Spatio Video Grounding via Graph Transformer

Kimia Afshari, Hannah Hagen

292F Course Project Report

1. Background

Most efforts to bridge the gap between natural language and visual content primarily revolve around question-answering tasks, where users provide textual prompts to describe objects of interest in video scenes. While significant progress has been made in image and video question-answering, a fundamental question remains unanswered. How can we understand whether the predictions are truly based on the relevance to visual content or biased toward the powerful language model?

Finding a connection between the textual answer and the pertinent visual information is a big step toward improving interpretability and trustworthiness. Localization within the context of visual question answering (VQA) plays a pivotal role in comprehending the model's ability in decision-making. Furthermore, beyond its role in model refinement, spatial localization holds immense practical value across various domains. In the realm of security and surveillance, the capability to accurately identify and localize humans or activities within video scenes is indispensable. For example, given a footage video, these models can assist humans in detecting anomalies or potential threats as easily as asking a question to reason for it. From understanding human activities in crowded environments to identifying suspicious behaviors, the implications of accurate localization extend far beyond the realm of research, stepping into real-world applications that prioritize safety, security, and informed decision-making.

In this project, we enhance a Video Question Answering (VQA) model to respond to natural language queries while localizing the answer within the video. Our proposed architecture extends CoVGT [1], an existing video QA system, by incorporating a space-time decoder specifically tailored for spatial localization. CoVGT adopts a graph-based representation for video elements, treating them as nodes and edges to capture dynamic interactions among objects for effective video reasoning. The use of graph transformers on nodes and edges enables the model to derive informative temporal relations across different timestamps. Their innovative approach to video question answering using graph neural networks motivated us to embark on this project.

2. Datasets

We use STAR dataset [2] as the benchmark for our spatial video grounding experiments. STAR serves as a widely recognized visual situated reasoning dataset, featuring videos that capture

human actions and interactions, such as drinking from a glass of water. The dataset comprises 22,000 video clips accompanied by 60,000 questions, each with four candidate choices. There are four categories of questions: interaction, sequence, prediction and feasibility. Figure 1 shows a data sample with all four question types. We choose this dataset for its strength in evaluating reasoning in real-world situations.

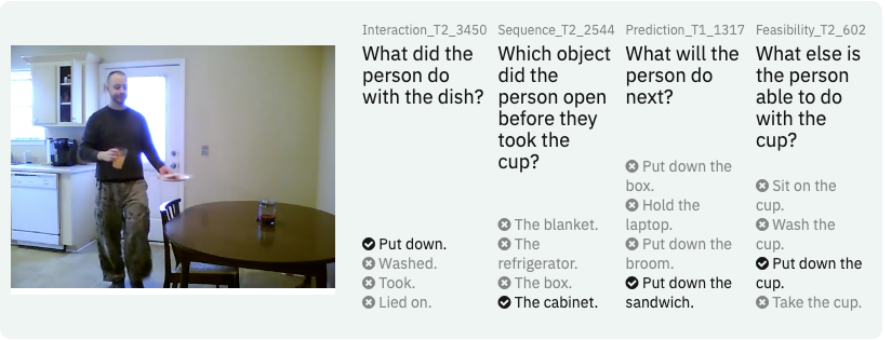


Figure 1. **A sample data in the STAR dataset**, showing four questions corresponding to the video. The highlighted choices are the answers.

The dataset does not include annotations for every individual frame of the video, since many adjacent frames represent the same action. Instead, dataset creators specifically annotated frames where a noticeable action or scene change takes place, which we refer to as "keyframes." Annotations include a question, answer, and bounding box locations for the subject/objects.

Our contribution to the dataset plays a crucial role in accomplishing the task. Before applying this dataset to our problem, we need to pre-process and transform the data. Specifically, we need to extract only the bounding boxes for the objects corresponding to the answer, ignoring the rest. This transformation is done by parsing each question and answer according to a pre-defined format based on its type. As shown in Figure 2, each question type follows a standard format, allowing us to parse the questions. In addition, we remove the samples whose answers do not contain an object (e.g., Interaction_T2 answer format is merely “[Verb]ed”).

Question Type	Subtype	Question Format	Answer Format
Interaction	T2	What did the person do with the [Obj]?	[Verb]ed.
Interaction	T3	What did the person do while they were [Contact_Rel] the [Obj]?	[Verb]ed the [Obj].
Sequence	T1	Which object did the person [Verb1] after they [Verb2]ed the [Obj2]?	The [Obj1].
Sequence	T3	What happened after the person [Verb1]ed the [Obj1]?	[Verb2]ed the [Obj2].
Prediction	T3	Which object would the person [Verb] next?	The [Obj].
Prediction	T4	Which object would the person [Verb2] next after they [Verb1] the [Obj1]?	The [Obj2].
Feasibility	T3	Which object is possible to be [Verb1]ed when the person is [Spatial_Rel] the [Obj2]?	The [Obj1].
Feasibility	T6	What is the person able to do after [Verb1]ing the [Obj1]?	[Verb2] the [Obj2].

Figure 2. **QA Template** showing a subset of the question and answer formats. There are 20 question types overall.

3. Methods

The model is comprised of two main components: a video-text encoder and a space-time decoder. The video-text encoder is responsible for capturing the video-language interaction, allowing for detailed spatial localization given a language query. The space-time decoder then models the informative visual contents by observing the temporal interactions over the entire video. The model overview is illustrated in Figure 3. More details on individual components are described below. In the results section, we also present the quantitative and qualitative results of our empirical study.

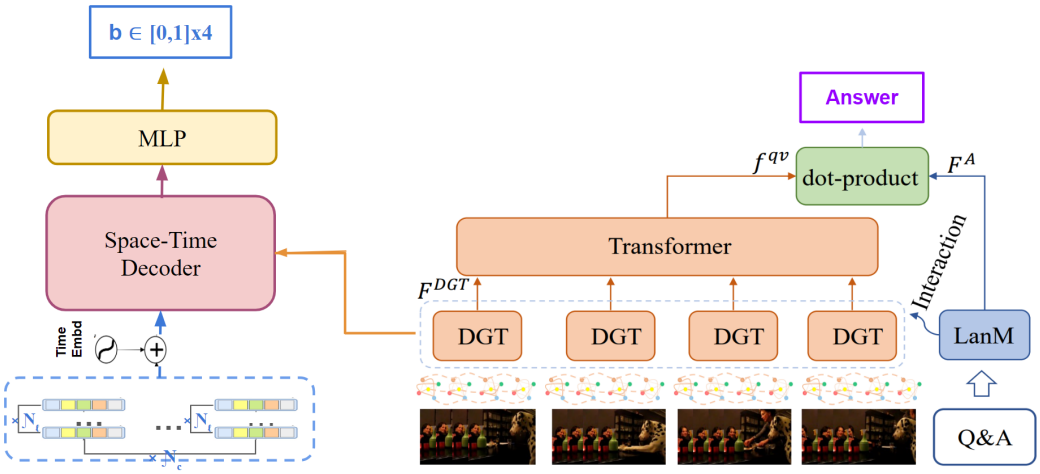


Figure 3. **Overview of the Proposed Model Architecture**

Video Sampling

As mentioned previously, the STAR dataset provides answer bounding boxes for only the keyframes. Additionally, we only have access to pre-computed features for a subset of the video frames. Frames with object location data (i.e keyframes) may not have pre-computed features, and vice versa. This mismatch poses a challenge to assembling a compelling dataset.

To address this challenge, we generated a custom dataset. The approach is to maximize the number of keyframes included under the condition that the frames have corresponding features. We achieve this by even sampling from the video's keyframes, ensuring that the corresponding offline features exist. If we need more samples to select, we begin sampling from the rest of the video frames that are not in the keyframes set. This is also constrained under the same matching condition. This way, we maximize the number of annotated frames that have corresponding features.

Video-text Encoder

The video-text encoder module jointly models the objects, their relations, and dynamics with respect to a text query for visual reasoning. It derives global visual clues from the local video contents and their relations over time.

We follow the video-text encoding method proposed in [1] to achieve this. First, we employ a video graph representation module to represent visual contents using graphs. Then, a dynamic graph transformer comes into play to capture object dynamics and their spatial interaction over time. Lastly, a global transformer component derives informative visual clues by reasoning over the local video contents. To enhance video-text correspondence, a cross-modal interaction is utilized. This component incorporates textual information into visual contents to obtain a query-aware representation for downstream tasks.

Space-Time Decoder

This module models temporal interaction between objects in all the frames to decode multi-modal features of video-text into spatial representations for localization. To capture the temporal interaction of the objects across the entire video, a temporal self-attention component is employed. A time-aligned cross-attention component is also used to derive pertinent visual information over time with respect to the multi-modal features obtained from the decoder.

In detail, the decoder operates on object queries across all T frames of a video and a positional encoding $\{q_t\}_{t=1}^{T}$, referred to as time queries. The decoder also takes clip-wise video-language embeddings as input from the video-text encoder to incorporate query-aware information. The decoder is a succession of N decoding blocks. Each block is composed of temporal self-attention, time-aligned cross-attention, and feed-forward layers interleaved with normalization, as shown in Figure 4. The decoder outputs object locations as the answer to ground video given an input query. The individual layers are described in detail next.

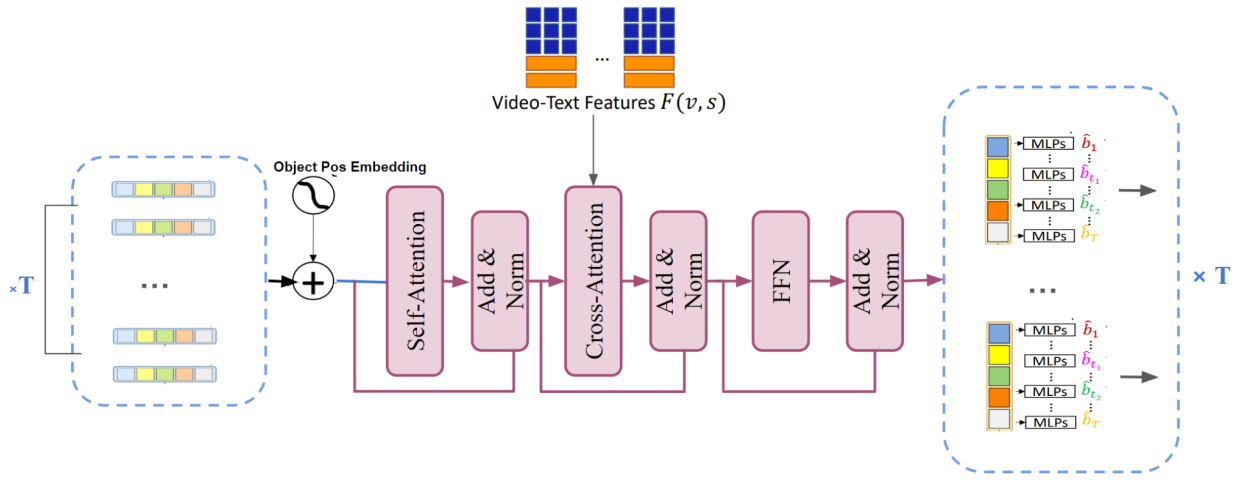


Figure 4. **Space-Time Decoder Architecture**

Temporal self-attention

The learnable position-aware object queries o_t attend to each other using the temporal self-attention layers. Note that the object queries are combined with the positional embeddings

to incorporate positional information. This layer is in each of the N blocks of the decoder and is responsible for modeling the long-range temporal interactions in the entire video.

Time-aligned cross-attention

In this stage, each object query cross-attends to the video-text features (output of the video-text encoder) to incorporate global visual contents with respect to the input language query. This time-aware cross-attention and temporal self-attention are essential elements of the decoder as they decode frames jointly for capturing video dynamics.

Output head

The output of the decoder is a set of object locations across all T frames of a video, denoted as $\{B_t\}_{t=1}^{T}$. For every t-th frame of a video, it predicts one object as the visual answer to a given text query. In detail, normalized coordinates of all bounding boxes (2D center and size) $\widehat{b} \in [0,1]^{T\times 4}$ are predicted with a 3-layer MLP.

Answer Prediction

Given a video, the model responds to an input language query in textual and visual answers. The space-time decoder outputs the spatial locations of the answer objects in each T frame of a video $\{\widehat{b_t}\}_{t=1}^{T}$. To infer the textual answer, we obtain the similarity between the global query-aware representations and all the candidate answers. Then, the candidate of the maximal similarity is returned as a prediction.

In detail, we first get a global representation of each candidate answer by mean-pooling its tokens from the language model (Roberta). Then, the similarity of each candidate with the query-aware representation is calculated by a dot-product between two vectors. Finally, the answer with maximum similarity is considered the final answer.

Training Loss

Each video is annotated with a question, answer, and bounding boxes denoting the answer's spatial locations. To train our model, we consider L_{vqa} and L_{vq} losses as the textual answer loss and two losses of $gIoU$ and L_1 for bounding box losses. The linear combination of these losses constructs out training loss for model optimization:

$$L = L_{vqa}(v, qa^{+}, qa^{-}) + \lambda_{vq} L_{vq}(v, q^{+}, q^{-}) + \lambda_{L_1} L_{L_1}(\widehat{b}, b) + \lambda_{gIoU} L_{gIoU}(\widehat{b}, b)$$

where L_{vqa} and L_{vq} are the contrastive objectives oriented for question answering in supervised and self-supervised manners introduced in [1], L_1 is a loss on bounding box coordinates, and L_{gIoU} is a generalized “Intersection over Union” (IoU) loss on the bounding boxes. In addition, $b \in [0, 1]^{T\times 4}$denotes the normalized ground truth box coordinates and $\widehat{b}$ the predicted bounding

boxes. Finally, different λ• are scalar weights of the individual losses that determine the contribution of each loss to the final loss.

4. Results

Evaluation metric

To evaluate the model performance, we evaluate the predicted textual answer and the bounding box locations separately. We follow standard protocol and report accuracy (percentage of correctly answered questions) as the evaluation metric for the textual question-answering part. Evaluation on spatial grounding is performed by defining a $vIoU$ as $vIoU = \frac{1}{|T|}\sum_{t=1}^{T} IoU(\widehat{b}, b)$ where T is a set of all frames in a video, and $\widehat{b_t}$ (respectively b_t) are the predicted (respectively GT) boxes at time t. Then, we define and report a m_vIoU as the average $vIoU$ for different question categories. To have a better understanding of the model's overall performance, we report the final evaluation as the average of $vIoU$ over all question categories. We also consider $vIoU@R$, the proportion of samples for which $vIoU > R$[1].

Implementation details

We follow the video sampling method proposed in [1] to sparsely sample 32 frames, distributing them into $k = 8$ clips of length $l_c = 4$. This will help reduce the training load, given our limited resources and budgets, while gaining our results. As the visual content features, we use pre-extracted frame-level appearance and object-level regional features of the STAR dataset used in [1]. The regional features contain the top 10 high-confident regions along with their corresponding appearance features for each frame. The dimension of the models' hidden states is set to $d = 512$, and the default number of graph layers is $U = 2$. Besides, the default number of layers and self-attention heads in transformers are $H = 4$ for video-text encoding, $H = 6$ for space-time decoding and $e = 8$ ($e = 5$ for the edge transformer in DGT), respectively. The contribution loss weights are set as $\lambda_{l1} = 2$, $\lambda_{gIoU} = 5$, $\lambda_{vq} = 5$ and $\lambda_{cl} = 1$. The model is initialized with pre-training weights of [1] and trained for 20 and 40 epochs on the STAR dataset and shows that training loss does not vary for epochs over 20. The optimization takes 12hrs on 1 Tesla V100 GPU.

Quantitative results

Table 1 shows the result of the proposed model on different question types of the STAR dataset. Since the dataset does not provide a test set, we report our evaluation on the validation set.
As shown in the table, the textual question-answering part achieved an accuracy of 44.6 in video reasoning tasks. We believe that this slight drop compared to its counterpart (CoVGT) is due to an increase in the model complexity as it tries to learn more. For the spatial localization part, the m_vIoU shows decent results. Since, at this moment, there is no prior experiment on this dataset for localization, we can not fully compare our model's performance. Overall, these results show a potential for further research to improve the performance of both parts.

[1] R is a predefined threshold. In our experiments, we report m_vIoU for R in [0.3, 0.5].

Model	Pre-training Data	STAR												
		Interaction			Sequence			Prediction			Feasibility			All
		m_vIoU	vIoU@0.3	vIoU@0.5	m_vIoU	vIoU@0.3	vIoU@0.5	m_vIoU	vIoU@0.3	vIoU@0.5	m_vIoU	vIoU@0.3	vIoU@0.5	Acc. %
CoVGT	WebVid2M	-	-	-	-	-	-	-	-	-	-	-	-	46.20
Ours	CoVGT	17.8	25.8	19.1	16.5	28.6	17.9	12.3	21.5	16.6	15.9	23.1	21.4	44.59

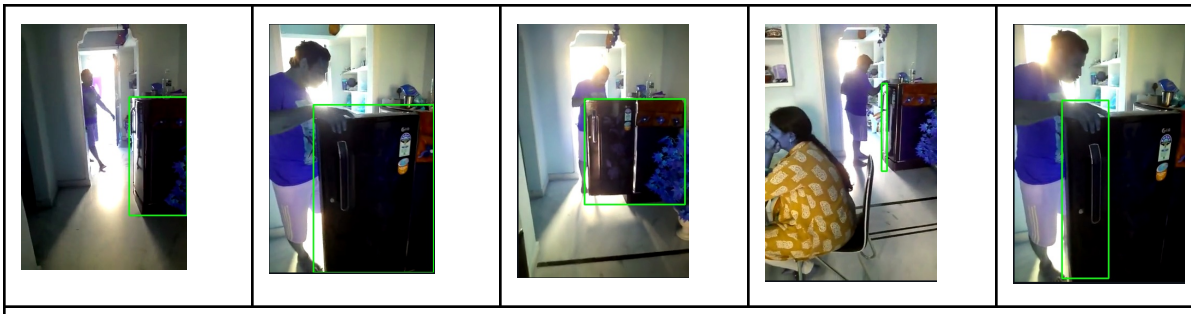
Table 1: **Quantitative results on STAR dataset**.

Qualitative samples

We show qualitative examples of our predictions on the STAR test set in Figure 5. These examples show that our model is able to predict the correct answer and approximate bounding boxes associated with the input question.

In the first example from the question type “sequence”, our model predicts the correct answer among 4 choices. It also localizes the clothes that the person is taking off. However, the location covers the area around the clothes slightly less well. The second example demonstrates the performance of the model in the feasibility category. The model is able to predict the correct answer precisely among the four given options, as well as the location of the refrigerator. However, in a few frames, it fails to localize the full refrigerator. The last example from the prediction category shows the model's failure in predicting the correct answer among the choices. Since the bounding box predictions depend on the answer prediction from the video-text encoder component, it is unable to output the correct location.

<table>
<tr><td colspan="5">  </td></tr>
<tr><td colspan="5">Which object did the person throw before they sat on the table?</td></tr>
<tr><td>1. The cup/glass/bottle.</td><td>2. The broom.</td><td>3. The book.</td><td colspan="2">4. The clothes.</td></tr>
</table>



What is the person able to do after walking through the doorway?

1. **Close the refrigerator.**	2. Throw the bag.	3. Take the shoe.	4. Close the door.

What will the person do next?

1. Eat the medicine.	2. **Take the dish.**	3. **Put down the bag.**	4. Take the paper/notebook.

Figure 5. **Qualitative examples** of our predictions on the STAR test set.

If we had more time, we would have evaluated the STAR dataset on one of the SOTA spatio-temporal grounding models to compare our localization. Our current results are also accessible in our GitHub repo.

5. Conclusion

This work combines video question answering with spatial grounding. It introduces an end-to-end pipeline that adds localization on top of the textual answer given a question to serve as visual evidence for the model's answer selection. The proposed space-time transformer decoder enables the model to learn time-aligned spatial locations of the predicted answer. Video graph representation and the utilization of transformers at different levels of granularity tackle the challenge of capturing temporal interactions, allowing the system to reason over the entire video. On the other hand, sparsely sampling of video frames and employing pre-computed visual features rather than having a separate feature extractor helps us conduct our experiments efficiently and gain results. Finally, despite encountering various challenges and limitations along the way, we successfully navigated the entire process, from designing the model to its deployment.

6. Future Works

There are some limitations with the existing model that we were unable to solve due to time constraints. One thing that needs to be done is to conduct more experiments; we are especially interested in evaluating the STAR dataset on TubeDETR [3] video grounding model to compare the effectiveness of our spatial localization. Secondly, this model outputs bounding boxes for all T frames of a video regardless of whether the object exists in the frame. One can work on predicting time segments when the answer exists and output predictions within the right video segment (start and end time that the answer lies in). This requires an additional time head to predict start and end time. The m_tIoU metric, the mean Intersection over Union on time tubes, is one of the potential solutions. Lastly, having the flexibility to efficiently extract all video frame features, rather than relying on the pre-computed features, can increase the model performance for the spatial localization part. We tried to dump frames at various fps and extract the regional features. However, due to the large-scale nature of the video data, it failed (we encountered a disk issue as well). In summary, tackling the aforementioned limitations is a promising direction for future research in this area.

7. References

[1] Xiao, Junbin, Pan Zhou, Angela Yao, Yicong Li, Richang Hong, Shuicheng Yan, and Tat-Seng Chua. "Contrastive Video Question Answering via Video Graph Transformer." arXiv preprint arXiv:2302.13668 (2023).

[2] B. Wu, S. Yu, Z. Chen, J. B. Tenenbaum, and C. Gan, “Star: A benchmark for situated reasoning in real-world videos,” in NeurIPS, 2021.

[3] Yang, Antoine, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. "Tubedetr: Spatio-temporal video grounding with transformers." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16442-16453. 2022.