Towards a Complete Benchmark on Video Moment Localization

Jinyeong Chae[1*] **Donghwa Kim**[1*] **Kwanseok Kim**[1] **Doyeon Lee**[1] **Sangho Lee**[1]
Seongsu Ha[1] **Jonghwan Mun**[2] **Woo-Young Kang**[2] **Byungseok Roh**[2] **Joonseok Lee**[1,3†]
[1]Seoul National University [2]Kakao Brain [3]Google Research

Abstract

In this paper, we propose and conduct a comprehensive benchmark on moment localization task, which aims to retrieve a segment that corresponds to a text query from a single untrimmed video. Our study starts from an observation that most moment localization papers report experimental results only on a few datasets in spite of availability of far more benchmarks. Thus, we conduct an extensive benchmark study to measure the performance of representative methods on widely used 7 datasets. Looking further into the details, we pose additional research questions and empirically verify them, including if they rely on unintended biases introduced by specific training data, if advanced visual features trained on classification task transfer well to this task, and if computational cost of each model pays off. With a series of these experiments, we provide multifaceted evaluation of state-of-the-art moment localization models. Codes are available at `https://github.com/snuviplab/MoLEF`.

1 INTRODUCTION

Moment localization task aims to retrieve a segment that corresponds to a text query from an untrimmed and unsegmented video. Fig. 1 shows an example to locate a moment corresponding to a query like "She is using a small vacuum cleaner." in a video, and the expected answer is the start/end times of [11.85, 48.6] for the query. This task is more challenging and complicated than other video tasks like action recognition, requiring comprehensive understanding of the video and the textual query, as well as alignment between them. This includes detecting and distinguishing objects and actions that appear in the video, multigranular abstraction of a scene from the discovered objects and actions, and general understanding of the scene transitions and relation between scenes.

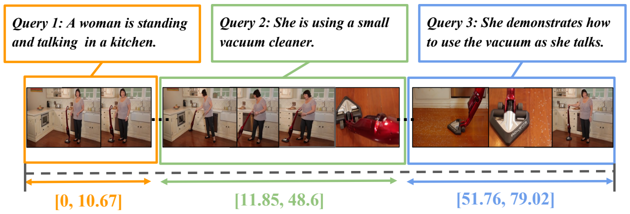

Query 1: A woman is standing and talking in a kitchen.
Query 2: She is using a small vacuum cleaner.
Query 3: She demonstrates how to use the vacuum as she talks.
[0, 10.67]
[11.85, 48.6]
[51.76, 79.02]

Figure 1: An example of moment localization.

Taking advantage of recent advances in computer vision and natural language processing, the existing studies on this task have argued that the performance has been significantly improved. Despite their claim, however, it is still questionable if these state-of-the-art models actually have advanced general capability on this task. Our study starts from an observation that most moment localization papers report experimental results on a few (usually two) datasets, although there are more widely-used standard benchmarks. A natural question after this observation is what if we test these models on other unreported datasets. Will they show a similar trend, outperforming other baseline models, or is this a result of more or less overfitting on a specific dataset?

Looking further into the details, one can easily realize notable diversity across benchmark datasets. First of all, each dataset contains videos covering diverse domain. For instance, action videos (*e.g.*, ActivityNet Captions) tend to be shorter and focus on relatively simpler actions, compared to cooking videos (*e.g.*, YouCook2), where a more complex sequence of steps is illustrated. Also, the observed actions or objects from different domain are quite different. We can easily imagine that the best moment localization model may not be the same for these different types of videos.

[*] Equal contribution.
[†] Corresponding author: joonseok@snu.ac.kr

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s).

Second, the nature of annotations is diverse across datasets. For example, one dataset tends to have longer annotations (30–60 sec), while another has shorter ones (2–3 sec). Annotations may be concentrated at the beginning of each video in a dataset, while more uniformly spread over the entire video in another. Observing such a variety, it is questionable if a model (implicitly) learns such a bias and overfits to a particular case.

Third, the query text is also diverse across datasets. One dataset may be annotated with more detailed descriptions, while another may contain more concise and abstract ones. The vocabulary in the annotations could also be diverse, considering various topics covered by different domains. In this paper, we conduct two experiments with modified query texts: 1) the action verb is masked out and 2) the action verb is left. Through the experiments, we reconfirm that the moment localization models take advantage of such correlations innate in the domain and the particular dataset.

If the moment localization models do their job without being biased, they would perform consistently well on most datasets, by precisely understanding the meaning of the query and the video and scoring candidate moments based on precisely estimated relevance, regardless of the peripheral conditions like domain or length of the video and query. If not, however, it may mean that the models achieve better performance than they deserve by somehow relying on those irrelevant conditions that should not be utilized for this task.

In this paper, we propose to scrutinize whether the existing moment localization methods solve the task legitimately. First, we compare the end-to-end performance of recent state-of-the-art moment localization models on a more complete set of benchmarks covering various domains, under controlled configurations (Sec. 5). Second, we examine further on how much each method utilizes specific biases, *e.g.*, the distribution of annotations and the query text (Sec. 6). Third, observing that each previous study uses particular but different feature representations for video and text, we breakdown the bottleneck for the performance of moment localization between the representations *vs.* the modeling aspects (Sec. 7). Finally, we complete the study by comparing computational cost of the competing models and seeing if increased model size or inference time can pay off the cost (Sec. 8). From extensive experimental comparisons, this study aims at providing precise diagnosis on the current state-of-the-art moment localization models and useful insights for the future research directions.

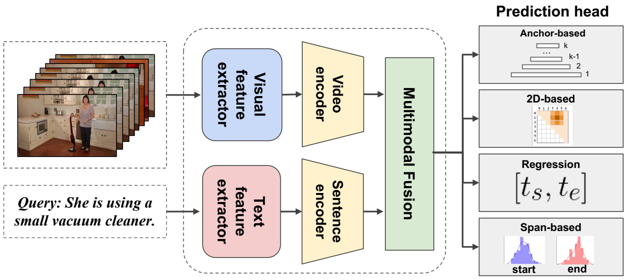


Figure 2: A common MLSV architecture.

2 THE MOMENT LOCALIZATION PROBLEM

The moment localization, or temporal video grounding, task aims at localizing a short fraction (segment) from a given entire video described by a query sentence. Formally, given a video $\mathbf{X}$ represented as a sequence of N frames, $\{\mathbf{X}_t \mid t = 1, ..., N\}$, where $\mathbf{X}_t \in \mathbb{R}^d$ is a visual feature vector representing the t-th frame, and a text query $\mathbf{w}$ represented as a sequence of L word tokens, $\{\mathbf{w}_i \mid i = 1, ..., L\}$, where $\mathbf{w}_i \in |V|$ is the i-th word token and V is the vocabulary set, the goal of moment localization task is to estimate the conditional probability $p(\mathbf{s}|\mathbf{w})$, where $\mathbf{s}$ is a video segment given by $\mathbf{s} = \{\mathbf{x}_t \mid t = t_s, ..., t_e\}$, with t_s and t_e as the indices of the starting and end frames of the segment in $\mathbf{X}$. The estimated scores for a handful number of candidate segments, proposed either explicitly or implicitly (see Sec. 3 for more details), are sorted and the top-k are evaluated.

Note that this task can be extended to moment localization in video corpus (MLVC) (Zhang et al., 2020a), where a corpus of videos (instead of a single video) is given and the task aims to retrieve video segments that are best described by the text query among all videos in this corpus. To distinguish from this, the single-video version is often referred to as moment localization in a single video (MLSV). We limit the scope of our comparative study to MLSV, leaving MLVC as an interesting future study.

3 RELATED WORK

3.1 Moment Localization Approaches

Existing MLSV methods are roughly classified into three categories: 1) proposal-based, 2) proposal-free, and 3) others. Most models consist of feature extractors, encoders, multimodal fusion modules, and prediction heads, as illustrated in Fig. 2.

Proposal-based approaches first generate multiple candidate segments before matching them with the query. Depending on how they generate the proposals, they are further classified into sliding-window-based (Wu and Han, 2018; Jiang et al., 2019; Ge et al., 2019), proposal-generated (Xu et al., 2019; Chen and Jiang, 2019; Xiao et al., 2021c; Liu et al., 2021a), anchor-based (Yuan et al., 2019a; Zhang et al., 2019a; Qu et al., 2020; Wang et al., 2020; Chen et al., 2018; Zhang et al., 2019b; Liu et al., 2020, 2021b), and 2D-based (Gao and Xu, 2021; Hu et al., 2021; Zhang et al., 2020c; Gao et al., 2021; Sun et al., 2022; Zheng et al., 2023). In this study, we experiment with anchor-based and 2D-based methods, since the sliding-window and proposal-generated methods suffer from high computational cost and show relatively weaker performance. The anchor-based methods are also limited in that the lengths of proposals are constrained to those of the pre-defined anchors.

Proposal-free approaches first compute a query-aware video representation as a sequence of features, and then predict the starting and ending frames of the segment described by the query. They are further classified into regression-based (Yuan et al., 2019b; Lu et al., 2019; Ghosh et al., 2019; Zeng et al., 2020; Mun et al., 2020; Li et al., 2021; Chen et al., 2020a,b; Chen and Jiang, 2020; Kim et al., 2021; Zhang et al., 2023b; Xiao et al., 2021a,b; Zhang et al., 2023c) and span-based approaches (Zhang et al., 2020b; Rodriguez et al., 2020; Yu et al., 2021; Zhang et al., 2021b,a; Woo et al., 2022). The regression-based methods are trained to directly predict moment scores, exhibiting greater computational efficiency than anchor-based methods. The span-based methods treat the input video as a text passage, advantageous for handling long videos.

In addition, weakly supervised approaches (Duan et al., 2018; Wang et al., 2021; Yang et al., 2021; Huang et al., 2021; Chen and Jiang, 2021; Song et al., 2020) introduce the MLSV task as an auxiliary task for representation learning or other downstream tasks. Several methods (Zheng et al., 2022a,b) propose a weakly supervised learning for MLSV, utilizing Gaussian masks to capture query-related events in proposal generation.

There are a few other highly relevant tasks concurrently studied. Spatio-temporal video grounding (Chen et al., 2019; Sadhu et al., 2020; Tang et al., 2020; Zhang et al., 2020d,e; Li et al., 2022) further extends the moment localization to the spatial dimension, localizing a spatio-temporal tube of an object described by a text query. Action segmentation (Lea et al., 2017; Farha and Gall, 2019) is another relevant task, temporally locating a set of pre-defined actions, instead of a free-form text query.

3.2 Comparative Studies on MLSV

There are a few existing works that have conducted a comparative study on MLSV methods. Yuan et al. (2021) have raised the annotation bias issue and reorganized ActivityNet and Charades-STA to differentiate the moment distribution in the training and test sets. Our work shares common motivation, but separates them even more strictly, completely non-overlapping the intervals. Otani et al. (2020) have raised the query text bias issue and performed action-aware blind experiments by sampling timestamps with the top verbs. Our work is distinguished from this in that we conduct experiments with modified query texts, masking out or solely leaving the action verbs. For a more complete understanding, the readers are encouraged to read recent surveys (Liu et al., 2021c, 2023; Zhang et al., 2023a) on general MLSV, or one focusing on activities (Yang et al., 2020).

4 COMPARATIVE STUDY DESIGN

Our study focuses on the following research questions:

1. Do the existing MLSV methods perform equally well on videos from a variety of domains? More generally, what are important factors to build an MLSV model that performs uniformly well without being affected by data characteristics? ▷ Sec. 5
2. How much are the MLSV methods affected by the annotation bias and the query text bias? How much do they get benefit from these biases? Which method is more robust from these biases? ▷ Sec. 6
3. For the MLSV task, which one is more important, between feature representations and modeling? How much benefit we get from stronger features? ▷ Sec. 7
4. How efficient are the existing MLSV methods in terms of model size and inference time? ▷ Sec. 8

4.1 Experimental Settings

Benchmark Datasets. We use 7 standard benchmark datasets for our experiments: ActivityNet Captions (Krishna et al., 2017), Charades-STA (Gao et al., 2017), TACoS (Regneri et al., 2013), DiDeMo (Anne Hendricks et al., 2017), YouCook2 (Zhou et al., 2018), MSR-VTT (Xu et al., 2016), and TVR (Lei et al., 2020). Tab. 1 summarizes general statistics of them, and Appendix D provides the detailed descriptions. We employ the common experimental settings (*e.g.*, video and text features) for each dataset. See Appendix B for details.

Competing Models. The methods of the MLSV task are classified into three categories: 1) proposal-based, 2) proposal-free, and 3) others. Most models

Dataset	**ActivityNet**	**Charades-STA**	**TACoS$_{\text{org}}$**	**DiDeMo**	**YouCook2**	**MSR-VTT**	**TVR**
Video source	YouTube	Homes	Lab Kitchen	Flickr	YouTube	YouTube	TV
Domain	Open	Indoor Activity	Cooking	Open	Cooking	Open	TV show
# Videos	14,926	6,664	127	10,641	1,790	5,127	19,614
# Moments	71,722	11,733	3,290	48,228	13,828	7,110	97,444
# Text queries	71,957	16,050	18,818	75,431	13,829	142,220	98,070
Average # annotations per video	4.8	1.8	25.9	7.1	7.7	1.4	5.0
Vocab size	15,505	1,303	2,287	7,785	13,079	28,041	57,103
Average video length (sec)	117.6	30.8	287.1	53.8	315.4	426.1	76.1
Min/Max video length (sec)	1.6 / 755.1	5.7 / 194.4	48.8 / 1402.7	22.0 / 2150.4	44.3 / 1106.1	5.0 / 3602.0	2.0 / 272.0
Average moment length (sec)	37.1	8.2	6.1	7.5	19.6	15.1	9.1
Min/Max moment length (sec)	0.05 / 408.8	1.0 / 80.8	0.3 / 167.0	2.0 / 25.0	1.0 / 264.0	10.0 / 30.0	0.3 / 239.4
Average query length (words)	13.2	6.2	8.8	7.6	8.8	9.4	12.2
Min/Max query length (words)	3 / 82	2 / 11	1 / 202	1 / 48	2 / 44	2 / 72	6 / 107

Table 1: Statistics of benchmark datasets used in this study.

consist of feature extractors, encoders, multimodal fusion modules, and prediction heads, as illustrated in Fig. 2. We summarize representative models in each category in Appendix A.

For our study, we select recent state-of-the-art methods among the anchor-based, 2D-based, regression-based, and span-based ones. We exclude methods with incompetent reported scores or those requiring excessive computational cost (*e.g.*, sliding-windows). If there are multiple models with almost identical structure or idea, we choose the most recent and competent one for our study. From these criteria, we select the following methods for our comparative study:

- Anchor-based approaches: TGN (Chen et al., 2018), CMIN (Zhang et al., 2019b), CSMGAN (Liu et al., 2020), IA-Net (Liu et al., 2021b)
- 2D-Map approaches: 2D-TAN (Zhang et al., 2020c), RaNet (Gao et al., 2021), MGPN (Sun et al., 2022), TRM (Zheng et al., 2023)
- Regression-based methods: DRN (Zeng et al., 2020), LGI (Mun et al., 2020), PLRN (Kim et al., 2021)
- Span-based methods: VSLNET (Zhang et al., 2020b), TMLGA (Rodriguez et al., 2020), ReLoCLNet (Zhang et al., 2021a), LVTR (Woo et al., 2022)
- Other models: CNM (Zheng et al., 2022a), CPL (Zheng et al., 2022b)

Detailed experimental settings for each method are detailed in Appendix C.

Evaluation Metrics. To measure the performance, we adopt a commonly used metric for MLSV: Recall@k with IoU=p (Escorcia et al., 2019; Lei et al., 2020). It measures the ratio of test queries, having at least one prediction within top-k retrieved moments whose Intersection over Union (IoU) with the ground truth is at least p. We set $k = 1$ and $p \in \{0.1, 0.3, 0.5, 0.7\}$ in our experiments.

5 OVERALL COMPARISON

We first compare the overall MLSV performance of competing models on all the benchmark datasets. Tab. 3 reports the evaluation results, and we summarize notable observations below.

Observation 1 *No specific method outperforms across all domains; current models are rather specialized on a specific domain or dataset.*

On open domains (MSR-VTT and TVR), 2DTAN and ReLoCLNet achieve the state-of-the-art across-the-board, while on cooking videos (TACOS$_{\text{org}}$ and YouCook2), MGPN, VSLNet, and CSMGAN perform relatively stronger. On activity videos (ActivityNet and Charades-STA), most models show decent performance, since the authors have developed and evaluated them on these datasets. Between the two action datasets, there is no general tendency among models; CSMGAN shows the strongest performance on ActivityNet, while PLRN and MGPN are the strongest on Charades-STA. However, they are not particularly competent on the open domains. Nevertheless, there are a few relatively robust models across datasets; *e.g.*, 2DTAN.

Observation 2 *Performance metrics measured at different IoU thresholds are usually in accordance, but not always.*

Tab. 3 indicates that the best performing models for each dataset achieves the state-of-the-art with any IoU threshold among $\{0.3, 0.5, 0.7\}$. However, there are a

[2] TGN is reproduced on the first 100 frames of MSR-VTT due to the limitation on our computational resources.

[3] TRM uses DistilBERT (Sanh et al., 2019) to extract sentence and phrase features concurrently, following the original setting.

[4] LVTR originally uses 4 sentences per video, but we use only 1 for each video for fair comparison. This may explain the lower performance.

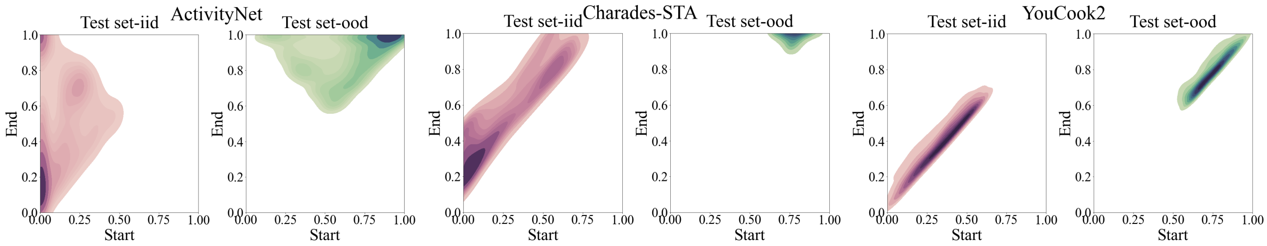
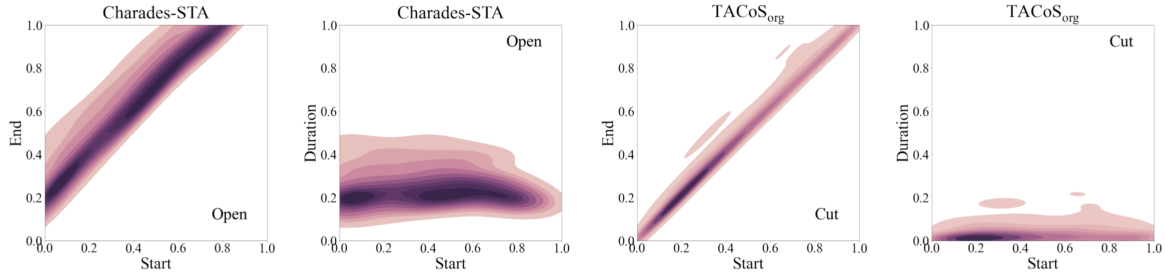


Figure 3: Distributions of the independent-identical and out-of-distribution test sets used for annotation bias experiments.

Cat	Dataset	ActivityNet						Charades-STA						YouCook2					
	R@1	IoU@0.3		IoU@0.5		IoU@0.7		IoU@0.3		IoU@0.5		IoU@0.7		IoU@0.3		IoU@0.5		IoU@0.7	
	Setting	IID	OOD	IID	OOD	IID	OOD	IID	OOD	IID	OOD	IID	OOD	IID	OOD	IID	OOD	IID	OOD
Anchor	CMIN	68.23	17.71	50.39	8.94	28.81	1.77	65.42	41.58	50.95	32.42	23.51	20.91	49.06	7.09	34.38	3.60	13.65	1.91
	CSMGAN	67.44	20.05	51.16	11.14	30.80	3.08	30.10	5.18	8.22	0.99	0.68	0.00	45.52	6.49	31.46	3.85	12.71	1.56
	IA-Net	66.74	9.08	35.48	1.46	13.35	0.18	62.64	31.01	49.36	20.67	22.59	9.32	25.94	1.91	14.90	0.68	4.79	0.23
2D	2D-TAN	63.25	22.73	49.09	13.81	34.38	2.82	62.44	42.44	45.52	24.82	23.40	6.58	34.90	19.14	21.15	11.26	9.69	4.73
	RaNet	63.99	20.49	50.97	11.96	34.36	4.42	61.90	38.61	46.12	23.10	24.81	9.01	43.44	19.82	27.50	11.26	11.88	5.29
Reg	DRN	63.51	19.33	47.56	8.96	28.98	3.17	59.24	11.67	36.96	3.70	18.34	0.32	13.02	0.00	6.77	0.00	1.77	0.00
	LGI	66.57	15.74	49.78	8.22	31.14	3.08	42.68	33.99	26.70	17.93	11.33	4.93	9.79	0.68	4.90	0.11	1.35	0.00
Span	VSLNet	60.76	20.46	44.24	12.14	28.71	5.14	63.12	35.71	45.31	21.30	26.84	10.73	32.81	2.48	20.00	1.35	7.60	0.56
	TMLGA	46.48	21.02	28.60	10.56	17.39	4.60	59.88	22.30	43.70	14.98	22.25	7.89	44.69	22.30	30.00	14.98	15.62	7.89

Table 2: **Annotation Bias.** Performance on the IID and OOD test sets. Most methods drop their performance on OOD, indicating they are relying on the annotation bias.



(a) On Charadse-STA: the top-1 word "Open"

(b) On TACoS$_{\text{org}}$: the top-1 word "Cut"

Figure 4: Labeled moments distribution with the top-1 words; (left): start-end points, (right): duration given the start point.

Cat	Approach	ActivityNet			Charades-STA			TACoS$_{\text{org}}$			DiDeMo			YouCook2			MSR-VTT			TVR		
	R@1,IoU	@0.3	@0.5	@0.7	@0.3	@0.5	@0.7	@0.3	@0.5	@0.7	@0.3	@0.5	@0.7	@0.3	@0.5	@0.7	@0.3	@0.5	@0.7	@0.3	@0.5	@0.7
Anchor	TGN[2]	45.51	28.47	17.90	50.51	34.38	16.26	21.77	18.90	10.53	29.70	14.54	4.15	5.15	2.20	0.65	0.13	0.08	0.01	15.70	9.08	4.30
	CMIN	63.61	43.40	23.88	69.36	52.52	27.49	24.64	18.05	7.40	26.84	13.98	5.31	55.10	37.89	16.35	8.80	3.33	1.04	22.03	11.06	4.15
	CSMGAN	**68.52**	**49.11**	29.15	55.95	40.95	20.56	33.90	27.09	10.64	32.12	16.33	5.88	**64.12**	**49.05**	**27.37**	8.16	3.04	0.91	25.62	13.75	5.98
	IANet	67.14	48.57	27.95	63.91	**61.29**	37.91	37.91	26.27	8.06	21.52	9.86	2.74	34.16	21.48	9.11	8.73	3.47	0.85	25.64	14.58	6.42
2D	2DTAN	59.45	44.51	27.38	60.32	39.81	23.31	37.29	25.32	9.57	30.80	16.75	8.72	42.67	26.26	10.60	**16.09**	**9.34**	**3.43**	42.74	28.92	**17.57**
	RaNet	60.09	45.59	28.67	67.01	60.40	39.65	43.34	33.54	11.44	32.07	15.68	6.66	50.57	35.28	18.21	7.85	2.59	0.32	24.19	12.35	6.77
	MGPN	60.41	47.92	30.47	64.50	60.82	**41.16**	**48.81**	**36.74**	8.77	29.24	16.86	9.88	41.75	28.38	13.83	8.06	3.92	1.34	32.98	23.24	13.55
	TRM[3]	66.41	50.44	**31.18**	60.67	47.77	28.01	18.44	13.35	7.59	30.16	17.10	8.94	42.10	28.84	14.49	6.37	3.06	1.20	41.09	29.66	17.14
Reg	DRN	60.99	45.45	24.36	55.89	53.09	31.75	9.47	5.65	3.20	29.47	15.19	6.81	12.87	5.65	1.72	8.05	3.95	1.36	17.37	8.23	2.92
	LGI	58.52	41.51	23.07	72.96	59.46	35.48	33.47	20.82	9.00	31.60	16.33	5.67	24.91	13.29	4.58	10.41	4.86	1.65	24.27	13.67	5.91
	PLRN	63.79	44.48	26.81	**73.17**	59.73	37.28	14.06	9.02	4.30	31.84	16.40	6.22	23.57	13.55	4.29	9.72	4.45	1.65	29.38	17.51	8.16
Span	VSLNet	63.16	43.22	26.16	70.46	54.19	35.22	29.61	24.27	**20.03**	31.25	14.11	7.09	32.73	20.02	9.42	7.31	3.33	1.55	42.31	28.52	15.86
	TMLGA	51.28	33.04	19.26	67.53	52.02	33.74	24.54	21.65	16.46	27.89	14.17	6.83	50.29	35.57	19.19	7.12	3.91	1.99	14.75	7.77	3.88
	ReLoCLNet	42.65	28.54	17.76	48.58	23.93	8.82	5.46	2.10	0.22	**43.31**	**29.10**	5.64	8.28	2.52	0.69	6.57	3.67	1.29	**49.87**	**31.88**	15.04
	LVTR[4]	28.73	15.96	6.57	51.72	34.48	18.10	5.08	1.95	0.00	32.75	20.82	**13.13**	8.68	3.65	1.83	6.40	2.40	0.42	16.89	9.40	4.11
etc	CNM	55.68	33.33	13.08	60.04	35.15	14.95	1.08	0.05	0.00	18.23	4.84	1.24	1.63	0.23	0.00	3.93	1.71	0.90	12.68	5.88	2.72
	CPL	53.67	31.24	10.67	55.49	35.34	17.24	1.03	0.02	0.02	26.60	11.59	4.08	12.71	2.61	0.32	5.83	1.78	0.50	16.93	6.85	2.49

Table 3: **Overall Comparison.** R@1 with IoU@$\{0.3, 0.5, 0.7\}$ of the representative methods on the end-to-end moment localization task. The shaded scores are the reported ones in the original paper, while the rest are our reproduction. The best reproduced score is marked with **boldface.**[2,3,4]

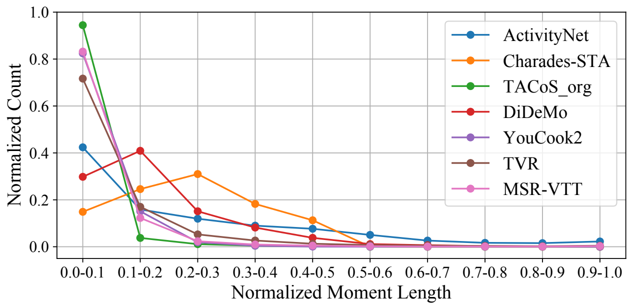


Figure 5: **Distribution of the normalized length of moments.** Most datasets have a strong prior on this distribution.

few exceptions. On ActivityNet and DiDeMo, CSMGAN and ReLoCLNet achieve the strongest performance with lower IoU thresholds (0.3, 0.5), showing relatively weaker performance with higher ones, respectively. On the other hand, TRM and LVTR are the strongest with higher IoU thresholds (0.7) on ActivityNet and DiDeMo, respectively.

In conclusion, it is necessary to evaluate methods with various IoU thresholds, since it is not always possible to estimate performance using one threshold for another. Appendix F provides mIoU metrics for this experiment.

6 EXPERIMENTATION ON BIASES

We further investigate if (and how much) more MLSV methods take advantage of biases in datasets to localize a moment. Specifically, we consider two types of biases: annotation bias (Sec. 6.1) and query text bias (Sec. 6.2).

6.1 Annotation Bias

Annotation bias indicates some underlying patterns among the labeled moments. If the labeled moments are selected uniformly randomly, *e.g.*, in terms of the start-end points and their length, the model may not rely on any prior. In reality, however, even carefully designed datasets may have unintentionally introduced biases; for instance, a dataset may contain more moments starting at the beginning than later, or most moments are shorter than 5 seconds, and so on. Fig. 5 shows the distribution of moments length, normalized by the corresponding video length. The labeled moments in most datasets (except for ActivityNet, YouCook2) indeed have a strong prior towards short moments, covering $< 20\%$ of the video. The model might learn this bias as prior, although the MLSV task itself is independent of such prior.

Experimental design. We first merge the training, validation, and test sets for each dataset and introduce a custom split. Specifically, we first normalize the start and end point of each moment for each video. Thus, every video starts at 0 and finish at 1 regardless of its length. All moments are sorted by their central point and the last 10% are chosen as the test-OOD set (see Appendix H for more details). The rest is divided into training (70%), validation (10%), and test-IID (10%) sets uniformly randomly. We use ActivityNet, Charades-STA, and YouCook2 for this experiment, as they contain sufficient number of long-enough videos. Fig. 3 illustrates their drastically different distributions with a kernel density plot with a Gaussian kernel.

Observation 3 *The existing approaches implicitly utilize the label distribution of the dataset.*

If the MLSV models are not affected by annotation bias, the test performance should be identical with the two splits. Tab. 2 indicates, however, all methods significantly drop their performance on all datasets, although the degree of drop varies. 2D-based and span-based models are relatively more robust than others.

6.2 Query Text Bias

The model is expected to comprehend the query text in full and use it to localize the corresponding moment in the given video. If the annotation process is not designed carefully, however, the model might focus only on a sub-part of the query text, rather than the full sentence to find the corresponding moment. Fig. 4 plots the distributions of the normalized moments with the most frequent verbs on Charades-STA and TaCoS$_{\text{org}}$, indicating that these datasets actually have a specific bias on the top-1 verbs. The moments containing the word "open" in their queries in Charades-STA, for example, have a consistent duration of around 20% of the video, regardless of their start point. As another example, the queries including "cut" in TACoS$_{\text{org}}$ are located mostly at the front of the video, probably because cutting is needed often for ingredient preparation at the beginning of cooking. From this observation, we hypothesize that the model might easily locate a moment based only on the action verb, instead of using all other clues in the query. We name this as *query text bias*, and investigate if the models rely on it.

Experimental design. We hypothesize that if the models rely on the full sentence, they will not perform well given only a part of the query. To verify this, we evaluate models on two test sets : with modified query texts 1) the action verb is masked out (M), 2) only the action verb is left (A). If the model relies purely on only one of them, it will maintain similar performance on one, while severely fail on the other setting.

Observation 4 *Models tend to rely more on the con-*

	R@1	IoU@0.3			IoU@0.5			IoU@0.7			IoU@0.3			IoU@0.5			IoU@0.7		
	Method	O	M	A	O	M	A	O	M	A	O	M	A	O	M	A	O	M	A
	Dataset	**ActivityNet**									**Charades-STA**								
Anchor	CMIN	64.5	58.9	34.0	45.8	38.4	18.4	25.1	20.2	8.8	69.4	62.0	54.3	52.5	44.3	35.9	27.5	22.0	16.5
	CSMGAN	66.1	64.8	46.7	46.5	42.5	25.9	26.2	22.7	14.1	56.0	32.6	23.1	41.0	9.0	5.2	20.6	1.2	0.5
	IA-Net	66.4	61.5	46.5	47.0	37.4	25.1	27.0	18.1	12.9	63.9	53.4	50.8	51.0	40.7	37.7	25.1	17.0	14.7
2D	2D-TAN	60.1	47.0	48.4	44.8	22.5	24.1	26.8	12.5	13.1	60.3	43.4	39.0	45.9	23.6	28.7	24.9	10.5	15.0
	RaNet	60.1	58.1	42.8	45.1	42.6	26.0	28.1	25.3	13.7	67.0	58.3	53.8	53.1	43.4	39.9	30.7	24.6	22.5
Reg	DRN	61.0	58.7	45.2	42.5	38.8	23.0	23.9	20.6	12.2	55.9	53.8	42.4	36.4	34.1	23.3	17.6	15.1	9.0
	LGI	58.0	56.0	43.0	39.8	37.9	27.6	21.7	19.6	14.2	70.9	60.4	57.9	58.2	46.2	43.6	35.2	25.4	23.8
Span	VSLNet	54.9	56.4	40.0	38.6	36.9	24.1	23.7	21.0	14.3	67.5	55.4	56.0	50.8	36.8	37.6	32.2	22.0	20.6
	TMLGA	51.3	50.6	48.1	26.6	25.8	21.3	15.3	14.8	12.3	70.0	59.9	7.7	49.0	37.1	1.5	31.5	16.5	0.5
	Dataset	**TACoS$_{\text{org}}$**									**YouCook2**								
Anchor	CMIN	27.4	21.4	10.4	18.4	13.5	6.4	7.4	4.6	2.6	55.1	44.2	17.0	37.9	30.0	10.4	16.4	12.1	3.9
	CSMGAN	25.4	18.9	7.3	20.0	14.2	5.3	10.6	6.6	2.3	64.1	55.5	19.3	49.1	41.1	10.9	27.4	20.6	4.5
	IA-Net	24.7	16.5	7.4	16.8	11.6	3.6	8.1	6.1	1.8	34.2	30.2	15.3	21.5	19.5	9.4	9.1	7.7	3.7
2D	2D-TAN	32.1	2.9	4.1	19.2	1.7	2.8	9.6	1.3	0.9	42.7	9.3	8.6	26.3	5.2	4.6	10.6	2.1	1.6
	RaNet	25.2	22.8	11.3	20.4	18.7	8.9	11.4	10.4	5.0	50.6	45.1	19.1	35.3	30.7	10.8	18.2	15.3	4.6
Reg	DRN	9.5	5.0	4.7	5.7	3.0	2.4	3.2	1.1	0.8	12.9	12.8	8.17	5.7	5.6	3.6	1.7	1.7	1.5
	LGI	33.5	2.5	2.3	20.8	1.2	0.9	9.0	0.2	0.2	24.9	22.5	12.0	13.3	11.7	5.1	4.6	4.0	1.3
Span	VSLNet	24.5	18.6	11.0	17.1	14.8	8.2	11.5	9.8	4.6	32.7	24.9	15.2	20.0	14.2	7.8	9.4	6.4	3.3
	TMLGA	21.6	18.9	7.5	19.0	16.1	5.7	13.4	10.2	3.3	50.3	43.6	18.9	35.6	28.7	11.3	19.2	15.1	5.2

Table 4: **Query Text Bias.** Performance on the original queries (O), those with action verb masked (M), and only the action verb term (A). Most methods degrade their performance both on M and A, where the degree is more severe on A, indicating they rely more on the context.

Category	Approach	IoU@0.1			IoU@0.3			IoU@0.5			IoU@0.7		
	R@1	VGG	C3D	I3D	VGG	C3D	I3D	VGG	C3D	I3D	VGG	C3D	I3D
AN-based	CMIN	34.05	35.44	**59.40**	15.32	19.79	**45.29**	4.52	7.37	**28.41**	0.96	2.19	**10.33**
	CSMGAN	29.38	34.95	**67.71**	5.09	19.83	**55.95**	0.57	6.51	**40.95**	0.03	2.10	**20.56**
	IA-Net	68.11	66.39	**72.92**	54.46	53.44	**63.91**	39.48	38.71	**50.98**	20.60	20.98	**25.06**
2D-based	2D-TAN	65.24	60.40	**69.30**	55.81	51.23	**60.32**	41.91	39.44	**45.86**	24.30	22.88	**24.87**
	RaNet	53.55	58.97	**74.75**	42.11	47.29	**67.01**	29.05	31.53	**53.09**	15.94	15.19	**30.70**
Regression	DRN	62.53	64.20	**66.34**	50.03	54.10	**55.89**	34.18	**41.50**	36.35	17.05	**23.90**	17.60
	LGI	65.31	63.06	**79.17**	49.57	43.93	**70.94**	31.47	24.52	**58.15**	13.52	9.20	**35.19**
Span-based	VSLNet	74.99	62.61	**78.61**	56.22	44.83	**67.49**	34.56	25.57	**50.77**	19.05	11.76	**32.16**
	TMLGA	11.05	22.98	**82.77**	5.99	12.47	**69.97**	2.28	5.16	**49.01**	0.73	1.51	**31.53**

Table 5: **Impact of Video Features.** Using more advanced features (I3D) significantly improves performance of most models (on Charades-STA).

texts than the action itself, even on action-centric datasets.

On most datasets and with most models, performance drop is more severe with the action-verb (A) than with the context-only (M) queries. This indicates that in most cases models do not excessively rely on the action. On cooking domain (*e.g.*, YouCook2), it is expected to perform well with the context than only with the verb, since particular tools or ingredients may give more hints than action like cutting to locate some cooking step. Tab. 4 indeed verifies that the performance significantly drops on the set A more than on the set M.

Surprisingly, a similar trend is observed even on the action-centric datasets. On ActivityNet, models suffer from poor performance only with the action queries (A), while most of them maintain performance with action-masked queries (M) similar to the original full queries (O).

7 ANALYSIS ON FEATURE REPRESENTATIONS

On most previous experiments including ours, a single designated video and text features (summarized in Appendix B) have been used for the sake of convenience and fair comparison. However, this practice hinders comparison between different feature representations on this task. With the advances in video representation learning, a natural research question arises: are the advanced features developed for video classification also powerful for MLSV task?

We choose Charades-STA dataset for this experiment, because relatively more number of previous works have utilized various features with this dataset. Tab. 5 compares the performance of each model on Charades-STA with commonly used 3 features: VGG (Simonyan and Zisserman, 2014), C3D (Tran et al., 2015), and I3D (Carreira and Zisserman, 2017).

Observation 5 *Most models enjoy significant performance boost with advanced video representations, implying that use of advanced features may be more important than improving the model.*

Tab. 5 indicates that each model not just achieves the best performance with I3D, but also any I3D-based model often outperforms all other models using simpler features (C3D, VGG). Although it is expected that I3D-based models would be stronger, the degree of difference is indeed surprising. This implies that using an advanced feature representation can be more important than any other modeling aspects. Thus, model development research is encouraged to be conducted on top of the most advanced features.

Among models, however, some are more severely affected while others still show relatively comparable performance with simpler features. Specifically, IA-Net, 2D-TAN, and VSLNet achieve reasonably good performance even with VGG and C3D, just slightly lagging behind its own I3D counter-pert. IA-Net achieves consistent performance across features probably thanks to its refinement of cross-modal correlation. TMLGA is an example of the opposite side, best-performing among I3D-based models, while the worst with simpler features. This model seems to heavily depend on feature representations since its cross-modal interaction only consists of the attention-guided dot-product. This observation implies that a robust cross-modal interaction may help to compensate for the drawbacks of the specific feature representations. As the advanced features tend to require longer inference time, performing well on lighter and simpler features would be another criterion when designing a model.

8 ANALYSIS ON COMPUTATIONAL COST

In this section, we analyze efficiency of the models in terms of the model size and inference time. Tab. 6 summarizes the number of parameters and per-query inference time of each model, measured on ActivityNet with batch size 1 on a single NVIDIA A100 GPU. Overall tendency is similar on other datasets as well.

Observation 6 *Span-based models tend to be lighter, while regression-based models (especially DRN) tend to be heavier. Additional computational cost with a heavier model does not always pay off.*

Category	Model	# params	Inference time
Anchor -based	CMIN	8M	29.5 ms
	CSMGAN	16M	33.6 ms
	IA-Net	23M	23.1 ms
2D-based	2D-TAN	92M	31.8 ms
	RaNet	91M	26.9 ms
Regression	DRN	681M	1019.4 ms
	LGI	54M	59.0 ms
Span-based	VSLNet	5M	15.7 ms
	TMLGA	4M	20.8 ms

Table 6: **Computational Cost.** Per-query inference time measured on ActivityNet dataset.

Overall, the model size and inference time are positively correlated, as expected. The actual time taken at inference is not significantly different among anchor-based and 2D-based models, while span-based models are 1.5–2 times faster.

Combining this cost analysis with Tab. 3, however, we observe that such a light model like VSLNet actually can be the strongest model on some dataset, *e.g.*, TACoS. This means larger and heavier models may not necessarily bring better performance, and it is crucial to design a proper scale suited for the target dataset.

9 QUALITATIVE RESULTS

We illustrate a few qualitative examples with an anchor-based (CSMGAN) and a span-based (ReLoCLNet) model, providing valuable insights into the behavior of these models.

First, in Fig. 6(a), we observe that CSMGAN provides a broad range of predictions that match with the text appearing throughout the entire video. In contrast, ReLoCLNet tends to predict only at the beginning of the video. Considering that the ground truth moments in the ActivityNet dataset often appear in the early parts, the model might take some advantage of it even if the model does not fully understand the video and query text. Second, due to the nature that CSMGAN generates multi-scale anchors as proposals, Fig. 6(b) shows that multiple proposals might confuse the model generating long predictions, in the DiDeMo dataset especially due to its short average moment length.

From these observations, we recognize that models not fine-tuned for specific granularity tend to rely on these biases, resulting in limited prediction varieties. Also, certain models exhibit sensitivity to specific hyperparameters, such as clip length and maximum length of the predicted span.

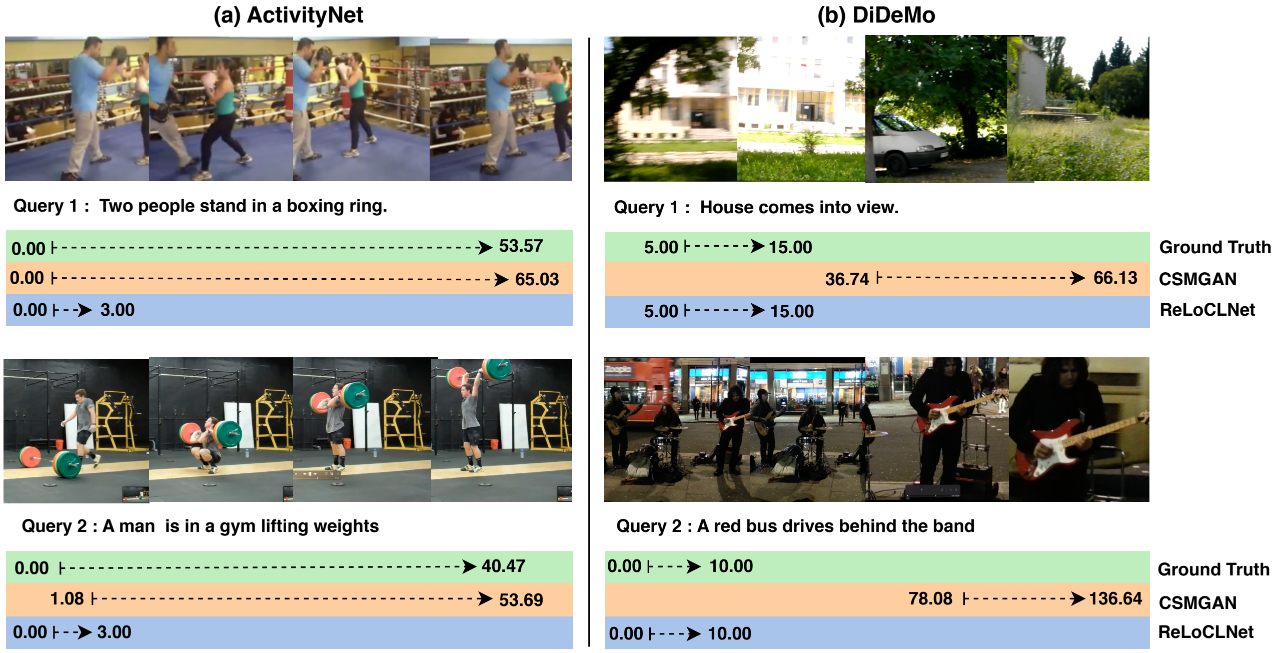


Figure 6: Qualitative results on ActivityNet and DiDeMo datasets

10 LIMITATION

We consider three limitations of our work as follows. First of all, in a rapidly evolving field like MLSV, new models are continuously developed, and thus the conclusions we draw in this paper may not last for a long time with the newly developed models. Second, although we have tuned hyperparameters to ensure reasonable performance of each model, there may be room for further improvement due to the limitations in grid search. We believe this unexplored gap would not change the overall conclusions, but the actual figure might be more or less different from the reported. Lastly, we had to make a conscious decision to neglect specific conditions of individual models in order to establish a standard evaluation setting for a fair comparison. For instance, TGN employs Inception-v4 video features in their original paper, while we use C3D features for all models to compare them under an identical setting. This standardization might overlook certain strategies employed by individual models, potentially negatively affecting their relative strengths.

11 SUMMARY

In this paper, we conduct a comprehensive comparative study of the representative moment localization (on a single video; MLSV) models on multiple benchmarks. We summarize our conclusions from our observations, answering to the research questions we pose in Sec. 4:

1. No current MLSV method performs equally well on videos from various domains. ▷ Obs. 1
2. Most current MLSV methods are significantly affected by the annotation bias, but not so much by the query text bias. ▷ Obs. 3 and 4
3. For the MLSV task, feature representation looks significantly more important than modeling. ▷ Obs. 5
4. Larger models may not necessarily bring better performance, so it is important to design a proper scale suited for the target data. ▷ Obs. 6

Empirical evaluations conducted in this study reveal that the current moment localization models are not as strong and robust as we expect, *e.g.*, over-engineered on a specific video domain and relying on annotation bias in the training set. This study also implies developing a stronger feature representation from genuine video understanding is essential for further improvement on moment localization.

Acknowledgements

This work was supported by the New Faculty Startup Fund from Seoul National University, by the Youlchon Foundation (Nongshim Corporation and affiliated companies), and by National Research Foundation (NRF) grant (No. 2021H1D3A2A03038607/50%, 2022R1C1C1010627/20%, RS- 2023-00222663/10%) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant (No. 2022-0-00264/20%) funded by the government of Korea.

References

Anne Hendricks, L., Wang, O., Shechtman, E., Sivic, J., Darrell, T., and Russell, B. (2017). Localizing moments in video with natural language. In *ICCV*.

Caba Heilbron, F., Escorcia, V., Ghanem, B., and Carlos Niebles, J. (2015). ActivityNet: A large-scale video benchmark for human activity understanding. In *CVPR*.

Carreira, J. and Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*.

Chen, J., Chen, X., Ma, L., Jie, Z., and Chua, T.-S. (2018). Temporally grounding natural sentence in video. In *EMNLP*.

Chen, L., Lu, C., Tang, S., Xiao, J., Zhang, D., Tan, C., and Li, X. (2020a). Rethinking the bottom-up framework for query-based video localization. In *AAAI*.

Chen, S., Jiang, W., Liu, W., and Jiang, Y.-G. (2020b). Learning modality interaction for temporal sentence localization and event captioning in videos. In *ECCV*.

Chen, S. and Jiang, Y.-G. (2019). Semantic proposal for activity localization in videos via sentence query. In *AAAI*.

Chen, S. and Jiang, Y.-G. (2020). Hierarchical visual-textual graph for temporal activity localization via language. In *ECCV*.

Chen, S. and Jiang, Y.-G. (2021). Towards bridging event captioner and sentence localizer for weakly supervised dense event captioning. In *CVPR*.

Chen, Z., Ma, L., Luo, W., and Wong, K.-Y. K. (2019). Weakly-supervised spatio-temporally grounding natural sentence in video. *arXiv:1906.02549*.

Duan, X., Huang, W., Gan, C., Wang, J., Zhu, W., and Huang, J. (2018). Weakly supervised dense event captioning in videos. In *NIPS*.

Escorcia, V., Soldan, M., Sivic, J., Ghanem, B., and Russell, B. (2019). Temporal localization of moments in video collections with natural language. *arXiv:1907.12763*.

Farha, Y. A. and Gall, J. (2019). MS-TCN: Multi-stage temporal convolutional network for action segmentation. In *CVPR*.

Gao, J., Sun, C., Yang, Z., and Nevatia, R. (2017). TALL: Temporal activity localization via language query. In *ICCV*.

Gao, J., Sun, X., Xu, M., Zhou, X., and Ghanem, B. (2021). Relation-aware video reading comprehension for temporal language grounding. *arXiv:2110.05717*.

Gao, J. and Xu, C. (2021). Fast video moment retrieval. In *ICCV*.

Ge, R., Gao, J., Chen, K., and Nevatia, R. (2019). MAC: Mining activity concepts for language-based temporal localization. In *WACV*.

Ghosh, S., Agarwal, A., Parekh, Z., and Hauptmann, A. (2019). Excl: Extractive clip localization using natural language descriptions. *arXiv:1904.02755*.

Hu, Y., Nie, L., Liu, M., Wang, K., Wang, Y., and Hua, X.-S. (2021). Coarse-to-fine semantic alignment for cross-modal moment localization. *IEEE Transactions on Image Processing*, 30:5933–5943.

Huang, J., Liu, Y., Gong, S., and Jin, H. (2021). Cross-sentence temporal and semantic relations in video activity localisation. In *ICCV*.

Jeon, M., Kang, M., and Lee, J. (2023). A unified framework for robustness on diverse sampling errors. In *ICCV*.

Jeon, M., Kim, D., Lee, W., Kang, M., and Lee, J. (2022). A conservative approach for unbiased learning on unknown biases. In *CVPR*.

Jiang, B., Huang, X., Yang, C., and Yuan, J. (2019). Cross-modal video moment retrieval with spatial and language-temporal attention. In *Proc. of the International Conference on Multimedia Retrieval (ICMR)*.

Kim, S., Yun, K., and Choi, J. Y. (2021). Position-aware location regression network for temporal video grounding. In *AVSS*.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *ICLR*.

Krishna, R., Hata, K., Ren, F., Fei-Fei, L., and Carlos Niebles, J. (2017). Dense-captioning events in videos. In *ICCV*.

Lea, C., Flynn, M. D., Vidal, R., Reiter, A., and Hager, G. D. (2017). Temporal convolutional networks for action segmentation and detection. In *CVPR*.

Lei, J., Yu, L., Berg, T. L., and Bansal, M. (2020). TVR: A large-scale dataset for video-subtitle moment retrieval. In *ECCV*.

Li, K., Guo, D., and Wang, M. (2021). Proposal-free video grounding with contextual pyramid network. In *AAAI*.

Li, M., Wang, T., Zhang, H., Zhang, S., Zhao, Z., Miao, J., Zhang, W., Tan, W., Wang, J., Wang, P., Pu, S., and Wu, F. (2022). End-to-end modeling via information tree for one-shot natural language spatial video grounding. In *ACL*.

Liu, D., Qu, X., Dong, J., and Zhou, P. (2021a). Adaptive proposal generation network for temporal sentence localization in videos. *arXiv:2109.06398*.

Liu, D., Qu, X., Liu, X.-Y., Dong, J., Zhou, P., and Xu, Z. (2020). Jointly cross-and self-modal graph attention network for query-based moment localization. In *Proc. of the ACM International Conference on Multimedia (MM)*.

Liu, D., Qu, X., and Zhou, P. (2021b). Progressively guide to attend: An iterative alignment framework for temporal sentence grounding. *arXiv:2109.06400*.

Liu, M., Nie, L., Wang, Y., Wang, M., and Rui, Y. (2023). A survey on video moment localization. *ACM Computing Surveys*, 55(9):1–37.

Liu, X., Nie, X., Tan, Z., Guo, J., and Yin, Y. (2021c). A survey on natural language video localization. *arXiv:2104.00234*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). RoBERTa: A robustly optimized bert pretraining approach. *arXiv:1907.11692*.

Lu, C., Chen, L., Tan, C., Li, X., and Xiao, J. (2019). Debug: A dense bottom-up grounding approach for natural language video localization. In *EMNLP*.

Miech, A., Alayrac, J.-B., Smaira, L., Laptev, I., Sivic, J., and Zisserman, A. (2020). End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*.

Miech, A., Zhukov, D., Alayrac, J.-B., Tapaswi, M., Laptev, I., and Sivic, J. (2019). HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*.

Mun, J., Cho, M., and Han, B. (2020). Local-global video-text interactions for temporal grounding. In *CVPR*.

Otani, M., Nakahima, Y., Rahtu, E., and Heikkilä, J. (2020). Uncovering hidden challenges in query-based video moment retrieval. In *BMVC*.

Pennington, J., Socher, R., and Manning, C. D. (2014). GloVe: Global vectors for word representation. In *EMNLP*.

Qu, X., Tang, P., Zou, Z., Cheng, Y., Dong, J., Zhou, P., and Xu, Z. (2020). Fine-grained iterative attention network for temporal language localization in videos. In *ACM MM*.

Raji, I. D., Gebru, T., Mitchell, M., Buolamwini, J., Lee, J., and Denton, E. (2020). Saving face: Investigating the ethical concerns of facial recognition auditing. In *Proc. of the AAAI/ACM Conference on AI, Ethics, and Society*.

Regneri, M., Rohrbach, M., Wetzel, D., Thater, S., Schiele, B., and Pinkal, M. (2013). Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics*.

Rodriguez, C., Marrese-Taylor, E., Saleh, F. S., Li, H., and Gould, S. (2020). Proposal-free temporal moment localization of a natural-language query in video using guided attention. In *WACV*.

Rohrbach, M., Regneri, M., Andriluka, M., Amin, S., Pinkal, M., and Schiele, B. (2012). Script data for attribute-based recognition of composite activities. In *ECCV*.

Sadhu, A., Chen, K., and Nevatia, R. (2020). Video object grounding using semantic roles in language description. In *CVPR*.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv:1910.01108*.

Shi, P. and Lin, J. (2019). Simple bert models for relation extraction and semantic role labeling. *arXiv:1904.05255*.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*.

Song, Y., Wang, J., Ma, L., Yu, Z., and Yu, J. (2020). Weakly-supervised multi-level attentional reconstruction network for grounding textual queries in videos. *arXiv:2003.07048*.

Sun, X., Wang, X., Gao, J., Liu, Q., and Zhou, X. (2022). You need to read again: Multi-granularity perception network for moment retrieval in videos. In *SIGIR*.

Tang, Z., Liao, Y., Liu, S., Li, G., Jin, X., Jiang, H., Yu, Q., and Xu, D. (2020). Human-centric spatio-temporal video grounding with visual transformers. *IEEE Transactions on Circuits and Systems for Video Technology*, 32:8238–8249.

Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., and Li, L.-J. (2016). YFCC100M: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). Learning spatiotemporal features with 3D convolutional networks. In *ICCV*.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In *CVPR*.

Wang, J., Ma, L., and Jiang, W. (2020). Temporally grounding language queries in videos by contextual boundary-aware prediction. In *AAAI*.

Wang, Y., Deng, J., Zhou, W., and Li, H. (2021). Weakly supervised temporal adjacent network for language grounding. *IEEE Transactions on Multimedia*.

Woo, S., Park, J., Koo, I., Lee, S., Jeong, M., and Kim, C. (2022). Explore and match: End-to-end video grounding with transformer. *arXiv:2201.10168*.

Wu, A. and Han, Y. (2018). Multi-modal circulant fusion for video-to-language and backward. In *IJCAI*.

Xiao, S., Chen, L., Shao, J., Zhuang, Y., and Xiao, J. (2021a). Natural language video localization with learnable moment proposals. *arXiv:2109.10678*.

Xiao, S., Chen, L., Shao, J., Zhuang, Y., and Xiao, J. (2021b). Natural language video localization with learnable moment proposals. *arXiv:2109.10678*.

Xiao, S., Chen, L., Zhang, S., Ji, W., Shao, J., Ye, L., and Xiao, J. (2021c). Boundary proposal network for two-stage natural language video localization. In *AAAI*.

Xie, S., Sun, C., Huang, J., Tu, Z., and Murphy, K. (2018). Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*.

Xu, H., He, K., Plummer, B. A., Sigal, L., Sclaroff, S., and Saenko, K. (2019). Multilevel language and vision integration for text-to-clip retrieval. In *AAAI*.

Xu, J., Mei, T., Yao, T., and Rui, Y. (2016). MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*.

Yang, W., Zhang, T., Zhang, Y., and Wu, F. (2021). Local correspondence network for weakly supervised temporal sentence grounding. *IEEE Transactions on Image Processing*, 30:3252–3262.

Yang, Y., Li, Z., and Zeng, G. (2020). A survey of temporal activity localization via language in untrimmed videos. In *Proc. of the IEEE International Conference on Culture-oriented Science & Technology (ICCST)*.

Yu, X., Malmir, M., He, X., Chen, J., Wang, T., Wu, Y., Liu, Y., and Liu, Y. (2021). Cross interaction network for natural language guided video moment retrieval. In *SIGIR*.

Yuan, Y., Lan, X., Wang, X., Chen, L., Wang, Z., and Zhu, W. (2021). A closer look at temporal sentence grounding in videos: Dataset and metric. In *Proc. of the International Workshop on Human-Centric Multimedia Analysis*.

Yuan, Y., Ma, L., Wang, J., Liu, W., and Zhu, W. (2019a). Semantic conditioned dynamic modulation for temporal sentence grounding in videos. *NIPS*.

Yuan, Y., Mei, T., and Zhu, W. (2019b). To find where you talk: Temporal sentence localization in video with attention based location regression. In *AAAI*.

Zeng, R., Xu, H., Huang, W., Chen, P., Tan, M., and Gan, C. (2020). Dense regression network for video grounding. In *CVPR*.

Zhang, B., Hu, H., Lee, J., Zhao, M., Chammas, S., Jain, V., Ie, E., and Sha, F. (2020a). A hierarchical multi-modal encoder for moment localization in video corpus. *arXiv:2011.09046*.

Zhang, D., Dai, X., Wang, X., Wang, Y.-F., and Davis, L. S. (2019a). MAN: Moment alignment network for natural language moment retrieval via iterative graph adjustment. In *CVPR*.

Zhang, H., Sun, A., Jing, W., Nan, G., Zhen, L., Zhou, J. T., and Goh, R. S. M. (2021a). Video corpus moment retrieval with contrastive learning. In *SIGIR*.

Zhang, H., Sun, A., Jing, W., Zhen, L., Zhou, J. T., and Goh, R. S. M. (2021b). Parallel attention network with sequence matching for video grounding. *arXiv:2105.08481*.

Zhang, H., Sun, A., Jing, W., and Zhou, J. T. (2020b). Span-based localizing network for natural language video localization. *arXiv:2004.13931*.

Zhang, H., Sun, A., Jing, W., and Zhou, J. T. (2023a). Temporal sentence grounding in videos: A survey and future directions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhang, S., Peng, H., Fu, J., and Luo, J. (2020c). Learning 2D temporal adjacent networks for moment localization with natural language. In *AAAI*.

Zhang, Y., Chen, X., Jia, J., Liu, S., and Ding, K. (2023b). Text-visual prompting for efficient 2D temporal video grounding. In *CVPR*.

Zhang, Y., Chen, X., Jia, J., Liu, S., and Ding, K. (2023c). Text-visual prompting for efficient 2D temporal video grounding. In *CVPR*.

Zhang, Z., Lin, Z., Zhao, Z., and Xiao, Z. (2019b). Cross-modal interaction networks for query-based moment retrieval in videos. In *SIGIR*.

Zhang, Z., Zhao, Z., Lin, Z., Huai, B., and Yuan, N. J. (2020d). Object-aware multi-branch relation networks for spatio-temporal video grounding. *arXiv:2008.06941*.

Zhang, Z., Zhao, Z., Zhao, Y., Wang, Q., Liu, H., and Gao, L. (2020e). Where does it exist: Spatio-temporal video grounding for multi-form sentences. In *CVPR*.

Zheng, M., Huang, Y., Chen, Q., and Liu, Y. (2022a). Weakly supervised video moment localization with contrastive negative sample mining. In *AAAI*.

Zheng, M., Huang, Y., Chen, Q., Peng, Y., and Liu, Y. (2022b). Weakly supervised temporal sentence grounding with Gaussian-based contrastive proposal learning. In *CVPR*.

Zheng, M., Li, S., Chen, Q., Peng, Y., and Liu, Y. (2023). Phrase-level temporal relationship mining for temporal sentence localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhou, L., Xu, C., and Corso, J. J. (2018). Towards automatic learning of procedures from web instructional videos. In *AAAI*.

CHECKLIST

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Not Applicable]
   (b) Complete proofs of all theoretical results. [Not Applicable]
   (c) Clear explanations of any assumptions. [Not Applicable]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes] See supplementary material and https://github.com/snuviplab/MoLEF.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes] See supplementary material.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Not Applicable]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes] See supplementary material.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes] See supplementary material.
   (b) The license information of the assets, if applicable. [Yes] See supplementary material and https://github.com/snuviplab/MoLEF.
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes] See supplementary material and https://github.com/snuviplab/MoLEF.
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Towards a Complete Benchmark on Video Moment Localization: Supplementary Materials

A SUMMARY OF MOMENT LOCALIZATION APPROACHES

The methods of the MLSV task are classified into three categories: 1) proposal-based, 2) proposal-free, and 3) others. Most models consist of feature extractors, encoders, multimodal fusion modules, and prediction heads, as illustrated in Fig. 2.

A.1 Proposal-based Approaches

Proposal-based methods first generate multiple candidate segments before matching them with the query. Depending on how they generate the proposals, they are further classified to sliding-window-based (Wu and Han, 2018; Jiang et al., 2019; Ge et al., 2019), proposal-generated (Xu et al., 2019; Chen and Jiang, 2019; Xiao et al., 2021c; Liu et al., 2021a), anchor-based, and 2D-based. In this study, we consider anchor-based and 2D-based methods for proposal-based approaches, because the sliding-window-based and proposal-generated methods suffer from high computational cost and show relatively weaker performance.

Anchor-based approaches (Yuan et al., 2019a; Zhang et al., 2019a; Qu et al., 2020; Wang et al., 2020) generate proposals with multi-scale anchors from input visual features before passing through encoders. TGN (Chen et al., 2018) introduces a dynamic single-stream architecture to capture fine-grained interactions between video frames and the query. CMIN (Zhang et al., 2019b) adopts multi-scale anchors based on different ratio of temporal scales centered at each time step, followed by cross-modal interaction with syntactic structure of queries and long-range semantic dependencies in videos. CSMGAN (Liu et al., 2020) adopts multi-scale anchors similar to CMIN, equipped with cross-modal and uni-modal graph attention networks. IA-Net (Liu et al., 2021b) introduces an iterative framework with co-attention and calibration to align semantics between the video and the query. Anchor-based methods also suffer from a limitation that the lengths of proposals are constrained to those of the pre-defined anchors.

2D-Map approaches (Gao and Xu, 2021; Hu et al., 2021) are proposed to overcome limitations of the anchor-based methods. They compute a two-dimensional map to flexibly consider proposals starting and ending at any point. 2D-TAN (Zhang et al., 2020c) models temporally adjacent relations of moments to match language and video features. RaNet (Gao et al., 2021) presents a graph-based network to capture visual and query representations by formulating this task as a video reading comprehension problem. MGPN (Sun et al., 2022) introduces reading strategies inspired by human reading habits through a multi-granularity perception network. TRM (Zheng et al., 2023) makes prediction at phrase-level, more fine-grained than the usual sentence-level, to better understand each semantic entity in the sentence.

A.2 Proposal-free Approaches

Proposal-free methods first compute a query-aware video representation as a sequence of features, and then predict the starting and ending frames of the segment described by the query. They are further classified into regression-based and span-based approaches.

Regression-based methods (Yuan et al., 2019b; Lu et al., 2019; Ghosh et al., 2019; Li et al., 2021; Chen et al., 2020a,b; Chen and Jiang, 2020) are trained to directly predict the scores of moments. DRN (Zeng et al., 2020) is composed of a dense regression network with a video-query interaction module to fuse the video and query features hierarchically. LGI (Mun et al., 2020) is another regression-based method, performing a sequential query attention to extract multiple semantic phrases from the text query and local-global video-text interaction to capture relations between candidate segments and the phrase.

PLRN (Kim et al., 2021) leverages semantic phrase representations following LGI (Mun et al., 2020), capturing comprehensive contextual information with one semantic phrase using an efficient center-width location regression loss.

Span-based methods (Yu et al., 2021; Zhang et al., 2021b), originally inspired by the reading comprehension task, compute the probability of a video segment to be the start and end point of a target moment. VSLNET (Zhang et al., 2020b) tackles the MLSV task as a multi-modal question answering problem by treating the video as a text passage and a moment as an answer span. A query-guided highlighting strategy is proposed to guide searching the target moment. TMLGA (Rodriguez et al., 2020) predicts the start and end frames by attending relevant features with guided attention and improves the capability of localization with a loss using the attended features. ReLoCLNet (Zhang et al., 2021a) is jointly trained on MLSV and MLVC tasks by utilizing both video-level and frame-level contrastive loss to align visual and textual representations. LVTR (Woo et al., 2022) randomly initializes proposals, explores time space, and then matches the corresponding target.

A.3 Other Approaches

In addition to the aforementioned approaches, weakly supervised learning approaches (Duan et al., 2018; Wang et al., 2021; Yang et al., 2021; Huang et al., 2021; Chen and Jiang, 2021; Song et al., 2020) introduce the MLSV task as an auxiliary task for representation learning or other downstream tasks. These methods are excluded from our study as they do not target the MLSV as their main task. However, CNM (Zheng et al., 2022a) proposes a weakly supervised learning for MLSV, generating a proposal based on a Gaussian mask, not sliding windows. Similarly, CPL (Zheng et al., 2022b) also incorporates Gaussian proposals, but utilizing multiple masks that jointly contribute to a unified positive proposal.

B DETAILED EXPERIMENTAL SETTINGS

Feature extractors. We employ the common experimental settings for each dataset. For video feature extractor, we use pre-trained I3D (Carreira and Zisserman, 2017) for Charades-STA, C3D (Tran et al., 2015) for ActivityNet, $\text{TACoS}_{\text{org}}$ and MSR-VTT, VGG (Simonyan and Zisserman, 2014) for DiDeMo, S3D (Xie et al., 2018) pre-trained on HowTo100M (Miech et al., 2019) with MIL-NCE (Miech et al., 2020) for YouCook2, and RestNet+I3D (Tran et al., 2018) for TVR. For Charades-STA, ActivityNet, $\text{TACoS}_{\text{org}}$ and MSR-VTT, the pre-trained network takes 16 video frames as one clip, and each clip overlaps 8 frames (the overlapping size is 0.5). For ActivityNet, we conduct dimension reduction from 4096 to 500 by PCA, following the standard practice. For DiDeMo and TVR, we use the officially provided visual features. In order to compare performance of different features (Sec. 7), we additionally use C3D and VGG features for Charades-STA.

For textual representation, we utilize the pre-trained GloVe (Pennington et al., 2014) embedding with $d = 300$ for all datasets except for TVR, where we use the officially provided RoBERTa (Liu et al., 2019) features with $d = 768$.

Training settings. We train all models with the Adam optimizer (Kingma and Ba, 2015). We adopt the optimal hyper-parameters reported in each original paper, and fine-tune them for several models whose performance is far from the reported one.

C IMPLEMENTATION DETAILS

In this section, we provide more implementation details that are useful for reproducing our results.

We generally follow the training setup of the existing works. Each model is trained for up to 100 epochs, on a single NVIDIA A100 GPU. The model and training hyperparameters are detailed for each benchmark dataset below:

TGN. We use an LSTM for sentence encoding, and the number of the anchors is set to 20 for Charades-STA and 100 for ActivityNet, respectively. Also, for TACoS, DiDeMo, YouCook2, MSR-VTT and TVR, we use 32 anchors. For training, we use the learning rate of 10^{-5} and clipping gradient norm of 100. To refine extracted samples, we set 0.85 as a threshold to extract positive samples and 0.15 for negative samples. The threshold for non-maximum suprression is set to 0.55.

CMIN. We cross-validate the anchor widths among {16, 24, 32, 40} for Charades-STA, {16, 32, 64, 96, 128, 160, 192} for ActivityNet, MSR-VTT, and TVR, and {8, 16, 32, 64} for TACoS, DiDeMo, and YouCook2. We construct the CMIN model with 2 GCN layers for the sentence encoder and 2 attention layers for the video encoder. We set the clearing threshold λ to 0.3, and the high-score threshold γ to 0.7. The final loss is controlled with balance weight of 0.001.

CSMGAN. We use the same anchor widths as described in CMIN. We use 2 attention layers for the video sentence encoder, and also 2 joint graph layers. The balance weight β in the final loss is set to 0.001 for ActivityNet and 0.005 for TACoS and Charades-STA, respectively, and we set the high-score threshold τ to 0.45 for all datasets.

IA-Net. This model also uses the same setting of anchors width adopted in CMIN. The balancing hyperparameter α of the final loss is set to 0.001 for ActivityNet and 0.005 for TACoS and Charades-STA datasets. Also, the high-score threshold λ is set to 0.45.

2D-TAN. We divide the dataset into 2 groups, *i.e.*, ActivityNet, MSR-VTT and TVR to group A, and TACoS, DiDeMo and YouCook2 to group B, and we apply separate hyper-parameters for each group. The min/max scaling thresholds, $t_{\min}$ and $t_{\max}$, are set to 0.5/1.0 for group A and 0.3/0.7 for group B, respectively. Besides, we use a 4-layer convolution network with kernel size of 9 for group A and an 8-layer convolution network with kernel size of 5 for group B. 2D-TAN adopts either max-pooling and stacked convolution for moment features extraction. We set corresponding clip-proposal modules for each, following the original paper.

RaNet. We use the same group division from 2D-TAN, and apply it to hyper-parameters of RaNet for each group. The min/max scaling threshold of group A is 0.5/1.0, while that of group B is 0.3/0.7.

MGPN. We use a bi-GRU for sentence encoding and the number of layers is 2 for ActivityNet, Charades-STA, DiDeMo, YouCook2, MSR-VTT, and TVR, while 3 for TACoS. Also, the size of all hidden states is 256 for all datasets. For training, we set the learning rate to 10^{-3}.

DRN. At the first stage, we use a learning rate of 10^{-4} for ActivityNet, DiDeMo, 10^{-5} for Charades-STA, TACoS, and 10^{-6} for MSR-VTT, TVR. At the second stage, the learning rate is decayed to 1/100 of the original value, following the original paper. This rate is also used at the fine-tuning stage.

LGI. We use 3 semantic phrases for sequential attention network for Charades-STA, TACoS, DiDeMo, and YouCook2 datasets (group A) and 5 semantic phrases for ActivityNet, MSR-VTT, and TVR datasets (group B). Also, we set the weight of 0.3 and 0.2 in the distinct query attention loss for group A and B, respectively. The number of heads of local-global video-text interaction is 4 and 5 for group A and B, respectively.

PLRN. We basically follow the training settings for LGI. We set a learning rate of 4×10^{-4} at training.

VSLNet. We set the dimensionality of all hidden layers to 128. We set the convolutional kernel size to 7, and the number of heads for multi-head attention to 8. We use an RNN for the predictor in the model. Also, to train the model, we adopt the gradient clip norm of 1 and the weight in the highlight loss of 5.0.

TMLGA. We set the learning rate to 10^{-4} and weight decay to 10^{-5}. Also, the dynamic filter consists of an LSTM with an average pooling layer, and for the prediction head, we use a multi-layer perceptron. The video feature size of the localization layer differs by dataset: 500 for ActivityNet, 1024 for Charades-STA, 4096 for TACoS, DiDeMo, and MSR-VTT, 512 for YouCook2, and 3072 for TVR.

ReLoCLNet. We set the learning rate to 10^{-4} for Charades-STA, DiDeMo, and YouCook2, and 10^{-5} for ActivityNet, TACoS, and TVR. The maximum length of video sequence is set to 128 for TVR and ActivityNet, 64 for Charades-STA, and 200 for other datasets. Besides, the maximum length of a text query is set to be 30 for TVR and 64 for others, following the original setting.

CNM. For the Transformer encoder, we set the dimensionality of token embeddings to 256, the number of attention heads to 4, and the number of layers to 3, respectively. At training, we set the contrastive loss parameters $\beta_1 = 0.1$ and $\beta_2 = 0.15$, respectively. The hyper-parameter α for controlling a variance over the Gaussian Mask is set to 5.

LVTR. We set the learning rate to 10^{-4} on every dataset and weight decay to 10^{-4} for ActivityNet, Charades-STA, TACoS, DiDeMo, MSR-VTT, and TVR, while 2×10^{-4} for YouCook2. We measure the correspondence between prediction and query based on dot product similarity for ActivityNet, Charades-STA, TACoS, DiDeMo, and

YouCook2, while cosine similarity for MSR-VTT and TVR. Although this model is designed to use multiple texts, we keep the number of input sentences per video to 1 to maintain the same experimental setting with other models. This setting might have led to degraded performance overall.

TRM. To create phrases for each dataset, we parse the sentence using the pre-trained SRLBERT (Shi and Lin, 2019), following the original paper. We retain only the semantic roles that occur in roughly more than 1/10 of the most frequent semantic role. The maximum number of frames is set to 64 for Charades and 256 for the others. Other hyperparameters remain consistent with the original implementation.

CPL. We set the number of positive proposals as 8. For the transformer encoder and decoder, we constructed 3 layers, each consisting of 4 attention heads with hidden dimensionality of 256. In terms of hyperparameters, we set $\sigma = 9, \gamma = 0, \beta_1 = 0.1, \beta_2 = 0.15$, and $\alpha_1 = 1$. α_2 was selected from $\{0.1, 1, 5\}$, and λ was set to be within the range $[0.13, 0.15]$ depending on datasets. We chose loss-based strategy for final prediction, as it demonstrated superior performance.

D DETAILS ON BENCHMARK DATASETS

ActivityNet Captions (Krishna et al., 2017) is originally introduced for dense video captioning task but is also adopted for MLSV. Its videos are originated from the ActivityNet dataset (Caba Heilbron et al., 2015), which aims to recognize human actions in videos. Since the test set is withheld for a competition, we utilize the 'val1' partition for validation and 'val2' for testing, following previous works.

Charades-STA is adapted from an indoor activity recognition dataset called Charades (Gao et al., 2017), by a semi-automatic process to make video-level descriptions to clip-level ones. As this dataset does not provide a set-aside validation set, we take 10% of the training set for validation used for righteous model selection.

TACoS contains 127 cooking scenario videos collected from MPII Cooking Composite Activities dataset (Rohrbach et al., 2012), originally designed to recognize human activities in a kitchen. There are two versions, TACoS$_{\text{org}}$ (Regneri et al., 2013) and TACoS$_{\text{2DTAN}}$ (Zhang et al., 2020c), and we use the former to compare the existing methods. Since only a small proportion of videos are accompanied by a long text query, TACoS is considered more challenging than other benchmarks. We follow the standard split of TACoS$_{\text{org}}$: 10,146 moment-query pairs for training, 4,589 for validation, and 4,083 for testing.

DiDeMo (Anne Hendricks et al., 2017) has its origin in an open domain dataset YFCC100M (Thomee et al., 2016), containing over 100k Flickr videos. Among them, 10k videos with an average length of 53.8 seconds are selected. Unlike other datasets, moments in DiDeMo start and end at a multiple of 5-seconds (*e.g.*, $t_s = 10, t_e = 15$). We follow the conventional split: 60,391, 7,679, and 7,361 video-query pairs for training, validation, and testing set, respectively.

YouCook2 (Zhou et al., 2018) contains 1,790 videos of 89 cooking recipes, taken at home with an unfixed camera. Up to first 20 minutes of each video is annotated with temporal boundaries and described in natural language. As this dataset does not provide a set-aside validation set, we take 10% of the training set for validation.

MSR-VTT (Xu et al., 2016) consists of clips and queries explaining each moment with annotations in 20 categories. From 7,180 videos, 10k clips are extracted and 200k sentences are labeled to them. This dataset is originally designed for clip retrieval, providing frames only within the moment with its start and end timestamp from its original video. We repurpose this dataset for MLSV by processing the entire video available on the web. Excluding currently unavailable videos, we use 5,127 videos.

TVR (Lei et al., 2020) is composed of 20k videos and 98k manually annotated text descriptions from TV shows. Since this dataset is stemmed from TV shows, videos and text queries contain rich social interaction between characters, making the benchmark more challenging. Ground truth is provided only for training and validation sets, while the test partition is withheld for a competition. Thus, we set-aside 1/8 of the training set for validation and use the original validation set for testing.

E FEATURE ENCODERS AND MULTIMODAL FUSION METHODS

Our Moment Localization Evaluation Framework (MoLEF) implements various visual and textual feature encoders as well as multimodal fusion methods that have been explored in literature.

E.1 Video Encoder

Various visual representations are used to capture temporal dependencies within a video. TGN adopts a simple LSTM to aggregate visual features. CMIN (Zhang et al., 2019b), CSMGAN (Liu et al., 2020), and IANet (Liu et al., 2021b) use multi-headed self-attention module to capture long-range semantic dependencies and then employ bidirectional GRU to incorporate the contextual information. 2D-TAN (Zhang et al., 2020c), RaNet (Gao et al., 2021), and MGPN (Sun et al., 2022) consist of variant convolution blocks with average pooling and max pooling, to capture the contextual information in video features. VSLNet (Zhang et al., 2020b) and ReLoCLNet (Zhang et al., 2021a) adopt Transformer blocks as a video encoder. LGI (Mun et al., 2020) and PLRN (Kim et al., 2021) use an embedding layer followed by a ReLU function.

E.2 Text Encoder

For text encoders, several variants of LSTMs, GRUs, or Transformers are used to integrate the sequential information. LSTM is employed for 2D-TAN (Zhang et al., 2020c) and TGN (Chen et al., 2018), while RNN is used for LGI (Mun et al., 2020) and PLRN (Kim et al., 2021). RaNet (Gao et al., 2021) adopts a bidirectional LSTM, and a bidirectional GRU is used for IANet (Liu et al., 2021b), TMLGA (Rodriguez et al., 2020), and MGPN (Sun et al., 2022). CMIN (Zhang et al., 2019b) uses a bidirectional GRU and graph convolution block networks to consider syntactic dependency graph. CSMGAN (Liu et al., 2020) uses convolutional blocks, LSTM, and GRU layers to encode textual reperentations. VSLNet (Zhang et al., 2020b) and ReLoCLNet (Zhang et al., 2021a) employ Transformer blocks to capture better contextual representations of the query.

E.3 Multi-modal Fusion

Various multimodal fusion methods have been introduced to aggregate textual and visual features. As a simple method, 2D-TAN (Zhang et al., 2020c), ReLoCLNet (Zhang et al., 2021a), LGI (Mun et al., 2020), RaNet (Gao et al., 2021), and PLRN (Kim et al., 2021) use a Hardmard product to encode both text and visual representations, and then employ additional layers such as convolution blocks, non-local blocks, or fully-connected layers. For a better aggregation, TMLGA (Rodriguez et al., 2020) and MGPN (Sun et al., 2022) adopt the attention mechanism. Especially, IANet (Liu et al., 2021b) propose inter- and intra-attention to consider modal relations and calibration module for alignment refinement. CMIN (Zhang et al., 2019b), TGN (Chen et al., 2018), and DRN (Zeng et al., 2020) apply the attention mechanism, in conjunction with additional layers such as LSTM, bidirectional LSTM, or cross-gating mechanism. VSLNet (Zhang et al., 2020b) considers two-way attention mechanism to encode visual-to-query and query-to-visual features, while CNM (Zheng et al., 2022a) and CPL (Zheng et al., 2022b) use a Transformer block with mask conditioned attention. Also, CSMGAN (Liu et al., 2020) applies a cross-modal relation graph and self-modal relation graph to integrate information among cross-modal relations and to capture self-attentive contexts within relations.

F MORE EXPERIMENTAL ANALYSIS

Tab. I reports the performance of competing models on an additional metric, the mean Intersection over Union (mIoU):

$$\text{mIoU} = \frac{1}{N}\sum \frac{\bigcap\left([\hat{t}_s, \hat{t}_e], [t_s, t_e]\right)}{\bigcup\left([\hat{t}_s, \hat{t}_e], [t_s, t_e]\right)}, \tag{1}$$

where $[t_s, t_e]$ and $[\hat{t}_s, \hat{t}_e]$ are the ground truth and predicted moments between start and end points, respectively, and N is the number of samples. Along with the Recall@k with IoU=p analyzed in Sec. 4, the mIoU metric is also often used for evaluation.

According to Tab. I, the overall trend is similar to Recall@k with IoU=p in Tab. 3; for instance, CSMGAN achieves the strongest performance on YouCook2, while LGI is the strongest on TACoS, and so on. However, some models achieve relatively stronger performance in mIoU. VSLNet, for example, achieves the best performance on DiDeMo and TVR, while it lags behind other competing models in Recall@k with IoU=p. 2D-TAN is an example of the opposite, achieving much stronger performance in Recall@k with IoU=p, while relatively weaker in mIoU. This results indicates that evaluating in multiple metrics does matter to draw a more trustworthy conclusion when compare models.

Table I: Comparison results of mIoU (%) of the competing models.

Category	Approach	mIoU ↑						
	Dataset	ActivityNet	Charades-STA	TACoS	DiDeMo	YouCook2	MSR-VTT	TVR
AN-based	TGN	33.46	36.75	15.03	20.53	3.80	0.60	10.47
	CMIN	43.74	46.74	17.71	17.55	35.17	5.32	15.3
	CSMGAN	46.18	22.87	13.03	20.5	**43.25**	1.32	17.94
	IA-Net	**46.30**	43.11	16.11	14.02	21.81	5.35	17.92
2D-based	2D-TAN	43.36	41.59	21.63	21.01	27.04	**7.92**	18.34
	RaNet	44.38	46.32	18.02	22.24	33.52	2.15	13.52
	MGPN	44.35	47.58	14.59	20.19	27.73	5.63	23.03
	TRM	45.78	44.92	13.10	20.36	27.98	4.74	28.55
Regression	DRN	36.16	43.01	4.09	20.83	8.84	5.72	11.69
	LGI	40.73	45.35	**22.51**	21.01	16.24	7.27	16.54
	PLRN	36.92	44.34	10.04	20.86	15.67	6.22	19.91
Span-based	VSLNet	40.48	47.61	17.32	**22.30**	22.77	5.88	**30.16**
	TMLGA	35.61	**48.21**	14.39	19.30	36.15	6.05	13.85
	ReLoCLNet	21.23	22.74	3.80	15.24	5.45	5.70	16.95
	LVTR	20.95	33.83	4.24	22.07	6.69	4.83	12.05
Others	CNM	36.77	32.90	3.09	17.86	6.60	7.46	13.12
	CPL	34.39	33.65	3.24	17.38	13.67	6.85	14.57

G MOMENTS DISTRIBUTION OF DATASETS

In addition to the length of a video, we investigate the distribution of moments of each dataset with the start and end point in Fig. I. Charades-STA and ActivityNet have a few dense points on the distribution, while others have widely spread dense areas. As seen in this visualization, a model is subject to be biased to these densely observed cases, instead of solving the task without prejudice. See Sec. 6.1 for more discussion.

H DETAILED EXPERIMENTAL DESIGN ON BIAS

For the annotation bias experiment (Sec. 6.1), we create test-IID and test-OOD sets as follows. First, we merge the training, validation, and test sets for each dataset and normalize the start and end point of each video moment with each length. For example, the start/end moment of $[0, 15]$ in a 30-seconds video is normalized with $[0, 0.5]$. Thus, all moments start as early as 0 and ends as late as 1, regardless of the length of the video they belong to. Second, the normalized moments are sorted by the central point of each and the last 10% of them are used for test-OOD set. For ActivityNet, Charades-STA, and YouCook2, we use the threshold of 0.55, 0.83, and 0.61, respectively. Third, the rest of the dataset is divided into training (70%), validation (10%) and test-IID sets (10%), uniformly randomly. (Here, the ratio means from the original distribution including the test-OOD partition.) Thus, the videos in different partitions do not overlap. Fig. II shows the normalized moment distributions of generated train and validation sets. As shown in Fig. 3 and Fig. II, the train, validation, test-IID sets have similar distributions, but the test-OOD set has a completely different distribution.

For the query text bias experiment (Sec. 6.2), we generate two test sets: 1) the action verb is masked out (M), and 2) only the action verb is left (A). To generate the sets, we use the part-of-speech tagging method of Spacy library.[1] If the part-of-speech tag of each token is a verb, the token is masked out for set M and left for set A. Specifically, we consider the first verb for this generation process because the first verb usually contains the key explanation of subjects in the described sentence.

I BROADER IMPACT

Video moment localization task requires a comprehensive understanding of the video and the textual query, as well as alignment between them. Our work basically provides a common evaluation framework for current

[1] `https://spacy.io/`

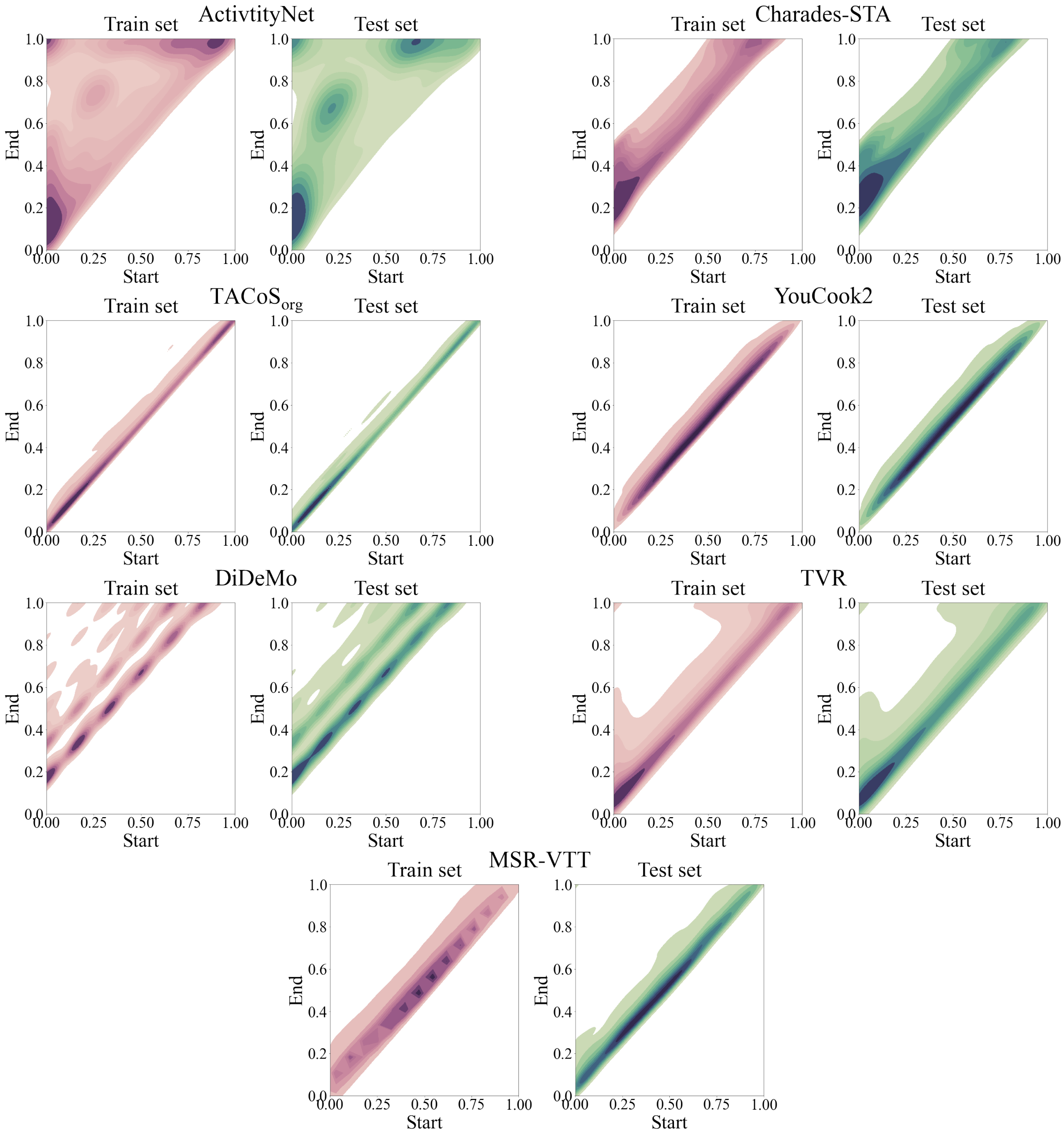


Figure I: Normalized moment distribution of all datasets.

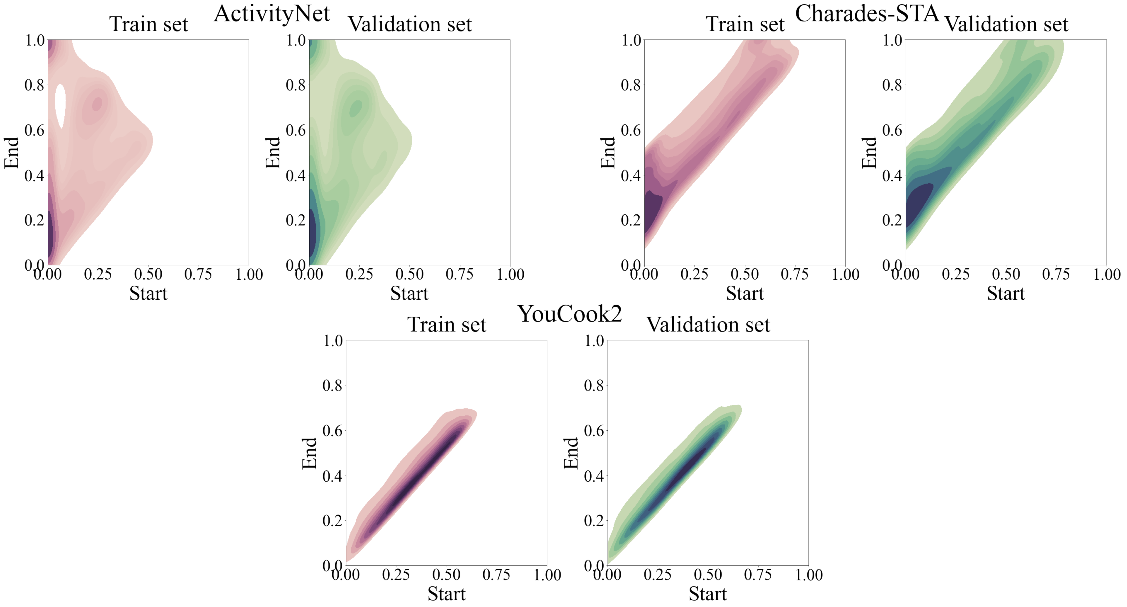


Figure II: Normalized moment distribution of ActivityNet, Charades-STA, and YouCook2 datasets, based on our split for query-text bias experiment.

state-of-the-art models and experimental analysis on them using public datasets. Thus, basically, this work does not bring any primary concern on fairness, privacy, or other foreseeable negative impacts on society. Potentially, this work may assist numerous applications, such as video surveillance, robotic navigation, autonomous driving, sports analytics, and more.

However, the datasets used in the comparison have not been created in a way considering fairness (Raji et al., 2020) or privacy (Jeon et al., 2022, 2023), although these datasets are widely used in computer vision research community. Thus, models trained on these datasets might be biased to specific races, countries, or societies, and the experimental conclusions that we have drawn on top of these models might have been affected as well. As our study does not directly address this problem, this potential issue might remain unresolved.

Also, the technology itself that enables easy retrieval of a visual content from videos from natural language can be ambivalent. This technology can be used for beneficial purposes (*e.g.*, identifying theft or violence in security camera, monitoring unapproved trespassing). However, if misused in scenarios such as unauthorized recording of videos involving unidentified individuals, many previously overlooked aspects could be more easily discovered. Also, video recordings should be more thoroughly protected and managed, since an information leak would cause much more serious problems including privacy issues when equipped with this technology. Addressing these aspects would likely require societal consensus and careful consideration.