SnAG: Scalable and Accurate Video Grounding (Supplementary Material)

Fangzhou Mu*, Sicheng Mo*, Yin Li
University of Wisconsin-Madison

In the supplementary material, we (1) elaborate on our cost analysis in the main paper (Section A); (2) compare SnAG to latest methods with more advanced learning paradigms (Section B); (3) analyze the effect of input snippet length in training on model performance (Section C); (4) conduct error analysis of our results in comparison to baseline results (Section D); (5) describe additional implementation details (Section E); and (6) discuss the limitations and future directions of our work (Section F). We hope that this document will complement our main paper.

For sections, figures, tables, and equations, we use numbers (*e.g.*, Sec. 1) to refer to the main paper and capital letters (*e.g.*, Sec. A) to refer to this supplement.

A. Cost Analysis

In Section 3.2 of the main paper, we have shown that late fusion, under the assumption of $\mathcal{C}(\mathcal{F}_L) \ll \mathcal{C}(g) + \mathcal{C}(h)$ and $\mathcal{C}(g) \ll \mathcal{C}(h)$, allows a maximum of N times speedup in theory at inference time. Here N is the number of text queries per video, $\mathcal{C}(g)$ and $\mathcal{C}(h)$ denotes the respective cost of video and text encoding, and $\mathcal{C}(\mathcal{F}_L)$ combines the cost of cross-modal fusion and moment decoding.

In practice, for any realistic instantiation of $\mathcal{F}_L$, like our SnAG, the cost of cross-modal fusion and moment decoding $\mathcal{C}(\mathcal{F}_L)$ can not be ignored. This partially breaks our assumption of $\mathcal{C}(\mathcal{F}_L) \ll \mathcal{C}(g) + \mathcal{C}(h)$. To account for non-trivial cost $\mathcal{C}(\mathcal{F}_L)$, we denote $\mathcal{C}(\mathcal{F}_L) = \alpha(\mathcal{C}(g) + \mathcal{C}(h))$, where the positive coefficient α reflects the cost of $\mathcal{F}_L$ relative to the total cost of both encoders g and h. For late fusion, α is typically in the range of $(0, 1)$. Following Section 3.2, we have

$$\mathbf{R}_{\text{inf}} = \frac{\mathcal{C}(\text{inference with early fusion})}{\mathcal{C}(\text{inference with late fusion})} \tag{A}$$

$$= \frac{MN\,\mathcal{C}(\mathcal{F}_L) + MN\,\mathcal{C}(g) + MN\,\mathcal{C}(h)}{MN\,\mathcal{C}(\mathcal{F}_L) + M\,\mathcal{C}(g) + N\,\mathcal{C}(h)} \tag{B}$$

$$= \frac{MN(1+\alpha)}{MN\alpha + M\mathbf{R}_{g/(g+h)} + N(1-\mathbf{R}_{g/(g+h)})} \tag{C}$$

$$\approx \frac{1+\alpha}{\frac{1}{N}+\alpha} \rightarrow \begin{cases} N, & \alpha \rightarrow 0 \\ 1+\frac{1}{\alpha}, & N \rightarrow \infty \end{cases}, \tag{D}$$

* indicates equal contribution

where the approximation in Equation D is a consequence of $\mathbf{R}_{g/(g+h)} = \mathcal{C}(g)/(\mathcal{C}(g)+\mathcal{C}(h)) \xrightarrow{\mathcal{C}(g) \gg \mathcal{C}(h)} 1$.

Substituting N with B_q (the number of queries drawn for each training snippet), we can derive similar efficiency result for our video-centric training scheme (Section 3.3):

$$\mathbf{R}_{\text{train}} \approx \frac{1+\alpha}{\frac{1}{B_q}+\alpha} \rightarrow \begin{cases} B_q, & \alpha \rightarrow 0 \\ 1+\frac{1}{\alpha}, & B_q \rightarrow \infty \end{cases}. \tag{E}$$

For long-form video grounding (*i.e.*, large N and B_q), our extended analysis indicates that late fusion remains highly efficient when α is reasonably small. For example, a late-fusion model with $\alpha \rightarrow 1$ can in theory bring a considerable $2\times$ cost reduction in both training and inference relative to early fusion. This is precisely the case with SnAG, as Figure 5(a) of the main paper shows that SnAG's encoders account for $\sim$50% of the total computation.

Importantly, our empirical findings confirms this estimate. Figure 5(b, d) of the main paper shows that video-centric training and inference of SnAG on MAD is $\sim$50% more efficient in terms of runtime and GPU memory. On TACoS and Ego4D, the saving in training is less prominent as we found that smaller B_q leads to slightly better results, yet the substantial reduction in inference time echoes our cost analysis and demonstrates the efficiency of SnAG.

B. Additional Comparisons

We further compare SnAG to latest approaches that (1) consume additional input modalities (*e.g.*, audio), (2) are trained end to end on raw videos (as opposed to using frozen video backbones); or (3) perform large-scale, multi-task pre-training. We note that these are not fair comparisons and put SnAG at a disadvantage. This comparative analysis, however, further highlights the strength of SnAG.

Comparison with Barrios *et al.* [1]. Barrios *et al.* [1] has demonstrated a method for long-form video grounding via Multimodal Guidance. The guidance model takes audio in addition to video and text queries as input, and produces a score to condition an existing grounding model.

Table A compares the performance of existing models with Multimodal Guidance and SnAG. While the audio input provides additional cues to facilitate accurate moment

Model	R@1			R@5			R@10			R@50		
	0.1	0.3	0.5	0.1	0.3	0.5	0.1	0.3	0.5	0.1	0.3	0.5
† CLIP [17]	9.30	4.65	2.16	18.96	13.06	7.40	24.30	17.73	11.09	39.79	32.23	23.21
† VLG-Net [21]	5.60	4.28	2.48	16.07	13.14	8.78	23.64	19.86	13.72	45.35	39.77	30.22
† Moment-DETR [10]	5.07	3.82	2.39	16.30	12.60	7.90	24.79	19.43	12.06	50.06	40.52	24.87
SnAG	**10.28**	**8.46**	**5.55**	**24.42**	**20.60**	**13.75**	**32.23**	**27.50**	**19.00**	**52.28**	**46.68**	**35.24**

(a) Results on MAD [20].

Model	mean R@1	mean R@5
† Moment-DETR [10]	7.28	22.14
† VSL-Net [23]	11.90	24.06
SnAG (EgoVLP)	**15.59**	**38.40**

(b) Results on Ego4D-NLQ [8].

Table A. **Comparison with Multimodal Guidance [6].** SnAG does not take audio as input, yet outperforms existing models with Multi-modal Guidance (†) using audio. Mean R@{1,5} are computed over tIoU={0.1,0.3,0.5}. For SnAG, we report the better results between video-centric and query-centric training.

Model	R@1		R@5	
	0.3	0.5	0.3	0.5
TCSF	49.82	38.53	68.60	59.89
SnAG (C3D)	**56.44**	**44.86**	**81.15**	**70.66**

(a) Results on TACoS [18].

Model	R@1		R@5	
	0.5	0.7	0.5	0.7
TCSF	53.85	37.20	90.86	58.95
SnAG (I3D)	**65.13**	**46.26**	**92.80**	**71.94**

(b) Results on Charades-STA [19].

Model	R@1		R@5	
	0.3	0.5	0.3	0.5
TCSF	**66.87**	48.38	88.75	80.24
SnAG (C3D)	63.58	**48.55**	**89.55**	**81.71**

(c) Results on ActivityNet-Captions [9].

Table B. **Comparison with TCSF [6].** SnAG outperforms TCSF despite using frozen video features. For SnAG, we report the better results between video-centric and query-centric training.

localization, it does not fully close the gap between the existing models and SnAG. SnAG is 3.07 absolute percentage points higher in R@1, tIoU=0.5 on MAD, and 1.67 points higher in mean R@1 on Ego4D. We hypothesize that Multi-modal Guidance will also boost the performance of SnAG.

Comparison with TCSF [6]. TCSF takes a compressed video and obtains features from keyframes without explicitly decoding the video. Importantly, the feature extractor is jointly trained with the grounding model. Joint training has

Model	R@1		
	0.3	0.5	avg
UniVTG	7.28	3.95	5.62
UniVTG w/ PT	11.74	7.54	9.64
SnAG (EgoVLP)	**15.87**	**11.26**	**13.57**

(a) Results on Ego4D-NLQ [8].

Model	R@1		
	0.3	0.5	avg
UniVTG	51.44	34.97	43.21
UniVTG w/ PT	56.11	43.44	49.78
SnAG (C3D)	**56.44**	**44.86**	**50.65**

(b) Results on TACoS [18].

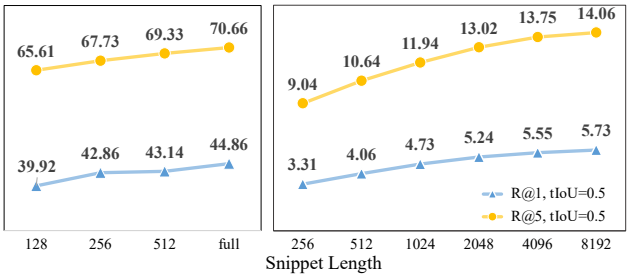
Model	R@1		
	0.5	0.7	avg
UniVTG	58.01	35.65	46.83
UniVTG w/ PT	60.19	38.55	49.27
SnAG (I3D)	**65.13**	**46.26**	**55.70**

(c) Results on Charades-STA [19].

Table C. **Comparison with UniVTG [11].** SnAG consistently outperforms UniVTG despite being simpler and learning from individual datasets without multi-task pre-training. w/ PT means fine-tuning after pre-training. For SnAG, we report the better results between video-centric and query-centric training.

significant benefit as it allows video features to adapt to the grounding task through end-to-end learning.

Table B compares the performance of TCSF and SnAG. Despite the extra capacity to learn more powerful features,



(a) TACoS (b) MAD

Figure A. **Ablations on snippet length in training.** On both TACoS (a) and MAD (b), SnAG delivers stronger test accuracy when trained on snippets of increased length.

TCSF lags behind SnAG by a significant margin. Consider the R@1, tIoU=0.5 scores — SnAG outperforms TCSF by 6.33 absolute percentage points on TACoS, and 11.28 absolute percentage points on Charades. The two methods are on par on ActivityNet (48.38 *vs*. 48.55).

Comparison with UniVTG [11]. UniVTG pools large-scale datasets and annotations from multiple video understanding tasks and learns a unified grounding model through multi-task pre-training. The pre-trained model supports fine-tuning and zero-shot inference on the training tasks, including temporal sentence grounding.

Table C compares the performance of UniVTG and SnAG. While UniVTG is capable of learning from diverse data and supervision signals, its performance lags behind SnAG, despite using stronger SlowFast [7] and CLIP [17] features. We again consider the R@1, tIoU=0.5 scores — SnAG achieves 11.26% on Ego4D, 44.85% on TACoS, and 64.62% on Charades, outperforming UniVTG by 3.72, 1.41 and 4.43 absolute percentage points, respectively.

C. Effect of Snippet Length

We conduct ablation studies on MAD and TACoS to understand the impact of snippet length in training.

Experiment setting. On TACoS, we emulate feature sub-sampling in previous methods [12, 21, 24] by resizing input video features to a uniform length of 128, 256 and 512, and compare to training using full-sized features (capped at a length of 2304). On MAD, we are able to vary snippet length from 256 (1 minute) to 8192 (1 hour) thanks to SnAG's scalable model design and training scheme. We report R@1 and R@5 at tIoU=0.5 for both datasets.

Results. Figure A presents our results. Training on longer snippets constantly leads to higher test accuracy for both datasets. On MAD, R@1, tIoU=0.5 grows by a significant 73% (from 3.31% to 5.73%) even though all moments are short (4.1 seconds on average) and can be fully covered by the shortest snippet length of 256. We conjecture that negative moments that are distant in time can provide strong learning signal to models like SnAG, which is capable of processing long-form video inputs.

D. Error Analysis

To better understand our results, we present an error analysis and compare our results with those from strong baselines of VLG-Net [21] and 2D-TAN [24]. This analysis is conducted on the test split of TACoS.

Experiment setting. In our first experiment, we divide the ground-truth moments into five bins according to their coverage (*i.e*., moment length relative to full video length): Extra Small (XS: (0, 0.02]), Small (S: (0.02, 0.04]), Medium (M: (0.04, 0.08]), Large (L: (0.08, 0.16]) and Extra Large (XL: (0.16, 1]). In our second experiment, we divide the videos into five bins based on their duration in seconds: Extra Small (XS: (0, 125]), Small (S: (125, 250]), Medium (M: (250, 500]), Large (L: (500, 750]), Extra Large (XL: (750, ∞)). We report mean R@1 averaged over the tIoU thresholds of 0.3 and 0.5 for each bin.

Results. Figure B presents our results. While our model consistently outperforms the baselines by a significant margin across all conditions, the relative percentage of improvement is especially large on low-coverage moments (XS and S) and long videos (L and XL). Note that both VLG-Net and 2D-TAN have a complexity of $\mathcal{O}(T^2)$ and thus need to sub-sample input video features to a small length (256 for VLG-Net and 128 for 2D-TAN) to reduce their computational cost. Intuitively, feature sub-sampling drops detailed information about event dynamics and makes precise localization of short moments more challenging. This becomes especially problematic for long videos as the drop rate can be extremely large (*e.g*., 10$\times$ for XL videos). Our results thus suggest that the ability to scale up to long videos is critical for enhancing localization accuracy.

E. Implementation Details

We now present the implementation details of SnAG. Table D summarizes key hyper-parameters we used in our experiments for feature extraction, training and inference.

Video features. For TACoS [18], Charades-STA [19] and ActivityNet-Captions [9], we extract clip-level video features using C3D network pre-trained on Sports-1M [22], following prior approaches (*e.g*., [12, 21, 24]). We decode TACoS videos at a frame rate of 29.4 frames per second (FPS) and compute the 4096-dimensional, post-ReLU activations of *fc6* layer as features using non-overlapping clips of 16 frames. For Charades-STA, we use the RGB frames provided on the official website . We use clips of 16 frames with a stride of 4 frames (*i.e*., 75% overlap), and compute the 4096-dimensional, post-ReLU activations of *fc7* layer as features. We use official C3D features for ActivityNet . For

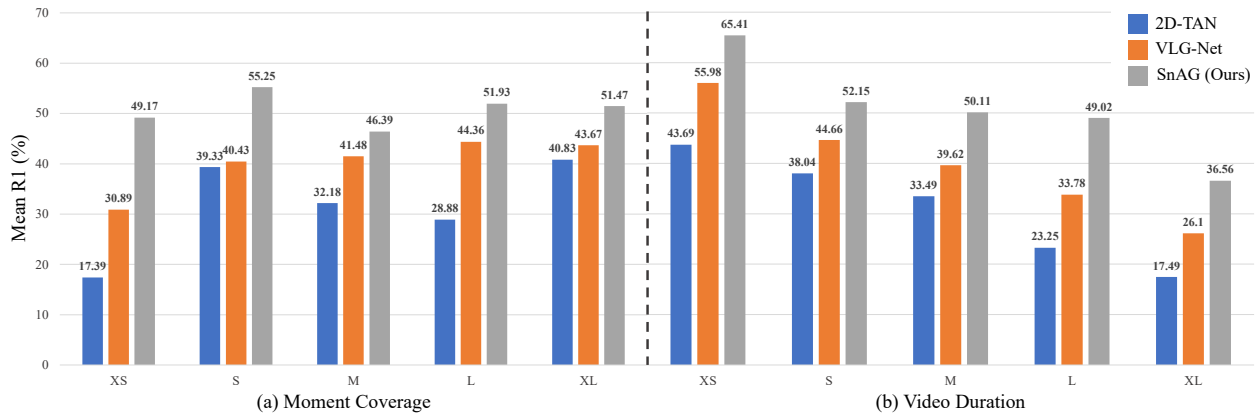


Figure B. **Error analysis on TACoS.** Our model consistently outperforms VLG-Net [21] and 2D-TAN [24] on moments with low (XS) to high (XL) coverage and videos with short (XS) to long (XL) duration. It is especially strong on low-coverage moments (a) and long videos (b) thanks to its scalability with respect to video duration.

Dataset	Video FPS	Video Feature	Clip Size	Stride	Snippet Length	Video Dim	Win Size	Video Layers	Text Feature	Text Dim	Text Layers	Batch Size	B_q	lr	Epochs
MAD	5	CLIP	1	1	4096	512	17	9	CLIP	512	1	32	8	1e-4	8
Ego4D-NLQ	30	SlowFast	32	16	2304	512	19	8	BERT	128	1	2	2	1e-4	10
		EgoVLP				384			EgoVLP	768		2	2		7
TACoS	29.4	C3D	16	16	2304	128	19	8	GloVe	128	5	16	4	1e-3	15
Charades-STA	24	C3D	16	4	256	256	5	7	GloVe	128	5	16	4	1e-3	10
		I3D										16	4		
ANet-Captions	varying	C3D	16	8	256	128	5	7	GloVe	128	5	16	2	1e-3	15

Table D. **Implementation Details.** We list key parameters for video encoding (left), text encoding (middle), and model training (right).

fair comparison with baselines, we additionally extract two-stream I3D features for Charades-STA from I3D network pre-trained on Kinetics [3] using clips of 16 frames with a stride of 4 frames. For Ego4D-NLQ [8], we use official SlowFast features [7] as well as EgoVLP features [16] , both extracted from 30 FPS videos using clips of 32 frames with a stride of 16 frames (*i.e.*, 50% overlap). For MAD [20], we use the official CLIP features [17] extracted at 5 FPS .

Textual features. We use GloVe word embeddings [15] with 6B vocabulary for TACoS, Charades-STA and ActivityNet-Captions. For Ego4D-NLQ, we use last-layer token embeddings (CLS token excluded) from BERT [5] or EgoVLP text encoder together with SlowFast or EgoVLP video features. For MAD, we use official CLIP token embeddings (SOT and EOT tokens excluded) from the last layer of text encoder.

Network architecture. For the video encoder, we apply a linear projection followed by two 1D convolutional layers with kernel size 3 on the input features prior to building the multi-scale video representation. The embedding dimension, attention window size and number of scales in the multi-scale Transformer network differ by dataset (see Table D for a summary). For the text encoder, we apply a single Transformer layer for MAD and Ego4D-NLQ, and 5 layers for the other datasets. We use 4 attention heads for all Transformer layers in the model. The classification and regression heads each contains two 1D convolutional layers with kernel size 3.

Training details. We randomly sample video snippets with a maximum sequence length of 4096, 2304 and 2304 for MAD, Ego4D-NLQ and TACoS, respectively. We resize all videos to a uniform length of 256 for Charades-STA and ActivityNet. We use the AdamW optimizer [13] with the default beta values of (0.9, 0.999) and a weight decay of 0.05. Mini-batch size (including B_q for video-centric training), learning rate and number of training epochs can be found in Table D. We maintain an exponential moving average (EMA) of model parameters over the entire course of training with a momentum of 0.999 and use the EMA version of the trained model for inference.

Inference details. Following our video-centric inference protocol, we first feed full videos into our model to compute the shared video representation, and then input text queries and fuse them with the video representation. Moments are then decoded from the fused representation and further merged using SoftNMS [2].

F. Discussion and Limitation

As we mentioned in Section 3.4 of the main paper, the design of SnAG is one of many possible instantiations of our analysis. As long as late fusion and video-centric training is employed, alternative designs can be considered, includ-

ing two-stage proposal-based methods or DETR-alike models. We anticipate that exploration of this design space may point to an interesting future avenue.

Similar to prior methods, SnAG's performance heavily depends on the quality of pre-extracted video and sentence features. These features are from networks pre-trained on a related yet different task (*e.g.*, action recognition or image-text matching) and may not be optimal for video grounding. Training of SnAG further requires moment and sentence annotations that are expensive and difficult to acquire. End-to-end training [4] of SnAG with less human annotation [14] is thus a promising future direction.

References

[1] Wayner Barrios, Mattia Soldan, Alberto Mario Ceballos-Arroyo, Fabian Caba Heilbron, and Bernard Ghanem. Localizing moments in long video via multimodal guidance. In *ICCV*, 2023. 1

[2] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-NMS–improving object detection with one line of code. In *ICCV*, 2017. 4

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 4

[4] Feng Cheng, Mingze Xu, Yuanjun Xiong, Hao Chen, Xinyu Li, Wei Li, and Wei Xia. Stochastic backpropagation: a memory efficient strategy for training video models. In *CVPR*, 2022. 5

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2018. 4

[6] Xiang Fang, Daizong Liu, Pan Zhou, and Guoshun Nan. You can ground earlier than see: An effective and efficient pipeline for temporal sentence grounding in compressed videos. In *CVPR*, 2023. 2

[7] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast networks for video recognition. In *ICCV*, 2019. 3, 4

[8] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022. 2, 4

[9] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017. 2, 3

[10] Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. *NeurIPS*, 2021. 2

[11] Kevin Qinghong Lin, Pengchuan Zhang, Joya Chen, Shraman Pramanick, Difei Gao, Alex Jinpeng Wang, Rui Yan, and Mike Zheng Shou. UniVTG: Towards unified video-language temporal grounding. In *ICCV*, 2023. 2, 3

[12] Daizong Liu, Xiaoye Qu, Xing Di, Yu Cheng, Zichuan Xu, and Pan Zhou. Memory-guided semantic learning network for temporal sentence grounding. In *AAAI*, 2022. 3

[13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 4

[14] Niluthpol Chowdhury Mithun, Sujoy Paul, and Amit K Roy-Chowdhury. Weakly supervised video moment retrieval from text queries. In *CVPR*, 2019. 5

[15] Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global vectors for word representation. In *EMNLP*, 2014. 4

[16] Kevin Qinghong Lin, Alex Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Zhongcong Xu, Difei Gao, Rongcheng Tu, Wenzhe Zhao, Weijie Kong, et al. Egocentric video-language pretraining. *NeurIPS*, 2022. 4

[17] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2, 3, 4

[18] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. Grounding action descriptions in videos. *TACL*, 2013. 2, 3

[19] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. 2, 3

[20] Mattia Soldan, Alejandro Pardo, Juan León Alcázar, Fabian Caba, Chen Zhao, Silvio Giancola, and Bernard Ghanem. MAD: A scalable dataset for language grounding in videos from movie audio descriptions. In *CVPR*, 2022. 2, 4

[21] Mattia Soldan, Mengmeng Xu, Sisi Qu, Jesper Tegner, and Bernard Ghanem. VLG-net: Video-language graph matching network for video grounding. In *ICCVW*, 2021. 2, 3, 4

[22] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 3

[23] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based localizing network for natural language video localization. In *ACL*, 2020. 2

[24] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. Learning 2d temporal adjacent networks for moment localization with natural language. In *AAAI*, 2020. 3, 4