RGB}}$) and that of optical-flow stream (dubbed $\hat{\boldsymbol{P}}^{\boldsymbol{V},\text{Flow}}$) as follows:

$$\hat{\boldsymbol{P}}^{\boldsymbol{V}} = 0.5\hat{\boldsymbol{P}}^{\boldsymbol{V},\text{RGB}} + 0.5\hat{\boldsymbol{P}}^{\boldsymbol{V},\text{Flow}}. \tag{7}$$

Then $\hat{\boldsymbol{P}}^{\boldsymbol{V}}$ is used for top-K selection. Notably, the results of the top-K selection also influences the definition of $\boldsymbol{Q}^{\boldsymbol{A}}$. we use Eq. (1) to determine the foreground and background snippets. , which can influence the learning of both the video classification module and attention module.

In Table 5, we present an evaluation of the effect of the two-stream co-labeling strategy on both the baseline model and our clustering-based F&B algorithm. It can be seen that the TSCL is important to the baseline, boosting its performance from 38.3% to 42.1%. However, the additional use of the TSCL in our algorithm results in only a small improvement compared to not using the TSCL in our algorithm (from 45.6% to 46.2%). This suggests that the main reason for the performance improvement of our algorithm over the baseline model is our proposed clustering-based approach, rather than the two-stream co-labeling strategy.

Method	mAP
Baseline *w/o* TSCL	38.3
Baseline *w/* TSCL	42.1
Baseline *w/* TSCL + Our algorithm *w/o* TSCL	45.6
Baseline *w/* TSCL + Our algorithm *w/* TSCL	46.2

Table 5: Ablation study of two-stream co-labeling (TSCL).

F.3. Training details

TVL1 [14] is applied to extract optical-flow stream from RGB stream in advance. Each stream is divided into 16-frame snippets. Following convention, we employ the I3D [3] network pre-trained on Kinetics-400 [3] to extract snippet-level features from each stream, where the channel dimension D is 1024. The number of sampled snippets T is set to 750 for THUMOS14 and 50 for ActivityNet v1.2 and v1.3. Both streams share the same structure but have separate parameters. The embedding encoders are comprised of a temporal convolution layer with 512 channels and a ReLU layer. The action classifier consists of a FC layer and a $\mathrm{Softmax}$ layer. The clustering head is composed of a linear cosine classifier [6] with a temperature of 10 and a $\mathrm{Softmax}$ layer. The attention layer consists of a FC layer and a $\mathrm{Sigmoid}$ layer. We set the classes K of the clustering head to 16 for THUMOS14 and 64 for ActivityNet v1.2 and v1.3. Following previous methods [10, 9], the k for top-k selection is set to $T//8$ in THUMOS14 and $T//2$ in ActivityNet v1.2 and v1.3, while the batch size B is set to 16, the γ is set to 0.7 and the ω is set to 0.25 for all datasets. Following [2], the ϵ is set to 20. The temperature ρ is set to 10. The standard deviation σ is set to 10. The loss weights are set as $\lambda_S = 1, \lambda_C = 0.3$ for all datasets. We utilize Adam optimizer with a learning rate of 10^{-4} for all datasets. We run each experiment three times and report their mean accuracy for reliability. The model implemented by Pytorch is trained on a Nvidia 1080Ti GPU.

F.4. Testing details

During inference, the video-level scores and snippet-level scores (*i.e.*, T-CAS) of both the RGB stream and optical-flow stream are fused by averaging. Then, a threshold is applied to the video-level scores to determine the action categories. For the selected action class, a threshold strategy is applied to the T-CAS, as done in [10, 5], to obtain action proposals. Next, the outer-inner-contrastive technique [11] is used to calculate the class-specific score for each proposal. To increase the pool of proposals, multiple thresholds are applied, and non-maximum suppression (NMS) is employed to remove duplicate proposals.

For multi-scale testing, following [7], we first rescale the input sequences to different scales $[1, 1.25, 1.5, 2]$, and then feed them into the model to generate action proposals. These proposals are then combined and subjected to NMS to obtain the final action detections.

G. Theoretical Derivation

Here we provide the derivation of the solution to the following optimal-transport problem in SCC:

$$\begin{aligned}&\min \langle \boldsymbol{Q^S}, -\log \boldsymbol{P^S}\rangle + \frac{1}{\epsilon}\mathrm{KL}(\boldsymbol{Q^S}, \boldsymbol{\hat{Q}^S}) \quad \textit{s.t.}, \boldsymbol{Q^S} \in \Omega^S \\ &\Omega^S = \{\boldsymbol{Q^S} \in \mathbb{R}_+^{N\times K} | \boldsymbol{Q^S}\boldsymbol{1}^K = \boldsymbol{\alpha^S}, \boldsymbol{Q^S}^\top \boldsymbol{1}^N = \boldsymbol{\beta^S}\}.\end{aligned} \tag{8}$$

For notation simplicity, we remove the superscript S. Then the problem is rewritten as

$$\begin{aligned}&\min \langle \boldsymbol{Q}, -\log \boldsymbol{P}\rangle + \frac{1}{\epsilon}\mathrm{KL}(\boldsymbol{Q}||\boldsymbol{\hat{Q}}) \quad \textit{s.t.}, \boldsymbol{Q} \in \Omega \\ &\Omega = \{\boldsymbol{Q} \in \mathbb{R}_+^{N\times K} | \boldsymbol{Q}\boldsymbol{1}^K = \boldsymbol{\alpha}, \boldsymbol{Q}^\top \boldsymbol{1}^N = \boldsymbol{\beta}\}.\end{aligned} \tag{9}$$

To address the problem, we first write the Lagrangian function of Eq. (9) as follows:

$$\begin{aligned}\mathcal{L}(\boldsymbol{Q}, \boldsymbol{\mu}, \boldsymbol{\nu}) =& \langle \boldsymbol{Q}, -\log \boldsymbol{P}\rangle + \frac{1}{\epsilon}\mathrm{KL}(\boldsymbol{Q}||\boldsymbol{\hat{Q}}) \\ &+ \boldsymbol{\mu}^\top(\boldsymbol{Q}\boldsymbol{1}^K - \boldsymbol{\alpha}) + \boldsymbol{\nu}^\top(\boldsymbol{Q}^\top\boldsymbol{1}^N - \boldsymbol{\beta}) \\ =& \sum_{n=1}^{N}\sum_{k=1}^{K}(-\boldsymbol{Q}_{n,k}\log \boldsymbol{P}_{n,k} + \frac{1}{\epsilon}\boldsymbol{Q}_{n,k}\log\frac{\boldsymbol{Q}_{n,k}}{\boldsymbol{\hat{Q}}_{n,k}} \\ &+ \boldsymbol{\mu}_n\boldsymbol{Q}_{n,k} + \boldsymbol{\nu}_k\boldsymbol{Q}_{n,k}) - \boldsymbol{\mu}^\top\boldsymbol{\alpha} - \boldsymbol{\nu}^\top\boldsymbol{\beta}\end{aligned} \tag{10}$$

where $\boldsymbol{\mu} \in \mathbb{R}^N$ and $\boldsymbol{\nu} \in \mathbb{R}^K$ are the dual variables so that $\boldsymbol{Q}\boldsymbol{1}^K = \boldsymbol{\alpha}$ and $\boldsymbol{Q}^\top\boldsymbol{1}^N = \boldsymbol{\beta}$. The derivative of $\mathcal{L}(\boldsymbol{Q}, \boldsymbol{\mu}, \boldsymbol{\nu})$ *w.r.t.* $\boldsymbol{Q}_{n,k}$ is:

$$\frac{\partial\mathcal{L}(\boldsymbol{Q}, \boldsymbol{\mu}, \boldsymbol{\nu})}{\partial\boldsymbol{Q}_{n,k}} = -\log \boldsymbol{P}_{n,k} + \frac{1}{\epsilon}\log\frac{\boldsymbol{Q}_{n,k}}{\boldsymbol{\hat{Q}}_{n,k}} + \frac{1}{\epsilon} + \boldsymbol{\mu}_n + \boldsymbol{\nu}_k. \tag{11}$$

Note that the optimal $\boldsymbol{Q}$ exists and is unique, as both the objective and the constraint in Eq. (9) are convex. Hence, to obtain the optimal $\boldsymbol{Q}$, we set $\frac{\partial\mathcal{L}(\boldsymbol{Q}, \boldsymbol{\mu}, \boldsymbol{\nu})}{\partial\boldsymbol{Q}_{n,k}} = 0$, and then get:

$$\boldsymbol{Q}_{n,k} = e^{-\frac{1}{2}-\epsilon\boldsymbol{\mu}_n-\frac{1}{2}}(\boldsymbol{\hat{Q}}_{n,k}\boldsymbol{P}_{n,k}^{\epsilon})e^{-\frac{1}{2}-\epsilon\boldsymbol{\nu}_k}. \tag{12}$$

Let us denote $\boldsymbol{S} = \boldsymbol{\hat{Q}} \cdot \boldsymbol{P}^\epsilon$. Obviously, all elements of $\boldsymbol{S}$ are strictly positive. According to [12, 1, 13], there exist diagonal matrices $\mathrm{diag}(\boldsymbol{u})$ and $\mathrm{diag}(\boldsymbol{v})$ with strictly positive diagonal elements so that $\mathrm{diag}(\boldsymbol{u})S\,\mathrm{diag}(\boldsymbol{v})$ belongs to Ω.

In summary, the optimal $\boldsymbol{Q}$ has the form as:

$$\boldsymbol{Q} = \mathrm{diag}(\boldsymbol{u})\boldsymbol{S}\,\mathrm{diag}(\boldsymbol{v}) = \mathrm{diag}(\boldsymbol{u})(\hat{\boldsymbol{Q}} \cdot \boldsymbol{P}^{\epsilon})\,\mathrm{diag}(\boldsymbol{v}), \tag{13}$$

where $\boldsymbol{u} \in \mathbb{R}^N$ and $\boldsymbol{v} \in \mathbb{R}^K$ are two renormalization vectors that make the resulting matrix $\boldsymbol{Q}$ to be a probability matrix. Throughout our work, we follow [2] to implement the algorithm due to its conciseness. Formally, Eq. (13) is replaced as follows:

$$\boldsymbol{Q} = \mathrm{diag}(\boldsymbol{u})(\hat{\boldsymbol{Q}} \cdot \exp(\epsilon\boldsymbol{L}))\,\mathrm{diag}(\boldsymbol{v}), \tag{14}$$

where $\boldsymbol{L}$ indicates the logits before the $\mathrm{Softmax}$ layer, namely $\boldsymbol{P} = \mathrm{Softmax}(\boldsymbol{L})$. Note that Eq. (13) and Eq. (14) are equivalent in principle. The main difference lies in the placement of the factor ϵ that sharpens the labels. In Eq. (13), the factor is applied before $\mathrm{Softmax}$, while in Eq. (14), it is applied after $\mathrm{Softmax}$. Similarly, for the cluster classification component (without $\hat{\boldsymbol{Q}}$), we can obtain the solution as follows:

$$\boldsymbol{Q} = \mathrm{diag}(\boldsymbol{u})\exp(\epsilon\boldsymbol{L})\,\mathrm{diag}(\boldsymbol{v}). \tag{15}$$

Both Eq. (14) and Eq. (15) can be efficiently computed using the iterative Sinkhorn-Knopp algorithm [4]. We refer to [4] for more details. This algorithm is highly efficient on GPU as it only involves a couple of matrix multiplication, enabling online computation.

References

[1] Alberto Borobia and Rafael Cantó. Matrix scaling: A geometric proof of sinkhorn's theorem. *Linear algebra and its applications*, 268:1–8, 1998. 5

[2] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 2020. 5, 6

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 5

[4] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*, 2013. 6

[5] Junyu Gao, Mengyuan Chen, and Changsheng Xu. Fine-grained temporal contrastive learning for weakly-supervised temporal action localization. In *CVPR*, 2022. 5

[6] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, 2018. 5

[7] Qinying Liu, Zilei Wang, Ruoxi Chen, and Zhilin Li. Convex combination consistency between neighbors for weakly-supervised action localization. *arXiv preprint arXiv:2205.00400*, 2022. 5

[8] Ziyi Liu, Le Wang, Wei Tang, Junsong Yuan, Nanning Zheng, and Gang Hua. Weakly supervised temporal action localization through learning explicit subspaces for action and context. In *AAAI*, 2021. 4

[9] Junwei Ma, Satya Krishna Gorti, Maksims Volkovs, and Guangwei Yu. Weakly supervised action selection learning in video. In *CVPR*, 2021. 4, 5

[10] 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. 4, 5

[11] Zheng Shou, Hang Gao, Lei Zhang, Kazuyuki Miyazawa, and Shih-Fu Chang. Autoloc: Weakly-supervised temporal action localization in untrimmed videos. In *ECCV*, 2018. 5

[12] Richard Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *The American Mathematical Monthly*, 74(4):402–405, 1967. 5

[13] Bing Su and Gang Hua. Order-preserving wasserstein distance for sequence matching. In *CVPR*, 2017. 3, 4, 5

[14] C Zach, T Pock, and H Bischof. A duality based approach for realtime tv-l 1 optical flow. *Pattern Recognition*, 2007. 5

[15] 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. 4