Method		$Rank1@$			$Rank5@$		
		0.3	0.5	0.7	0.3	0.5	0.7
MCN		39.35	21.36	6.43	68.12	53.23	29.70
CTRL		47.43	29.01	10.34	75.32	59.17	37.54
TGN		43.81	27.93	–	54.56	44.20	–
ACRN		49.70	31.67	11.25	76.50	60.34	38.57
CMIN		**63.61**	*43.40*	*23.88*	*80.54*	*67.95*	*50.73*
QSPN		52.13	33.26	13.43	77.72	62.39	40.78
ABLR		55.67	36.79	–	–	–	–
TripNet		48.42	32.19	13.93	–	–	–
2D-TAN	Pool	*59.45*	**44.51**	26.54	85.53	**77.13**	61.96
	Conv	58.75	44.05	**27.38**	**85.65**	76.65	**62.26**

Table 2: Performance comparison on ActivityNet Captions.

$IoU@0.5$ and 0.7. It validates that 2D-TAN is able to localize the moment boundary more precisely.

In more details, by comparing 2D-TAN with other related methods, we obtain serveral observations. First, we compare 2D-TAN with previous sliding window based methods: MCN, CTRL, ACRN, ACL-K and VAL. From the results in Table 1–3, we observe that our 2D-TAN achieves superior results than sliding window methods. The reason is that independently matching the sentence with moment candidates ignores the temporal dependencies, and cannot distinguish the small differences between overlapped moments. Differently, our proposed 2D-TAN models the dependencies between moment candiates by a 2D temporal map, and enables the network to perceive more context information from the adjacent moment candidates. Hence, it gains large improvements compared to sliding window based methods.

Moreover, we compare our approach with RNN-based methods, i.e. TGN and CMIN. Due to the involvement of context information during prediction, the RNN-based approaches perform better than the sliding window approaches, however, inferior to our proposed 2D-TAN method. RNN-based approaches implicitly update the context information through a recurrent memory module, while our 2D-TAN explicitly exploit the context information via a 2D temporal map. It further verifies the effectiveness of our model in high quality moment localization.

Last, we compare our method with graph convolutional netowrk (GCN) based method MAN (**?**), which achieves the state-of-the-art on Charades-STA. It utilizes a GCN to model the relations between moment pairs. Differently, our 2D-TAN models the temporal dependencies through a 2D convolution network. From Table 1, we can see that 2D-TAN performs better at higher $IoU@0.7$, while slightly inferior to MAN at lower $IoU@0.5$.

Ablation Study

In this section, we evaluate the effects of different factors in our proposed 2D-TAN. The experiments are conducted on the ActivityNet Captions dataset, as shown in Table 4.

Number of Moment Candidates. The number of moment candidates is a vital factor in moment localization models. We first tune this factor in our 2D-TAN approach, and show its impacts on final performance. Then, we compare different approaches with respect to this factor.

Method		$Rank1@$			$Rank5@$		
		0.1	0.3	0.5	0.1	0.3	0.5
MCN		14.42	–	5.58	37.35	–	10.33
CTRL		24.32	18.32	13.30	48.73	36.69	25.42
MCF		25.84	18.64	12.53	52.96	37.13	24.73
TGN		41.87	21.77	18.9	53.40	39.06	*31.02*
ACRN		24.22	19.52	14.62	47.42	34.97	24.88
ROLE		20.37	15.38	9.94	45.45	31.17	20.13
VAL		25.74	19.76	14.74	51.87	38.55	26.52
ACL-K		31.64	24.17	*20.01*	57.85	*42.15*	30.66
CMIN		32.48	*24.64*	18.05	*62.13*	38.46	27.02
QSPN		25.31	20.15	15.23	53.21	36.72	25.30
SM-RL		26.51	20.25	15.95	50.01	38.47	27.84
SLTA		23.13	17.07	11.92	46.52	32.90	20.86
ABLR		*34.70*	19.50	9.40	–	–	–
SAP		31.15	–	18.24	53.51	–	28.11
TripNet		–	23.95	19.17	–	–	–
2D-TAN	Pool	**47.59**	**37.29**	**25.32**	70.31	**57.81**	**45.04**
	Conv	46.44	35.22	25.19	**74.43**	56.94	44.21

Table 3: Performance comparison on TACoS.

We vary the number of sampled clips N from 16 to 64 in our 2D-TAN approach. The results are shown in Table 4 (Row $4-6$). We observe that, increasing N from 16 to 64 brings improvements (57.79 $v.s$ 59.66 $v.s$ 60.18 in $Rank5@0.7$). This observation is also consistent with the theoretical upper bound, as listed in Table 4 (Row $1-3$). Here, the upper bound represents the performance of an ideal model that can provide a correct prediction on all the sampled video clips. The upper bound is smaller than 100% since that the sampling of video clips introduces errors.

Furthermore, we compare the number of moment candidates with the previous state-of-the-art method CMIN. Row 16 in Table 4 shows that CMIN use 1400 (N=200) moment candidates. However, our 2D-TAN only uses 136 (N=16) candidates, and achieves comparable results to CMIN (Row 4 *v.s* 16). Moreover, with larger number of moment candidates (N=64) and stacked convolution layers for moment representations, the performance of our method can be further boosted, as listed in Row 14. Noted that the number of moment candidates in Row 14 is 1200, which is still smaller than the ones used in CMIN. This comparison validates that our 2D-TAN gains improvement from the context modeling, rather than the dense sampling of moment candidates.

Receptive Field Size. We vary the depth and kernel size of convolution layers in our 2D-TAN to study the impact of receiptive field size. The results in terms of different kernel sizes and layer depths are reported in Table 4 Row $7-9$. We observe that the performance increases significantly as the receptive field enlarges. However, it becomes saturated when it is large enough, as listed in Row 6. Moreover, if the receptive field size is fixed, changing the depth of layers and kernel sizes has limited impacts on final performance, as shown in Row 11-13. This verifies the importance of receiptive field size in our 2D-TAN model. Large receiptive field

Row#	Method		N	2D-TAN		$Rank1@$			$Rank5@$		
				Kernel	Layer	0.3	0.5	0.7	0.3	0.5	0.7
1	Upper Bound		16	–	–	97.16	93.58	89.14	97.16	93.58	89.14
2	Upper Bound		32	–	–	99.10	96.88	94.38	99.10	96.88	94.38
3	Upper Bound		64	–	–	99.84	98.94	97.34	99.84	98.94	97.34
4	**2D-TAN**	Enum	16	9	4	58.82	42.45	23.93	85.07	75.99	57.79
5		Enum	32	9	4	58.26	43.18	25.47	84.82	75.45	59.66
6		Enum	64	9	4	58.15	42.80	25.76	84.53	75.39	60.18
7		Enum	64	1	1	45.90	26.20	14.27	70.72	56.14	37.13
8		Enum	64	5	1	54.78	35.27	18.81	81.80	69.76	50.68
9		Enum	64	5	4	58.20	40.45	23.25	83.76	73.97	57.46
10		Enum	64	9	4	58.15	42.80	25.76	84.53	75.39	60.18
11		Pool	64	9	4	59.45	44.51	26.54	85.53	77.13	61.96
12		Pool	64	5	8	57.86	41.68	25.13	85.26	75.74	58.90
13		Pool	64	17	2	58.19	43.09	26.09	84.22	75.16	60.02
14		Conv	64	9	4	58.75	44.05	27.38	85.65	76.65	62.26
15	CTRL		–	–	–	47.43	29.01	10.34	75.32	59.17	37.54
16	CMIN		200	–	–	63.61	43.40	23.88	80.54	67.95	50.73

Table 4: Ablation Study. N is the number of sampled clips. Row $1-3$ show the upper bound of an ideal model under different N. Row $4-6$ demonstrate how our model perform under different N. Row $6-13$ compare the performance under different kernel and layer settings. Row 14 show the performance using moment features extracted by stacked convolution. Row $15-16$ are two previous methods for comparison.

is able to model temporal dependencies, resulting in performance improvements. If we set the kernel size to 1 (Row 7), the 2D-TAN model is equivalent to treat each moment independently. In this case, it achieves similar performance with CTRL method (Row 15), which also treats each moment individually. This phenomenon further proves our hypothesis that modeling the moment candidates as a whole enables the network to distinguish similar moments. *Sparse Sampling* v.s. *Enumeration.* We further compare the effectiveness of our sparse sampling strategy with the dense enumeration for moment candidate selection. The results are reported in Table 4 (Row 10-11). It is observed that these two strategies achieve similar performance. The underlying reason is that the designed sparse sampling removes nearly 50% redundant moment candidates. Thus, it reduces the computation cost without performance decrease.

Stacked Convolution v.s. *Max-Pooling.* Stacked convolution and pooling have been applied for extracting moment features in previous works (**?**; **?**). We compare their performance on three datasets, as shown in Table 1-3 (2D-TAN: Pool *v.s.* Conv). It is observed that stacked convolution (Conv) performs better than max-pooling (Pool) on ActivityNet Captions, while comparable on Charades-STA and TACoS. We recommend to adopt the max-pooling operation, since it is fast in calculation, while does not contain any parameters.

Conclusion

In this paper, we study the problem of moment localization with natural language, and propose a novel 2D Temporal Adjacent Networks(2D-TAN) method. The core idea is to retrieve a moment on a two-dimensional temporal map, which considers adjacent moment candidates as the temporal context. 2D-TAN is capable of encoding adjacent temporal relation, while learning discriminative feature for matching video moments with referring expressions. Our model is simple in design and achieves competitive performance in comparison with the state-of-the-art methods on three benchmark datasets. In the future, we would like to extend our model to other temporal localization tasks, such as temporal action localization, video re-localization, etc.

Acknowledgement

We thank the support of NSF awards IIS-1704337, IIS-1722847, IIS-1813709, and the generous gift from our corporate sponsors. Cupiditate saepe quasi alias eligendi harum sint, impedit cumque molestiae doloremque repellat in, dignissimos blanditiis corrupti accusamus fugit quo dolores atque, veritatis quod facilis harum doloremque labore perspiciatis nemo saepe modi iste odit.Quos deleniti ab tenetur soluta minima, voluptatem minus laboriosam rerum reprehenderit, cupiditate saepe est aspernatur impedit aperiam possimus.Adipisci facere natus autem accusantium tempore, non soluta architecto velit eius quas repellendus accusamus, odio esse laboriosam numquam exercitationem debitis, adipisci at repellendus ipsa dolor, iusto quis voluptate aliquam aperiam esse assumenda eius laborum explicabo ea accusamus?Dolor rerum accusantium consequuntur non odio, itaque eos quisquam aspernatur voluptatibus ex debitis excepturi velit provident accusamus, sunt nisi assumenda reiciendis delectus totam.Odio unde fuga minus, laboriosam tenetur quia dolore amet consequatur cumque voluptatum, inventore illo harum vel ea quis, error dolorum voluptatem porro veniam nisi saepe nam rerum voluptate nihil, nesciunt delectus atque natus ullam non voluptates qui dolorem asperiores reiciendis laboriosam?Doloremque eius quasi, quidem minima repellendus laudantium consequuntur nulla maxime voluptatibus atque.Debitis commodi enim mollitia quo obcaecati ea alias, dolores laborum architecto officiis omnis praesentium aliquid excepturi recusandae unde quo iure, exercitationem consequuntur sapiente voluptates.Dignissimos sint odio