TubeDETR: Spatio-Temporal Video Grounding with Transformers Supplementary Material

Antoine Yang[1,2], Antoine Miech[3], Josef Sivic[4], Ivan Laptev[1,2], Cordelia Schmid[1,2]

[1]Inria Paris [2]Département d'informatique de l'ENS, CNRS, PSL Research University [3]DeepMind [4]CIIRC CTU Prague

https://antoyang.github.io/tubedetr.html

In this Supplementary Material, we present additional visualizations of the different attention mechanisms in our space-time decoder in Section 1. Section 2 provides additional implementation details. We then give detailed results for ablations in the main paper on the VidSTG dataset [4] split by sentence type in Section 3. Next we present an ablation of our fast and aggregation modules in Section 4. Finally we discuss broader impact in Section 5. Code and trained models are publicly available at [1].

1. Visualization of space, time and language attention patterns in the decoder

This section illustrates attention mechanisms of our space-time decoder over space, language and time for the spatio-temporal video grounding example presented in Figure 2. For this example the time-aligned cross-attention for the visual modality is also shown in Figure 2. We note that spatially, attention at each timestep is particularly focused on humans that are receiving the sports ball and gesturing. Additionally, the time-aligned cross-attention for the textual modality is illustrated in Figure 1. We observe that the words *adult* and *grabs* are the most attended overall, and that attention weights on the different words (*e.g.* *sports* and *ball*) vary over time. $\hat{t}_s$ and $\hat{t}_e$ in Figure 1 denote the predicted start and end times of the output tube. Next, the temporal self-attention is illustrated in Figure 3. We notice long-range temporal interactions: a certain number of time queries attend to various temporally distant time queries, *e.g.* time queries located around the start of the video between the eighth and sixteenth frames.

2. Additional implementation details

In our transformer, the number of heads is 8 and the hidden dimension of the feed-forward layers is 2048. We set the initial learning rates to $1e^{-5}$ for the visual backbone, and $5e^{-5}$ for the rest of the network. The learning rate fol-

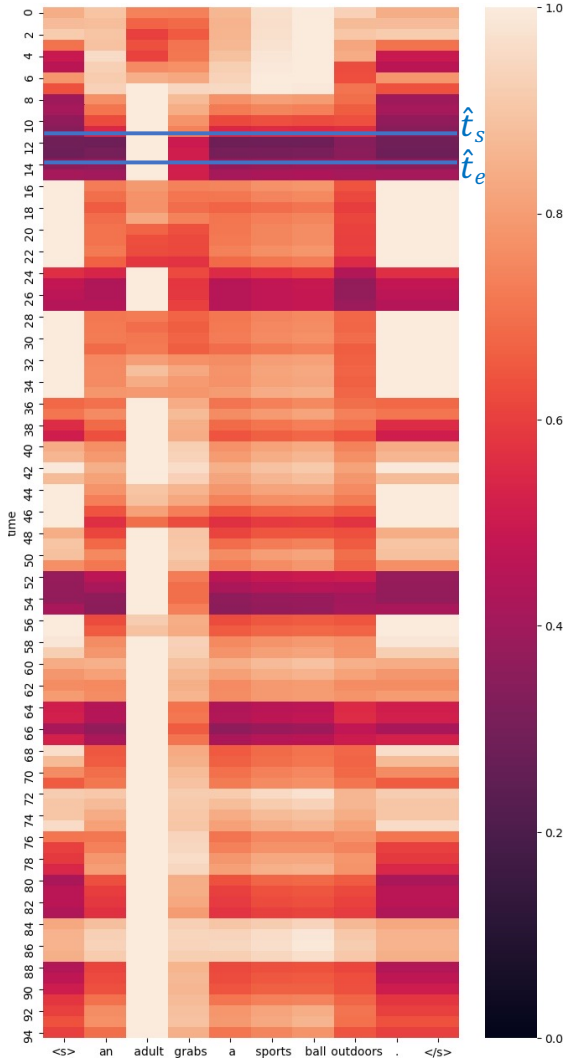

Figure 1. **Time-aligned cross-attention visualization (textual modality).** Visualization of the attention weights between the time query (y-axis) and its time-aligned visually-contextualized text features (x-axis) at different times in our space-time decoder. These attention weights are averaged across all 8 heads and all 6 layers, and renormalized by the maximum weight at each timestep (*i.e.* each row) for the purpose of visualization. Lighter colors correspond to higher attention weights (see the colorbar on the right).

[4]Czech Institute of Informatics, Robotics and Cybernetics at the Czech Technical University in Prague.

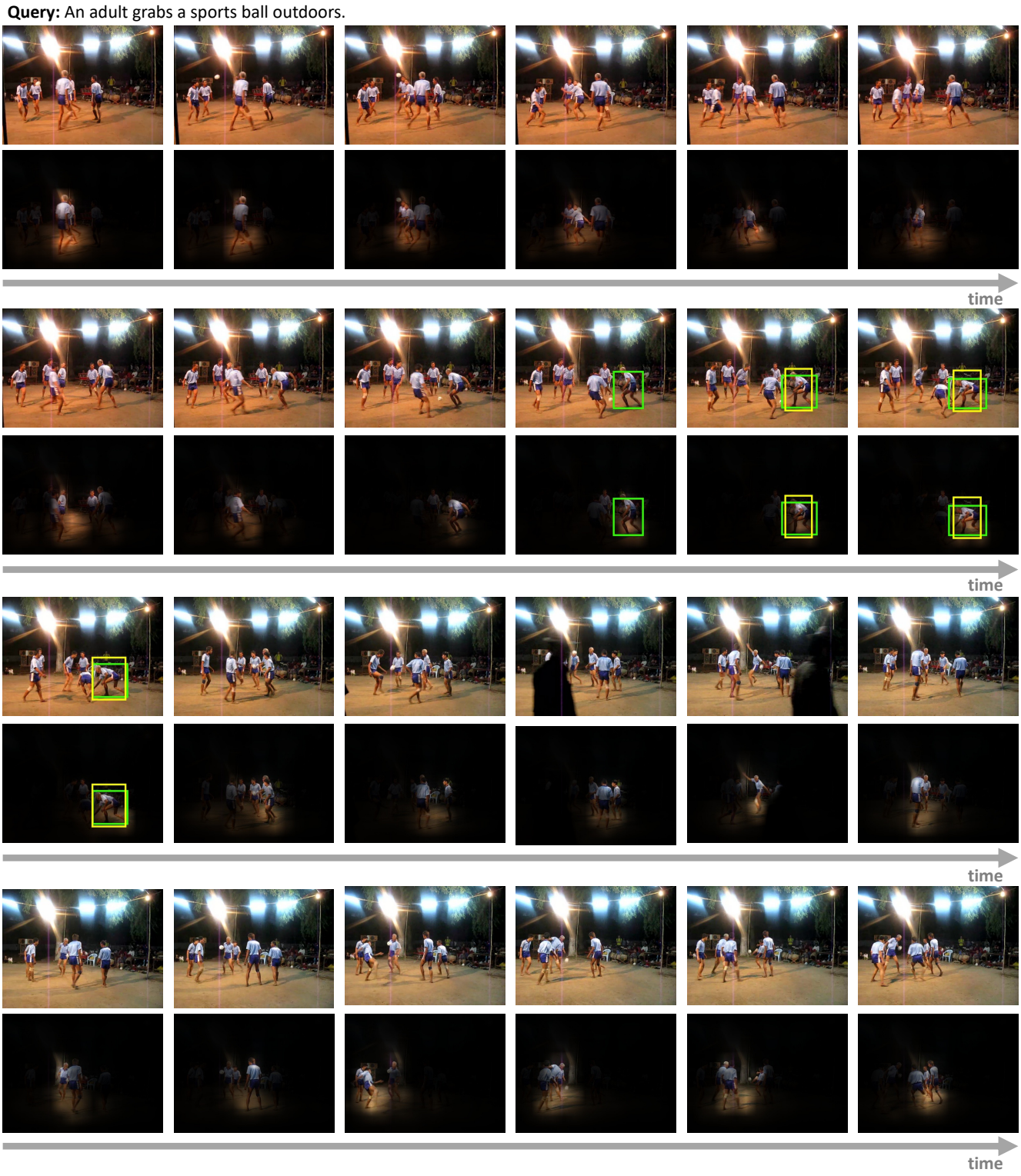

Figure 2. **Time-aligned cross-attention visualization (visual modality).** Top rows: Input frames with the predicted (yellow) and ground truth (green) spatio-temporal tubes overlaid. Bottom rows: Visualization of the attention weights between the time query and its time-aligned text-contextualized visual features at different times in our space-time decoder. These attention weights are averaged across all 8 heads and all 6 layers, and renormalized by the maximum weight at each timestep for the purpose of visualization. Attention at each timestep is particularly focused on humans that are receiving the sports ball and gesturing.

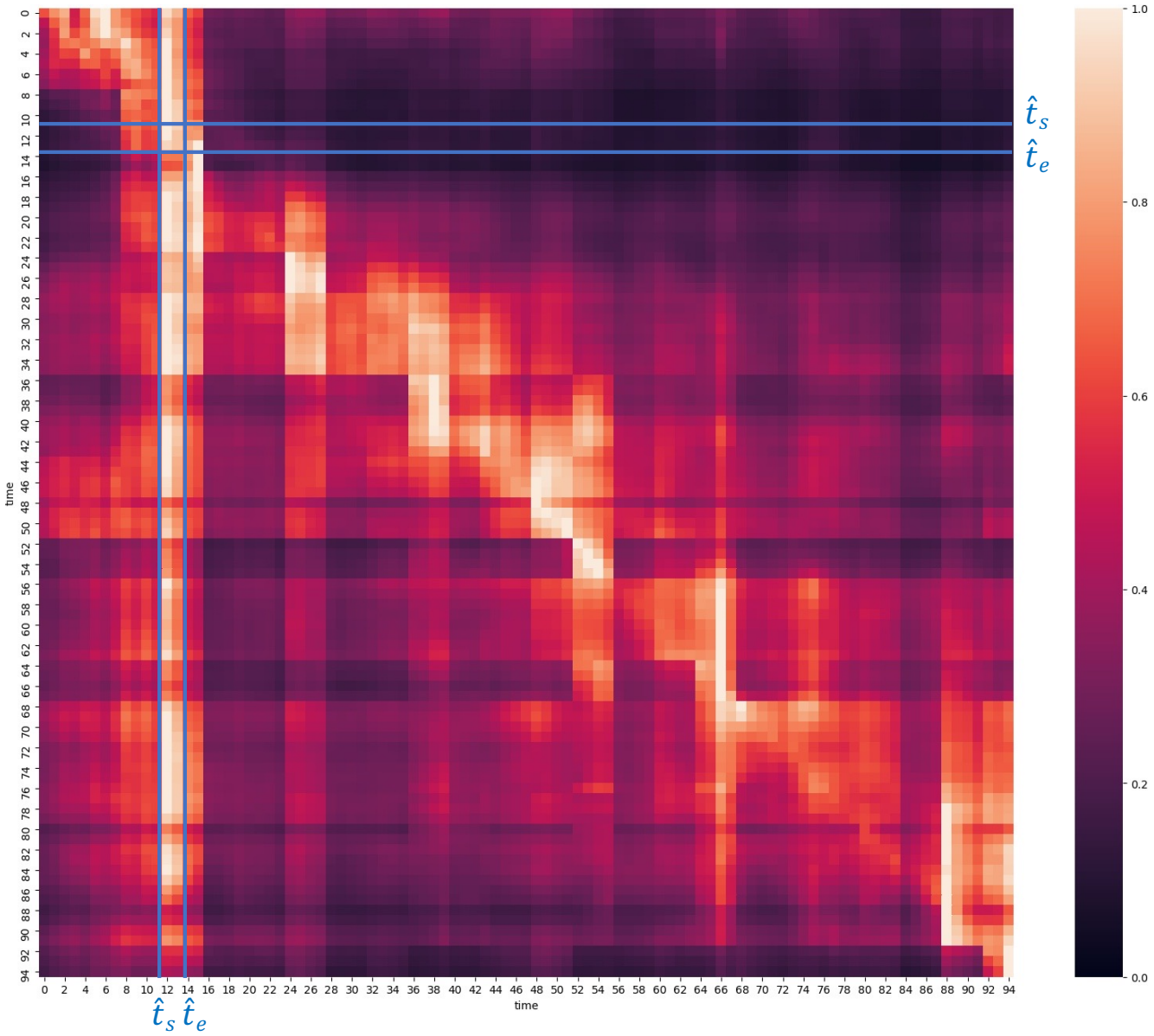

Figure 3. **Temporal self-attention visualization.** Visualization of the attention weights between the different time queries in our space-time decoder. The column t corresponds to the weights of the different time queries for the time query at time t. These attention weights are averaged across all 8 heads and all 6 layers, and renormalized by the maximum weight at each timestep (*i.e.* each column) for the purpose of visualization. $\hat{t}_s$ and $\hat{t}_e$ denote the predicted start and end times of the output tube. Lighter colors correspond to higher attention weights (see the colorbar on the right).

lows a linear schedule with warm-up for the text encoder and the learning rate is constant for the rest of the network. We use the AdamW optimizer [2] and weight-decay $1e^{-4}$. Video data augmentation includes spatial random resizing, spatial random cropping preserving box annotations, and temporal random cropping preserving the annotated time interval. Dropout [3] with probability 0.1 is applied in our transformer layers, and dropout with probability 0.5 is applied in the temporal localization head. We use exponential moving average with a decay rate of 0.9998, and an effective batch size of 16 videos. For temporal stride $k = 1$ the fast and aggregation modules in the encoder are not active, as their goal is to recover local spatial and temporal information when $k > 1$.

3. Detailed ablation results

In this section, we provide detailed results split by sentence type (declarative, interrogative) on the VidSTG dataset for the ablation studies presented in the main paper.

Space-time decoder. We first provide detailed results for

	Time Encoding	Self Attention	Declarative Sentences					Interrogative Sentences				
			m_tIoU	m_vIoU	vIoU @0.3	vIoU @0.5	m_sIoU	m_tIoU	m_vIoU	vIoU @0.3	vIoU @0.5	m_sIoU
1.	✗	-	24.4	20.4	29.9	16.6	51.9	23.5	16.9	23.4	12.8	43.1
2.	✗	Temporal	25.3	21.4	32.2	18.0	52.2	25.0	18.6	26.6	14.9	43.3
3.	✓	-	42.1	30.0	42.1	27.9	51.3	41.5	25.6	35.6	23.0	42.5
4.	✓	Temporal	**46.4**	**33.2**	**46.6**	**33.4**	**52.8**	**45.6**	**27.9**	**38.9**	**27.0**	**43.6**

Table 1. Effect of the time encoding and the temporal self-attention in our space-time decoder on the VidSTG validation set.

	Pre-Training	Decoder Self-Attention Transfer	Declarative Sentences					Interrogative Sentences				
			m_tIoU	m_vIoU	vIoU @0.3	vIoU @0.5	m_sIoU	m_tIoU	m_vIoU	vIoU @0.3	vIoU @0.5	m_sIoU
1.	✗	✗	42.9	24.9	35.5	22.7	41.1	42.8	22.4	31.3	19.5	36.5
2.	✓	✗	44.2	31.3	43.9	30.4	51.5	43.5	26.5	36.6	24.9	42.7
3.	✓	✓	**46.4**	**33.2**	**46.6**	**33.4**	**52.8**	**45.6**	**27.9**	**38.9**	**27.0**	**43.6**

Table 2. Effect of the weight initialization for our model on the VidSTG validation set.

	Fast	Res.	Temp. Stride	Declarative Sentences					Interrogative Sentences					Mem.
				m_tIoU	m_vIoU	vIoU@0.3	vIoU@0.5	m_sIoU	m_tIoU	m_vIoU	vIoU@0.3	vIoU@0.5	m_sIoU	(GB)
1.	—	224	1	46.9	34.6	49.1	34.8	54.2	46.1	28.8	40.3	27.8	44.9	23.9
2.	✓	224	2	46.6	34.3	48.9	35.2	54.3	45.5	28.5	40.1	27.9	44.7	16.2
3.	✓	224	5	46.4	33.2	46.6	33.4	52.8	45.6	27.9	38.9	27.0	43.6	11.8
4.	✓	288	2	47.0	**35.4**	49.7	35.7	55.7	46.0	**29.9**	**42.1**	**29.5**	46.3	23.7
5.	✓	320	3	46.9	35.2	50.0	**36.8**	56.0	45.9	29.7	41.7	**29.5**	**46.4**	23.6
6.	✓	352	4	**47.2**	**35.4**	**50.1**	36.7	**56.4**	**46.6**	29.8	41.9	**29.5**	46.2	24.4
7.	✗	352	4	47.1	33.8	47.9	33.8	53.7	46.2	28.3	40.1	27.1	44.0	18.1
8.	✓	384	5	47.1	34.8	48.8	35.6	55.4	46.3	29.7	42.0	29.3	46.1	26.1

Table 3. Comparison of performance-memory trade-off with various temporal strides k, frame spatial resolutions (Res.), with or without the fast branch in our video-text encoder, on the VidSTG validation set.

the ablation on our space-time decoder. The analysis is similar for both declarative and interrogative sentences. In detail, Table 1 shows that there is a substantial improvement over the space-only decoder when using both time encoding and temporal self-attention (+16.7% on $vIoU$@0.3 for declarative sentences and +15.5% on $vIoU$@0.3 for interrogative sentences between rows 1 and 4). The gain comes mostly from the temporal localization (+22.0% on m_tIoU for declarative sentences and +22.1% on m_tIoU for interrogative sentences), while the spatial grounding moderately increases (+0.9% in m_sIoU for declarative sentences and +0.5% in m_sIoU for interrogative sentences). Furthermore, we can observe that the time encoding brings most of the gain (+12.2% on $vIoU$@0.3 for declarative sentences and +12.2% on $vIoU$@0.3 for interrogatives sentences between rows 1 and 3). Finally, the temporal self-attention results in an additional improvement (+4.5% on $vIoU$@0.3 for declarative sentences and +3.3% on $vIoU$@0.3 for interrogative sentences between rows 3 and 4) over using time encoding only.

Initialization. We now provide detailed results for the ablation on our weight initialization. The analysis is similar for both declarative and interrogative sentences. In detail, Table 2 shows that pretraining is highly beneficial for spatio-temporal video grounding (+11.1% on $vIoU$@0.3 for declarative sentences and +7.6% on $vIoU$@0.3 for interrogative sentences between rows 1 and 3). The gain mainly comes from the spatial grounding performance (+11.7% on m_sIoU for declarative sentences and +7.1% on m_sIoU for interrogative sentences). Additionally, we observe the benefit of using the spatial self-attention weights from the MDETR decoder to initialize the temporal self-attention in our decoder (+2.7% on $vIoU$@0.3 for declarative sentences and +2.3% on $vIoU$@0.3 for interrogative sentences between rows 2 and 3).

Impact of spatial resolution and temporal stride k. In this section, we provide detailed results on the VidSTG dataset for the ablation on the impact of the spatial frame resolution and the temporal stride k. The analysis is similar for both declarative and interrogative sentences. In detail, Table 3 shows that increasing the resolution is an important factor of performance for spatio-temporal video grounding (see rows 2 and 4). Our proposed video-text encoder enables us to train on higher resolutions at a given memory usage. This leads to a better performance-memory trade-off (rows 4, 5, 6, 8) compared to the baseline variant with temporal stride $k = 1$ (row 1). In particular, the best spatio-temporal video grounding results (m_vIoU and $vIoU@R$) are obtained with temporal stride $k = 4$ and resolution 352 (row 6).

	Slow	Spatial Pool.	f	g	Declarative Sentences					Interrogative Sentences				
					m_tIoU	m_vIoU	vIoU@0.3	vIoU@0.5	m_sIoU	m_tIoU	m_vIoU	vIoU@0.3	vIoU@0.5	m_sIoU
1.	✗	✗	Linear	Sum + Linear	42.7	24.0	34.0	21.7	39.6	42.5	21.6	30.0	18.8	35.1
2.	✓	-	0	0	46.2	31.6	45.0	30.9	49.7	45.1	26.0	36.2	24.6	40.5
3.	✓	✓	Linear	Sum + Linear	45.8	31.2	44.6	30.5	50.2	44.9	26.3	37.2	24.8	40.9
4.	✓	✗	Linear	Product + σ	46.2	32.9	**47.3**	32.2	52.0	45.4	27.7	38.9	25.9	43.0
5.	✓	✗	Transformer	Sum + Linear	**46.4**	33.0	46.8	33.0	**52.8**	45.3	27.6	**39.1**	26.3	43.3
6.	✓	✗	Linear	Sum + Linear	**46.4**	**33.2**	46.6	**33.4**	**52.8**	**45.6**	**27.9**	38.9	**27.0**	**43.6**

Table 4. Comparison of designs for the video-text encoder, with or without the slow branch, with or without spatial pooling in the fast branch, with variants of the fast module f and aggregation module g, on the VidSTG validation set.

Impact of the fast branch. Finally, we provide detailed results on the VidSTG dataset for the ablation on the importance of our fast branch where we compare, for the best variant, temporal stride $k = 4$ and resolution 352, our slow-fast video-text encoder to a slow-only variant. The analysis is similar for both declarative and interrogative sentences. By comparing rows 6 and 7 in Table 3, our fast branch significantly improves the spatio-temporal video grounding performance (+2.2% $vIoU$@0.3 for declarative sentences and +1.8% $vIoU$@0.3 for interrogative sentences) with low computational memory overhead. This shows that the fast branch recovers useful spatio-temporal details lost by the temporal sampling operation in the slow branch.

4. Additional Experiments

In this section, we provide additional ablation studies. As in the ablations presented in the main paper, unless stated otherwise, we use spatial frame resolution of 224 pixels and temporal stride $k = 5$.

Design of the fast and aggregation modules. Here we further ablate the fast and aggregation modules f and g used in our dual-branch encoder. We report results in Table 4. The comparison between our slow-fast design (row 6) and the slow-only variant (row 2) is discussed in the main paper. Likewise, we compare our slow-fast design to a fast-only variant (row 1). The fast-only variant does not use the slow multi-modal branch, in which case the video-text features are the fast visual-only features concatenated with the text features. As shown in Table 4, our slow-fast design outperforms the fast-only variant, showing the importance of the slow multi-modal branch. We further compare the design of our fast and aggregation modules f and g (row 6) to other alternatives: row 3, a variant with the same primitives f and g but with f operating on features pooled over the spatial dimension; row 4, a variant which uses the same fast module f but a gating aggregation module $g(h_v(v,t), f(v)) = \sigma(h_v(v,t) * f(v))$ where σ is the sigmoid function; row 5, a variant that uses the same aggregation module g but a fast temporal transformer module f, which models temporal interactions between spatially-detailed features. As shown in Table 4, our design outperforms row 3, showing that preserving spatial information for each frame is crucial for the effectiveness of the fast branch. Additionally, our design slightly improves over row 4, indicating that further forcing the network to use the slow branch is not helpful. Finally, our design slightly improves over row 5, suggesting that additional modeling of temporal interactions in our encoder is not necessarily helpful.

5. Broader Impact

This work is a contribution to spatio-temporal video grounding and its potential positive or negative impacts depend on the application. Such models may be used for video surveillance and hence lead to questionable use. On the other hand, we believe that such methods could improve explainability of vision and language models which may help to understand some of their biases. This work also ablates memory usage when learning such models and thus could help promote development of lighter models with a reduced impact on the environment.

References

[1] TubeDETR project webpage. https://antoyang.github.io/tubedetr.html. 1

[2] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 3

[3] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 2014. 3

[4] Zhu Zhang, Zhou Zhao, Yang Zhao, Qi Wang, Huasheng Liu, and Lianli Gao. Where does it exist: Spatio-temporal video grounding for multi-form sentences. In *CVPR*, 2020. 1