Multimedia Systems (2023) 29:2181–2191
https://doi.org/10.1007/s00530-023-01091-0

REGULAR PAPER

Centralized sub-critic based hierarchical-structured reinforcement learning for temporal sentence grounding

Yingyuan Zhao[1,2] · Zhiyi Tan[1] · Bing-Kun Bao[1] · Zhengzheng Tu[2]

Received: 14 January 2023 / Accepted: 9 April 2023 / Published online: 28 April 2023

Abstract
Temporal sentence grounding is to localize the corresponding video clip of a sentence in video. Existing study based on hierarchical-structured reinforcement learning treats the task as training an agent learn its strategy, decomposed into a master-policy and several sub-policies, to adjust the prediction boundary progressively heading for the target clip. They adopt a decentralized-sub-critic framework, equipping every sub-policy with its own sub-critic network to perceive the current environment for enhancing its training. However, massive sub-critics result in massive network parameters. In addition, each decentralized sub-critic only considers the action of its sub-policy and fails to model the impact of other sub-policies' actions on the environment, which would mislead sub-policies' learning. To handle this, we contribute a novel solution composed of a centralized sub-critic based hierarchical-structured reinforcement learning (CSC-HSRL). The key is to train a centralized sub-critic network to evaluate the effects of all sub-policies' actions. Furthermore, centralized sub-critic helps sub-policies to determine whether their actions are beneficial to localize target clip more precisely and support their training. Also, centralized sub-critic has fewer parameters. Experiments on Charades-STA and ActivityNet dataset show that compared with the decentralized sub-critic based model TSP-PRL, CSC-HSRL has higher accuracy and reduces model parameters in the meantime.

1 Introduction

Temporal sentence grounding aims at localizing a video clip described by a sentence from a long video. Existing studies are generally divided into two categories: proposal-based methods and end-to-end methods. Proposal-based models [2, 8, 16] randomly generate candidates through a sliding window over the original video. They sort those proposals by their alignment scores, which are quite time-consuming due to the numerous candidates. To get rid of the drawbacks of proposals, end-to-end models [11, 12, 30, 32] are proposed to directly predict the start/end timestamp of the target video. Some works [11, 12, 30, 32] turn to deep reinforcement learning, which formulates the task as training an agent of single policy to learn the strategy of gradually adjusting the prediction boundary by interacting with the environment.

Restricted by neural network parameters and the action space, it is difficult for an agent of single policy to make precise and interpretable action choices. Recent works propose hierarchical-structured deep reinforcement learning to realize the complex-decision task. Inspired by human's coarse-to-fine decision-making paradigm, Wu et al. [32] proposes a tree-structured agent consisting of a master policy and five sub-policies to decompose the complex action. The master policy decides the category of adjustment and the sub-policies are responsible for the scale (see Fig. 1). If the adjustment makes the boundary closer to the target clip, the participating policies which make the actions will receive positive rewards, otherwise, they will receive negative rewards. By rewarding policies respectively, each policy will take responsibility for adjustment in its corresponding way.

✉ Zhiyi Tan
tzy@njupt.edu.cn

Yingyuan Zhao
1020010623@njupt.edu.cn

Bing-Kun Bao
bingkunbao@njupt.edu.cn

Zhengzheng Tu
zhengzhengahu@163.com

[1] College of Telecommunications Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing, China

[2] Anhui Provincial Key Laboratory of Multimodal Cognitive Computation, School of Computer Science and Technology, Anhui University, Hefei, China

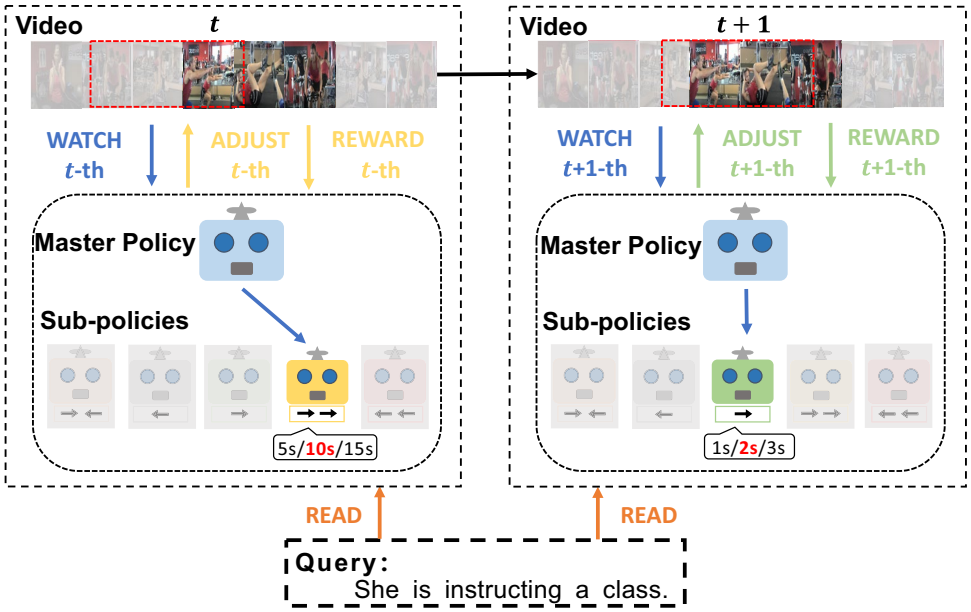

Fig. 1 An illustration of how hierarchical-structured RL based model works. For one episode, the master policy observes the current prediction boundary and decides to shift the boundary right markedly. So it chooses the yellow sub-policy which is in charge of shifting right markedly. Then the chosen sub-policy will observe the environment and choose its primitive action, shifting the boundary right by 10 s. After this, the participating policies will receive rewards based on the impact of their actions. Then, the next episode will go on until the last step

Though the framework of tree-structured policies perfectly matches the task, the corresponding training algorithm of sub-policies remains to be completed. Previous work chooses a decentralized sub-critics framework [18] for sub-policies: each sub-policy has its sub-critic network to estimate the trend of the current environment, whether it is making the prediction boundary head for the target clip. The principal idea behind this is to represent the trend of the environment's change with the value of the potential accumulated reward that a sub-policy will obtain in the future. They tend to train each sub-critic to directly predict its sub-policy's potential accumulated reward. There are two problems in the decentralized-structured sub-critics. First, it is impractical for each sub-critic to directly predict its sub-policy's potential accumulated reward. In each episode, the master policy will choose a sub-policy to change the environment, which is blind to the decentralized sub-critics. Since the environment is changed by different sub-policy sequentially, each sub-policy could randomly influence the trend, which in turn affects other sub-policies' potential accumulated rewards. We argue that the decentralized sub-critics may mislead the sub-policies about the real state of the current environment because their estimations/predictions are unreliable. The other problem is that decentralized sub-critics lead to waste and parameters-consuming. Each sub-critic costs a certain number of computing and storage resources due to the transforming high-dimensional feature into numerical value and the storage of its network's parameters. The framework of decentralized sub-critics magnifies the cost by equipping each sub-policy with a sub-critic, thus reducing the mobile deployability of the model.

To this end, we follow the tree-structured decision-making framework mentioned above and propose Centralized Sub-Critic based hierarchical-structured reinforcement learning (CSC-HSRL), to realize temporal sentence grounding. Instead of one sub-critic for one sub-policy in decentralized sub-critics, we propose a centralized sub-critic to guide all sub-policies, which estimates the trend of the environment by predicting the sum of all sub-policies' potential accumulated rewards. On one hand, since the environment could only be changed by sub-policy group, It is practical and easy to match the trend of environment with the sum of all sub-policies' potential accumulated rewards. Also, the prediction from our centralized sub-critic takes the group's impact on the environment into consideration, avoiding the difficulty of modeling each sub-policy's impact on others. On the other hand, centralizing sub-critics decreases the number of sub-critics to one, which frees the suffering of massive parameters that extra sub-critics network cost in a decentralized structure. Correspondingly, we propose the Advantage Actor Centralized Sub Critic algorithm (A2CSC) to train sub-policies, which is a variation of Advantage Actor-Critic (A2C) [19]. We formulate the difference between the accumulated reward of sub-policy group and the prediction of the centralized sub-critic as the advantage function, meaning that sub-policies adjust the environment better than expected. Each sub-policy will optimize itself to maximize the advantage function. The centralized sub-critic enables each sub-policy to perceive the impact of the group's actions so that it can learn the strategy to achieve its final goal.

Different from the works based on the decentralized-structured sub-critics, CSC-HSRL utilizes centralized sub-critic to offer each sub-policy the precise estimation of the environment's trend, helping them learn to adjust the prediction boundary. It solves the problem that sub-policies

are unable to perceive the difference in the environment changed by others. Meanwhile, the framework of the centralized sub-critic alleviates the struggle of massive parameters and accelerate the speed of prediction. This paper conducts extensive experiments on the Charade-STA [8] and ActivityNet [15] dataset. Experiments show that compared with the existing hierarchical-structured RL method, CSC-HSRL improves the accuracy while reducing the parameters of the model.

2 Related works

2.1 Temporal sentence grounding

In recent years, more and more researchers have shown great interest in temporal sentence grounding tasks. Earlier studies [2, 8, 9, 34] use a sliding window to generate proposals, thus treating the task as a ranking or regression task. Gao et al. [8] proposes a cross-modal temporal regression locator to jointly model sentences and videos. Similarly, Anne Hendricks et al. [2] designs a temporal context network to measure semantic distance by mapping sentence features and video features to a common space, which expands to enhance the visual information. Such methods are mostly inefficient and time-consuming due to the need to generate candidate fragments. Therefore, researchers tend to seek a solution in an end-to-end manner. Reinforcement learning is adopted to formulate the task as a sequential decision process. He et al. [12] is the first to transform the task into a sequential decision problem, which alleviates the above problems to a certain extent. The following works [11, 30] enhance the structure in different ways: feature fusion and agent optimization. To free the model from the restriction of agent with single policy, Wu et al. [32] first proposes a tree-structured, hierarchical reinforcement learning framework. It decouples complex actions into actions of two levels, executed by a master policy and sub policies. However, existing method based on hierarchical-structured reinforcement learning equips every sub-policy with a sub-critic, which expands the model's parameters. To this end, this paper proposes a centralized sub-critic based hierarchical-structured reinforcement learning framework, which reduces the model's parameters while improving performance.

2.2 Deep reinforcement learning

Deep reinforcement learning is a kind of effective machine learning method, aiming at learning the strategy to maximize the cumulative future rewards. When it comes to computer vision, DRL has been successfully applied to many tasks in [22, 31, 33]. To satisfy the demands of different tasks, DRL has been modified to large amount of variations such as multi-agent reinforcement learning in [7, 18, 23] and hierarchical reinforcement learning in [1, 3, 27]. Recent works [22, 31, 33] based on multi-agent reinforcement learning usually decompose their tasks into multiple sequential decision-making processes, which are then handed over to each agent for cooperation or confrontation to achieve the final goal. Wu et al. [33] proposes a multi-agent solution for video recognition, which samples video frames at different positions by multiple agents in simple cooperation. For hierarchical reinforcement learning, present methods solve problems by a divide-and-conquer approach. Wang et al. [31] proposes a hierarchical reinforcement learning-based method for video description. The manager module observes high-dimensional information and set a goal for the worker module. The worker module adjusts actions to achieve the goal, and the two cooperate to complete the video description.

Instead of setting a sub-goal for the low-level module, Wu et al. [32] proposes a tree-structured agent to decompose the complex grounding strategy into a master policy and several sub-policies. Its hierarchical-structured policies provide a perfect paradigm for temporal sentence grounding. Although the tree-structured policies conform to the human perception mechanism from coarse to fine, the corresponding training for sub-critics is outdated. The ability of modeling other sub-policies and the perception of overall are missed. To this end, this paper proposes a centralized sub-critic based hierarchical-structured reinforcement learning framework, which enriches the environment perception.

3 Methods

3.1 Task formulation

We treat the temporal sentence grounding task as a Markov decision process: states are represented as $s \in S$, actions of the master policy and sub-policies are represented as a^m, a^{sub} respectively. For one episode, the master policy $\pi^m\left(a_t^m \mid s_t;\theta_{\pi^m}\right)$ observes the current prediction boundary and chooses which kind of adjustment should be done. Then the corresponding sub-policy $\pi^{sub}\left(a_t^{sub} \mid s_t;\theta_{\pi^{sub}}\right)$, $sub \in \{A, B, C, D, E\}$, which is in charge of it, will observe the environment and choose its primitive action to adjust the prediction boundary. r^m and r^{sub} represent the rewards obtained by the current actions of the master policy and the chosen sub-policy. The state transition function is represented as $\mathrm{T} : \left(s, \left\langle a^m, a^{sub}\right\rangle\right) \rightarrow s'$. Figure 2 shows the process of the model training phase. An episode at t timestep in reinforcement learning [25]:

$$\tau = \{s_t, \pi^m(a_t^m \mid s_t;\theta_{\pi^m}), \mathrm{a}_t^m, r_t^m, a_t^m * \pi^{sub} (a_t^{sub} \mid s_t;\theta_{\pi^{sub}}), a_t^{sub}, r_t^{sub}\} \quad (1)$$

3.2 Hierarchical-structured agent

State encoder

As shown in Fig. 2, query Q, global video V_t^g, current predicted video V_t^c and prediction boundary L_t are fused as the agent's current state S_t. Videos are sampled units by units [8]. To fetch the video-unit feature and the sentence feature, pre-trained backbone [14, 26, 29] are used to process them. The global video feature V_t^g and current predicted video feature V_t^c are the concatenation of ten unit-level features which are uniformly sampled from their segments. To integrate the cross-modal information into a united representation, gated-attention [4], a fully-connected layer and GRU [6] are used to obtain the state vector S_t:

$$s_t = GRU\left(f\left(V_t^{gq}, V_t^{cq}, L_t^{lq}\right), s_{t-1}\right) \quad (2)$$

in which

$$V_t^{qg} = h(Q) \odot V_t^g \quad (3)$$

where $f(.)$ and $h(.)$ denotes the fully-connected layer with activation function. V_t^{qc} and L_t^{ql} are calculated in a similar way.

Action space

As the master policy is responsible for deciding which category of adjustment should be done in the current state, its action space consists of five sub-policies:

$$a_t^m \in \{A, B, C, D, E\} \quad (4)$$

Each sub-policy is responsible for different adjustment categories: scaling transformation A, large-scale left shift B, large-scale right shift C, small-scale left shift D and small-scale right shift E. Their action space are set to provide corresponding adjustments in different scale. The details of sub-policies' action space can be found in Table 1.

Table 1 Some notations used in this paper and sub-policies' action space

Notations	Description
s, s'	The current state and the next state
s_t	The state at t timestep
m	The master policy
sub	One of the five sub-policies, $sub \in \{A, B, C, D, E\}$
a_t^m	The master policy's action at t timestep
a_t^{sub}	The chosen sub policy's action at t timestep
r_t^m	The master policy's reward at t timestep
r_t^{sub}	The chosen sub policy's reward at t timestep
$V^m(s_t)$	The master critic network
$V^{csc}(s_t)$	The centralized sub-critic network
SG	The sub-policy group $\{A, B, C, D, E\}$
IoU_t	The intersection over union between the boundary and the ground truth at t timestep
$\pi^m(a_t^m \mid s_t;\theta_{\pi^m})$	The master policy network which outputs the probability of actions
$\pi^{\mathrm{sub}}(a_t^{\mathrm{sub}} \mid s_t;\theta_{\pi^{sub}})$	One of the sub-policies networks which output the probability of actions
R_t^{sub}	The accumulated rewards of chosen sub-policy since t timestep
R_t^m	The accumulated rewards of the master policy since t timestep
β	The constant discount factor using in calculating the sum of rewards in Eq 6, 8, 9, 14
μ	The positive constant value that determines the degree of reward in Eq 7, 15
α, γ, λ	Weight parameters to balance the two loss functions in training in Eq 11, 12, 16, 18
N	The number of the clips of the entire video
Sub-policy	Action space
A	Extending/shortening 1.2/1.5 times w.r.t center point
B	Shifting start/ end/ start & end point backward N/10
C	Shifting start/ end/ start & end point forward N/10
D	Shifting start/ end/ start & end point backward Z frames
E	Shifting start/ end/ start & end point forward Z frames

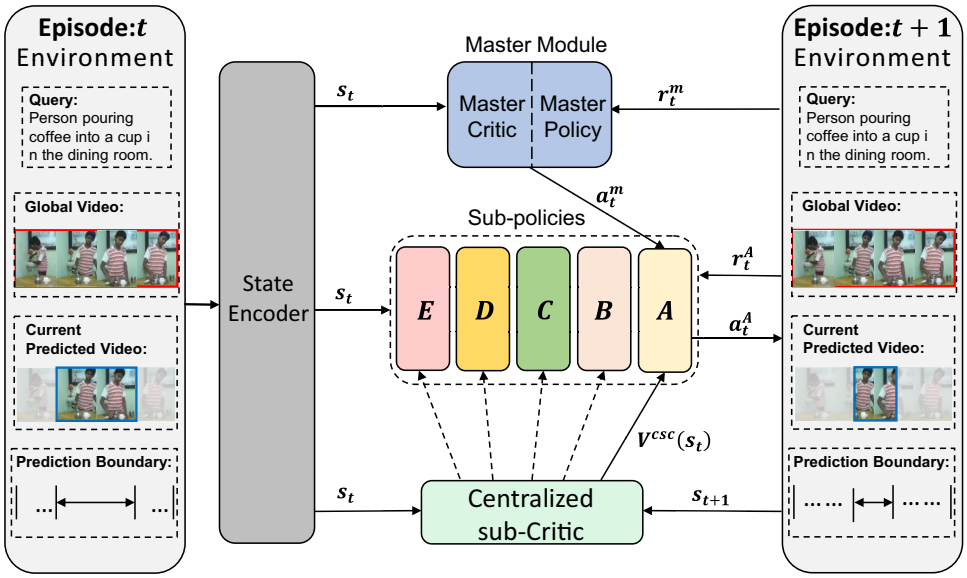

Fig. 2 The graphical representation of the workflow of CSC-HSRL in one episode. After the state encoder outputs the state vector s_t, the master policy observes it and decides which kind of adjustment should be done. It activates the corresponding sub-policy by action a_t^m and gets the reward r_t^m calculated by the reward function. Similarly, the activated sub-policy A will observe the state s_t, conduct its action a_t^A and get its rewardr_t^A. After that, the centralized sub-critic network observes the state s_t and makes the prediction $V^{csc}(s_t)$ of the cumulative future rewards of the sub-policy group. At last, the environment is changed into the next state s_{t+1}

3.3 Advantage actor centralized sub-critic for sub-polices

CSC-HRSL is based on Actor-Critic [19], where critic network aims to help the agent's training by estimating the current state and predicting the expectation of the cumulative future rewards: $V(s_t) \rightarrow R_t$. Since the master policy and sub-policies are supposed to learning different strategies to adjust the boundary, their views of the environment also vary a lot. We set different critic networks for them.

The recent works [32] have decentralized sub-critic for sub-policies. In fact, their decentralized sub-critic fails to estimate the trend of the environment due to its decentralized structure. In order to offer sub-policies a better perception of others influence, we propose the centralized sub-critic shown in Fig. 2 and its corresponding training algorithm: advantage actor centralized sub-critic (A2CSC) to maximize the effect of this structure.

Different from the decentralized sub-critic, our proposed centralized sub-critic $V^{csc}(s_t)$ predicts the cumulative potential rewards that sub-policy group will obtain based on the current environment:

$$V^{csc}(s_t) \rightarrow \sum_{sub \in SG} R_t^{sub} \tag{5}$$

where SG represents the sub-policy group and R_t^{sub} represents the cumulative potential rewards of one sub-policy, which is calculated as:

$$R_t^{sub} = \sum_{k=t} \sum_{a_k^m = sub} \beta^{k-t} r_k^{sub} \tag{6}$$

where β is the constant discount factor. For each sub-policy, its reward is calculated based on degree of alignment between the prediction boundary and the ground truth. IoU is a popular metric which calculates the intersection over union of two segments. We follow the recent works [24, 32] and formulate the reward function of sub-policies:

$$r_t^{sub} = \begin{cases} \mu + \emptyset(IoU_t) & IoU_t > IoU_{t-1} \\ -\mu/10 & IoU_{t-1} \geq IoU_t \geq 0 \\ -\mu & otherwise \end{cases} \tag{7}$$

where μ is the positive constant value for rewarding. $\emptyset(IoU_t)$ takes the value of IoU_t only if IoU_t is larger than 0.5, otherwise it takes 0. This extra reward is to award sub-policy for its great contribution for positive adjustment.

To achieve balance between accuracy and efficiency, we set a limitation where the agent could only do actions in finite steps $T_{\max}$. We minimize the squared difference between the predictions and the true values:

$$\mathcal{L}_{SG}(\theta_{V^{csc}}) = \sum_{t=1}^{T_{\max}} \left(\sum_{sub \in SG} R_t^{sub} - V^{csc}(s_t) + \beta^{T_{\max}-t+1} V^{csc}(s_{T_{\max}}) \right)^2 \tag{8}$$

where $V^{csc}(s_{T_{\max}})$ is added for consistency which represents the cumulative potential rewards of sub-policy group after $T_{\max}$.

For fully utilize the centralized sub-critic, we propose the advantage actor centralized sub-critic which is a variation of A2C. Following the A2C [19], each sub-policy has its objective function which contains its advantage function. In this task, each sub-policy participates in the adjustment and have the same goal of localizing the target segment. As mentioned above, the centralized sub-critic give the estimation of the sum of sub-policy group's potential rewards. Inspired by A2C, we formulate the difference between real potential rewards and the estimation as the sub-policy's advantage function:

$$adv_t^{sub} = \sum_{sub \in SG} R_t^{sub} - V^{csc}(s_t) + \beta^{T_{\max}-t+1} V^{csc}(s_{T_{\max}}) \tag{9}$$

meaning its objective is making actions to maximize the future reward of the sub-policy group. To inspire policies to explore more, researchers use entropy regularization. The greater the entropy, the more ability of exploration a policy will have. Therefore, we follow the practice of using the entropy of policy to increase the exploration ability by:

$$\mathcal{L}_H(\theta_\pi) = -\sum_{t=1}^{T_{max}} \pi(a_t \mid s_t;\theta_\pi) \log \pi(a_t \mid s_t;\theta_\pi) \tag{10}$$

CSC-HSRL uses Monte Carlo sampling [25] to optimize the objective function of the sub-policies. Here, the overall loss of one sub-policy *sub* is:

$$\begin{aligned}\mathcal{L}_{sub}(\theta_{\pi^{sub}}) = &-\sum_{t=1}^{T_{\max}} \sum_{a_t^m = sub} \log \pi^{sub}(a_t^{sub} \mid s_t;\theta_{\pi^{sub}}) adv_t^{sub} \\ &+ \alpha L_H(\theta_{\pi^{sub}})\end{aligned} \tag{11}$$

where α is a weight parameter used to balance the two loss functions.

3.4 A2C for master policy

For the master policy, the loss function is calculated similarly:

$$\begin{aligned}\mathcal{L}_{\mathrm{m}}(\theta_{\pi^m}) = &-\sum_{t=1}^{T_{\max}} \log \pi^m(a_t^m \mid s_t;\theta_{\pi^m})(R_t^m - V^m(s_t)) \\ &+ \alpha L_H(\theta_{\pi^m})\end{aligned} \tag{12}$$

For the master critic $V^m(s_t)$, the loss function is formulated similarly. The only difference is that it minimizes the difference between the predicted value and the accumulated reward that the master policy will achieve in the following episodes:

$$\mathcal{L}_{\mathrm{m}}(\theta_{V^m}) = \sum_{t=1}^{T_{\max}} (R_t^m - V^m(s_t))^2 \tag{13}$$

where R_t^m is calculated as:

$$R_t^m = \begin{cases} r_t^m + \beta V^m(s_t) & t = T_{\max} \\ r_t^m + \beta R_{t-1}^m & t = 1, 2, \ldots, T_{\max} - 1 \end{cases} \tag{14}$$

in which master policy's reward function r_t^m:

$$r_t^m = \begin{cases} \mu + \Delta IoU & IoU_t = IoU_t^{\max} \\ IoU_t - IoU_t^{\max} + \Delta IoU & otherwise \end{cases} \tag{15}$$

where ΔIoU represents the reward based on the difference of IoU between the current state and the next state. $IoU_t - \mathrm{IoU}_t^{\max}$ and μ are the rewards to measure the decision of choosing a sub-policy. Note that for the master policy, its reward function should not only contain reward based on IoU but also reward of awarding choosing the most suitable sub-policy.

3.5 Joint training

Simultaneous training of the master module and the sub module will lead to the instability of the training process. This paper adopts the optimization method of training the two in turn (Wake-Sleep Mode): the master module and the sub module will be updated in turn for M iterations. The loss function of the reinforcement learning part of the model:

$$\mathcal{L}_{\mathrm{CSC-HSRL}} = \begin{cases} \mathcal{L}_{\mathrm{m}}(\theta_{\pi^m}) + \gamma \mathcal{L}_{\mathrm{m}}(\theta_{V^m}) \\ \sum_{sub \in SG} \mathcal{L}_{sub}(\theta_{\pi^{sub}}) + \gamma \mathcal{L}_{SG}(\theta_{V^{csc}}) \end{cases} \tag{16}$$

where γ is a weight parameter to control the update speed of value network. The master policy and the sub-policy group are alternately trained while maintaining the training stability.

In this paper, an additional supervised learning [12] is used to help the policy perceive boundaries. We follow the IoU prediction network based on the IoU and the loss function is:

$$\mathcal{L}_{\mathrm{SL}} = BCELoss(IoU_{t-1}, P_{IoU}^{(t)}) \tag{17}$$

where $P_{IoU}^{(t)}$ represents the IoU predicted by the SL prediction network. In inference phase, when the agent interacts with the environment $T_{\max}$ times, and obtains a series of $P_{IoU}^{(t)}$ predicted IoU. The agent takes the moment with the highest predicted $P_{IoU}^{(t)}$ as the best predicted moment.

The overall loss function in this paper is as follows:

$$\mathcal{L} = \mathcal{L}_{\text{CSC-HSRL}} + \lambda \mathcal{L}_{\text{SL}} \tag{18}$$

where λ is the weight parameter used to balance the two loss functions.

4 Experiments

In order to verify the effectiveness of our work, we evaluate our proposed CSC-HSRL method on Charades-STA [8] and ActivityNet [15].

4.1 Implementation details

In this paper, 0.25 and 0.75 of the original video duration are used as the initial normalized positions of the prediction boundary. The scale of boundary adjustment Z is set to 16 and 80 for Charades-STA and ActivityNet. We choose Two-Stream [29] and C3D [26] as the video feature extractor for Charades-STA and only C3D [26] for ActivityNet dataset. The size of GRU's middle hidden layer is set to 1024. The training is performed by Adam optimizer with a learning rate of 0.001. The reward parameter μ is set to 1, and the other hyperparameters α, β, γ and λ are set to 0.1, 0.4, 1 and 1. The master module and sub module are alternately trained every 200 rounds. We choose two wide-using metrics: "IoU@ϵ" and "mIoU". "IoU@ϵ" means the ratio of the predicted segment to the ground truth is higher than the ratio; "mIoU" means the average overlap ratio of the predicted segment to the ground truth.

Table 2 Performance comparison in ActivityNet [15]

Paradigm	Methods	IoU@0.3	IoU@0.5	IoU@0.7	mIoU
SL	ACRN	31.75	16.53	–	24.49
	QSPN	45.3	27.7	13.6	–
	ABLR	–	36.79	–	36.99
	MLVI	45.3	27.7	13.6	–
	SCDM	54.8	36.75	19.86	–
	TMLGA	–	33.04	19.26	–
	CBP	–	35.76	17.8	36.85
	ASST	–	37.04	18.04	–
RL	RWM	53.00	34.91	–	36.25
	Trip-Net	48.42	32.19	–	–
	MABAN	–	37.20	18.87	–
	TSP-PRL	**56.08**	**38.76**	–	**39.21**
	TSP-PRL*	53.99	36.89	20.16	37.67
	CSC-HSRL	55.43	37.57	**20.29**	38.44

"*" Denotes the result reproduced in our experiments

Table 3 Performance comparison in charades-STA [8]

Paradigm	Feature	Methods	IoU@0.5	IoU@0.7	mIoU
SL	C3D	ACRN	20.26	7.64	–
	C3D	ROLE	25.26	12.12	–
	C3D	SLTA	22.81	8.25	–
	C3D	MAC	30.48	12.2	–
	C3D	QSPN	35.6	15.8	–
	C3D	ABLR	24.36	9.01	–
	C3D	SAP	27.42	13.36	–
	C3D	MLVI	35.6	15.8	–
	C3D	CBP	36.8	18.87	35.74
RL	C3D	SM-RL	24.36	11.17	32.22
	C3D	RWM	34.12	13.74	35.09
	C3D	Trip-Net	36.61	14.5	–
	C3D	TSP-PRL	37.39	17.69	**37.22**
	C3D	TSP-PRL*	36.85	18.14	34.5
	C3D	CSC-HSRL	**38.23**	**18.92**	35.42
	Two-stream	RWM	37.23	17.72	36.22
	Two-stream	TSP-PRL	45.3	**24.73**	40.93
	Two-stream	TSP-PRL*	44.91	23.68	40.32
	Two-stream	CSC-HSRL	**46.48**	23.76	**41.11**

"*": denotes the result reproduced in our experiments

4.2 Comparison algorithm

The experimental results are shown in Tables 2 and 3. The best performance of the methods in same feature is highlighted in bold and the second-best underline. "*": denotes the result reproduced in our experiments. Other results are collected from their papers. ACRN [16], ROLE [17], QSPN [35], ABLR [38], SLTA [13], MAC [10], SAP [5], MLVI [36], SCDM [37], TMLGA [21], ASST [20], CBP [28], READ [12], SM-RL [30], Trip-Net [11], MABAN [24] and TSP-PRL [32] are compared methods. Since TSP-PRL [32] is our baseline model, we follow its released source code and re-implement it. For fair comparison, we use TSP-PRL's reproduced result, which is noted as TSP-PRL*. As shown in Tables 2 and 3, our proposed method CSC-HSRL achieves the competitive performance with both C3D and Two-streamed visual features on the Charades-STA dataset and Activitynet dataset. Compared with CBP [28] in C3D-based Charades-STA, our model is slightly lower than CBP in metric *mIoU*. This may be because CBP has finer feature fusion than RL based-methods. The method of this paper is based on hierarchical-structured reinforcement learning. Compared with the current hierarchical-structured RL method TSP-PRL* [32], our method improves the metric IoU@0.5, IoU@0.7, and mIoU by 1.57, 0.08, and 0.75 respectively in two-streamed Charades-STA dataset. For ActivityNet dataset, our method improves the metric IoU@0.3, IoU@0.5,

Table 4 Comparison of performance between CSC-HSRL and DSC-HSRL in Charades-STA

Methods	Charades-STA			
	IoU@0.1	IoU@0.3	IoU@0.5	IoU@0.7
DSC-HSRL-*It*@10	70.69	58.31	44.01	22.39
DSC-HSRL-*It*@20	70.13	58.76	44.91	23.68
DSC-HSRL-*It*@30	70.10	58.63	44.87	23.54
CSC-HSRL-*It*@10	70.89	60.16	45.53	22.39
CSC-HSRL-*It*@20	71.23	60.82	46.23	23.47
CSC-HSRL-*It*@30	**71.34**	**60.99**	**46.48**	**23.76**

Table 5 Comparison of performance between CSC-HSRL and DSC-HSRL in ActivityNet

Methods	ActivityNet			
	IoU@0.1	IoU@0.3	IoU@0.5	IoU@0.7
DSC-HSRL-*It*@10	71.71	53.62	36.45	19.98
DSC-HSRL-*It*@20	71.83	53.99	36.95	20.12
DSC-HSRL-*It*@30	72.22	54.03	37.05	20.16
CSC-HSRL-*It*@10	73.35	54.66	36.60	19.90
CSC-HSRL-*It*@20	73.41	55.43	37.50	20.24
CSC-HSRL-*It*@30	**73.85**	**55.52**	**37.58**	**20.29**

IoU@0.7, and mIoU by 1.44, 0.68, 0.13, and 0.77 respectively compared with TSP-PRL* [32].

4.3 Comparison between decentralized sub-critic and centralized sub-critic

To verify the effectiveness of the centralized sub-critic network in the hierarchical-structure RL paradigm, this paper selects the decentralized-sub-critic-based method TSP-PRL* [32] (marked as DSC-HSRL) as a comparison.

Analysis of localization performance

We compare the accuracy under different $T_{\max}$ (maximum number of steps) in Charades-STA and ActivityNet. The experimental results are shown in Tables 4 and 5 and the best performance is highlighted in bold. It@10, It@20, and It@30 indicate that the maximum number of steps the model can make decisions is 10, 20 and 30. The accuracy of the prediction in this paper is higher than the DSC-HSRL-based model in the case of different $T_{\max}$. Take the result in Charades-STA as an example. In the first 10 steps, our model achieves 45.53 on IoU@0.5, which is higher than 44.01 of the DSC-HSRL-based model when $T_{\max}$ is 10, 20 and 30. This indicates that our model can adjust the prediction boundary quickly to approach the target segment.

When $T_{\max}$ increases to 20 and 30, CSC-HSRL's prediction accuracy increases: the matrix IoU@0.5 and IoU@0.7 are increased by 0.7, 1.08, and 0.25, 0.29, which indicates

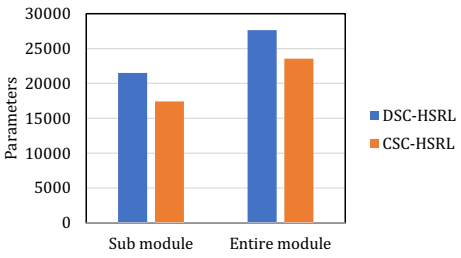

Fig. 3 Comparison of parameters numbers between DSC-HSRL and CSC-HSRL

Table 6 Comparison of time–cost of inference between CSC-HSRL and DSC-HSRL

Dataset	Methods	Number of inference steps in average			
		IoU@0.9	IoU@0.7	IoU@0.5	*Stop*
Activitynet	DSC-HSRL	6.33	4.52	3.51	11.02
	CSC-HSRL	**5.93**	**4.15**	**3.18**	**10.15**
Charades-STA	DSC-HSRL	8.79	7.07	5.02	11.61
	CSC-HSRL	**7.43**	**5.87**	**4.21**	**8.63**

that our model's strategy is in the correct direction, and the prediction boundary slowly approaches the target segment with subsequent adjustments. In contrast, although the DSC-HSRL is higher than the CSC-HSRL in IoU@0.7 in the first 10 steps, its accuracy drops in the subsequent decisions by 20 steps. It indicates that the adjustment strategy of DSC-HSRL has not reached the optimal level. The process produces fluctuations. This shows that during the training process, the centralized sub-critic method CSC-HSRL has a more comprehensive perception of the changes in the environment, which helps the sub-policy group learn the correct adjustment strategy.

Analysis of the parameters numbers of Agent Methods based on reinforcement learning contain a state encoder and a decision-making network. The method in this paper mainly aims at the decision inference part of models, that is, the improvement of master module and sub module. Therefore, we compare the amount of decision network parameters. We calculate the decision network parameters of the DSC-HSRL model based on decentralized sub-critic and our CSC-HSRL model based on centralized sub-critic, as shown in Fig. 3. Experiments show that, compared with the DSC-HSRL, the CSC-HSRL in this paper reduces the parameters of the sub module network by 19.05 % and the parameters of the entire decision network by 14.82 % while improving the overall prediction accuracy.

Analysis of speed and efficiency in inference During inference, the speed and efficiency of temporal sentence grounding is determined by two modules: state encoder, decision

module. Although the state encoder takes an important role in this process, which is the key to fuse the multi-modal information, the decision module is the core part to achieve the task. In this paper, we propose a more reasonable and efficient decision module which gains better and faster performance. Table 6 illustrates the inferring time–cost of CSC-HSRL and DSC-HSRL and the least steps of reaching IoU@n is highlighted in bold. Since RL based methods need take a few steps to adjust the predicting boundary and determine the prediction by *Stop*, we analyse their speed of inference by comparing how many steps they reach the situation where IoU is larger than n in average. As shown in Table 6, compared with DSC-HSRL, CSC-HSRL takes fewer steps to reach the same IoU and reach its determination stage *Stop* earlier, which accelerate the inference to a certain extent.

Analysis of environment sensing To verify the effectiveness of centralized sub-critic which provides finer environment perception, we accumulate how many times their reach the situation where IoU is larger than n during inference on one test set in Table 7, where the most times of reaching IoU@n is highlighted in bold. Reaching the situation of high IoU frequently represents that the model has good sense of locating target segment. For the sake of fairness, we experiment on the same test sets of two datasets to ensure the same test samples. Obviously, CSC-HSRL reaches the same status more times than DSC-HSRL, which means CSC-HSRL has greater ability of distinguish the target segment from irrelevant segment. The reason behind this is that centralized sub-critic estimates the trend of environment better than the decentralized one.

Table 7 Comparison of times of reaching IoU@n during inference between CSC-HSRL and DSC-HSRL

Dataset	Methods	Times of reaching IoU@n		
		IoU@0.9	IoU@0.7	IoU@0.5
Activitynet	DSC-HSRL	2728	6959	9327
	CSC-HSRL	**3097**	**7912**	**10532**
Charades-STA	DSC-HSRL	497	1460	2343
	CSC-HSRL	**510**	**1556**	**2406**

4.4 Analysis of reward factor

Since our decentralized sub-critic estimates the environment by predicting the sub-policy group's accumulated reward, we investigate the sensibility of the reward factor μ. The reward factor μ is the measurement of how much each action affects, deciding the pace at which we train policies. The parameter tuning results of μ are revealed in Fig.4. Experimental results show that the model with μ equal to 1 has the best performance. Except from IoU@0.7-It@20, models with μ closer to 1 perform better.

4.5 Localizing visualization

To get a visible representation of our method's effectiveness, we conduct some case studies. In particular, we choose two video-query pairs which were cast into DSC-HSRL and CSC-HSRL to observe their localizing process and performance. As mentioned before, we choose TSP-PRL* [32] to represent the DSC-HSRL-based model. The green arrows with lines are the ground truth of this sample. The red arrows with lines represent the best prediction made by models. Figure 5 indicates that CSC-HSRL is more effective than TSP-PRL no matter of performance or process. Take the first pair as an example. CSC-HSRL reaches its best performance at the 11th time step whose IoU is 0.90, and its localization precision increases along with the time step. By contrast, TSP-PRL only gets the largest IoU of 0.69 at the 7th time step and struggles in the next time steps, demonstrating that its decentralized sub-critics mislead its sub-policies about the environment. It could be easily captured that CSC-HSRL adjusts better than TSP-PRL at the 7th time step when they face the same state.

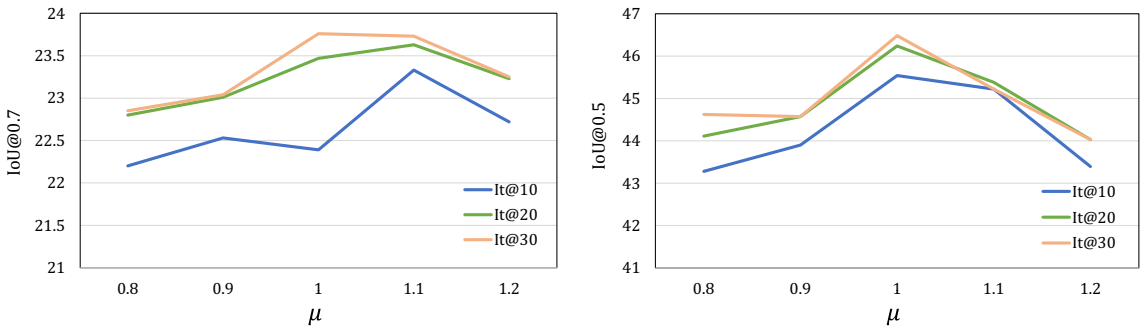

Fig. 4 Performance of CSC-HSRL w.r.t the reward factor μ. The lines in different colours represent the performance in different maximum steps

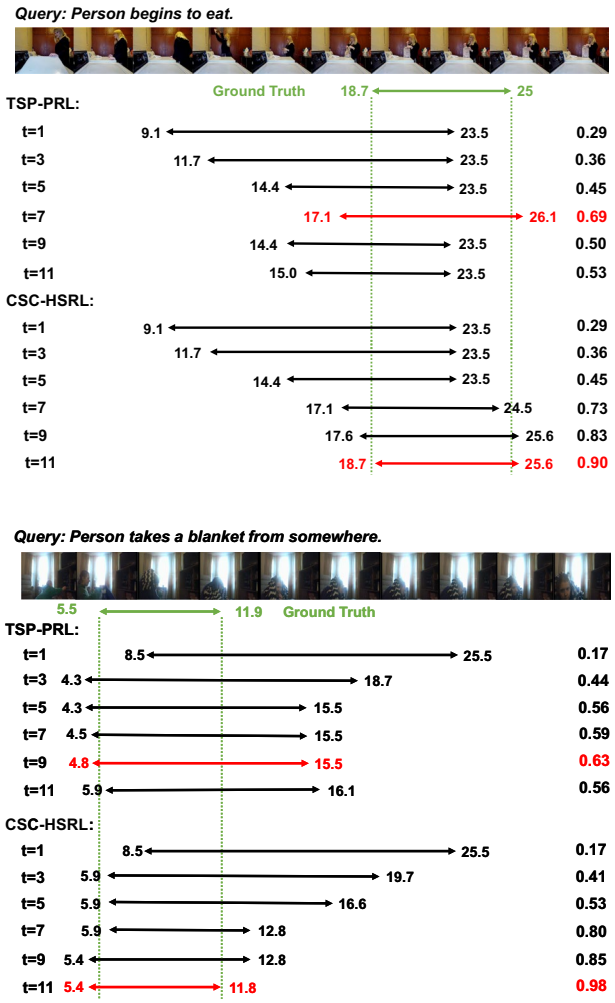

Fig. 5 Visualization of the localization process

5 Conclusion

The present hierarchical-structured-RL-based method remains the problem that sub-policy cannot model others' impacts and contains massive network's parameters. To handle this, we propose a centralized sub-critic based hierarchical-structured reinforcement learning. Specifically, we train a centralized sub-critic to evaluate the effect of all sub-policies' actions, thus helping them learn their strategies according to the overall environment changes. Meanwhile, the centralization of sub-critics decreases the number of sub-critic networks, alleviating the stress of the network's parameters. A large number of experimental results show that our method makes the policies' adjustment more reasonable and improves the accuracy of grounding while reducing the parameters of the agent network.

Acknowledgements Thiswork was supported by National Key Research and Development Project (No.2020AAA0106200), the National Nature Science Foundation of China under Grants (No.61936005, 61872424), and the Natural Science Foundation of Jiangsu Province (Grants No. BK20200037). And the Natural Science Foundation of Jiangsu Province (Grants No. BK20200037 and BK20210595) and the Open Project of Anhui Provincial Key Laboratory of Multimodal Cognitive Computation, Anhui University (Grant No MMC202010).

Author contributions YZ performed the experiment and wrote the main manuscipt text and ZT, ZT and B-KB guided and modified this manuscipt. All authors reviewed the manuscript.

Data availability The data that support this study are available in Charades-STA at https://doi.org/10.1109/ICCV.2017.563 and ActivityNet dataset at https://doi.org/10.1109/ICCV.2017.83. These data were derived from the following resources available in the public domain: https://github.com/jiyanggao/TALL.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Pieter Abbeel, O., Zaremba, W.: Hindsight experience replay. Advances in neural information processing systems **30** (2017)
2. Anne Hendricks, L., Wang, O., Shechtman, E., Sivic, J., Darrell, T., Russell, B.: Localizing moments in video with natural language. In: Proceedings of the IEEE international conference on computer vision, pp. 5803–5812 (2017)
3. Bacon, P.L., Harb, J., Precup, D.: The option-critic architecture. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)
4. Chaplot, D.S., Sathyendra, K.M., Pasumarthi, R.K., Rajagopal, D., Salakhutdinov, R.: Gated-attention architectures for task-oriented language grounding. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
5. Chen, S., Jiang, Y.G.: Semantic proposal for activity localization in videos via sentence query. Proc AAAI Conf Artif Intell **33**, 8199–8206 (2019)
6. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
7. Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., Whiteson, S.: Counterfactual multi-agent policy gradients. In: Proceedings of the AAAI conference on artificial intelligence, vol. 32 (2018)
8. Gao, J., Sun, C., Yang, Z., Nevatia, R.: Tall: Temporal activity localization via language query. In: Proceedings of the IEEE international conference on computer vision, pp. 5267–5275 (2017)
9. Gao, J., Xu, C.: Fast video moment retrieval. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1523–1532 (2021)
10. Ge, R., Gao, J., Chen, K., Nevatia, R.: Mac: Mining activity concepts for language-based temporal localization. In: 2019 IEEE winter conference on applications of computer vision (WACV), pp. 245–253. IEEE (2019)
11. Hahn, M., Kadav, A., Rehg, J.M., Graf, H.P.: Tripping through time: Efficient localization of activities in videos. arXiv preprint arXiv:1904.09936 (2019)

12. He, D., Zhao, X., Huang, J., Li, F., Liu, X., Wen, S.: Read, watch, and move: Reinforcement learning for temporally grounding natural language descriptions in videos. Proc AAAI Conf Artif Intell **33**, 8393–8400 (2019)
13. Jiang, B., Huang, X., Yang, C., Yuan, J.: Cross-modal video moment retrieval with spatial and language-temporal attention. In: Proceedings of the 2019 on international conference on multimedia retrieval, pp. 217–225 (2019)
14. Kiros, R., Zhu, Y., Salakhutdinov, R.R., Zemel, R., Urtasun, R., Torralba, A., Fidler, S.: Skip-thought vectors. Advances in neural information processing systems **28** (2015)
15. Krishna, R., Hata, K., Ren, F., Fei-Fei, L., Carlos Niebles, J.: Dense-captioning events in videos. In: Proceedings of the IEEE international conference on computer vision, pp. 706–715 (2017)
16. Liu, M., Wang, X., Nie, L., He, X., Chen, B., Chua, T.S.: Attentive moment retrieval in videos. In: The 41st international ACM SIGIR conference on research & development in information retrieval, pp. 15–24 (2018)
17. Liu, M., Wang, X., Nie, L., Tian, Q., Chen, B., Chua, T.S.: Cross-modal moment localization in videos. In: Proceedings of the 26th ACM international conference on Multimedia, pp. 843–851 (2018)
18. Lowe, R., Wu, Y.I., Tamar, A., Harb, J., Pieter Abbeel, O., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. Advances in neural information processing systems **30** (2017)
19. Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. In: International conference on machine learning, pp. 1928–1937. PMLR (2016)
20. Ning, K., Cai, M., Xie, D., Wu, F.: An attentive sequence to sequence translator for localizing video clips by natural language. IEEE Transact Multimedia **22**(9), 2434–2443 (2019)
21. Rodriguez, C., Marrese-Taylor, E., Saleh, F.S., Li, H., Gould, S.: Proposal-free temporal moment localization of a natural-language query in video using guided attention. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 2464–2473 (2020)
22. Ryu, H., Kang, S., Kang, H., Yoo, C.D.: Semantic grouping network for video captioning. Proc AAAI Conf Artificial Intell **35**, 2514–2522 (2021)
23. Su, J., Adams, S., Beling, P.: Value-decomposition multi-agent actor-critics. Proc AAAI Conf Artif Intell **35**, 11352–11360 (2021)
24. Sun, X., Wang, H., He, B.: Maban: Multi-agent boundary-aware network for natural language moment retrieval. IEEE Transact Image Proc **30**, 5589–5599 (2021)
25. Sutton, R.S., Barto, A.G.: Reinforcement learning: an introduction. MIT press (2018)
26. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pp. 4489–4497 (2015)
27. Vezhnevets, A.S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., Kavukcuoglu, K.: Feudal networks for hierarchical reinforcement learning. In: International Conference on Machine Learning, pp. 3540–3549. PMLR (2017)
28. Wang, J., Ma, L., Jiang, W.: Temporally grounding language queries in videos by contextual boundary-aware prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 12168–12175 (2020)
29. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Gool, L.V.: Temporal segment networks: Towards good practices for deep action recognition. In: European conference on computer vision, pp. 20–36. Springer (2016)
30. Wang, W., Huang, Y., Wang, L.: Language-driven temporal activity localization: A semantic matching reinforcement learning model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 334–343 (2019)
31. Wang, X., Chen, W., Wu, J., Wang, Y.F., Wang, W.Y.: Video captioning via hierarchical reinforcement learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4213–4222 (2018)
32. Wu, J., Li, G., Liu, S., Lin, L.: Tree-structured policy based progressive reinforcement learning for temporally language grounding in video. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 12386–12393 (2020)
33. Wu, W., He, D., Tan, X., Chen, S., Wen, S.: Multi-agent reinforcement learning based frame sampling for effective untrimmed video recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6222–6231 (2019)
34. Xiao, S., Chen, L., Shao, J., Zhuang, Y., Xiao, J.: Natural language video localization with learnable moment proposals. arXiv preprint arXiv:2109.10678 (2021)
35. Xu, H., He, K., Plummer, B.A., Sigal, L., Sclaroff, S., Saenko, K.: Multilevel language and vision integration for text-to-clip retrieval. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 9062–9069 (2019)
36. Xu, H., He, K., Plummer, B.A., Sigal, L., Sclaroff, S., Saenko, K.: Multilevel language and vision integration for text-to-clip retrieval. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 9062–9069 (2019)
37. Yuan, Y., Ma, L., Wang, J., Liu, W., Zhu, W.: Semantic conditioned dynamic modulation for temporal sentence grounding in videos. Advances in Neural Information Processing Systems **32** (2019)
38. Yuan, Y., Mei, T., Zhu, W.: To find where you talk: Temporal sentence localization in video with attention based location regression. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 9159–9166 (2019)