Learning Disentangled Classification and Localization Representations for Temporal Action Localization

Zixin Zhu[1], Le Wang[1]*, Wei Tang[2], Ziyi Liu[3], Nanning Zheng[1], Gang Hua[3]

[1]Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
[2]University of Illinois at Chicago
[3]Wormpex AI Research
zhuzixin@stu.xjtu.edu.cn, {lewang,nnzheng}@xjtu.edu.cn, tangw@uic.edu, iair_lzy@foxmail.com, ganghua@gmail.com

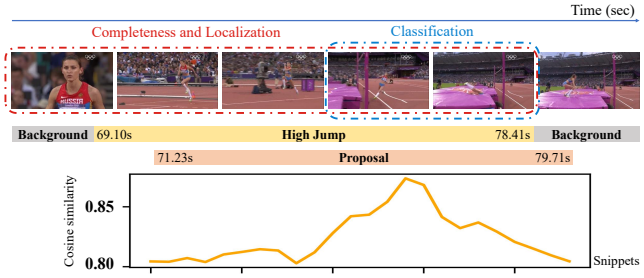
Abstract

A common approach to Temporal Action Localization (TAL) is to generate action proposals and then perform action classification and localization on them. For each proposal, existing methods universally use a shared proposal-level representation for both tasks. However, our analysis indicates that this shared representation focuses on the most discriminative frames for classification, *e.g.*, "take-offs" rather than "run-ups" in distinguishing "high jump" and "long jump", while frames most relevant to localization, such as the start and end frames of an action, are largely ignored. In other words, such a shared representation can not simultaneously handle both classification and localization tasks well, and it makes precise TAL difficult. To address this challenge, this paper disentangles the shared representation into classification and localization representations. The disentangled classification representation focuses on the most discriminative frames, and the disentangled localization representation focuses on the action phase as well as the action start and end. Our model can be divided into two sub-networks, *i.e.*, the disentanglement network and the context-based aggregation network. The disentanglement network is an autoencoder to learn orthogonal hidden variables of classification and localization. The context-based aggregation network aggregates the classification and localization representations by modeling local and global contexts. We evaluate our proposed method on two popular benchmarks for TAL, which outperforms all state-of-the-art methods.

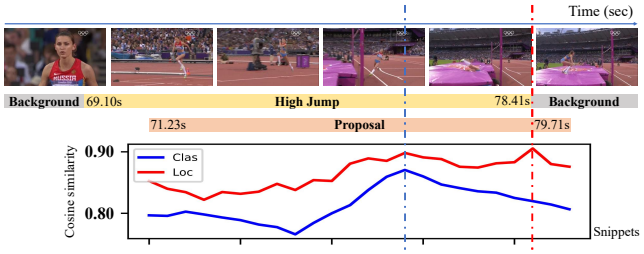
Introduction

Temporal Action Localization (TAL) aims to seek out actions of interest in a video and locate the temporal start and end of each action instance. It is inherently a heterogeneous multi-task learning problem, with action classification and action localization as different tasks. Recently, TAL has received more attention as a fundamental tool for several applications such as video summarization (Xiao et al. 2020), action recognition (Meng et al. 2020), and video captioning (Wang et al. 2018).

Inspired by the great success of the proposal-based object detection framework, *i.e.*, object proposal generation followed by proposal classification and boundary regression, many current TAL methods (Huang et al. 2018; Bai et al. 2020; Wang et al. 2021; Liu et al. 2021; Li and Yao 2021; Li et al. 2020) have followed this pipeline. They first generate category-agnostic action proposals (Lin et al. 2019, 2018;



(a) Attention of the original shared proposal features to different snippets.



(b) Attentions of our disentangled classification and localization features to different snippets.

Figure 1: We visualize the cosine similarity between proposal features and features of snippets within the corresponding proposal. We assume that if the proposal features are more similar to the feature of a snippet, it indicates the proposal features pay more attention to that temporal segment corresponding to the snippet. (a) The original shared proposal features (the orange line) focus on the most discriminative frames for classification. Therefore, it is not the best for judging completeness and locating the action. (b) Our localization features (the red line) focus on the start and end moments of the action and all snippets within the action, which are useful for completeness scoring and localization. Besides, our classification features (the blue line) also focus on the most discriminate frames.

*Corresponding author.

Gao, Chen, and Nevatia 2018; Liu et al. 2019; Lin et al. 2020; Zhao et al. 2020; Qing et al. 2021). Subsequently, action classification and temporal boundary refinement are performed for each proposal, taking the shared proposal-level features as input for both tasks (Zeng et al. 2019; Wang et al. 2017; Xu, Das, and Saenko 2017; Chao et al. 2018; Dai et al. 2017).

However, a shared proposal-level representation cannot account for both classification and localization well. As shown in Figure 1(a), it focuses on "take-offs" for the action "high jump" rather than "run-ups". As a result, this shared representation is more beneficial for classification, since some other actions such as "long jump" also include "run-ups", making "run-ups" a distraction for classification. But this representation is not suitable for judging the completeness and locating the action. Judging completeness requires the consideration of both "run-ups" and "take-offs" and locating the action requires the consideration of its start and end moments.

Therefore, it is critical to generate apposite proposal-level representations for classification and localization tasks, respectively. This paper addresses this challenge by disentangling the original shared representation of a proposal into two representations respectively for classification and localization. Our disentangled representations are shown in Figure 1(b). In addition to considering both "run-ups" and "take-offs" more evenly, our localization representation focuses on the snippets capturing the start and end frames of "high jump", indicating that they are more appropriate for the localization task. Besides, our classification representation still focuses on the most discriminative frames ("take-offs") of "high jump".

Our method consists of two sub-networks, *i.e.*, the disentanglement network and the context-based aggregation network (Zhu et al. 2021). The disentanglement network is an autoencoder to divide original snippet-level features into three parts: the unique classification part, the unique localization part, and the common part. Specially, the unique classification and localization parts are orthogonal. Afterwards, the disentangled classification features and localization features are obtained by decoding the corresponding unique classification part and unique localization part, respectively. Subsequently, the context-based aggregation network aggregates and enhances our disentangled proposal-level representations via fine-grained modeling of snippet-level representations and higher-level modeling of the video-level representation. After obtaining the final disentangled proposal-level representations, the classification and localization tasks are performed on their corresponding features instead of using shared features.

In summary, our contributions lie in three folds:

- Previous proposal-based TAL approaches take it for granted that proposal-level features should be shared for both classification and localization. To our knowledge, we are the first to identify the problem of this practice: the shared representation can not handle both tasks well.
- We design a novel disentanglement network to obtain representations suitable for classification and representations suitable for localization, respectively. This is achieved by learning disentangled classification parts and localization parts, which are orthogonal to each other, from the original shared proposal-level features.
- Our model achieves the state-of-the-art performance on two popular TAL benchmarks, *i.e.*, THUMOS14 (Jiang et al. 2014) and ActivityNet v1.3 (Heilbron et al. 2015).

Related Work

Action Recognition. Action recognition is essentially a classification task in video understanding and has been widely studied in the past few years. Two-stream networks (Simonyan and Zisserman 2014; Shi et al. 2019; Zhu et al. 2018) as a type of popular methods adopt RGB frames and optical flows to capture appearance and motion information. Specially, Simonyan *et al.* (Simonyan and Zisserman 2014) fuse the predictions made on RGB frames and optical flows. We also adopt this strategy in our model.

There are a few methods exploring the feature disentanglement for action recognition. Liu *et al.* (Liu et al. 2015) disentangle the features of similar actions (fencing, sword, draw sword) and let them contribute to the maximum extent. Liu *et al.* (Liu et al. 2020) disentangle the skeleton node features at separate spatial-temporal neighborhoods in graph convolutions. Their disentanglement aims to obtain better representations for one task while our disentanglement is to obtain eligible representations for different tasks.

Temporal Action Localization. Prior TAL methods can be divided into two categories. One-stage approaches (Huang, Dai, and Lu 2019; Long et al. 2019; Liu and Wang 2020; Lin et al. 2021) classify and locate action instances from an input video in a single shot. Their advantage is that they can be easily trained in an end-to-end fashion.

Two-stage approaches (Gao et al. 2017; Bai et al. 2020) as described in Section usually obtain superior performance. Some approaches focus on the proposal generation. For example, Gao *et al.* (Gao et al. 2020) capture global contextual information and simultaneously detect actions with different durations. Lin *et al.* (Lin et al. 2018) locate temporal boundaries with high probabilities, then directly combine these boundaries into proposals and predict whether a proposal contains an action. Some other works focus on proposal classification and localization. For example, Li *et al.* (Li and Yao 2021) design two auxiliary tasks by reconstructing the available label information and then facilitate the learning of the temporal action detection model. They focus on network construction and context modeling in videos.

Huang *et al.* (Huang, Dai, and Lu 2019) design two branches to learn representations separately for localization or classification. In contrast, we explicitly disentangle the original shared representation of a proposal into a unique classification part, a unique localization part and a common part, and then carefully recompose them into two representations respectively suitable for localization and classification. To our knowledge, feature disentanglement has never been studied in TAL.

Disentanglement in Videos. Disentanglement networks are widely used in various video understanding tasks but

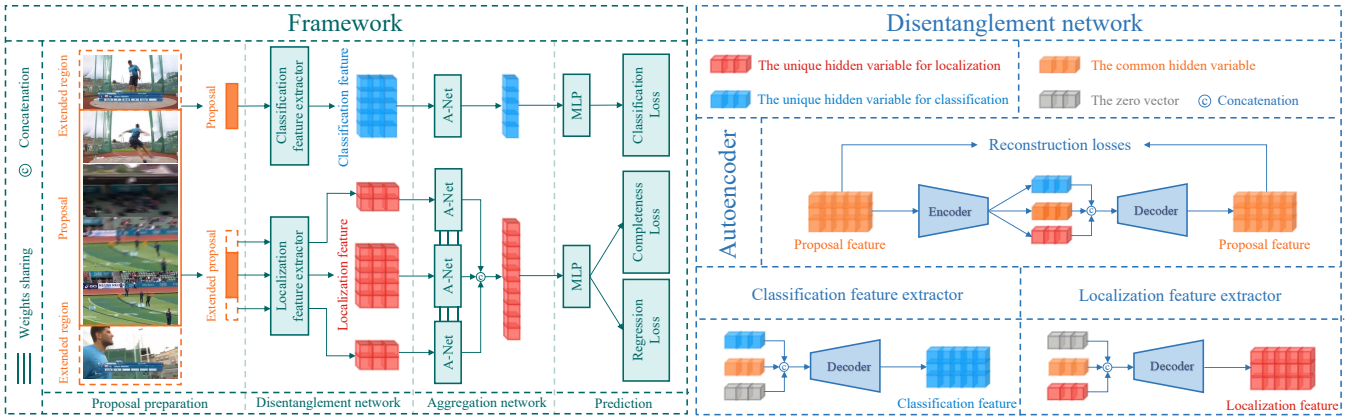


Figure 2: The framework of the proposed method (left) and the structure of the disentanglement network (right). Left: Taking an untrimmed video consisting of non-overlapping snippets as the input, snippet-level features are first extracted. Afterwards, the features of snippets within a proposal are disentangled to classification and localization features by corresponding feature extractors. The aggregation network (A-Net) aggregates and enhances these disentangled features to obtain better proposal-level features. Finally, instead of using shared features for both classification and localization, different features are fed into corresponding prediction layers. Right: The disentanglement network leverages an autoencoder structure to divide original features into three components. By substituting the localization (or classification) component with zero vector, the learned decoder outputs the classification (or localization) features.

not in TAL, such as video facial authentication (Kim et al. 2020), video representation learning (Denton and Birodkar 2017) and video generation (Wang et al. 2020). Common disentanglement networks are divided into Generative Adversarial Networks (GANs) (Sun, Xu, and Saenko 2020) and Autoencoders (Bhagat et al. 2020; Ding et al. 2020).

Specially, Bhagat *et al.* (Bhagat et al. 2020) use Gaussian processes to model the latent space for the unsupervised learning of disentangled representations in video sequences. Ding *et al.* (Ding et al. 2020) use an adversarial excitation and inhibition mechanism to encourage the disentanglement of the latent variables. Different form the prior disentanglement networks based on Autoencoders, we replace the unrelated latent variables with zero vectors in latent space to implement the disentanglement.

Our Approach

Framework

The proposed framework is presented in Figure 2. The input untrimmed video is denoted as $V = \{s_t\}_{t=1}^{T}$, where s_t is a video snippet consisting of a small number of consecutive frames and T denotes the number of non-overlapping snippets. The snippet-level features $\{\mathbf{x}_t \in \mathbb{R}^{D\times 1}\}_{t=1}^{T}$ are extracted snippet-by-snippet, where $\mathbf{x}_t$ denotes the features of s_t, and D is the feature dimension. Assuming V contains N action proposals, denoting as $\mathbf{P} = \{\mathbf{p}_i \mid \mathbf{p}_i = (t_{i,s}, t_{i,e})\}_{i=1}^{N}$, the i-th proposal $\mathbf{p}_i$ is parameterized by its start time $t_{i,s}$ and end time $t_{i,e}$.

For $\mathbf{p}_i$, its features $\mathbf{Y}_i \in \mathbb{R}^{D\times K_i}$ are obtained by concatenating features of K_i snippets, where K_i denotes the number of snippets within $\mathbf{p}_i$. $\mathbf{Y}_i$ is disentangled to the classification features $\mathbf{F}_i^{cls} \in \mathbb{R}^{D\times K_i}$ and the localization features $\mathbf{F}_i^{loc} \in \mathbb{R}^{D\times K_i}$ by the disentanglement network. Afterwards, $\mathbf{F}_i^{cls}$ and $\mathbf{F}_i^{loc}$ are fed to different context-based aggregation networks (A-Nets) to obtain the aggregated features, denoted as $\mathbf{f}_i^{cls} \in \mathbb{R}^D$ and $\mathbf{f}_i^{loc} \in \mathbb{R}^D$, respectively.

To better consider the temporal contextual information for localization, we follow the common practice in TAL (Shou et al. 2017; Zeng et al. 2019; Lin et al. 2018, 2020) to use the extended proposal to achieve localization predictions. We extend $\mathbf{p}_i$ on both ends by 50% of its temporal duration. The extended regions on the two sides are also treated as two proposals to obtain extended features $\mathbf{Y}_i^L \in \mathbb{R}^{D\times(K_i/2)}$ and $\mathbf{Y}_i^R \in \mathbb{R}^{D\times(K_i/2)}$ on the left and right extended regions, respectively. Afterwards, the corresponding localization features $\mathbf{f}_i^{L,loc} \in \mathbb{R}^D$ and $\mathbf{f}_i^{R,loc} \in \mathbb{R}^D$ are obtained by feeding $\mathbf{Y}_i^L \in \mathbb{R}^{D\times(K_i/2)}$ and $\mathbf{Y}_i^R \in \mathbb{R}^{D\times(K_i/2)}$ into the localization feature extractor and the A-Net. The final localization features of the i-th proposal $\mathbf{p}_i$ are obtained by concatenating $\mathbf{f}_i^{loc}$, $\mathbf{f}_i^{L,loc}$ and $\mathbf{f}_i^{R,loc}$.

After obtaining the final classification features ($\mathbf{f}_i^{cls}$) and localization features, the classification and localization predictions are achieved by feeding them to Multilayer Perceptrons (MLPs).

The Disentanglement Network

According to our discussion in Section , the representations that are most suitable for classification and localization are different. For classification, the representation should focus on the most discriminative features of the action phase that are relevant to the action category, *e.g.*, the "take-off" motion to distinguish "high jump" and "long jump", and the background "pool" to classify the action as "diving" instead of "gymnastics". For localization, the representation is sup-

posed to reflect the completeness and extent of an action, which include the action phase and the action start and end. Note features of action start and end alone are not sufficient for localization as action boundaries are often ambiguous, and the action phase can help reduce this ambiguity and also infer the action completeness.

The analysis above motivates us to divide the representation of a proposal into three parts: the unique classification part, the unique localization part, and the common part. The unique classification part includes features that are suitable for classification but not localization, *e.g.*, the background context shared by both action and non-action frames. The unique localization part includes features that are suitable for localization but not classification, *e.g.*, the action start and end. The common part include features that are suitable for both tasks, *e.g.*, discriminative features of the action phase.

As shown in Figure 2 (right), our disentanglement network is divided into an autoencoder, a classification feature extractor, and a localization feature extractor. Instead of performing explicit disentanglement, we implicitly disentangle features via different tasks. In other words, the network learns the representations suitable for different tasks by itself.

Autoencoder. We divide a proposal-level representation $\mathbf{Y}_i$ into three parts: the unique classification part, the unique localization part and the common part. We use the autoencoder to learn the hidden variables corresponding to these three parts. These hidden variables can be formulated by

$$\mathbf{H}_i^{cls} = \mathbf{W}_e^{cls}\mathbf{Y}_i, \mathbf{H}_i^{loc} = \mathbf{W}_e^{loc}\mathbf{Y}_i, \mathbf{H}_i^{com} = \mathbf{W}_e^{com}\mathbf{Y}_i, \tag{1}$$

where $\{\mathbf{H}_i^{cls}, \mathbf{H}_i^{loc}, \mathbf{H}_i^{com}\} \in \mathbb{R}^{D_h \times K_i}$ are the hidden variables of classification, localization and common parts, respectively. D_h is the feature dimension of hidden variables. $\{\mathbf{W}_e^{cls}, \mathbf{W}_e^{loc}, \mathbf{W}_e^{com}\} \in \mathbb{R}^{D_h \times D}$ are learnable weights. We hope that these hidden variables can successfully reconstruct the original proposal features$\mathbf{Y}_i$ through

$$\widehat{\mathbf{Y}}_i = \sigma(\mathbf{W}_d \cdot [\mathbf{H}_i^{cls}, \mathbf{H}_i^{com}, \mathbf{H}_i^{loc}]), \tag{2}$$

where $\widehat{\mathbf{Y}}_i \in \mathbb{R}^{D \times K_i}$ denotes the reconstructed proposal features, $[\cdot]$ denotes the concatenation, σ is the ReLU function, and $\mathbf{W}_d \in \mathbb{R}^{D \times 3D_h}$ is the learnable weights. We adopt the Mean Squared Error loss and the Cosine Similarity loss as the reconstruction losses:

$$\mathcal{L}_{rec} = \mathcal{L}_{\text{mse}}(\mathbf{Y}_i, \widehat{\mathbf{Y}}_i) + \frac{1}{K_i}\sum_{t=1}^{K_i}\mathcal{L}_{\cos}(\mathbf{Y}_i(t), \widehat{\mathbf{Y}}_i(t)), \tag{3}$$

where $\{\mathbf{Y}_i(t), \widehat{\mathbf{Y}}_i(t)\}$ denotes the t-th element in $\{\mathbf{Y}_i, \widehat{\mathbf{Y}}_i\}$.

Classification Feature Extractor. In order to weaken the interference of the unique localization part on the classification task (*e.g.*, the interference of "run-ups" to distinguish the "long jump" and the "high jump"), we use the zero matrix instead of the localization hidden variables when we decode the classification features. It is calculated as

$$\mathbf{F}_i^{cls} = \sigma(\mathbf{W}_d \cdot [\mathbf{H}_i^{cls}, \mathbf{H}_i^{com}, \mathbf{H}^{zero}]), \tag{4}$$

where $\mathbf{H}^{zero} \in \mathbb{R}^{D_h \times K_i}$ is a zero matrix and $\mathbf{F}_i^{cls} \in \mathbb{R}^{D \times K_i}$ denotes the disentangled classification features.

Localization Feature Extractor. In order to weaken the interference of the unique classification part on the localization task, we also adopt the zero matrix instead of classification hidden variables when we decode the localization features. It is calculated as

$$\mathbf{F}_i^{loc} = \sigma(\mathbf{W}_d \cdot [\mathbf{H}^{zero}, \mathbf{H}_i^{com}, \mathbf{H}_i^{loc}]), \tag{5}$$

where $\mathbf{F}_i^{loc} \in \mathbb{R}^{D \times K_i}$ denotes the disentangled localization features.

The Context-based Aggregation Network

Since the duration of each proposal is uncertain, we need to aggregate proposal-level features $\{\mathbf{F}_i^{cls}, \mathbf{F}_i^{loc}\} \in \mathbb{R}^{D \times K_i}$ of variable lengths into fixed-length features $\{\mathbf{f}_i^{cls}, \mathbf{f}_i^{loc}\} \in \mathbb{R}^D$. Inspired by Zhu *et al.* (Zhu et al. 2021), we use the context-based aggregation network to achieve the aggregation. Since the representations of classification and localization are aggregated in the same way, below we use $\mathbf{F}_i$ to represent either $\mathbf{F}_i^{cls}$ or $\mathbf{F}_i^{loc}$ and use $\mathbf{f}_i$ to represent either $\mathbf{f}_i^{cls}$ or $\mathbf{f}_i^{loc}$. Our aggregation process can be formulated as

$$\mathbf{f}_i^O = \text{Max-pooling}(\mathbf{F}_i), \tag{6}$$

$$\mathbf{a}_i(t) = cos(\mathbf{f}_i^O, \mathbf{F}_i(t)), \tag{7}$$

$$\mathbf{f}_{i,a} = \mathbf{W}_a^O\mathbf{f}_i^O + \mathbf{W}_a\mathbf{F}_i\mathbf{a}_i^T, \tag{8}$$

where $\mathbf{f}_i^O \in \mathbb{R}^D$ denotes the original proposal-level features. $\{\mathbf{a}_i(t), \mathbf{F}_i(t)\}$ denotes the t-th element in $\{\mathbf{a}_i, \mathbf{F}_i\}$. $\mathbf{a}_i \in \mathbb{R}^{1 \times K_i}$ denotes the attention vector. *cos* denotes the cosine similarity. $\mathbf{f}_{i,a} \in \mathbb{R}^{D/2}$ is the aggregated proposal-level features. $\{\mathbf{W}_a^O, \mathbf{W}_a\} \in \mathbb{R}^{(D/2) \times D}$ are learnable weights.

Our global context can be computed by

$$\mathbf{f}_g = \text{Max-pooling}(\{\mathbf{x}_t \in \mathbb{R}^D\}_{t=1}^T), \tag{9}$$

$$\mathbf{f}_{i,g} = \mathbf{W}_g\mathbf{f}_g + \text{Avg-pooling}(\mathbf{W}_l\mathbf{F}_i), \tag{10}$$

where $\mathbf{f}_g \in \mathbb{R}^D$ denotes the original global context. $\mathbf{f}_{i,g} \in \mathbb{R}^{D/2}$ denotes the global context adapted to $\mathbf{p}_i$. $\{\mathbf{W}_g, \mathbf{W}_l\} \in \mathbb{R}^{(D/2) \times D}$ are learnable weights capturing global and local contexts. We obtain the final proposal-level features $\mathbf{f}_i \in \mathbb{R}^D$ by

$$\mathbf{f}_i = [\mathbf{f}_{i,a}, \mathbf{f}_{i,g}]. \tag{11}$$

Prediction

With the disentanglement network and context-based aggregation network, we can obtain $\mathbf{f}_i^{cls}$ and $\mathbf{f}_i^{loc}$ from the $\mathbf{p}_i$. In addition, we can obtain $\mathbf{f}_i^{L,loc}$ and $\mathbf{f}_i^{R,loc}$ from the extended regions of it. The final predictions can be formulated as

$$\{\hat{c}_{i,m}\}_{m=0}^C = \mathcal{M}^{cls}(\mathbf{f}_i^{cls}), \tag{12}$$

$$\{\hat{z}_{i,m}\}_{m=1}^C = \mathcal{M}^{z}([\mathbf{f}_i^{L,loc}, \mathbf{f}_i^{loc}, \mathbf{f}_i^{R,loc}]), \tag{13}$$

$$(\hat{t}_{i,s}, \hat{t}_{i,e}) = \mathcal{M}^{loc}([\mathbf{f}_i^{L,loc}, \mathbf{f}_i^{loc}, \mathbf{f}_i^{R,loc}]). \tag{14}$$

Specially, $\mathcal{M}^{cls}$ is the classification head consisting of an MLP followed by a fully-connected layer. $\mathcal{M}^z$ and $\mathcal{M}^{loc}$ are heads for completeness prediction and temporal boundary regression, respectively. They share an MLP followed

by different fully-connected layers. $\{\hat{c}_{i,m}\}_{m=0}^{C}$ is the classification prediction, where $\hat{c}_{i,m}$ denotes the probability of the m-th action category and C is the number of action categories. Here, $m = 0$ represents the background category. $\{\hat{z}_{i,m}\}_{m=1}^{C}$ is the completeness prediction, where $\hat{z}_{i,m}$ denotes the completeness score of the m-th action category. $(\hat{t}_{i,s}, \hat{t}_{i,e})$ are the localization predictions, where $\hat{t}_{i,s}$ and $\hat{t}_{i,e}$ are the predicted start and end time of the action for $\mathbf{p}_i$. Below we write $\{\hat{c}_{i,m}\}_{m=0}^{C}$, $\{\hat{z}_{i,m}\}_{m=1}^{C}$ and $(\hat{t}_{i,s}, \hat{t}_{i,e})$ compactly as $\hat{\mathbf{c}}_i$, $\hat{\mathbf{z}}_i$ and $\hat{\mathbf{t}}_i$, respectively.

Loss Function

Our model not only predicts the action category but also refines the proposal's temporal boundary via regression. We use a multi-task loss function to train our autoencoder, action classifier, completeness classifier and boundary regressor. For $\mathbf{p}_i$, the multi-task loss can be defined by

$$\begin{aligned}\mathcal{L} = &\sum_i \mathcal{L}_{cls}\left(\mathbf{c}_i, \hat{\mathbf{c}}_i\right) + \lambda_1 \sum_i \mathcal{L}_{reg}\left(\mathbf{t}_i, \hat{\mathbf{t}}_i\right) \\ &+ \lambda_2 \sum_i \mathcal{L}_{com}\left(\mathbf{z}_i, \hat{\mathbf{z}}_i\right) + \mathcal{L}_{rec},\end{aligned} \tag{15}$$

where $\mathbf{c}_i$ is the ground truth action label of the i-th proposal, $\mathbf{z}_i$ is the completeness label and $\mathbf{t}_i$ is the start and end time of the action which is closest to the $\mathbf{p}_i$. The classification loss $\mathcal{L}_{cls}$ is the cross-entropy loss. The regression loss $\mathcal{L}_{reg}$ is the smooth L_1 loss. The completeness loss $\mathcal{L}_{com}$ is the hinge loss. In addition, we do not consider the completeness of the background proposals. In all experiments, we set $\lambda_1 = \lambda_2 = 0.5$.

Inference

For $\mathbf{p}_i$, we can obtain the classification scores $\{s_{i,m}^{cls}\}_{m=1}^{C} = \{\hat{c}_{i,m}\}_{m=1}^{C}$ by removing the score of the background category. Then we get the corresponding completeness scores $\{s_{i,m}^{com}\}_{m=1}^{C} = \exp(\{\hat{z}_{i,m}\}_{m=1}^{C})$. The two sets of scores are fused to obtain the final scores by element-wise multiplication. Concretely, for each proposal, the final confidence score of category m is computed as

$$s_{i,m}^{final} = s_{i,m}^{cls} \times s_{i,m}^{com}. \tag{16}$$

Moreover, we fuse the predictions including the final confidence scores and the regressed boundaries from the RGB and optical flow streams. With the scores and boundaries, we then use Non-Maximum Suppression (NMS) to obtain the final predicted temporal proposals for each action category separately.

Experiments

Datasets. The THUMOS14 (Jiang et al. 2014) dataset provides temporal annotations for 20 action categories. The videos of verification, background and test sets are untrimmed. Following the common setting in THUMOS14, we apply 200 videos (including 3,007 action instances) in the validation set for training and conduct evaluation on the 213 annotated videos (including 3,358 action instances) from the test set. ActivityNet v1.3 (Heilbron et al. 2015) is currently the largest dataset of action analysis in videos, including 20,000 Youtube videos with 200 action categories. The training set contains about 10,000 untrimmed videos. Both the validation set and the test set contain about 5,000 untrimmed videos. On average, each video has 1.5 action instances. Following the standard practice, we train our method on the training set and test it on the validation set.

Method	RGB	Flow	Fusion	#Params
Baseline	44.58	51.42	53.50	10.1M
+ MLP_1	46.73	52.15	54.70	11.7M
+ MLP_2	46.31	52.70	55.09	11.7M
+ D-Net	47.28	55.10	57.02	11.7M

Table 1: Ablation study on the effectiveness of the disentanglement network (D-Net). The context-based aggregation network (A-Net) is taken as a strong baseline. MLP_1 denotes a shared MLP replacing D-Net. MLP_2 denotes two different MLPs replacing D-Net. The mAP of tIoU@0.5 are reported on the THUMOS14 test set.

Cls.	Loc.	RGB	Flow	Fusion
F^{cls}	F^{loc}	47.28	55.10	57.02
F^{cls}	F^{cls}	40.26	44.85	47.16
F^{loc}	F^{loc}	17.02	23.57	25.53
F^{loc}	F^{cls}	14.81	17.89	21.26

Table 2: Ablation study on whether we successfully disentangle the classification and localization features. "Cls." and "Loc." in the first row denote the classification task and localization task, respectively. F^{cls} and F^{loc} denote the disentangled classification features and localization features, respectively. The mAP of tIoU@0.5 are reported on the THUMOS14 test set.

Evaluation Metric. We evaluate the performance of our model using mean average precision (mAP) values at different tIoU thresholds. On THUMOS14, the tIoU thresholds are chosen from $\{0.3, 0.4, 0.5, 0.6, 0.7\}$. On ActivityNet v1.3, the tIoU thresholds are from $\{0.5, 0.75, 0.95\}$.

Implementation Details. The interval between snippets is set to 16 frames. The I3D network (Carreira and Zisserman 2017) is used to extract snippet-level features. BSN (Lin et al. 2018) is responsible for generating proposals. The ratio of fusing the RGB and optical flow predictions is 5:6. The video-level classification results generated by UntrimmedNet (Wang et al. 2017) as an common practice (Zeng et al. 2019; Su et al. 2021; Qing et al. 2021) improves the performance. Moreover, we appropriately add some incomplete proposals as positive samples for classification to help our disentanglement network find the most discriminative parts.

Ablation Study

In order to explore the effectiveness of our disentanglement network and how disentangled features are better than original features, we conducted in-depth ablation experiments.

Cls.	Loc.	RGB	Flow	Fusion
F^{pro}	F^{pro}	44.58	51.42	53.50
F^{pro}	F^{loc}	46.85	54.84	56.76
F^{cls}	F^{pro}	44.99	51.47	53.72
F^{cls}	F^{loc}	47.28	55.10	57.02

Table 3: Ablation study on advantages of the disentangled features over the original features. F^{pro} denotes the original proposal-level features. The mAP of tIoU@0.5 are reported on the THUMOS14 test set.

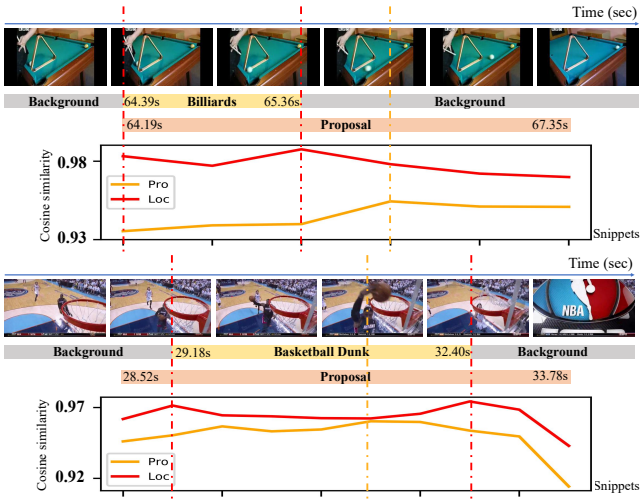


Figure 3: The visualization of frames concerned by the original features (the orange lines) and our disentangled localization features (the red lines) on the THUMOS14 test set. The original features focus on the background frames (top) useful for classification and the most discriminative frames (bottom). Nevertheless, Our disentangled localization features focus the frames at the start and end of the action.

All results are reported on THUMOS14.

Sub-networks. The direct use of max-pooling to aggregate disentangled features will blur features and cause great damage to the process of disentanglement. Compared with max-pooling, the context-based aggregation network (A-Net) can better aggregate disentangled features through a *query-and-retrieval* procedure. Therefore, we take it as a strong baseline to validate the effectiveness of our disentanglement network. The results are shown in Table 1.

Based on the context-based aggregation network, our disentanglement network improves 2.70% on the RGB stream and 3.68% on the optical flow stream at tIoU 0.5. In addition, the performance after fusion is improved by 3.52%.

The model size of our model is 11.7M. Specially, the context-based aggregation network has 10.1M parameters and the disentanglement network only has 1.6M parameters. Although the additional number of parameters brought by the disentanglement network is quite small, for a fair comparison, we still add some MLPs to the context-based aggregation network to fill up the gap. Table 1 reports the results.

Method	RGB	Flow	Fusion
Random vectors	45.05	53.94	56.00
Only common	46.64	51.70	54.26
W/o common	44.75	53.20	55.75
Ours	47.28	55.10	57.02

Table 4: Ablation study on alternative design choices. The mAP of tIoU@0.5 are reported on the THUMOS14 test set.

MLP_1 denotes we add an MLP for both classification and localization branches. MLP_2 means we add two different MLPs for classification and localization branches, respectively. In both cases, the performance improvement of MLP is far less than that of our disentanglement network.

Several conclusions can be drawn from this experiment. (1) The disentanglement of classification and localization representations via the disentanglement network benefits TAL. (2) The performance improvement of our disentanglement network is not caused by a deeper or larger network. (3) Simply using two MLPs to obtain different representations for classification and localizations is not as effective as our disentanglement network.

Successful Disentanglement. To explore whether we successfully disentangle the classification and localization features, we regroup the tasks and features in testing. Table 2 reports the results. When the disentangled localization features are used for the classification task, the performance drops a lot. It indicates the disentangled localization features do not focus on the most discriminative parts.

When the disentangled classification features are used for the localization task, the performance does not drop so much. Therefore, up to a certain extent, the disentangled classification features can handle the localization task. This may also be the reason why previous work using shared I3D features suitable for classification can work on classification and localization tasks. Meanwhile, it also indicates that the disentangled classification features are indeed not the most suitable for the localization task.

Advantages of Disentangled Features. Prior approaches perform classification and localization tasks on the shared proposal-level features. In order to explore how the disentangled features are better than the original features, we add classification and localization branches based on the original features. These branches are independent of our network. Then we combine the results of the two new branches and our disentangled branches. The results are shown in Table 3.

Compared with the original features, the advantages of our disentangled classification features are not obvious. The reason is that the original I3D features are extracted with the classification network.

However, in the localization task, our disentangled localization features have obvious advantages compared with the original features. In both two cases, it brings 3.26% and 3.30% improvements on tIoU 0.5, respectively. This again verifies that our disentangled localization features are more suitable for the localization task.

Design Choices. We experiment with the design choices

Method	THUMOS14						ActivityNet v1.3			
	0.3	0.4	0.5	0.6	0.7	Avg.	0.5	0.75	0.95	Avg.
BSN (Lin et al. 2018)	53.5	45.0	36.9	28.4	20.0	36.8	46.45	29.96	8.02	30.03
TAL-Net (Chao et al. 2018)	53.2	48.5	42.8	33.8	20.8	39.8	38.23	18.30	1.30	20.22
P-GCN (Zeng et al. 2019)	63.6	57.8	49.1	—	—	—	48.26	33.16	3.27	31.11
GTAN (Long et al. 2019)	57.8	47.2	38.8	—	—	—	52.61	34.14	8.91	34.31
MGG (Liu et al. 2019)	53.9	46.8	37.4	29.5	21.3	37.8	—	—	—	—
BMN (Lin et al. 2019)	56.0	47.4	38.8	29.7	20.5	38.5	50.07	34.78	8.29	33.85
Li *et al.* (Li et al. 2020)	57.1	51.6	38.6	28.9	17.0	38.6	—	—	—	—
G-TAD (Xu et al. 2020)	54.5	47.6	40.2	30.8	23.4	39.3	50.36	34.60	9.02	34.09
BU-MR (Zhao et al. 2020)	53.9	50.7	45.4	38.0	28.5	43.3	43.47	33.91	**9.21**	30.12
BSN++ (Su et al. 2021)	59.9	49.5	41.3	31.9	22.8	41.1	51.27	35.70	8.33	34.88
TCANet (Qing et al. 2021)	60.6	53.2	44.6	36.8	26.7	44.4	51.91	34.92	7.46	34.43
Lin *et al.* (Lin et al. 2021)	67.3	62.4	55.5	43.7	**31.1**	52.0	52.40	35.30	6.50	34.40
ContextLoc (Zhu et al. 2021)	68.3	63.8	54.3	41.8	26.2	50.9	56.01	35.19	3.55	34.23
MUSES (Liu et al. 2021)	68.9	64.0	56.9	**46.3**	31.0	53.4	50.02	34.97	6.57	33.99
Ours	**72.1**	**65.9**	**57.0**	44.2	28.5	**53.5**	**58.14**	**36.30**	6.16	**35.24**

Table 5: Action localization results on THUMOS14 and ActivityNet v1.3. On the THUMOS14 test set, the mAP (%) at different tIoU thresholds and the average mAP of IoU thresholds from 0.3 to 0.7 are reported. On the ActivityNet v1.3 validation set, the mAP (%) at different tIoU thresholds and the average mAP of IoU thresholds from 0.5 to 0.95 are reported. **Bold** fonts indicate the best performance.

Method	0.5	0.75	0.95	Average
TCANet [BSN]	51.91	34.92	7.46	34.43
Ours [BSN]	58.14	36.30	6.16	35.24
TCANet [BMN]	54.33	39.13	**8.41**	37.56
Ours [BMN]	**61.94**	**39.51**	6.58	**38.45**

Table 6: Results on the ActivityNet v1.3 validation set. For fair comparison, we combine proposals with the scores of BMN (Lin et al. 2019).

and report the results in Tab. 4. (1) Comparing "random vectors" and "ours" indicates replacing zero vectors with random vectors degrades the performance. The reason might be that random vectors not only remove information (like zero vectors) but also bring noise. (2) Comparing "only common" and "ours" indicates using only the common hidden variables degrades the performance. It verifies the necessity of feature disentanglement. (3) Comparing "w/o common" and "ours" indicates not having the common hidden variables also degrades the performance. It verifies the importance of modeling the common features in addition to the unique classification features and the unique localization features.

More Visualization Results about Advantages. Quantitative experiments (Table 3) show our disentangled localization features are better than the original features. In order to explain it better, we perform more visualizations.

In the action "billiards", the original features focus on the background frame. But our disentangled localization features can accurately focus on the start and end frames.

In the action "basketball dunk", the original features focus on the most discriminative frames. In contrast, our disentangled localization features focus on the start and end frames. The above two examples once again illustrate the advantages of our disentangled localization features. More qualitative results can be found in the supplementary material.

Comparison with State-of-the-Art Methods

This section will compare the proposed network with state-of-the-art methods on THUMOS14 and ActivityNet v1.3.

THUMOS14. We compare our model with the state-of-the-art methods on THUMOS14 in Table 5. At tIoU 0.3, our model outperforms the previously best method MUSES (Liu et al. 2021) by 3.2% absolute improvement. At tIoU 0.5, our model achieves the best performance. This demonstrates the benefit of finding the appropriate features for the localization and classification tasks.

ActivityNet v1.3. Table 5 compares our model with other methods on ActivityNet v1.3. Our model outperforms all other methods on average mAP for tIoU thresholds {0.5:0.05:0.95}. At tIoU 0.5, our model reaches an mAP of 58.14% which is 2.13% higher than the current best 56.01% achieved by ContextLoc (Zhu et al. 2021).

TCANet (Qing et al. 2021) combines proposals with the scores of BMN (Lin et al. 2019) for better performance. For fair comparison, we additionally conduct experiments with the same strategy on ActivityNet v1.3. Table 6 shows our model still has obvious advantages.

Conclusion

This paper introduces a novel disentanglement network to solve TAL. It disentangles features that are more suitable for classification and localization tasks from the original features. Ablation experiments under controlled settings indicate that (1) our model succeeds in disentanglement and (2) our model indeed disentangles features that are more suitable for the localization task compared to the original features. Results on two benchmark datasets demonstrate that our model outperforms state-of-the-art TAL methods.

Acknowledgments

This work was supported partly by National Key R&D Program of China under Grant 2018AAA0101400, NSFC under Grants 62088102, 61976171, and 61773312, and Fundamental Research Funds for the Central Universities under Grant XTR042021005.

References

Bai, Y.; Wang, Y.; Tong, Y.; Yang, Y.; Liu, Q.; and Liu, J. 2020. Boundary content graph neural network for temporal action proposal generation. In *ECCV*, 121–137.

Bhagat, S.; Uppal, S.; Yin, V.; and Lim, N. 2020. Disentangling Multiple Features in Video Sequences Using Gaussian Processes in Variational Autoencoders. In *ECCV*, 102–117.

Carreira, J.; and Zisserman, A. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 6299–6308.

Chao, Y.-W.; Vijayanarasimhan, S.; Seybold, B.; Ross, D. A.; Deng, J.; and Sukthankar, R. 2018. Rethinking the Faster R-CNN Architecture for Temporal Action Localization. In *CVPR*, 1130–1139.

Dai, X.; Singh, B.; Zhang, G.; Davis, L. S.; and Chen, Y. Q. 2017. Temporal Context Network for Activity Localization in Videos. In *ICCV*, 5727–5736.

Denton, E. L.; and Birodkar, V. 2017. Unsupervised Learning of Disentangled Representations from Video. In *NIPS*, 4415–4424.

Ding, Z.; Xu, Y.; Xu, W.; Yang, Y.; Welling, M.; and Tu, Z. 2020. Guided Variational Autoencoder for Disentanglement Learning. In *CVPR*, 7917–7926.

Gao, J.; Chen, K.; and Nevatia, R. 2018. Ctap: Complementary temporal action proposal generation. In *ECCV*, 68–83.

Gao, J.; Shi, Z.; Li, J.; Wang, G.; Yuan, Y.; Ge, S.; and Zhou, X. 2020. Accurate Temporal Action Proposal Generation with Relation-Aware Pyramid Network. In *AAAI*, 10810–10817.

Gao, J.; Yang, Z.; Sun, C.; Chen, K.; and Nevatiawqer, R. 2017. TURN TAP: Temporal Unit Regression Network for Temporal Action Proposals. In *ICCV*, 3648–3656.

Heilbron, F. C.; Escorcia, V.; Ghanem, B.; and Niebles, J. C. 2015. ActivityNet: A large-scale video benchmark for human activity understanding. In *CVPR*, 961–970.

Huang, J.; Li, N.; Zhang, T.; Li, G.; Huang, T.; and Gao, W. 2018. SAP: Self-Adaptive Proposal Model for Temporal Action Detection Based on Reinforcement Learning. In *AAAI*, 6951–6958.

Huang, Y.; Dai, Q.; and Lu, Y. 2019. Decoupling Localization and Classification in Single Shot Temporal Action Detection. In *ICME*, 1288–1293.

Jiang, Y.-G.; Idrees, H.; Zamir, A.; Gorban, A.; Laptev, I.; Shah, M.; and Sukthankar, R. 2014. Thumos challenge: Action recognition with a large number of classes.

Kim, M.; Lee, H. J.; Lee, S.; and Ro, Y. M. 2020. Robust Video Facial Authentication With Unsupervised Mode Disentanglement. In *ICIP*, 1321–1325.

Li, J.; Liu, X.; Zong, Z.; Zhao, W.; Zhang, M.; and Song, J. 2020. Graph Attention Based Proposal 3D ConvNets for Action Detection. In *AAAI*, 4626–4633.

Li, Z.; and Yao, L. 2021. Three Birds with One Stone: Multi-Task Temporal Action Detection via Recycling Temporal Annotations. In *CVPR*, 4751–4760.

Lin, C.; Li, J.; Wang, Y.; Tai, Y.; Luo, D.; Cui, Z.; Wang, C.; Li, J.; Huang, F.; and Ji, R. 2020. Fast Learning of Temporal Action Proposal via Dense Boundary Generator. In *AAAI*, 11499–11506.

Lin, C.; Xu, C.; Luo, D.; Wang, Y.; Tai, Y.; Wang, C.; Li, J.; Huang, F.; and Fu, Y. 2021. Learning Salient Boundary Feature for Anchor-free Temporal Action Localization. In *CVPR*, 3320–3329.

Lin, T.; Liu, X.; Li, X.; Ding, E.; and Wen, S. 2019. BMN: Boundary-Matching Network for Temporal Action Proposal Generation. In *ICCV*, 3888–3897.

Lin, T.; Zhao, X.; Su, H.; Wang, C.; and Yang, M. 2018. Bsn: Boundary sensitive network for temporal action proposal generation. In *ECCV*, 3–19.

Liu, J.; Huang, Y.; Peng, X.; and Wang, L. 2015. Multi-view descriptor mining via codeword net for action recognition. In *ICIP*, 793–797.

Liu, Q.; and Wang, Z. 2020. Progressive Boundary Refinement Network for Temporal Action Detection. In *AAAI*, 11612–11619.

Liu, X.; Hu, Y.; Bai, S.; Ding, F.; Bai, X.; and Torr, P. H. 2021. Multi-shot Temporal Event Localization: a Benchmark. In *CVPR*, 12596–12606.

Liu, Y.; Ma, L.; Zhang, Y.; Liu, W.; and Chang, S.-F. 2019. Multi-granularity generator for temporal action proposal. In *CVPR*, 3604–3613.

Liu, Z.; Zhang, H.; Chen, Z.; Wang, Z.; and Ouyang, W. 2020. Disentangling and Unifying Graph Convolutions for Skeleton-Based Action Recognition. In *CVPR*, 140–149.

Long, F.; Yao, T.; Qiu, Z.; Tian, X.; Luo, J.; and Mei, T. 2019. Gaussian temporal awareness networks for action localization. In *CVPR*, 344–353.

Meng, Y.; Lin, C.-C.; Panda, R.; Sattigeri, P.; Karlinsky, L.; Oliva, A.; Saenko, K.; and Feris, R. 2020. AR-Net: Adaptive Frame Resolution for Efficient Action Recognition. In *ECCV*, 86–104.

Qing, Z.; Su, H.; Gan, W.; Wang, D.; Wu, W.; Wang, X.; Qiao, Y.; Yan, J.; Gao, C.; and Sang, N. 2021. Temporal Context Aggregation Network for Temporal Action Proposal Refinement. In *CVPR*, 485–494.

Shi, L.; Zhang, Y.; Cheng, J.; and Lu, H. 2019. Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition. In *CVPR*, 12018–12027.

Shou, Z.; Chan, J.; Zareian, A.; Miyazawa, K.; and Chang, S.-F. 2017. CDC: Convolutional-De-Convolutional Networks for Precise Temporal Action Localization in Untrimmed Videos. In *CVPR*, 1417–1426.

Simonyan, K.; and Zisserman, A. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. In *NeurIPS*, 568–576.

Su, H.; Gan, W.; Wu, W.; Yan, J.; and Qiao, Y. 2021. BSN++: Complementary Boundary Regressor with Scale-Balanced Relation Modeling for Temporal Action Proposal Generation. In *AAAI*, 2602–2610.

Sun, X.; Xu, H.; and Saenko, K. 2020. TwoStreamVAN: Improving Motion Modeling in Video Generation. In *WACV*, 2733–2742.

Wang, B.; Ma, L.; Zhang, W.; and Liu, W. 2018. Reconstruction Network for Video Captioning. In *CVPR*, 7622–7631.

Wang, L.; Xiong, Y.; Lin, D.; and Van Gool, L. 2017. Untrimmednets for weakly supervised action recognition and detection. In *CVPR*, 4325–4334.

Wang, X.; Zhang, S.; Qing, Z.; Shao, Y.; Gao, C.; and Sang, N. 2021. Self-supervised learning for semi-supervised temporal action proposal. In *CVPR*, 1905–1914.

Wang, Y.; Bilinski, P.; Brémond, F.; and Dantcheva, A. 2020. G3AN: Disentangling Appearance and Motion for Video Generation. In *CVPR*, 5263–5272.

Xiao, S.; Zhao, Z.; Zhang, Z.; Yan, X.; and Yang, M. 2020. Convolutional Hierarchical Attention Network for Query-Focused Video Summarization. In *AAAI*, 12426–12433.

Xu, H.; Das, A.; and Saenko, K. 2017. R-C3D: Region Convolutional 3D Network for Temporal Activity Detection. In *ICCV*, 5794–5803.

Xu, M.; Zhao, C.; Rojas, D. S.; Thabet, A.; and Ghanem, B. 2020. G-TAD: Sub-Graph Localization for Temporal Action Detection. In *CVPR*, 10153–10162.

Zeng, R.; Huang, W.; Tan, M.; Rong, Y.; Zhao, P.; Huang, J.; and Gan, C. 2019. Graph convolutional networks for temporal action localization. In *CVPR*, 7094–7103.

Zhao, P.; Xie, L.; Ju, C.; Zhang, Y.; Wang, Y.; and Tian, Q. 2020. Bottom-up temporal action localization with mutual regularization. In *ECCV*, 539–555.

Zhu, Y.; Lan, Z.; Newsam, S.; and Hauptmann, A. 2018. Hidden Two-Stream Convolutional Networks for Action Recognition. In *ACCV*, 363–378.

Zhu, Z.; Tang, W.; Wang, L.; Zheng, N.; and Hua, G. 2021. Enriching Local and Global Contexts for Temporal Action Localization. In *ICCV*.