TubeDETR: Spatio-Temporal Video Grounding with Transformers

Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, Cordelia Schmid

TubeDETR: Spatio-Temporal Video Grounding with Transformers

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

Abstract

We consider the problem of localizing a spatio-temporal tube in a video corresponding to a given text query. This is a challenging task that requires the joint and efficient modeling of temporal, spatial and multi-modal interactions. To address this task, we propose TubeDETR, a transformer-based architecture inspired by the recent success of such models for text-conditioned object detection. Our model notably includes: (i) an efficient video and text encoder that models spatial multi-modal interactions over sparsely sampled frames and (ii) a space-time decoder that jointly performs spatio-temporal localization. We demonstrate the advantage of our proposed components through an extensive ablation study. We also evaluate our full approach on the spatio-temporal video grounding task and demonstrate improvements over the state of the art on the challenging VidSTG and HC-STVG benchmarks.

1. Introduction

Grounding natural language in visual content is a fundamental skill to build powerful and explainable vision and language models. In particular, understanding the association of language with spatial regions and temporal boundaries in videos is particularly important to analyze and improve multi-modal video models. This goes beyond associating a global visual representation with a textual representation [57, 62], as it requires to reason about detailed spatio-temporal visual representations and their association with natural language, as illustrated in Figure 1.

Spatio-temporal video grounding, recently introduced in [102], is an interesting and challenging task that lies at the intersection of visual grounding [33, 59, 74] and temporal localization [9, 25, 30]. Given an untrimmed video and a textual description of an object, spatio-temporal video grounding aims at localizing a spatio-temporal tube (*i.e.*, a sequence of bounding boxes) for the target object described by the input text. This task is particularly challenging as

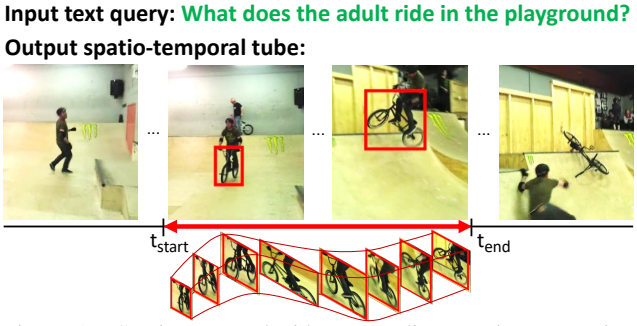

Figure 1. Spatio-temporal video grounding requires reasoning about space, time, and language.

videos are highly diverse and often present challenging scenarios where different entities have similar appearance or perform similar actions within one scene.

The success of attention-based models in natural language processing [21, 75] has recently inspired approaches to integrate transformers into computer vision tasks, such as image classification [22], object detection [8], semantic segmentation [52] or action recognition [3, 7, 60, 100]. Notably, with DETR [8], transformers have shown competitive performance on object detection while removing the need of multiple hand-designed components encoding a prior knowledge about this task. More recently, MDETR [37] has extended this framework for various text-conditioned object detection tasks in the image domain, such as phrase grounding, referring expression comprehension and segmentation.

Inspired by these works, and the fact that attention-based architectures are an intuitive choice for modelling multi-modal and spatio-temporal contextual relationships in videos, we develop a transformer encoder-decoder model for spatio-temporal video grounding, as illustrated in Figure 2. While existing approaches for this task rely on pre-extracted object proposals [102], tube proposals [72] or upsampling layers [68], our architecture simply reasons about abstractions called *time queries* to jointly perform temporal localization and visual grounding. Our framework enables to use the same representations for both subtasks in order to learn powerful contextualized representations.

More specifically, our architecture includes key components to jointly model temporal, spatial and multi-modal in-

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

teractions. Our video-text encoder efficiently encodes spatial and multi-modal interactions by computing these interactions over sparsely sampled frames, and separately recovers temporally local information with a lightweight fast branch. Our space-time decoder models temporal interactions with temporal self-attention layers, and spatial and multi-modal interactions with time-aligned cross-attention layers. Spatio-temporal video grounding is then tackled with multiple heads on top of the decoder outputs, which predict the object boxes and temporal start and end probabilities. We conduct various ablation studies, where we notably show the benefit of our video-text encoder in terms of performance-memory trade-off, and the efficiency of our space-time decoder in terms of spatio-temporal grounding results. Finally, we show that our method significantly improves over state-of-the-art methods on two benchmarks, VidSTG [102] and HC-STVG [72].

In summary, our contributions are three-fold: *(i)* We propose a novel architecture for spatio-temporal video grounding that performs this task with a space-time transformer decoder. *(ii)* We propose a dual-stream encoder that efficiently encodes spatial and multi-modal interactions, based on a slow multi-modal stream and a lightweight fast visual stream. *(iii)* We conduct comprehensive experiments on two benchmarks, VidSTG and HC-STVG, showing the effectiveness of our framework for the spatio-temporal video grounding task. Our approach, referred to as TubeDETR, outperforms all state-of-the-art methods by a large margin. Code and trained models are publicly available at [1].

2. Related Work

Spatio-temporal video grounding. Visual grounding consists in spatially localizing an object given a referring expression, and has been an active area of research both in the image domain [18, 32, 33, 51, 59, 77, 83, 91, 97, 107] and the video domain [35, 66, 74]. A standard paradigm consists in using pre-extracted object proposals [48, 49, 78, 84, 86, 87, 90], while some recent works [19, 34, 37, 46, 56, 88, 89] have proposed one-stage approaches which do not rely on such proposals. Our work follows the one-stage framework of MDETR [37], but extends it to spatio-temporal video grounding with temporal localization losses (see Equation 1), slow-fast encoding (see Figure 3), and space-time decoding (see Figure 4).

A separate line of work focuses on temporally localizing moments in a video given a natural language query [9, 10, 12, 25, 27, 30, 31, 47, 58, 64, 76, 80, 92, 95, 96, 98, 99, 101]. These works build architectures that reason about time but do not preserve spatial information. Spatio-temporal video grounding lies at the intersection of temporal localization and visual grounding. While some approaches [15, 72, 84] rely on pre-extracted tube proposals, or object proposals [102], our method does not require any pre-extracted proposals. A recent work [68] proposes STVGBert, a one-stage approach that extends the VilBERT model [54] pre-trained on Conceptual Captions [65] to this task. STVGBert uses deconvolutions to perform visual grounding, and symmetrically models temporal and spatial interactions. In contrast, our architecture performs visual grounding with a transformer decoder, and separately reasons about the temporal and spatial dimensions.

Temporal modeling for video understanding. The rise of powerful models for image understanding such as ViT [22] or DETR [8] has fostered research extending these models to the video domain [3, 7, 29, 41, 60, 100]. In particular, Lei *et al*. [41] propose an architecture that views moment retrieval as a direct set prediction problem, but is unsuitable to visual grounding as it does not preserve spatial information. He *et al*. [29] extend the DETR framework to videos, and propose an architecture built with sequentially added modules on top of Deformable DETR [106], while ours is built on inner modifications of a pretrained encoder and decoder and also reasons about language. Our dual-branch encoder is also related to SlowFast networks [23, 82] which combine fast and slow video streams. In contrast, in our case, both streams operate on features extracted from the same backbone, and our dual-stream architecture is motivated by the computational complexity related to multi-modal modeling.

Vision and language. Transformer-based architectures have become ubiquitous in various vision and language tasks [11, 14, 17, 20, 36, 38, 43, 45, 54, 55, 69, 71, 103]. Most video-text transformers rely either on pre-extracted object features [105], or spatially pooled features [24, 26, 44, 70, 85, 104], which do not preserve detailed spatial information. In contrast, our architecture is designed to preserve spatial information to perform visual grounding. Some recent works propose transformer-based architectures reasoning on videos and text that do preserve spatial information [2, 5, 42, 94]. However, these works typically aim to learn global video representations to tackle video-level prediction tasks, while we focus on learning detailed frame-level representations to address a dense prediction task requiring spatial and temporal localization.

3. Method

We first give an overview of our model in Section 3.1. Next, we describe in detail the two main components of our model, the video-text encoder (Section 3.2) and the space-time decoder (Section 3.3). Then in Section 3.4 we explain the loss used to train our model. Finally in Section 3.5 we present how we initialize our model weights.

3.1. Overview

Our objective is, given a video and a language query, to output a spatio-temporal tube, *i.e*. a sequence of bounding boxes with temporal boundaries, grounding the language

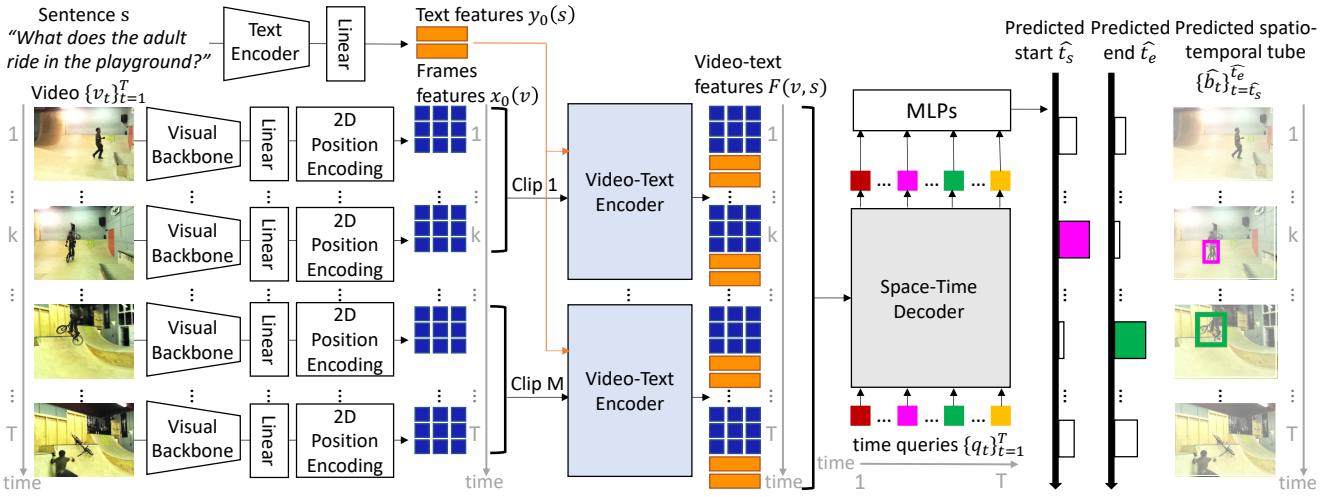

Figure 2. **TubeDETR model overview**. All input video frames v_t and the sentence s are first processed with a Visual Backbone and a Text Encoder. The resulting text and video features $y_0(s)$ and $x_0(v)$ are then jointly encoded with a Video-Text Encoder that computes spatial and multi-modal interactions for M short clips of k frames (about 1 second). The resulting video-text features $F(v,s)$ are then decoded into the output spatio-temporal tube $\hat{b}$ using a Space-Time Decoder that jointly reasons about time, space and text over the entire video.

query in the video. This is challenging as it requires modelling long-range *spatial* and *temporal* interactions between the language query and the video where the video may have hundreds of frames represented by tens of thousands spatio-temporal video features. Hence efficiency is a major challenge. To address this issue we design an encoder-decoder architecture, illustrated in Figure 2, that enables accurate yet efficient modelling of video-language spatial and temporal interactions across the entire video. In particular, our two-stream video-text encoder (Section 3.2) models video-language interactions only over short clips of about one second but allows for detailed spatial localization. Our space-time decoder (Section 3.3) then models long-range temporal interaction over the entire video to produce a temporally consistent output and accurate predictions of the start and end times of the output spatio-temporal tube.

3.2. Video-Text Encoder

Our encoder is illustrated in Figure 3 and described next. Its objective is to model spatial and multi-modal interactions between the language query and the video to accurately spatially ground the query in each frame. To achieve this, we leverage the ability of the self-attention layers to jointly model spatial and visual-linguistic interactions [36, 37, 42]. However, computing self-attention between visual features and textual features for every frame is computationally expensive. For this reason, we propose to compute spatial and multi-modal interactions only for every k-th frame. We denote the resulting stream as *slow multi-modal* branch. We use a separate lightweight *fast visual-only* branch that preserves the original frame rate and allows us to recover some of the high frequency spatio-temporal details lost by the sparse sampling in the slow branch.

Formally, our encoder takes as input a set of 2D flattened image features $x_0(v) \in \mathbb{R}^{T\times HW\times d}$ from the visual backbone for all T frames of the input video together with a set of L text features $y_0(s) \in \mathbb{R}^{L\times d}$ extracted by the text encoder from the query sentence, and outputs a set of video-text features $F(v,s) \in \mathbb{R}^{T\times(HW+L)\times d}$, one for each frame. Next we give the details of the Slow and Fast branches, and the final feature aggregation module.

Slow multi-modal branch. The goal of this branch (see top of Figure 3) is to model interactions between visual and textual representations. This branch first samples features from *one* frame for a short clip of k consecutive frames. A typical clip length is one second, *i.e.* $k = 5$ with a standard frame rate of 5 frames per second [102]. Formally, the resulting feature map is written as $x^p \in \mathbb{R}^{M\times HW\times d}$ where $M = \lceil \frac{T}{k} \rceil$ is the number of clips, k is the length of the clip and T is the length of the entire video. We then concatenate, for each clip m, its visual features x_m^p with text features $y_0(s)$ and forward it to a N-layer transformer encoder. The outputs are contextualized visual-text representations $h^p(v,s) \in \mathbb{R}^{M\times(HW+L)\times d}$, which effectively combine information from the input video v and the query sentence s.

Fast visual-only branch. The previously explained temporal sparse sampling scheme reduces significantly the memory requirements of the video-text encoder but results in a loss of spatio-temporal details which are important for spatio-temporal video grounding. To alleviate this issue, we introduce module f (see bottom of Figure 3) which operates on *2D flattened image features for all frames*. Formally, given feature map $x_0(v)$, this module outputs visual features $f(v) \in \mathbb{R}^{T\times HW\times d}$. This *fast* branch preserves the spatial and temporal resolution of the features but is computationally light as it does not compute any multi-modal

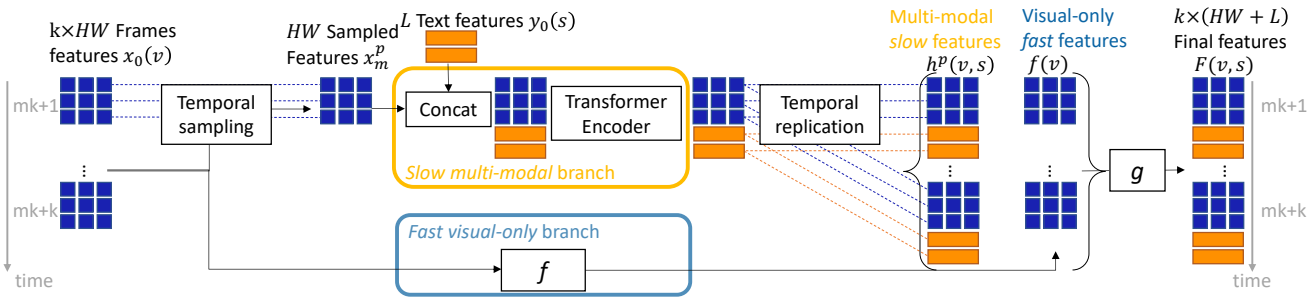

Figure 3. **Video-Text Encoder** takes as input a set of 2D flattened image features $x_0(v)$ together with a set of text features $y_0(s)$ from the query sentence, and outputs a set of video-text features $F(v,s)$, one for each frame. Top: the *Slow multi-modal* branch first samples video features x_m^p, one from every k frames. Then it computes multi-modal interactions between the sampled features x_m^p and text features y_0 using a transformer encoder. The temporal sampling reduces the number of video features in order to efficiently compute the attention-based interactions. Bottom: lightweight "Fast visual-only" branch f processes features from *all* frames but without any attention layers for increased efficiency. Features from both branches are then combined in module g into the final set of per-frame features $F(v,s)$.

or spatial interactions. For additional efficiency, at training time, this branch does not back-propagate gradients to the visual backbone. Furthermore, we show in Section 4.2 that it is able, when combined with the temporally sparse features obtained from the slow branch, to recover some of the temporal information lost during the temporal sampling.

Slow-Fast feature aggregation. We now describe the *slow* and *fast* branches aggregation module (see Figure 3, right), which fuses information from both branches and outputs final video-text features. To match the temporal dimension of the output from the *fast* branch $f(v)$, the output of the *slow* multi-modal branch $h^p(v,s)$ is temporally replicated k times for each clip resulting in video-text encodings $h(v,s) \in \mathbb{R}^{T\times(HW+L)\times d}$. These encodings are a concatenation of text-contextualized visual encodings $h_v(v,s) \in \mathbb{R}^{T\times HW\times d}$ and visually-contextualized textual encodings $h_s(v,s) \in \mathbb{R}^{T\times L\times d}$. The text-contextualized visual encodings $h_v(v,s)$ are combined with the outputs of the *fast* branch with an additional aggregation module g and a residual connection, resulting in aggregated visual encodings $F_v(v,s) = g(h_v(v,s), f(v)) + h_v(v,s)$. The final output of our video-text encoder is obtained by concatenating these aggregated visual encodings with the visually-contextualized textual encodings *i.e.* $F(v,s) = [F_v(v,s), h_s(v,s)] \in \mathbb{R}^{T\times(HW+L)\times d}$. In detail, the module g is implemented as a sum followed by a linear layer, *i.e.* $g(h_v(v,s), f(v)) = \texttt{Linear}(h_v(v,s) + f(v))$.

3.3. Space-Time Decoder

Our decoder is illustrated in Figure 4 and detailed next. Its objective is to model the temporal interactions within the entire video of T frames and decode the multi-modal features from the encoder into a temporally coherent output tube with accurate start and end times. This is achieved by an efficient decoder architecture that alternates (i) *temporal self-attention* layers, which model *temporal* interactions across the entire video, with (ii) *time-aligned cross attention* layers, which efficiently incorporate the video-text features for individual frames obtained from the encoder. In detail, the decoder operates on T positional encodings $\{q_t\}_{t=1}^T$, one per frame, referred to as time queries. The initial encoding of each time query is obtained by summing a learnt object encoding common to all frames, and a frozen sinusoidal time encoding. The decoder also takes as input $T \times (HW + L)$ video-language embeddings $F(v,s)$ output from the video-text encoder. The decoder is a succession of N decoder blocks. Each block is composed of temporal self-attention, time-aligned cross-attention, and feed-forward layers, interleaved with normalization [4], as shown in Figure 4. The decoder outputs refined time queries $\{Q_t\}_{t=1}^T$, which are contextualized across all frames in the video together with video-text features produced by the encoder. The refined time queries are then jointly used for outputting the spatio-temporal video tube that grounds the input sentence in the video. The individual layers are described in detail next.

Temporal self-attention. The T input time queries q_t attend to each other using the temporal self-attention layer. This layer is in each of the N blocks of the decoder and is responsible for modelling the long-range temporal interactions in the entire video. This is possible because of the relatively low complexity of this layer, which does not depend on the spatial resolution of the input video.

Time-aligned cross-attention. Allowing each time query to cross-attend to all $T \times (HW + L)$ video-text features can be highly computationally expensive due to the large number of video frames T and a large spatial resolution HW of the video features. Instead, in our cross-attention module, each time query q_t only cross-attends to its temporally corresponding multi-modal features $F(v,s)[t]$ at frame t. Note that with our time-aligned cross-attention formulation, the time encoding and the temporal self-attention layers are all the more important, as they are responsible for the temporal modelling across the entire video. Without them, our decoder would be decoding each frame independently. Their importance is ablated in Section 4.2.

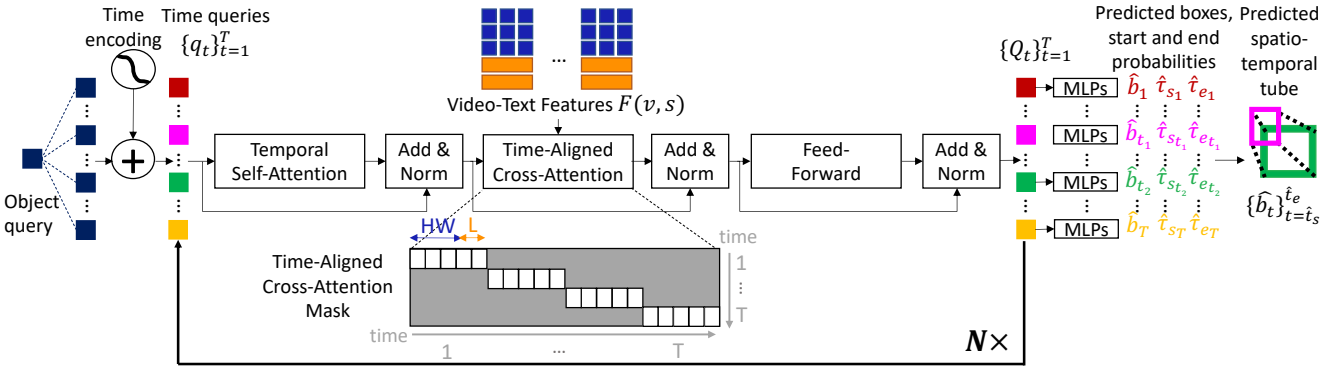

Figure 4. **Space-Time Decoder**. The decoder is composed of N repeated blocks. In each block, time queries q_t successively attend to each other via *temporal self-attention* and to their respective time-aligned video-text features $F(v,s)$ via *time-aligned cross-attention*. The cross-attention mask (bottom) indicates the non-zero weights (white) between the input $HW+L$ video-language features for each of the T input frames (x-axis) and T time queries (y-axis). The cross-attention mask ensures that each time query q_t only cross-attends to video-text features $F(v,s)$ at the corresponding frame t, which significantly increases efficiency of the decoder and enables decoding entire videos of T frames. The temporal modelling over the entire length of the video is ensured by the temporal self-attention layers.

Prediction heads. The output of the decoder is a set of refined time queries $\{Q_t\}_{t=1}^T$. They are jointly used for visual grounding and temporal localization to simultaneously obtain predictions for *all frames of the video*. In detail, normalized coordinates of all bounding boxes (2D center and size) $\hat{b} \in [0,1]^{T\times 4}$ are predicted with a 3-layer MLP. Probabilities of the start and the end of the output video tube, $\hat{\tau}_s \in [0,1]^T$ and $\hat{\tau}_e \in [0,1]^T$, respectively, are predicted with 2-layer MLPs. At inference time, the start and end times of the output tube, $\hat{t}_s$ and $\hat{t}_e$, are computed by choosing the maximum of the joint start and end probability distribution $(\hat{\tau}_s, \hat{\tau}_e) \in [0,1]^{T\times T}$ with invalid combinations where $\hat{t}_e \leq \hat{t}_s$ masked out. The predicted spatio-temporal tube $\{\hat{b}_t\}_{t=\hat{t}_s}^{\hat{t}_e}$ is composed from bounding boxes $\hat{b}_t$ predicted within the chosen start and end times $\hat{t}_s$ and $\hat{t}_e$.

3.4. Training loss

The input training data is in the form of a set of videos, where each video is annotated with a query sentence s and the corresponding video tube b composed of a set of bounding boxes and corresponding start and end times, t_s and t_e. Inspired by [64], we construct a target start (respectively end) distribution $\tau_s \in [0,1]^T$ (respectively τ_e) which follows a quantized Gaussian centered at $t_s \in [0, T-1]$ (respectively t_e) with standard deviation 1. We train our architecture with a linear combination of four losses

$$\begin{aligned}\mathcal{L} = \lambda_{\mathcal{L}_1}\mathcal{L}_{\mathcal{L}_1}(\hat{b}, b) + \lambda_{gIoU}\mathcal{L}_{gIoU}(\hat{b}, b) \\ +\lambda_{KL}\mathcal{L}_{KL}(\hat{\tau}_s, \hat{\tau}_e, \tau_s, \tau_e) + \lambda_{att}\mathcal{L}_{att}(A) \quad (1)\end{aligned}$$

where $b \in [0,1]^{4(t_e - t_s + 1)}$ denotes the normalized ground truth box coordinates and $\hat{b}$ the predicted bounding boxes and $A \in [0,1]^{T\times T}$ denotes the temporal self-attention matrix. Finally, different $\lambda_\bullet$ are scalar weights of the individual losses. $\mathcal{L}_{\mathcal{L}_1}$ is a $\mathcal{L}_1$ loss on bounding box coordinates. $\mathcal{L}_{gIoU}$ is a generalized "intersection over union" (IoU) loss [63] on the bounding boxes. Both $\mathcal{L}_1$ and $\mathcal{L}_{gIoU}$ are used for spatial grounding. $\mathcal{L}_{KL}(\hat{\tau}_s, \hat{\tau}_e, \tau_s, \tau_e)$ is the Kullback-Leibler divergence loss measuring the distance between the predicted and the target start distribution as well as the distance between the predicted and the target end distribution [64]. $\mathcal{L}_{att}(A)$ is a guided attention loss [64] that encourages weights corresponding to time queries outside of the temporal boundaries to be lower than the weights inside these boundaries. $\mathcal{L}_{KL}$ and $\mathcal{L}_{att}(A)$ are both used for temporal grounding. Losses are computed at each layer of the decoder following [8].

3.5. Weight initialization

We initialize our architecture with weights from MDETR [37] pretrained on Flickr30k [61], MS COCO [13] and Visual Genome [40]. In detail, weights of our video-text encoder are initialized from the MDETR multi-modal encoder, except for the fast and aggregation modules. We also use the weights from the MDETR single-image multi-object decoder to initialize our multi-frame single-object space-time decoder, except for the temporal localization head. We show the benefit of this initialization notably by comparing it to an ImageNet initialization, *i.e.* using a visual backbone pretrained on ImageNet with a randomly initialized transformer, in Section 4.2. We also evaluate a MDETR-equivalent baseline in Section 4.2.

4. Experiments

This section demonstrates the effectiveness of our architecture and compares our method to the state of the art. We first introduce the datasets, evaluation metrics and implementation details in Section 4.1. We then present ablation studies in Section 4.2. The comparison to the state of the art in spatio-temporal video grounding is given in Section 4.3. Finally, we show qualitative results in Section 4.4.

4.1. Experimental setup

Datasets. We evaluate our approach on the VidSTG [102] and HC-STVG [72] datasets. Both are annotated with spatio-temporal tubes corresponding to text queries. **VidSTG** consists of 99,943 sentence descriptions with 44,808 declarative sentences and 55,135 interrogative sentences describing 79 types of objects appearing in 10,303 different videos. The dataset is divided into training, validation and test subsets with 80,684, 8,956 and 10,303 distinct sentences respectively, and 5,436, 602 and 732 distinct videos respectively. **HC-STVG** consists of videos in multi-person scenes, each annotated with one sentence referring to a person. For ablation, we use the second improved version of the dataset **HC-STVG2.0** which is divided into training and validation subsets with 10,131 and 2,000 video-sentence pairs, respectively. The test set is not publicly available at the time of writing. To compare with prior work, we use the first version of the dataset **HC-STVG1** which is divided into training and test subsets with 4,500 and 1,160 video-sentence pairs, respectively.

Evaluation metrics. We follow [102] and define $vIoU$ as $vIoU = \frac{1}{|S_u|}\sum_{t\in S_i} IoU(\hat{b}_t, b_t)$ where S_u (respectively S_i) is the set of frames in the union (respectively intersection) between the ground truth (GT) and the predicted timestamps. $\hat{b}_t$ (respectively b_t) are the predicted (respectively GT) boxes at time t. To evaluate spatio-temporal video grounding, we use m_vIoU, which is the average of $vIoU$. We also use $vIoU@R$, the proportion of samples for which $vIoU > R$. To isolate the evaluation of temporal localization, we use m_tIoU which is the average of temporal IoU between the GT start and end and the predicted start and end. Likewise, to evaluate spatial grounding only, we use m_sIoU, which is computed by using the GT start and end times. For ablations we report results averaged over all samples. More detailed ablation results presented separately for declarative and interrogative sentences in VidSTG are reported in Appendix Section C. We also report peak GPU memory usage during training (Mem.) to measure the memory footprint of alternative models.

Implementation details. The visual backbone is ResNet-101 [28], the text encoder is RoBERTa [50] and the fast module f is a linear layer. Following [102], we sample 5 frames per second for videos, and for videos with more than 200 sampled frames we uniformly sample 200 frames. We use hyper-parameters $T = 200$, $N = 6$, $d = 256$, $\lambda_{\mathcal{L}_1} = 5$, $\lambda_{giou} = 2$, $\lambda_{KL} = 10$ and $\lambda_{att} = 1$. We train our networks for 10, 20 and 40 epochs on VidSTG, HC-STVG2.0 and HC-STVG1, respectively. The final model is selected based on the best spatio-temporal video grounding performance on the validation set. For the largest dataset VidSTG, the optimization takes 2 days on 16 Tesla V100 GPUs. Further details are included in Appendix Section B.

	Time Encoding	Self Attention	m_tIoU	m_vIoU	vIoU @0.3	vIoU @0.5	m_sIoU
1.	✗	-	23.9	18.5	26.3	14.5	47.0
2.	✗	Temporal	25.2	19.8	29.1	16.3	47.3
3.	✓	-	41.7	27.5	38.5	25.2	46.5
4.	✓	Temporal	**45.9**	**30.3**	**42.3**	**29.8**	**47.7**

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	m_tIoU	m_vIoU	vIoU @0.3	vIoU @0.5	m_sIoU
1.	✗	✗	42.8	23.5	33.2	20.9	38.5
2.	✓	✗	43.8	28.6	39.8	27.3	46.6
3.	✓	Temporal	**45.9**	**30.3**	**42.3**	**29.8**	**47.7**

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

4.2. Ablation studies

In this section, we ablate the hyper-parameters of our model and evaluate alternative design choices of the encoder and decoder. Unless stated otherwise, we use spatial frame resolution of 224 pixels and temporal stride $k = 5$.

Space-time decoder. We first ablate the design choices of the proposed space-time decoder. We compare our full decoder model with variants without time encoding, without temporal self-attention and without both. The variant without both corresponds to a space-only decoder, similar to MDETR [37] applied independently to every frame. Table 1 shows that there is a substantial improvement over the space-only decoder when using both time encoding and temporal self-attention (+16.0% on $vIoU@0.3$ between rows 1 and 4). The gain comes mostly from the temporal localization (+22.0% on m_tIoU), while the spatial grounding moderately increases (+0.7% in m_sIoU). Furthermore, we can observe that the time encoding brings most of the gain (+12.2% on $vIoU@0.3$ between rows 1 and 3). Finally, the temporal self-attention results in an additional improvement (+3.8% on $vIoU@0.3$ between rows 3 and 4) over using time encoding only.

Initialization. We now ablate the importance of initializing our model with pretrained MDETR [37] weights. In Table 2, we compare this initialization to ImageNet initialization, and a variant that does not transfer the spatial self-attention weights from MDETR decoder to the temporal self-attention in our space-time decoder. At pretraining time, this self-attention was used to model spatial relationships between different objects in the same image, while the temporal self-attention in our decoder models temporal relationships between the same object in different frames of a video. We find that pretraining is highly beneficial (+9.1% on $vIoU@0.3$ between rows 1 and 3), especially for the spatial grounding performance (+9.2% on m_sIoU). 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 (+2.5% on $vIoU@0.3$ between rows 2 and 3).

(a) VidSTG

	Fast	Res.	Temp. Stride	m_tIoU	m_vIoU	vIoU@0.3	vIoU@0.5	m_sIoU	Mem. (GB)
1.	—	224	1	46.5	31.4	44.2	30.9	49.1	23.9
2.	✓	224	2	46.0	31.1	44.0	31.1	49.0	16.2
3.	✓	224	5	45.9	30.3	42.3	29.8	47.7	11.8
4.	✓	288	2	46.4	**32.4**	**45.5**	32.3	50.5	23.7
5.	✓	320	3	46.4	32.1	45.4	**32.8**	**50.7**	23.6
6.	✓	352	4	**46.9**	32.3	**45.5**	32.7	**50.7**	24.4
7.	✗	352	4	46.6	30.7	43.6	30.1	48.3	18.1
8.	✓	384	5	46.7	32.0	45.0	32.1	50.2	26.1

(b) HC-STVG2.0

	Fast	Res.	Temp. Stride	m_tIoU	m_vIoU	vIoU@0.3	vIoU@0.5	m_sIoU	Mem. (GB)
1.	—	224	1	52.8	45.0	68.0	46.9	63.9	14.3
2.	✓	224	2	53.7	46.3	70.5	49.5	64.3	10.2
3.	✓	224	5	53.2	45.2	69.0	48.0	63.2	8.0
4.	✓	288	2	**53.9**	46.6	71.3	49.6	**65.4**	13.9
5.	✓	320	3	53.6	46.5	70.5	48.8	65.2	13.8
6.	✓	352	4	**53.9**	**46.7**	71.3	**49.7**	64.9	14.3
7.	✗	352	4	53.1	45.0	69.4	47.3	63.0	11.3
8.	✓	384	5	53.6	46.6	**71.6**	48.9	65.3	15.2

Table 3. Comparison of performance-memory trade-off with various temporal strides k, spatial resolutions (Res.), with or without the fast branch in our video-text encoder, on the VidSTG validation set (left, Table 3a) and the HC-STVG2.0 validation set (right, Table 3b).

Method	Pretraining Data	VidSTG								HC-STVG1		
		Declarative Sentences				Interrogative Sentences						
		m_tIoU	m_vIoU	vIoU@0.3	vIoU@0.5	m_tIoU	m_vIoU	vIoU@0.3	vIoU@0.5	m_vIoU	vIoU@0.3	vIoU@0.5
1. STGRN [99]	Visual Genome	**48.5**	19.8	25.8	14.6	**47.0**	18.3	21.1	12.8	—	—	—
2. STGVT [69]	Visual Genome + Conceptual Captions	—	21.6	29.8	18.9	—	—	—	—	18.2	26.8	9.5
3. STVGBert [65]	ImageNet + Visual Genome + Conceptual Captions	—	24.0	30.9	18.4	—	22.5	26.0	16.0	20.4	29.4	11.3
4. TubeDETR (Ours)	ImageNet	43.1	28.0	39.9	26.6	42.3	25.1	35.7	22.4	23.7	36.0	17.2
5. TubeDETR (Ours)	ImageNet + Visual Genome + Flickr + COCO	48.1	**37.8**	**53.5**	**39.9**	46.9	**31.8**	**44.9**	**31.8**	**43.7**	**65.0**	**46.1**

Table 4. Comparison to the state of the art on the VidSTG test set and the HC-STVG1 test set.

Impact of spatial resolution and temporal stride k. In this section, we analyze the impact of the frame resolution and the temporal stride k. In Table 3, we show that increasing the resolution is an important factor of performance for spatio-temporal video grounding, on both the VidSTG and HC-STVG2.0 datasets (see rows 2 and 4). However, it also results in significantly higher memory usage (16.2GB vs 23.7GB). As a consequence, the variant using temporal stride $k = 1$ is challenging to train on VidSTG with a resolution higher than 224 on a Tesla V100 32GB GPU. At a fixed 224 resolution, increasing the temporal stride k to 2 or 5 reduces the peak memory usage by 7.7GB or 12.1GB, respectively (see row 1 vs 2 or 3, respectively). 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) than the baseline variant with temporal stride $k = 1$ (row 1). In particular, the best spatio-temporal video grounding results (m_vIoU and $vIoU@R$) over the two datasets are obtained with temporal stride $k = 4$ and resolution 352 (row 6).

We note that as the resolution increases, performance gains obtained by its further increase are expected to be lower as they are limited by the original video resolution. For instance, the average video pixel height in VidSTG and HCSTVG2.0 is 440 and 490 pixels, respectively.

Impact of the fast branch. Finally, we validate the importance of our fast branch by comparing, for the best variant, temporal stride $k = 4$ and resolution 352, our slow-fast video-text encoder to a slow-only variant that corresponds to $f = 0$ and $g = 0$. In this case the video-text features are the slow video-text features. By comparing rows 6 and 7 in Table 3, our fast branch significantly improves the spatio-temporal video grounding performance (+1.9% $vIoU@0.3$ on VidSTG and +1.9% $vIoU@0.3$ on HC-STVG2.0) 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. We further ablate the design of the fast and aggregation modules f and g in Appendix Section D.

4.3. Comparison to the state of the art

In this section, we compare our approach to state-of-the-art methods in spatio-temporal video grounding. We report results for the model achieving the best validation results in the previous ablation studies, *i.e.*, our space-time decoder with time encoding and temporal self-attention, temporal stride $k = 4$ and resolution 352. The focus of our work is on the spatio-temporal video grounding metrics (m_vIoU and $vIoU@R$). As shown in Table 4, *only using ImageNet to initialize the visual backbone* (row 4), our TubeDETR significantly improves over the state-of-the-art approaches, including those using large-scale image-text pretraining (rows 2 and 3), on both VidSTG and HC-STVG1. Furthermore, if we use MDETR initialization (row 5), our TubeDETR outperforms by a large margin all previous methods (rows 1, 2 and 3) on both datasets. STGRN [102] achieves similar m_tIoU (measuring only temporal localization), but it defines a handcrafted set of possible window widths to tackle temporal localization, while we consider all possible windows, *i.e.* any starting frame i and ending frame j with $i < j$. These results demonstrate the excellent performance of our architecture for spatio-temporal video grounding.

4.4. Qualitative examples

We show qualitative examples of our predictions on the VidSTG test set in Figure 5. These examples show that our

Figure 5. Qualitative examples of spatio-temporal tubes predicted by our model (light yellow), compared with ground truth (light green), on the VidSTG test set. The first three examples illustrate successful predictions of our method. In the last example the method confuses the small sports ball in the background with a balloon.

model is able to predict meaningful and accurate spatio-temporal tubes associated with the input text queries. In particular, in the first example, our model correctly detects the temporal moment corresponding to the cat biting the adult. In the second example, our model localizes the spatio-temporal tube corresponding to a man quickly grabbing a very small sports ball and in the third example it is able to localize the skis under the adult while skiing. However, as shown in the last example, it may fail to understand fine details in the query and the video. Note that the balloon and the ball are visually and semantically similar. A careful analysis is required to understand the difference. Furthermore, we provide visualizations of the different attention mechanisms of TubeDETR in Appendix Section A.

5. Conclusion

We have proposed TubeDETR, a novel transformer-based architecture for spatio-temporal video grounding. TubeDETR tackles this task with a space-time transformer decoder combined with a video-text encoder that efficiently encodes spatial and multi-modal interactions. We have demonstrated the effectiveness of our space-time decoder, and the benefits of our video-text encoder in terms of performance-memory trade-off. Finally, our approach outperforms state-of-the-art methods on two benchmarks, VidSTG and HC-STVG. Future work could extend our space-time decoder to detect multiple objects per frame or multiple events per video. Investigating more efficient alternatives to self-attention, such as the ones studied for natural language [6, 16, 39, 73, 79, 81, 93], is another promising direction for future research.

Acknowledgements. This work was granted access to the HPC resources of IDRIS under the allocation 2021-AD011011670R1 made by GENCI. The work was funded by a Google gift, the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute), the Louis Vuitton ENS Chair on Artificial Intelligence, the European Regional Development Fund under project IMPACT (reg. no. CZ.02.1.01/0.0/0.0/15 003/0000468). We thank S. Chen and J. Chen for helpful discussions and O. Bounou and P.-L. Guhur for proofreading.

References

[1] TubeDETR project webpage. https://antoyang.github.io/tubedetr.html. 2

[2] Hassan Akbari, Linagzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. VATT: Transformers for multimodal self-supervised learning from raw video, audio and text. *arXiv preprint arXiv:2104.11178*, 2021. 2

[3] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A video vision transformer. In *ICCV*, 2021. 1, 2

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[5] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021. 2

[6] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. 8

[7] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021. 1, 2

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1, 2, 5

[9] Jingyuan Chen, Xinpeng Chen, Lin Ma, Zequn Jie, and Tat-Seng Chua. Temporally grounding natural sentence in video. In *EMNLP*, 2018. 1, 2

[10] Jingyuan Chen, Lin Ma, Xinpeng Chen, Zequn Jie, and Jiebo Luo. Localizing natural language in videos. In *AAAI*, 2019. 2

[11] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. In *NeurIPS*, 2021. 2

[12] Shaoxiang Chen and Yu-Gang Jiang. Semantic proposal for activity localization in videos via sentence query. In *AAAI*, 2019. 2

[13] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 5

[14] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: Universal image-text representation learning. In *ECCV*, 2020. 2

[15] Zhenfang Chen, Lin Ma, Wenhan Luo, and Kwan-Yee K Wong. Weakly-supervised spatio-temporally grounding natural sentence in video. In *ACL*, 2019. 2

[16] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. In *ICLR*, 2021. 8

[17] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-memory transformer for image captioning. In *CVPR*, 2020. 2

[18] Chaorui Deng, Qi Wu, Qingyao Wu, Fuyuan Hu, Fan Lyu, and Mingkui Tan. Visual grounding via accumulated attention. In *CVPR*, 2018. 2

[19] Jiajun Deng, Zhengyuan Yang, Tianlang Chen, Wengang Zhou, and Houqiang Li. TransVG: End-to-end visual grounding with transformers. In *ICCV*, 2021. 2

[20] Karan Desai and Justin Johnson. VirTex: Learning visual representations from textual annotations. In *CVPR*, 2021. 2

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019. 1

[22] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 2

[23] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 2

[24] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *ECCV*, 2020. 2

[25] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. TALL: Temporal activity localization via language query. In *ICCV*, 2017. 1, 2

[26] Simon Ging, Mohammadreza Zolfaghari, Hamed Pirsiavash, and Thomas Brox. COOT: Cooperative hierarchical transformer for video-text representation learning. In *NeurIPS*, 2020. 2

[27] Dongliang He, Xiang Zhao, Jizhou Huang, Fu Li, Xiao Liu, and Shilei Wen. Read, watch, and move: Reinforcement learning for temporally grounding natural language descriptions in videos. In *AAAI*, 2019. 2

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 6

[29] Lu He, Qianyu Zhou, Xiangtai Li, Li Niu, Guangliang Cheng, Xiao Li, Wenxuan Liu, Yunhai Tong, Lizhuang Ma, and Liqing Zhang. End-to-end video object detection with spatial-temporal transformers. *Proceedings of the 29th ACM International Conference on Multimedia*, 2021. 2

[30] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. *ICCV*, 2017. 1, 2

[31] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with temporal language. In *EMNLP*, 2018. 2

[32] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. In *CVPR*, 2017. 2

[33] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. Natural language object retrieval. In *CVPR*, 2016. 1, 2

[34] Binbin Huang, Dongze Lian, Weixin Luo, and Shenghua Gao. Look before you leap: Learning landmark features for one-stage visual grounding. In *CVPR*, 2021. 2

[35] De-An Huang, Shyamal Buch, Lucio Dery, Animesh Garg, Li Fei-Fei, and Juan Carlos Niebles. Finding" it": Weakly-supervised reference-aware visual grounding in instructional videos. In *CVPR*, 2018. 2

[36] Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. Pixel-BERT: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849*, 2020. 2, 3

[37] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. MDETR - modulated detection for end-to-end multi-modal understanding. In *ICCV*, 2021. 1, 2, 3, 5, 6

[38] Wonjae Kim, Bokyung Son, and Ildoo Kim. ViLT: Vision-and-language transformer without convolution or region supervision. In *ICML*, 2021. 2

[39] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *ICLR*, 2020. 8

[40] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual Genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2016. 5

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

[42] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *CVPR*, 2021. 2, 3

[43] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, Daxin Jiang, and Ming Zhou. Unicoder-VL: A universal encoder for vision and language by cross-modal pre-training. In *AAAI*, 2020. 2

[44] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. HERO: Hierarchical encoder for video+language omni-representation pre-training. In *EMNLP*, 2020. 2

[45] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, 2020. 2

[46] Yue Liao, Si Liu, Guanbin Li, Fei Wang, Yanjie Chen, Chen Qian, and Bo Li. A real-time cross-modality correlation filtering method for referring expression comprehension. In *CVPR*, 2020. 2

[47] Zhijie Lin, Zhou Zhao, Zhu Zhang, Qi Wang, and Huasheng Liu. Weakly-supervised video moment retrieval via semantic completion network. In *AAAI*, 2020. 2

[48] Jingyu Liu, Liang Wang, and Ming-Hsuan Yang. Referring expression generation and comprehension via attributes. In *ICCV*, 2017. 2

[49] Xihui Liu, Zihao Wang, Jing Shao, Xiaogang Wang, and Hongsheng Li. Improving referring expression grounding with cross-modal attention-guided erasing. In *CVPR*, 2019. 2

[50] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 6

[51] Yongfei Liu, Bo Wan, Lin Ma, and Xuming He. Relation-aware instance refinement for weakly supervised visual grounding. In *CVPR*, 2021. 2

[52] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 1

[53] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 13

[54] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019. 2

[55] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *CVPR*, 2020. 2

[56] Gen Luo, Yiyi Zhou, Xiaoshuai Sun, Liujuan Cao, Chenglin Wu, Cheng Deng, and Rongrong Ji. Multi-task collaborative network for joint referring expression comprehension and segmentation. In *CVPR*, 2020. 2

[57] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, 2020. 1

[58] Niluthpol Chowdhury Mithun, Sujoy Paul, and Amit K Roy-Chowdhury. Weakly supervised video moment retrieval from text queries. In *CVPR*, 2019. 2

[59] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. Modeling context between objects for referring expression understanding. In *ECCV*, 2016. 1, 2

[60] Mandela Patrick, Dylan Campbell, Yuki M Asano, Ishan Misra Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, Jo Henriques, et al. Keeping your eye on the ball: Trajectory attention in video transformers. In *NeurIPS*, 2021. 1, 2

[61] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015. 5

[62] 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. *arXiv preprint arXiv:2103.00020*, 2021. 1

[63] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union. In *CVPR*, 2019. 5

[64] Cristian Rodriguez, Edison Marrese-Taylor, Fatemeh Sadat Saleh, HONGDONG LI, and Stephen Gould. Proposal-free temporal moment localization of a natural-language query in video using guided attention. In *WACV*, 2020. 2, 5

[65] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual Captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 2

[66] Jing Shi, Jia Xu, Boqing Gong, and Chenliang Xu. Not all frames are equal: Weakly-supervised video grounding with contextual similarity and visual clustering losses. In *CVPR*, 2019. 2

[67] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 2014. 13

[68] Rui Su, Qian Yu, and Dong Xu. STVGBert: A visual-linguistic transformer based framework for spatio-temporal video grounding. In *ICCV*, 2021. 1, 2

[69] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: Pre-training of generic visual-linguistic representations. In *ICLR*, 2019. 2

[70] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. VideoBERT: A joint model for video and language representation learning. In *ICCV*, 2019. 2

[71] Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019. 2

[72] Zongheng Tang, Yue Liao, Si Liu, Guanbin Li, Xiaojie Jin, Hongxu Jiang, Qian Yu, and Dong Xu. Human-centric spatio-temporal video grounding with visual transformers. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021. 1, 2, 6

[73] Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers. In *ICLR*, 2021. 8

[74] Arun Balajee Vasudevan, Dengxin Dai, and Luc Van Gool. Object referring in videos with language and human gaze. In *CVPR*, 2018. 1, 2

[75] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1

[76] Jingwen Wang, Lin Ma, and Wenhao Jiang. Temporally grounding language queries in videos by contextual boundary-aware prediction. In *AAAI*, 2020. 2

[77] Liwei Wang, Jing Huang, Yin Li, Kun Xu, Zhengyuan Yang, and Dong Yu. Improving weakly supervised visual grounding by contrastive knowledge distillation. In *CVPR*, 2021. 2

[78] Peng Wang, Qi Wu, Jiewei Cao, Chunhua Shen, Lianli Gao, and Anton van den Hengel. Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In *CVPR*, 2019. 2

[79] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. 8

[80] Weining Wang, Yan Huang, and Liang Wang. Language-driven temporal activity localization: A semantic matching reinforcement learning model. In *CVPR*, 2019. 2

[81] Zhanghao Wu, Zhijian Liu, Ji Lin, Yujun Lin, and Song Han. Lite transformer with long-short range attention. In *ICLR*, 2020. 8

[82] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. Audiovisual slowfast networks for video recognition. *arXiv preprint arXiv:2001.08740*, 2020. 2

[83] Fanyi Xiao, Leonid Sigal, and Yong Jae Lee. Weakly-supervised visual grounding of phrases with linguistic structures. In *CVPR*, 2017. 2

[84] Masataka Yamaguchi, Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Spatio-temporal person retrieval via natural language queries. In *ICCV*, 2017. 2

[85] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. In *ICCV*, 2021. 2

[86] Sibei Yang, Guanbin Li, and Yizhou Yu. Cross-modal relationship inference for grounding referring expressions. In *CVPR*, 2019. 2

[87] Sibei Yang, Guanbin Li, and Yizhou Yu. Dynamic graph attention for referring expression comprehension. In *ICCV*, 2019. 2

[88] Zhengyuan Yang, Tianlang Chen, Liwei Wang, and Jiebo Luo. Improving one-stage visual grounding by recursive sub-query construction. In *ECCV*, 2020. 2

[89] Zhengyuan Yang, Boqing Gong, Liwei Wang, Wenbing Huang, Dong Yu, and Jiebo Luo. A fast and accurate one-stage approach to visual grounding. In *ICCV*, 2019. 2

[90] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. MAttNet: Modular attention network for referring expression comprehension. In *CVPR*, 2018. 2

[91] Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L Berg. A joint speaker-listener-reinforcer model for referring expressions. In *CVPR*, 2017. 2

[92] Yitian Yuan, Tao Mei, and Wenwu Zhu. To find where you talk: Temporal sentence localization in video with attention based location regression. In *AAAI*, 2019. 2

[93] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. In *NeurIPS*, 2020. 8

[94] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. MERLOT: Multimodal neural script knowledge models. In *NeurIPS*, 2021. 2

[95] Runhao Zeng, Haoming Xu, Wenbing Huang, Peihao Chen, Mingkui Tan, and Chuang Gan. Dense regression network for video grounding. In *CVPR*, 2020. 2

[96] Da Zhang, Xiyang Dai, Xin Wang, Yuan-Fang Wang, and Larry S Davis. MAN: Moment alignment network for natural language moment retrieval via iterative graph adjustment. In *CVPR*, 2019. 2

[97] Hanwang Zhang, Yulei Niu, and Shih-Fu Chang. Grounding referring expressions in images by variational context. In *CVPR*, 2018. 2

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

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

[100] Yanyi Zhang, Xinyu Li, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. VidTr: Video transformer without convolutions. In *ICCV*, 2021. 1, 2

[101] Zhu Zhang, Zhijie Lin, Zhou Zhao, and Zhenxin Xiao. Cross-modal interaction networks for query-based moment retrieval in videos. In *SIGIR*, 2019. 2

[102] 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, 2, 3, 6, 7, 13

[103] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and VQA. In *AAAI*, 2020. 2

[104] Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *CVPR*, 2018. 2

[105] Linchao Zhu and Yi Yang. ActBERT: Learning global-local video-text representations. In *CVPR*, 2020. 2

[106] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 2

[107] Bohan Zhuang, Qi Wu, Chunhua Shen, Ian Reid, and Anton Van Den Hengel. Parallel attention: A unified framework for visual object discovery through dialogs and queries. In *CVPR*, 2018. 2

Appendix

In this Appendix, we present additional visualizations of the different attention mechanisms in our space-time decoder in Section A. Section B provides additional implementation details. We then give detailed results for ablations in Section 4.2 on the VidSTG dataset [102] split by sentence type in Section C. Next we present an ablation of our fast and aggregation modules in Section D. Finally we discuss broader impact in Section E.

A. 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 7. For this example the time-aligned cross-attention for the visual modality is also shown in Figure 7. 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 6. 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 6 denote the predicted start and end times of the output tube. Next, the temporal self-attention is illustrated in Figure 8. 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.

B. 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 follows 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 [53] 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 [67] 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 $k > 1$.

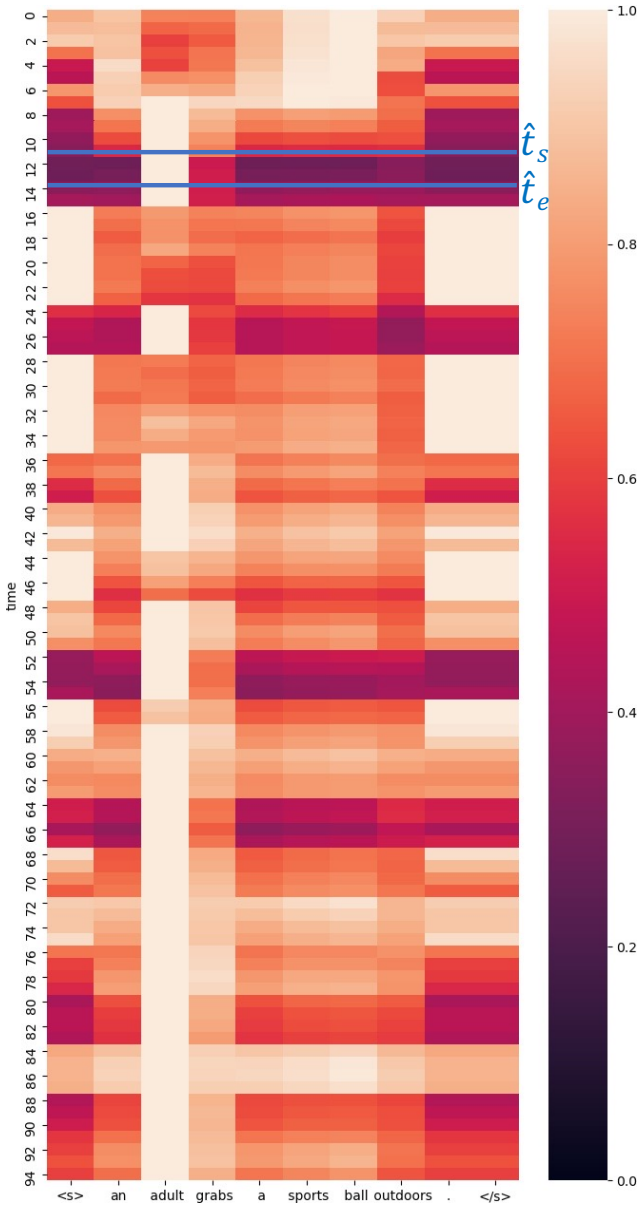

Figure 6. **Time-aligned cross-attention visualization (textual modality).** Visualization of the attention weights between the time query (y-axis) and its time-aligned visually-contextualized text features (x-axis) at different times in our space-time decoder. These attention weights are averaged across 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).

C. 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 Section 4.2.

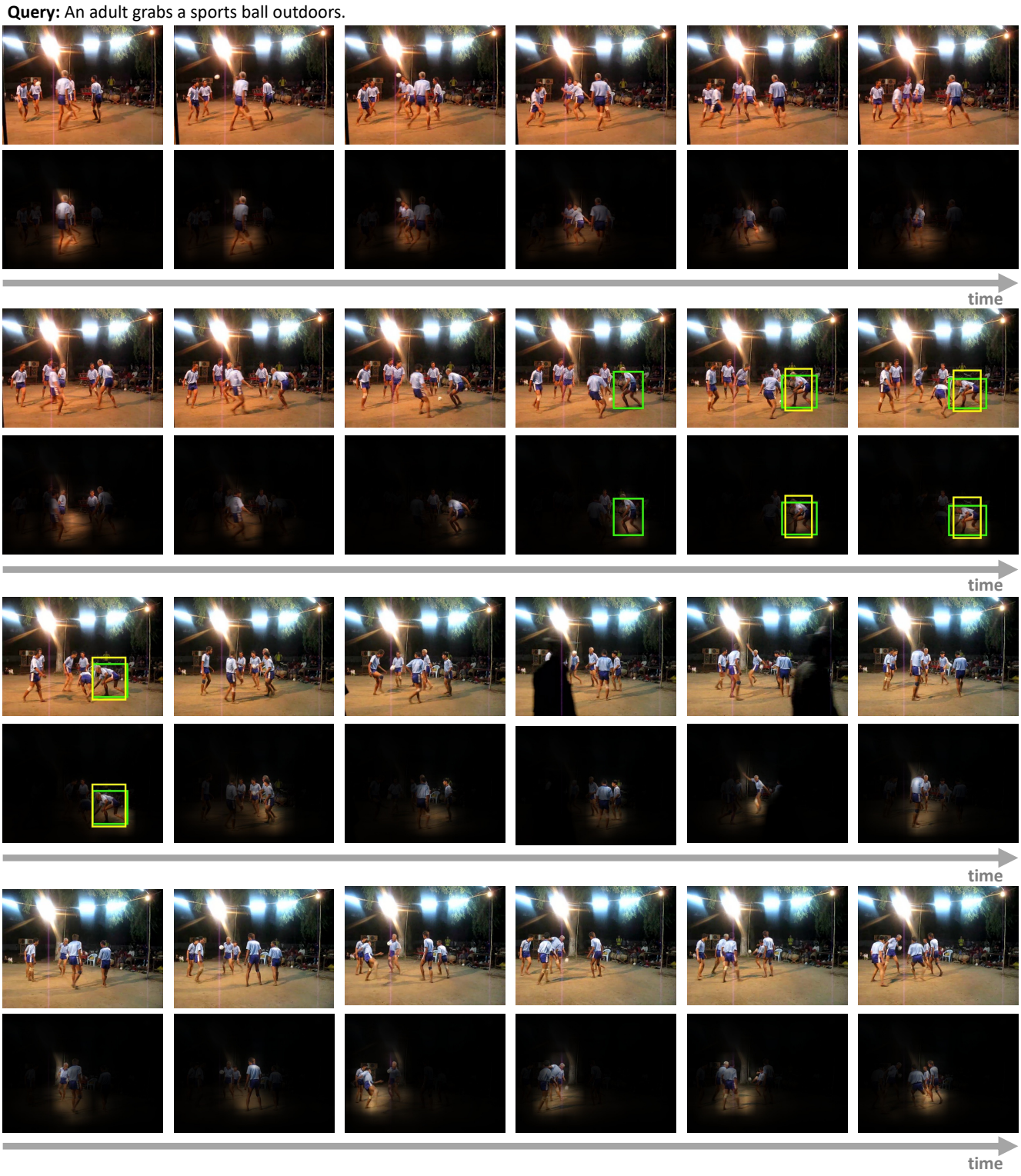

Figure 7. **Time-aligned cross-attention visualization (visual modality).** Top rows: Input frames with the predicted (yellow) and ground truth (green) spatio-temporal tubes overlaid. Bottom rows: Visualization of the attention weights between the time query and its time-aligned text-contextualized visual 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.

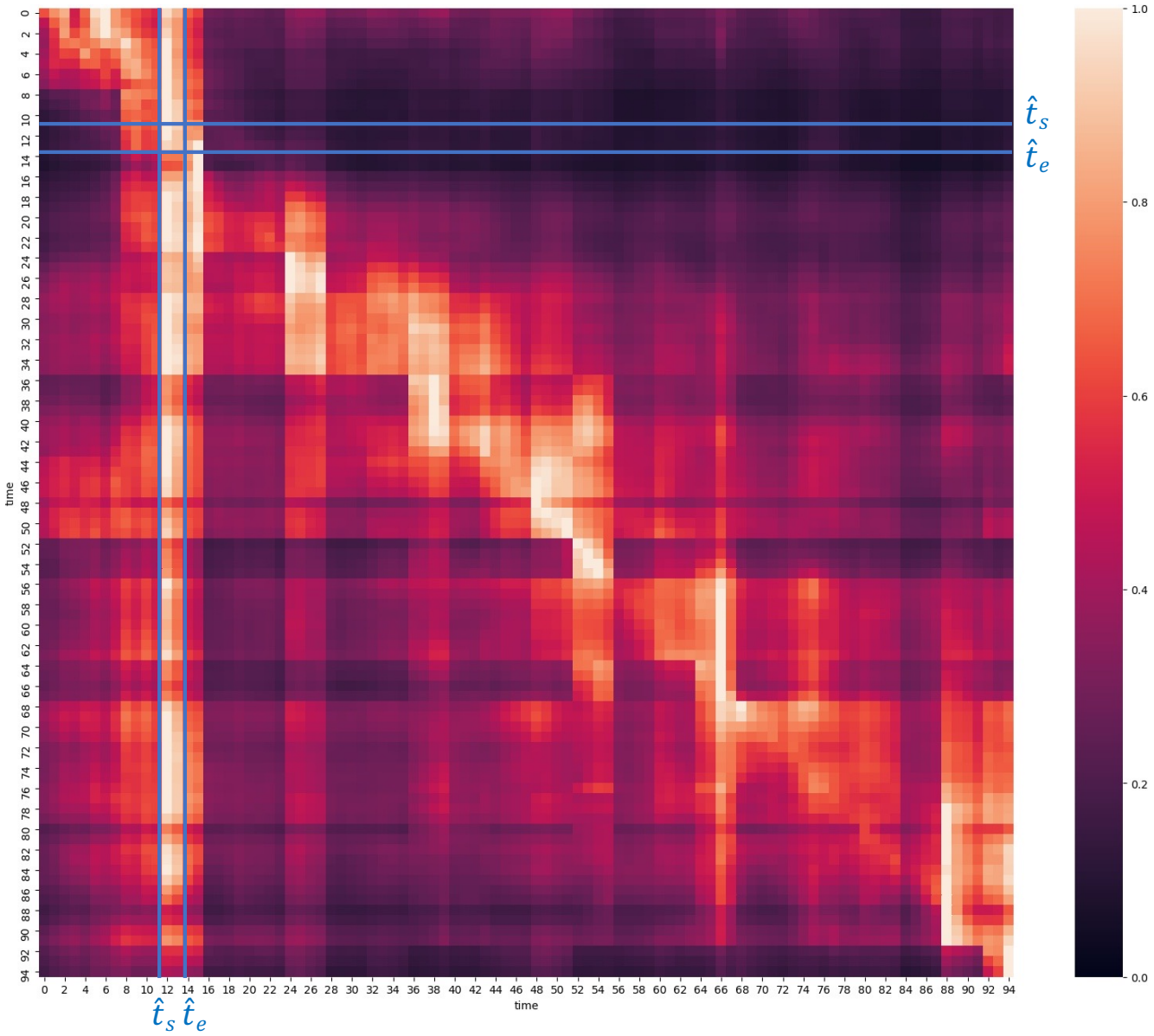

Figure 8. **Temporal self-attention visualization.** Visualization of the attention weights between the different time queries in our space-time decoder. The column t corresponds to the weights of the different time queries for the time query at time t. These attention weights are averaged across all 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).

	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 5. 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 6. Effect of the weight initialization for our model on the VidSTG validation set.

	Fast	Res.	Temp. Stride	Declarative Sentences					Interrogative Sentences					Mem. (GB)
				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.	—	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 7. 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.

Space-time decoder. We first provide detailed results for the ablation on our space-time decoder. The analysis is similar for both declarative and interrogative sentences. In detail, Table 5 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 6 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 7 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).

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 7, 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.

	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 8. 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.

D. Additional Experiments

In this section, we provide additional ablation studies. As in the ablations presented Section 4.2, 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 8. The comparison between our slow-fast design (row 6) and the slow-only variant (row 2) is discussed in Section 4.2. 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 8, 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 8, 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.

E. 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.