Supplementary Material for: Can Shuffling Video Benefit Temporal Bias Problem: A Novel Training Framework for Temporal Grounding

Jiachang Hao, Haifeng Sun*, Pengfei Ren, Jingyu Wang*, Qi Qi, and Jianxin Liao

State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications
{haojc, hfsun, rpf, wangjingyu, qiqi8266}@bupt.edu.cn jxlbupt@gmail.com

In the supplementary material, we present

1. the details of our baseline grounding model in Sec. A
2. the details of our implementation in Sec. B
3. the computational complexity of our method in Sec. C
4. the performance on original splits in Sec. D
5. sanity check on visual input of our baseline grounding model in Sec. E
6. length distributions of the predicted moments in Sec. F
7. ablation study of the temporal gating in Sec. G
8. more visualization examples in Sec. H

A Baseline Grounding Model Architecture

Our baseline grounding model applies the architecture proposed by [6]. As illustrated in Fig. a, our baseline consists of three componets, namely the query encoder, the video encoder, and the span predictor.

A.1 Query Encoder

Query encoder models the sentence query with multi-layered bidirectional LSTM with a pre-trained language model (GloVe [10]) embeddings as input,

$$w_n = \text{BiLSTM}(q_n, w_{n-1}). \tag{1}$$

The encoded word-level embeddings are denoted as $W = \{w_n\}_{n=1}^{N}$. Same as [5], we use the last hidden state of the bidirectional LSTM as the global sentence-level representation s.

* Corresponding author

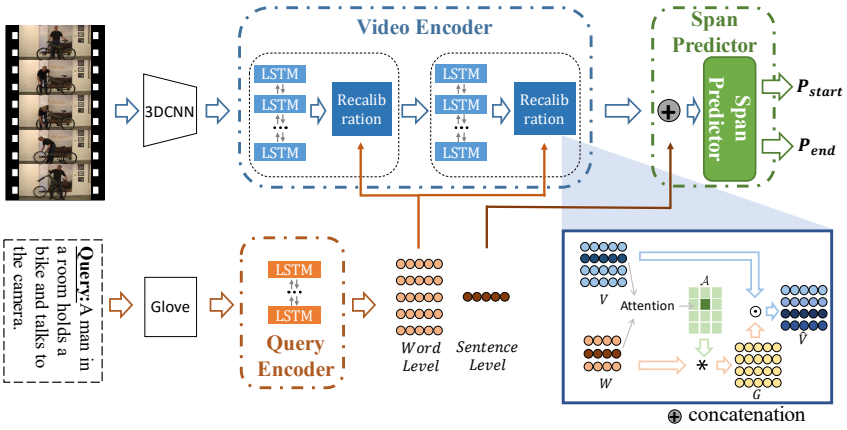

Fig. a: An overview of our span-based grounding model. Query encoder models the sentence query. Video encoder is guided by word-level query features to encode video features over time. The word-level query features are used to recalibrate the frame features from the BiLSTM layers. The span predictor predicts the boundary scores for each frame. The feature extractor is fixed during training.

A.2 Vidoe Encoder

Video encoder models the video modality with multi-layered bidirectional LSTM with the features extraxted from a pre-trained video classification model (C3D [12] or I3D [1]) as input. As shown in Fig. a, to highlight the query-related features, we leverage the word-level query features W to recalibrate each frame feature encoded by BiLSTM, same as [6].

Formally, given query features $W = \{w_i\}_{i=1}^{N}$ and frame features $V = \{v_t\}_{t=1}^{T}$, where N is the number of words, we first compute a frame-to-word attention map $\mathcal{A}$, as same as [16,2]. For the frame at the time step t and the $i-$th word in the query, the attention score is calculated as,

$$A_i^t = p^T(\tanh(W^V v_t + W^Q w_i + b)), \tag{2}$$

$$\mathcal{A}_i^t = \text{softmax}_i(A_i^t), \tag{3}$$

where $W^V \in \mathbb{R}^{d_v \times d_v}$ and $W^Q \in \mathbb{R}^{d_v \times d_q}$ are the projection matrices, $p \in \mathbb{R}^{d_v}$ projects the vector to a scalar, and d_v and d_q are dimensions of frame and word features, respectively. A column-wise softmax is applied on $A \in \mathbb{R}^{T \times N}$ to obtain the attention map $\mathcal{A}$.

Then, for each frame feature v_t, we attentively summarize the word features to generate the specific channel-level gating weights G_t. Finally, we gate the raw frame features $V = \{v_t\}_{t=1}^{T}$ with the generated gating weights G_t to obtain the recalibrated frame features $\hat{V} = \{\hat{v}_t\}_{t=1}^{T}$ as follows:

$$g_t = \sum_{i}^{n} \mathcal{A}_i^t c_i \in \mathbb{R}^{d_q}, \tag{4}$$

$$G_t = \text{sigmoid}(W^G g_t) \in \mathbb{R}^{d_v}, \tag{5}$$

$$\hat{v}_t = G_t \odot v_t \in \mathbb{R}^{d_v}. \tag{6}$$

A.3 Span Predictor

We use two multi-layered perceptrons (MLPs) with tanh activation in hidden layers as the span predictor. At each time step t, we first concatenate the frame feature $\dot{v}_t$ with the sentence-level query representation S and then pass the concatenated feature into the span predictor to obtain the start and end scores $S_{start}(t)$, $S_{end}(t)$ as follows,

$$S_{start}(t) = W_2^s \text{tanh}(W_1^s(\dot{v}_t||s) + b_1^s) + b_2^s, \tag{7}$$

$$S_{end}(t) = W_2^e \text{tanh}(W_1^e(\dot{v}_t||s) + b_1^e) + b_2^e, \tag{8}$$

where W_1^s, W_2^s, b_1^s and b_2^s are the learnable parameters of MLPs and shared across all time steps.

A.4 Inference

Given the video duration is $\mathcal{T}$ and the video has T frames, the start and end frame indices are calculated by $t^{s(e)} = \langle \tau^{s(e)}/\mathcal{T} \times T \rangle$, where $\langle \cdot \rangle$ denotes the rounding operation. During inference, we predict the span of frames $(\hat{t}^s, \hat{t}^e)$ for each sentence query by maximizing the joint probability of start and end frames:

$$\begin{aligned} \text{span}(\hat{t}^s, \hat{t}^e) &= \underset{\hat{t}^s, \hat{t}^e}{\text{argmax}} P_{start}(\hat{t}^s) P_{end}(\hat{t}^e) \\ &= \underset{\hat{t}^s, \hat{t}^e}{\text{argmax}} S_{start}(\hat{t}^s) + S_{end}(\hat{t}^e) \\ &\textbf{s.t. } \hat{t}^s < \hat{t}^e. \end{aligned} \tag{9}$$

Then we map the predicted span of frames back to the time values by $\hat{\tau}^{s(e)} = \hat{t}^{s(e)}/T \times \mathcal{T}$.

B Implementation Details

Our experiments are conducted within PyTorch1.6 on a computer with an AMD Ryzen 9 3900X 3.80GHz CPU, 64GB of RAM, and one Nvidia 2080TI GPU.

We conduct hyperparameters search on the validation sets for all dataset splits. (As the original split of Charades-STA does not have a validation set, we

Table I: Running times on one RTX 2080Ti GPU.

Mode	Method	Charades-STA		ActivityNet Captions	
		Memory	Time/batch	Memory	Time/batch
Train (batch size=32)	baseline	1705MB	0.1066s	2529MB	0.2012s
	ours	2483MB	0.2415s	4373MB	0.4090s
Infer (batch size=1)	baseline	957MB	0.0163s	959MB	0.0261s
	ours	957MB	0.0184s	963MB	0.0284s

Table II: Parameter size comparison between various models.

Models	2DTAN	LG	DCM	Baseline	Ours
Params(M)	59.353	33.411	28.42	12.268	13.846

use the same hyperparameter setting searched on the validation set of Charades-CD split.)

For natural language modality, the vocabulary sizes are 1,294 and 13,745 for Charades-STA and ActivityNet Captions datasets, respectively. We truncate all sentence queries with a maximum of 25 words for ActivityNet Captions dataset and 15 for Charades-STA dataset. The number of LSTM layers in the query encoder is 2, and the dimension of the hidden layers is set to 256 for both datasets.

For video modality, we set the length of video feature sequences to 128 for Charades-STA and 240 for ActivityNet Captions, respectively. Longer is truncated, and shorter is padded. Video encoder consists of four LSTM layers, and we recalibrate the video features once after every two lstm layers. The dimension of the hidden layers in LSTM is set to 256 for both datasets. A layer normalization is performed on the encoded video features before they are feed forward to the span predictor.

The dimension of the hidden layers in the span predictor is 256. The dimension of the hidden layers in cross-modal semantic matching module and temporal order discriminator is 1024.

We decay the learning rate to one-tenth of the original at epoch 20 for Charades-STA and epoch 15 for ActivityNet Captions, respectively.

C Computational Complexity

We report the comparsion results between the baseline and our method on the running time and memory size in Table I. As shown in Table I, our method increases the computional complexity during training stage. The major computational cost is from the video encoding procedure. Compared to the baseline, our method needs to encode both original and shuffled videos, which means that our method perform the video encoding twice. During inference stage, our method achieve similar inference speed and memory cost to the baseline. Because our method only needs to encode videos once as same as the baseline.

Table III: Comparison with state-of-the-art methods on Charades-STA with the original split.

Model	Feature	IoU=0.5	IoU=0.7	mIoU
CTRL [4]	c3d	23.63	8.89	-
ABLR [17]	c3d	24.36	9.01	-
CBP [13]	c3d	36.8	18.87	35.74
PMI-LOC [2]	c3d	39.73	19.27	-
HVTG [3]	-	47.27	23.3	-
DRN [18]	i3d	53.09	31.75	-
ExCL [5]	i3d	44.1	23.3	-
TMLGA [11]	i3d	52.02	33.74	-
LG [8]	i3d	59.46	35.48	51.38
VSLNet [19]	i3d	54.19	35.22	50.02
CBLN [7]	i3d	**61.13**	38.22	-
DeNet [23]	i3d	59.70	38.52	-
CPN [22]	i3d	56.70	36.62	**53.14**
Baseline	i3d	55.40	36.51	50.61
Ours	i3d	57.69	**39.27**	51.65

We report the parameter size comparison in Tab. II. Compared to the baseline, our model has only a slight increase. Because our added CSMM and TOD modules are achieved by two simple and light two-layered MLPs, respectively. Compared to the prior arts, our model is much smaller. It shows that the performance boost is from the de-bias design of our approach.

D Performance on Original Splits

Table III and Table IV summarize the results of our method on the original splits of Charades-STA and ActivityNet Captions, respectively. Our method achieves competitive performances on both datasets, especially on Charades-STA. For instance, our method performs the best in IoU=0.7 and second best in mIoU.

However, the original splits have some drawbacks. First, Charades-STA does not have a validation set. Many methods conduct hyperparameters search on the test set. This setting only measures the capability of models to overfit the test set, and no generalization measurement is performed. Second, the samples in test and training sets have similar temporal distributions in both datasets. Therefore, if a model is proficient in the temporal biases of queries in the training set, it can achieve state-of-the-art performance on the test set [9,15]. For instance, some methods [20,19,8] do not make any use of visual input but achieve the state-of-the-art performance on ActivityNet Captions as shown in **Sanity check on visual input** in our paper. So the performance on the original splits cannot fully reflect the model's ability to address the temporal grounding task.

Table IV: Comparison with state-of-the-art methods on ActivityNet Captions with the original split using C3D features.

Model	IoU=0.3	IoU=0.5	IoU=0.7	mIoU
ABLR [17]	55.67	36.79	-	36.99
QSPN [14]	52.13	33.26	13.43	-
CMIN [21]	63.61	43.40	23.88	-
2D-TAN [20]	59.45	44.51	27.38	-
DRN [18]	-	45.45	24.36	-
LG [8]	58.52	41.51	23.07	41.13
HTVG [3]	57.60	40.15	18.27	-
CBLN [7]	**66.34**	**48.12**	27.60	-
DeNet [23]	61.93	43.79	-	-
CPN [22]	62.81	45.10	**28.10**	**45.70**
Baseline	60.56	42.55	25.36	43.54
Ours	62.13	43.11	25.84	44.14

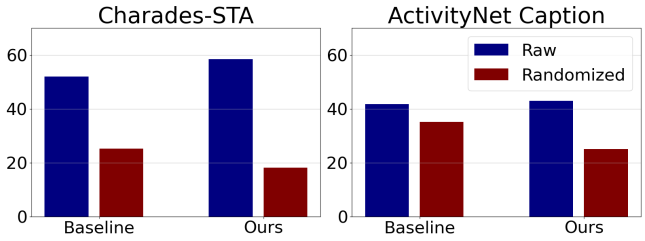

Fig. b: R@1 (IoU=0.5) scores for baseline and our method when the original input videos and randomized ones are fed into these models.

E Sanity Check on Visual Input of Baseline Grounding Model

We also test the sanity check on visual input on the baseline as same as the experiment **Sanity check on visual input** in our paper. The results are shown in Fig. b. On Charades-STA, the baseline shows a significant performance drop for randomized videos as same as most methods. On ActivityNet Captions, the baseline's score for randomized input videos is on a par with the original input video, which shows that the baseline also suffers the temporal bias problem on this dataset. Compared to the baseline, our method shows more significant drops on both datasets, especially on ActivityNet Captions. It demonstrates the effectiveness of our method for mitigating the reliance on temporal biases.

F Length Distribution of Predicted Moments

In the ablation study **Loss terms** in our paper, we study the impact of each loss term on the grounding model. Here we give the distribution of the length of

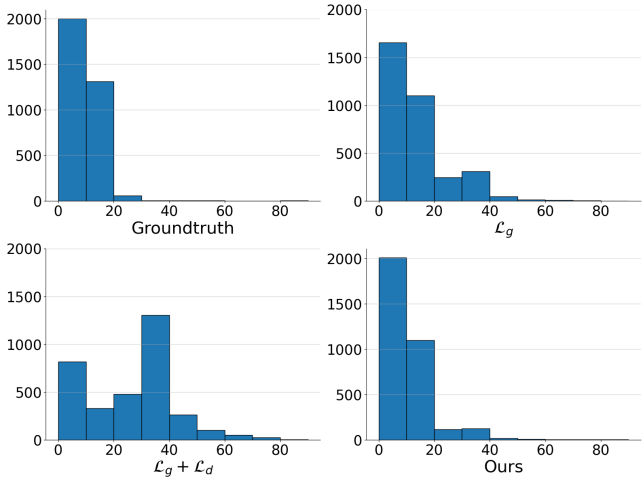

Fig. c: Comparison of the histogram of the moment duration on test-ood of Charades-CD. Horizontal axis denotes the moment duration (s). Vertical axis denotes the number of predicted moments. The top left is the groundtruth. The top right is the prediction of the model supervised by only grounding loss $\mathcal{L}_d$. The bottom left is the prediction of the model adding the discriminator loss $\mathcal{L}_d$. The bottom right is the prediction of our method.

predictions in Fig. c. As shown in Fig. c, compared to the model supervised by only the grounding loss $\mathcal{L}_g$, adding the discrimination loss $\mathcal{L}_d$ makes the model predict much more long-span locations. Specifically, nearly 1,300 predictions span 30s to 40s while the the average length of videos and target moments in Charades-STA are 30s and 8s, respectively. Thus adding the $\mathcal{L}_d$ makes the model perform poorly on the high IoU as shown Table 5 in our paper. It is worthy noting that, with the combination of loss terms, the length distribution of our method gets similar to the groundtruth. It shows that our cross-modal matching task can work well with the temporal order discrimination task and prevent the model from the degradation due to the lack of temporal position information.

G Ablation Study about Temporal Gating

In our method, we use the cross-modal semantic matching scores to temporally gate the encoded frame features. The temporal gating and the supervision on the temporal gating scores are widely used by temporal grounding methods [17,19,11]. Here we study the effects of these operations. We first apply temporal gating (TemG) on the baseline. However, as shown in Table V, we find that only applying the temporal gating (Baseline + TemG) cannot bring performance improvement from the baseline and even gets inferior to the baseline. With the supervison $\mathcal{L}_{intra}$ on the matching scores (only the part of the original

Table V: Effect of temporal gating on Charades-CD.

Model	test-ood			
	IoU=0.3	IoU=0.5	IoU=0.7	mIoU
Baseline	58.96	38.22	20.50	39.52
Baseline + TemG	54.61	29.75	14.90	35.17
Baseline + TemG + $\mathcal{L}_{intra}$	59.82	42.81	23.91	41.36
Ours w/o TemG	62.03	44.33	24.76	42.66
Ours	64.95	46.67	27.08	44.30

video), the temporal gating (Baseline + TemG + $\mathcal{L}_{intra}$) makes improvement from the baselines but there are clear margins over all metrics compared to our methods. It shows that the root of the success of our method is not simply from the temporal gating operation and its supervision but from our design of mining the content consistency between shuffled and original videos. We also test our methods without temporal gating. As shown in Table V, all metrics, especially for IoU=0.7, have performance drops, which verifies the effectiveness of that we use the matching scores to highlight the impact of the matching results on the final reasoning.

H Visualization Examples

Figure d shows some words' distribution maps and the corresponding grounding example results produced by the baseline and our methods. Left is the words' probability distributions of temporal locations on the training set of Charades-CD. Color represents value of probability. Right is the examples of grounding results for the queries containing one of the words.

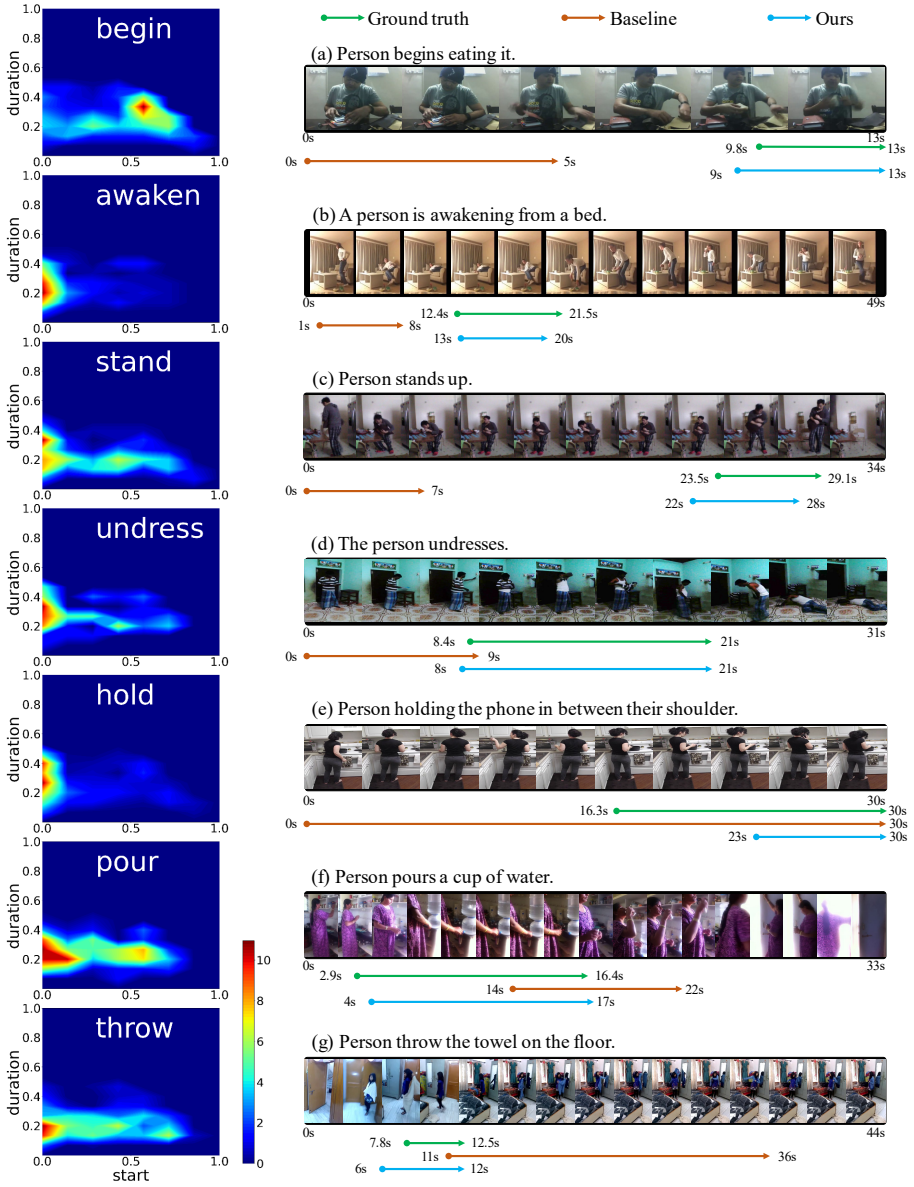

Fig. d: Qualitative Results on Charades-CD.

References

1. Carreira, J., Zisserman, A.: Quo vadis, action recognition? A new model and the kinetics dataset. In: CVPR (2017) 2
2. Chen, S., Jiang, W., Liu, W., Jiang, Y.: Learning modality interaction for temporal sentence localization and event captioning in videos. In: ECCV (2020) 2, 5
3. Chen, S., Jiang, Y.G.: Hierarchical visual-textual graph for temporal activity localization via language. In: ECCV (2020) 5, 6
4. Gao, J., Sun, C., Yang, Z., Nevatia, R.: TALL: temporal activity localization via language query. In: ICCV (2017) 5
5. Ghosh, S., Agarwal, A., Parekh, Z., Hauptmann, A.G.: Excl: Extractive clip localization using natural language descriptions. In: NAACL (2019) 1, 5
6. Hao, J., Sun, H., Ren, P., Wang, J., Qi, Q., Liao, J.: Query-aware video encoder for video moment retrieval. Neurocomputing (2022) 1, 2
7. Liu, D., Qu, X., Dong, J., Zhou, P., Cheng, Y., Wei, W., Xu, Z., Xie, Y.: Context-aware biaffine localizing network for temporal sentence grounding. In: CVPR. pp. 11235–11244 (2021) 5, 6
8. Mun, J., Cho, M., Han, B.: Local-global video-text interactions for temporal grounding. In: CVPR (June 2020) 5, 6
9. Otani, M., Nakashima, Y., Rahtu, E., Heikkilä, J.: Uncovering hidden challenges in query-based video moment retrieval. In: BMVC (2020) 5
10. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: EMNLP (2014) 1
11. 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: WACV (2020) 5, 7
12. Tran, D., Bourdev, L.D., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: ICCV. pp. 4489–4497 (2015) 2
13. Wang, J., Ma, L., Jiang, W.: Temporally grounding language queries in videos by contextual boundary-aware prediction. In: AAAI (2020) 5
14. Xu, H., He, K., Plummer, B.A., Sigal, L., Sclaroff, S., Saenko, K.: Multilevel language and vision integration for text-to-clip retrieval. In: AAAI (2019) 6
15. Yuan, Y., Lan, X., Chen, L., Liu, W., Wang, X., Zhu, W.: A closer look at temporal sentence grounding in videos: Datasets and metrics. CoRR **abs/2101.09028** (2021), https://arxiv.org/abs/2101.09028 5
16. Yuan, Y., Ma, L., Wang, J., Liu, W., Zhu, W.: Semantic conditioned dynamic modulation for temporal sentence grounding in videos. In: NIPS (2019) 2
17. Yuan, Y., Mei, T., Zhu, W.: To find where you talk: Temporal sentence localization in video with attention based location regression. In: AAAI (2019) 5, 6, 7
18. Zeng, R., Xu, H., Huang, W., Chen, P., Tan, M., Gan, C.: Dense regression network for video grounding. In: CVPR (June 2020) 5, 6
19. Zhang, H., Sun, A., Jing, W., Zhou, J.T.: Span-based localizing network for natural language video localization. In: ACL (2020) 5, 7
20. Zhang, S., Peng, H., Fu, J., Luo, J.: Learning 2d temporal adjacent networks for moment localization with natural language. In: AAAI (2020) 5, 6
21. Zhang, Z., Lin, Z., Zhao, Z., Xiao, Z.: Cross-modal interaction networks for query-based moment retrieval in videos. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019. pp. 655–664 (2019) 6

22. Zhao, Y., Zhao, Z., Zhang, Z., Lin, Z.: Cascaded prediction network via segment tree for temporal video grounding. In: CVPR. pp. 4197–4206 (2021) 5, 6
23. Zhou, H., Zhang, C., Luo, Y., Chen, Y., Hu, C.: Embracing uncertainty: Decoupling and de-bias for robust temporal grounding. In: CVPR. pp. 8445–8454 (2021) 5, 6