Supplemental Material
DORi: Discovering Object Relationships for Moment Localization of a Natural Language Query in a Video

Cristian Rodriguez-Opazo[1,2] Edison Marrese-Taylor[3] Basura Fernando[4]
Hongdong Li[1,2] Stephen Gould[1,2]

[1] Australian National University, [2] Australian Centre for Robotic Vision (ACRV)
{cristian.rodriguez, hongdong.li, stephen.gould}@anu.edu.au
[3] Graduate School of Engineering, The University of Tokyo [4] A*STAR Singapore
emarrese@weblab.t.u-tokyo.ac.jp fernando_basura@ihpc.a-star.edu.sg

Appendices

A. Ablated models

In the following sub-sections we give details about the ablated models presented in Section **??**.

A.1. No Node Type

This ablation experiment is intended to show the importance of considering the Faster-RCNN features related to human labels as a different source of information. The experiment consists of assigning the same 15 object features extracted for each of the keyframes only to the Object node $\mathcal{O}$. In this way we limit the ability of the network to only be able to find relations between objects and activity representations, but without reducing the total amount of data that is available to it. We consider this experiment is very relevant as it shows that the additional information provided by the objects detected is not the only reason to explain the performance improvements, but rather the way in which this data is used is more relevant. In fact, enabling the model to obtain state-of-the-art performance in different and challenging benchmarks.

A.2. No Language Attention

In this case we replace the set of linguistic nodes by a single query node $\mathcal{Q}$. It receives a high-dimensional representation (denoted by q) of the natural language query Q, as can be seen in Figure 1. This high-dimensional representation is constructed using a function $F_Q : Q \mapsto q$ that first maps each word w_j for $j = 1, \ldots, m$ in the query to a semantic embedding vector $h_j \in \mathbb{R}^{d_w}$, where d_w defines the hidden dimension of the word embedding. Representations for each word are then aggregated using mean pooling to get a semantically rich representation of the whole query.

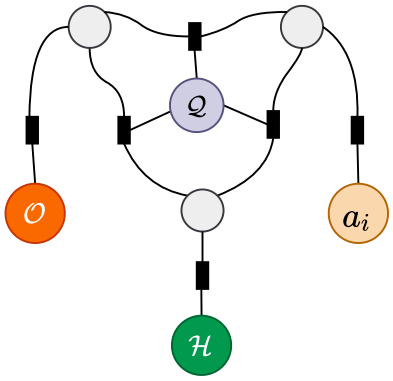

Figure 1. Spatial graph with a single query node $\mathcal{Q}$

Although the query node is generic, in this work we use a bi-directional GRU [1] on top of GLoVe word embeddings, which are pre-trained on a large collection of documents, for computing the h_j. Therefore, our query function F_Q is parameterized by both GLoVe embedding and the GRU.

Again we capture capture the relationship between this high-dimensional representation of the query and any observation of the nodes human $\mathcal{H}$, object $\mathcal{O}$ and activity $\mathcal{A}$, using a linear mapping function f specific for each node, as follows:

$$\Phi^{n}_{\mathcal{Q},\mathcal{A}} = f_{\mathcal{Q},\mathcal{A}}(q, a^n) \quad (1)$$

$$\Phi^{j,n}_{\mathcal{Q},\mathcal{O}} = f_{\mathcal{Q},\mathcal{O}}(q, o^{j,n}) \quad (2)$$

$$\Phi^{k,n}_{\mathcal{Q},\mathcal{H}} = f_{\mathcal{Q},\mathcal{H}}(q, h^{k,n}) \quad (3)$$

where functions $f_{\mathcal{Q},\mathcal{A}}, f_{\mathcal{Q},\mathcal{O}}, f_{\mathcal{Q},\mathcal{H}}$ are simple linear projections. For example, in the case of the object observations we have $f_{\mathcal{Q},\mathcal{O}}(q, o^{j,n}) = W_{qo}[q; o^{j,n}] + b_{qo}$, where the subindex

qo denotes the dependency of the parameters of the linear function which are specific for each relation. To compute the messages that are passed between the nodes, we utilize the following functions:

$$\Psi^{j,n}_{\mathcal{H},\mathcal{Q},\mathcal{O}} = f_{\mathcal{H},\mathcal{Q},\mathcal{O}}(\Phi^{j,n}_{\mathcal{Q},\mathcal{O}}, \textstyle\sum_{k=1}^{K} \Phi^{k,n}_{\mathcal{Q},\mathcal{H}}) \quad (4)$$

$$\Psi^{j,n}_{\mathcal{A},\mathcal{Q},\mathcal{O}} = f_{\mathcal{A},\mathcal{Q},\mathcal{O}}(\Phi^{j,n}_{\mathcal{Q},\mathcal{O}}, \Phi^{n}_{\mathcal{Q},\mathcal{A}}) \quad (5)$$

$$\Psi^{n}_{\mathcal{H},\mathcal{Q},\mathcal{A}} = f_{\mathcal{H},\mathcal{Q},\mathcal{A}}(\Phi^{n}_{\mathcal{Q},\mathcal{A}}, \textstyle\sum_{k=1}^{K} \Phi^{k,n}_{\mathcal{Q},\mathcal{H}}) \quad (6)$$

$$\Psi^{n}_{\mathcal{O},\mathcal{Q},\mathcal{A}} = f_{\mathcal{O},\mathcal{Q},\mathcal{A}}(\Phi^{n}_{\mathcal{Q},\mathcal{A}}, \textstyle\sum_{j=1}^{J} \Phi^{j,n}_{\mathcal{Q},\mathcal{O}}) \quad (7)$$

$$\Psi^{k,n}_{\mathcal{O},\mathcal{Q},\mathcal{H}} = f_{\mathcal{O},\mathcal{Q},\mathcal{H}}(\Phi^{k,n}_{\mathcal{Q},\mathcal{H}}, \textstyle\sum_{j=1}^{J} \Phi^{j,n}_{\mathcal{Q},\mathcal{O}}) \quad (8)$$

$$\Psi^{k,n}_{\mathcal{A},\mathcal{Q},\mathcal{H}} = f_{\mathcal{A},\mathcal{Q},\mathcal{H}}(\Phi^{k,n}_{\mathcal{Q},\mathcal{H}}, \Phi^{n}_{\mathcal{Q},\mathcal{A}}) \quad (9)$$

where again $f_{\mathcal{H},\mathcal{Q},\mathcal{O}}, f_{\mathcal{A},\mathcal{Q},\mathcal{O}}, f_{\mathcal{H},\mathcal{Q},\mathcal{A}}, f_{\mathcal{O},\mathcal{Q},\mathcal{A}}, f_{\mathcal{O},\mathcal{Q},\mathcal{H}}$ and $f_{\mathcal{A},\mathcal{Q},\mathcal{H}}$ are linear mappings, each receiving as input a concatenations of the corresponding features capturing. Finally, we update the representation of the human, action and object nodes based on the following formulas.

$$o^{j,n+1} = \sigma(m_o(\Psi^{j,n}_{\mathcal{H},\mathcal{Q},\mathcal{O}} \odot \Psi^{j,n}_{\mathcal{A},\mathcal{Q},\mathcal{O}}) \odot o^{j,0}) \quad (10)$$

$$a^{n+1} = \sigma(m_a(\Psi^{n}_{\mathcal{H},\mathcal{Q},\mathcal{A}} \odot \Psi^{n}_{\mathcal{O},\mathcal{Q},\mathcal{A}}) \odot a^{0}) \quad (11)$$

$$h^{k,n+1} = \sigma(m_h(\Psi^{k,n}_{\mathcal{O},\mathcal{Q},\mathcal{H}} \odot \Psi^{k,n}_{\mathcal{A},\mathcal{Q},\mathcal{H}}) \odot h^{k,0}) \quad (12)$$

where $\odot$ is the element-wise product and m_o, m_a, m_h are again linear functions.

B. Language Attention

In the following Figures 2 and 3, we present a set of samples of the multihead attention to the query sentence on the Charades-STA dataset.

C. Examples

In the following Figures, we present success and failure cases of our method on Charades-STA, YouCookII and TACoS dataset. Each visualization is showing a subsample of the keyframes inside of the prediction with their corresponding spatial observations. In green observations associated with the human node $\mathcal{H}$ and orange for the object node $\mathcal{O}$. Moreover, each visualization is presenting the ground-truth and predicted localization in seconds of the given query.

C.1. Charades-STA

Success cases of our algorithm on the Charades-STA dataset can be seen in Figure 6. In Figure 4, given the query "a person cooks a sandwich on a panini maker" our method could localize the moment at a tIoU of 99.56%. The label of the features extracted by Faster-RCNN to localize the query are *'bottle', 'counter', 'door', 'drawer', 'faucet', 'floor', 'glasses', 'hair', 'jacket', 'jeans', 'kitchen', 'microwave', 'pants', 'shelf', 'shirt', 'sink', 'stove', 'sweater',* ***'toaster'****, 'wall', 'window', 'woman'*.

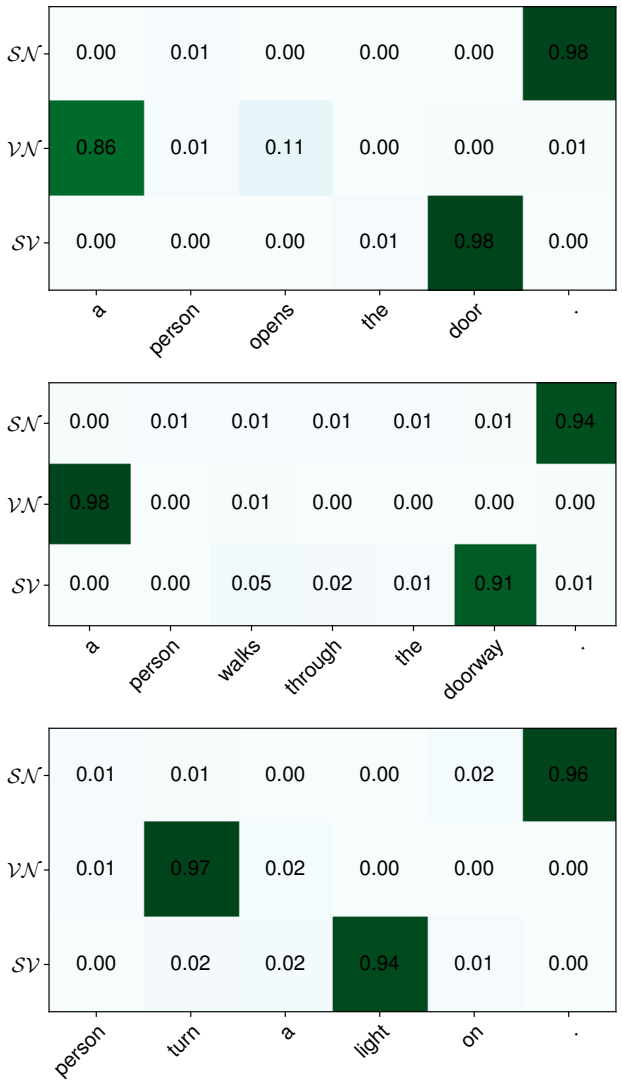

Figure 2. Linguistic nodes attentions on Charades-STA.

In the case of Figure 5, given the query "the person closes a cupboard door." our method could localize the moment at a tIoU of 97.88%. The features extracted by Faster RCNN for this query are *'arm', 'building',* ***'cabinet'****, 'counter', 'door', 'faucet', 'hair', 'hand', 'head', 'jacket', 'kitchen', 'man', 'microwave', 'refrigerator', 'shirt', 'sink', 'sleeve', 'stove', 'sweater', 'wall', 'window', 'woman'*.

Failure cases of our method are presented in Figure 9. In the first example, given a query "a person opens a door goes into a room." our method could detect correct spatial features, such as 'door' and 'knob', and the correct span of the query, according to our qualitative evaluation. However, in this case, the annotation for the query is localized incorrectly in the video. It refers to the last part of the video, where a person is using a laptop, as can be seen at the right of Figure 7. In Fig. 8 we can see our method localizing

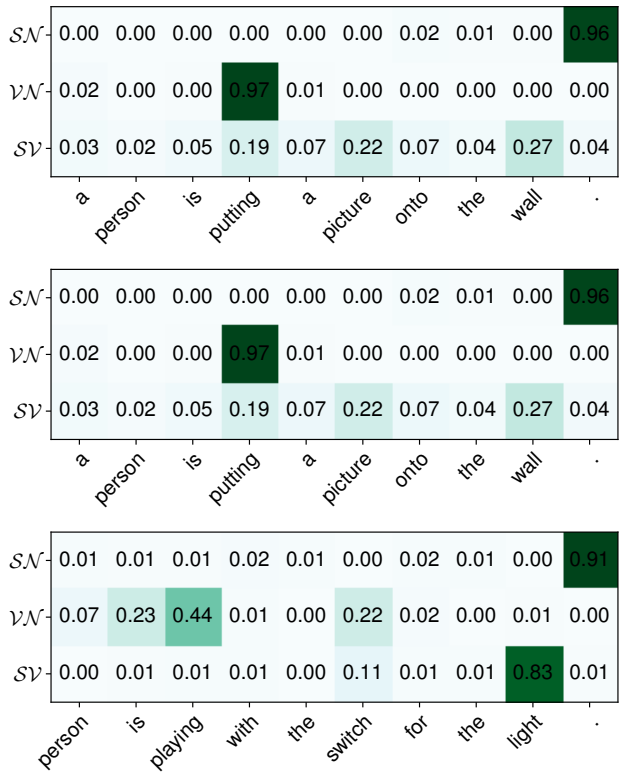

Figure 3. Linguistic nodes attentions on Charades-STA.

the query "person walks over to the refrigerator open it up", however, the annotation is not considering that the moment is performed two times in the video.

C.2. YouCookII

Although videos in YouCookII are much longer than videos in Charades-STA, our method still can get good localization performance. In Figure 10 given the query "spread the sauce onto the dough" our method localize the query at a tIoU of 98.57%. The label of the feature extracted by Faster-RCNN on this case are *'bacon', 'bird', 'board', 'bottle', 'bowl', 'cabinet', 'cake', 'cherry', 'chocolate', 'cookie', 'counter', 'cutting board', 'dessert', 'door', 'drawer', 'finger', 'floor', 'fork', 'fruit', 'glass', 'grape', 'ground', 'hand', 'handle', 'jeans', 'ketchup', 'knife', 'meat', 'olive', 'pancakes', 'pepperoni', 'person', 'phone',* ***'pizza'****, 'plant', 'plate',* ***'sauce'****, 'saucer', 'shirt', 'sleeve', 'spoon', 'table', 'towel', 'tree', 'wall'*.

Figure 11 shows the query "cook the pizza in the oven", which belong to the same video. In this case the label of the features extracted by Faster-RCNN are *'arm', 'bar', 'board', 'building', 'cabinet', 'car', 'ceiling', 'cheese', 'cord', 'counter', 'crust', 'cucumber', 'curtain', 'door', 'drawer', 'fireplace', 'floor', 'food', 'fork', 'glass', 'grill', 'hand', 'hotdog', 'key', 'keyboard', 'kitchen', 'knife', 'knob', 'laptop', 'leaf', 'leaves', 'leg', 'light', 'man', 'microwave', 'mouse',* ***'oven'****, 'oven door', 'person',* ***'pizza'****, 'plate', 'pole', 'rack', 'roof', 'room', 'salad', 'screen', 'shadow', 'sleeve', 'slice', 'spinach', 'stove', 'table', 'television', 'thumb', 'tracks', 'train', 'tray', 'vegetable', 'vegetables', 'wall', 'window', 'wood'* and our method could localize the query with a temporal intersection over union of 97.60%.

Failure cases of our method on YouCookII dataset are presented in Figure 15. In these cases, it is possible to see that our approach is able to recognize the activity *add* and *mix* correctly. However, the objects "dressing, ginger and garlic" are not detected by Faster-RCNN, probably given that the object detector has not been trained to deal with some of the kinds of objects present on this dataset. We think this naturally hinders the disambiguation capabilities of our model, specially in terms of the repetitive actions such as as adding, mixing and pouring, which are often performed throughout recipes like the one depicted in the example.

C.3. TACoS

Figures 18 and 21 show two examples of success and failure cases on the TaCoS dataset, respectively. It is possible to see the how challenging this dataset is in general, as in the the cases where our approach fails it is in fact difficult even for us to localize the given query.

D. Experimental Information

Our models are implemented using PyTorch [3] and are trained using the Adam [2] optimizer, with a batch size of 6. Experiments for different datasets were run in two different machines:

- First server machine with an Intel Core i7-6850K CPU with two NVIDIA Titan Xp (Driver 430.40, CUDA 10.1) GPUs, and one NVIDIA Quadro P5000, running ArchLinux
- An additional server machine with an Intel Xeon 4215 CPU, with three NVIDIA RTX8000 (Driver 430.44, CUDA 10.1) GPUs, running Ubuntu 16.04

We used PyTorch version 1.4. Our method has 10.865.155 trainable parameters. In training takes 1.56 hours per epoch in Charades-STA, 4.3 hours per epoch in TACoS, 5.4 hours per epoch in YouCookII and 6.7 hours per epoch in ActivityNet. In average our method takes 0.015 seconds to localize one query.

References

[1] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN

encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, Oct. 2014. Association for Computational Linguistics.

[2] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[3] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. cite arxiv:1912.01703Comment: 12 pages, 3 figures, NeurIPS 2019.

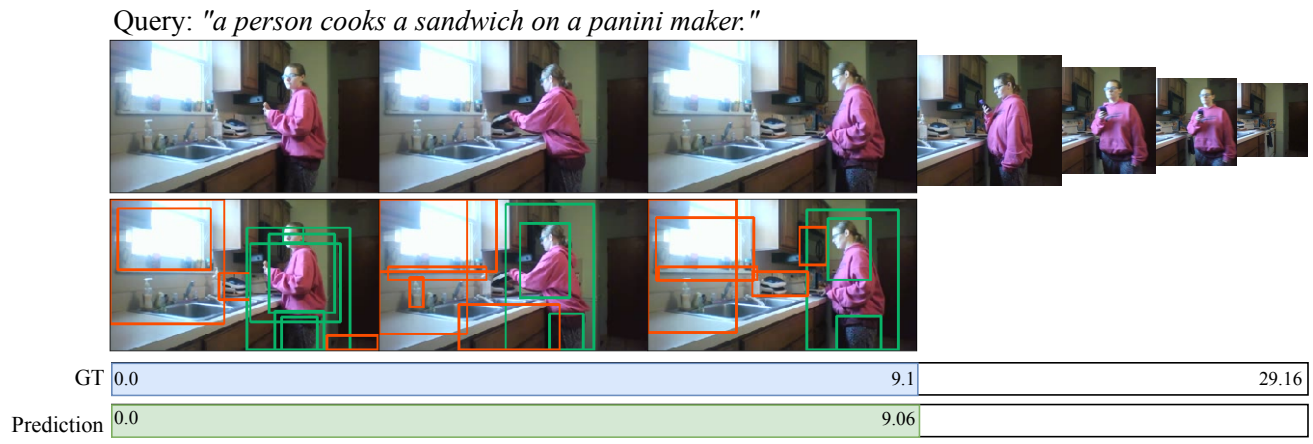

Figure 4. Example of success 1.

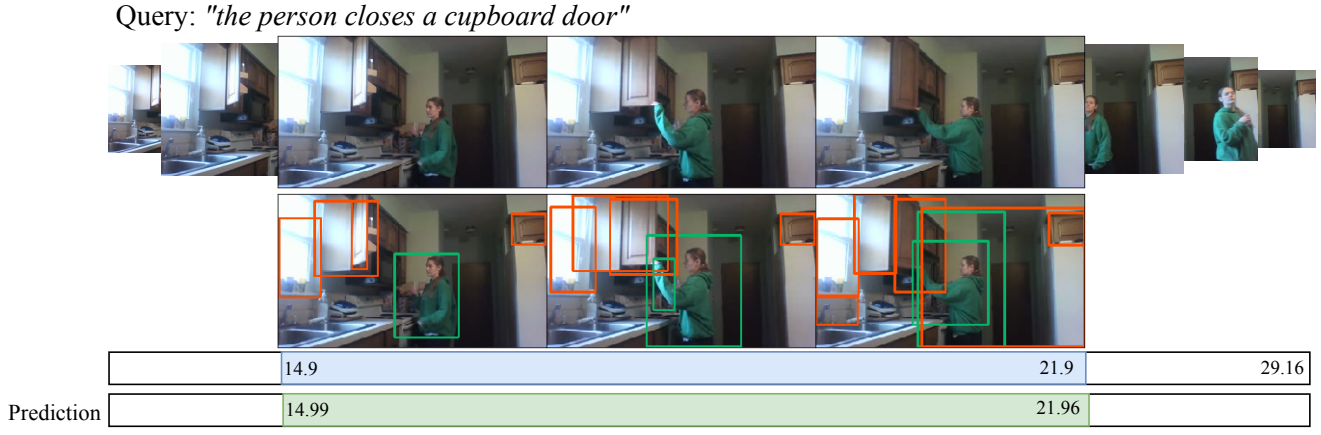

Figure 5. Example of success 2.

Figure 6. Success examples of our method on Charades-STA dataset.

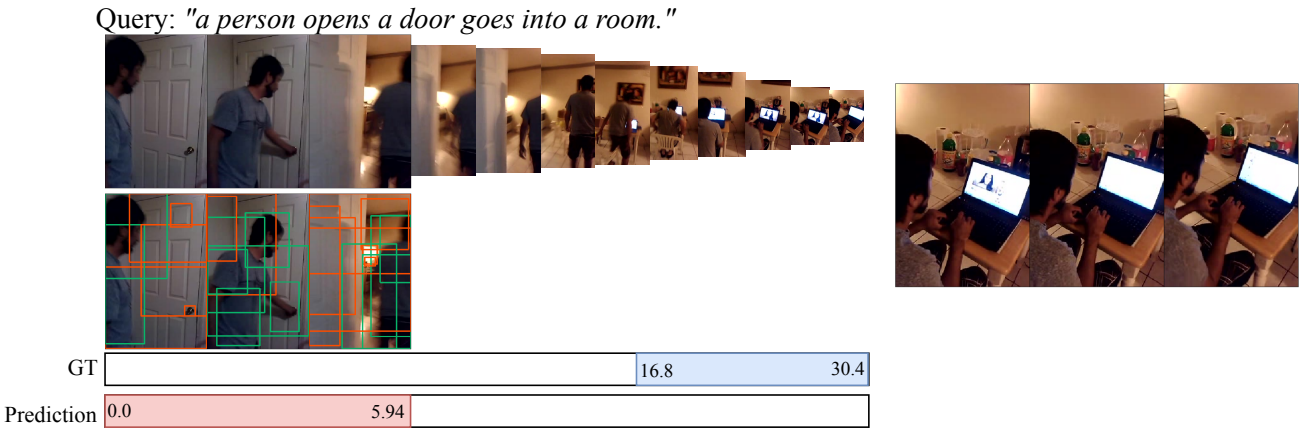

Figure 7. Example of failure 1.

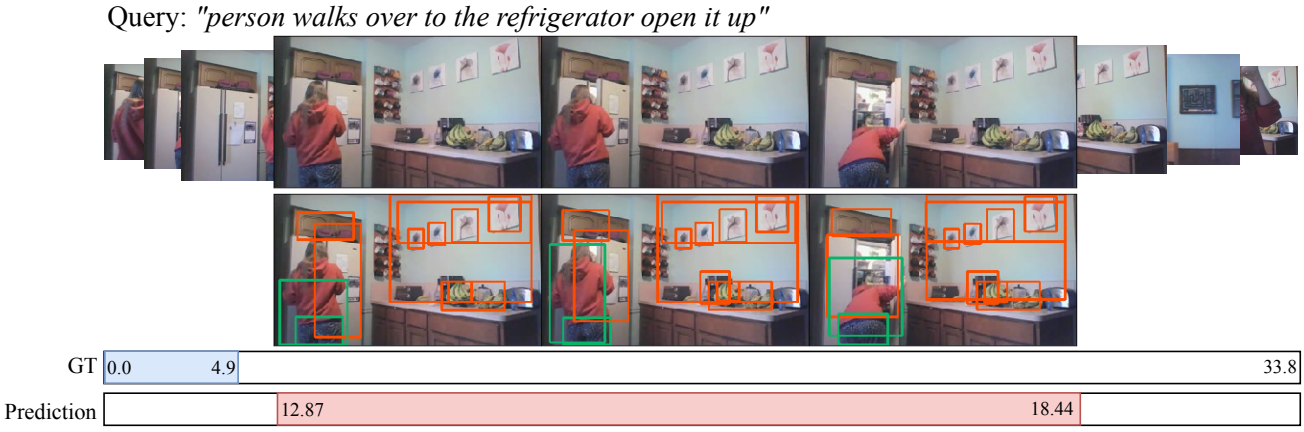

Figure 8. Example of failure 2.

Figure 9. Failure examples of our method on Charades-STA.

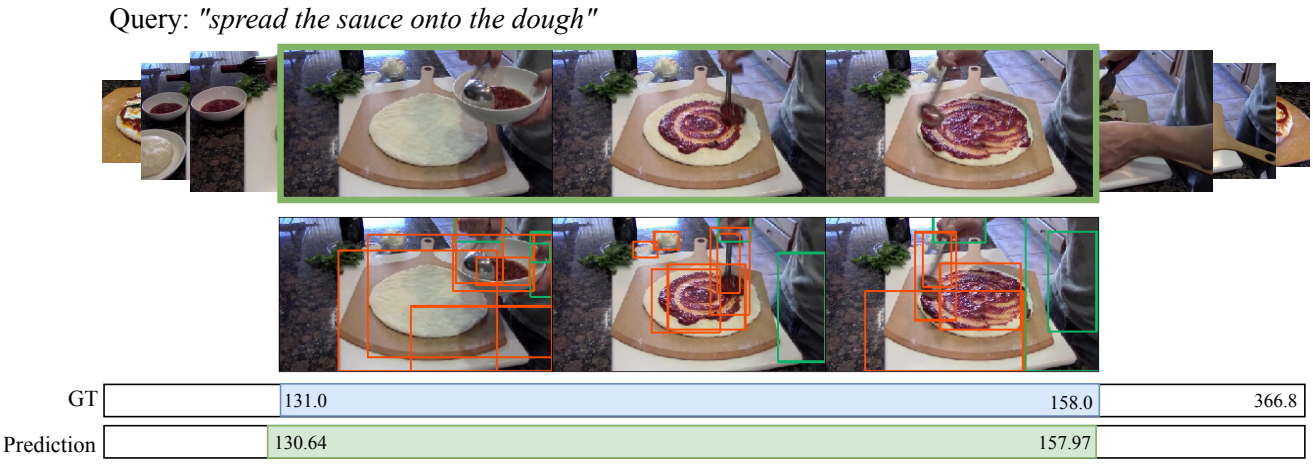

Figure 10. Success example 1.

Figure 11. Success example 2.

Figure 12. Success examples of our method in the YouCookII dataset.

Query: *"pour the dressing over the salad and mix"*

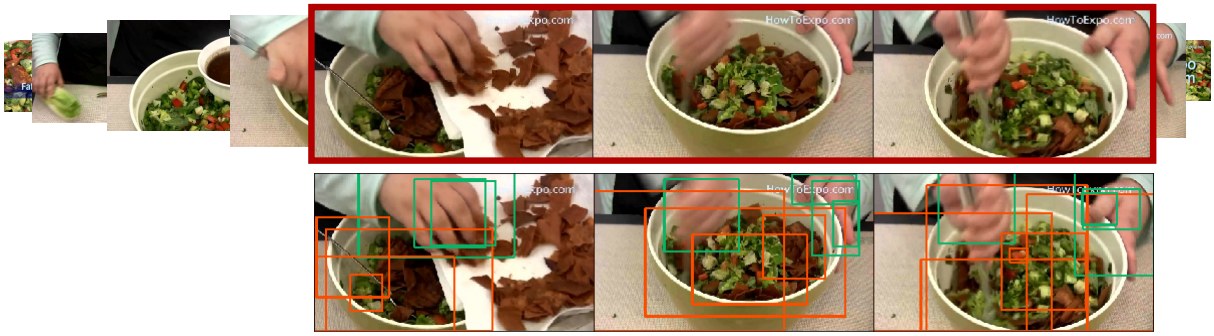

Figure 13. Failure case 1.

Query: *"add oil ginger and garlic to a pot"*

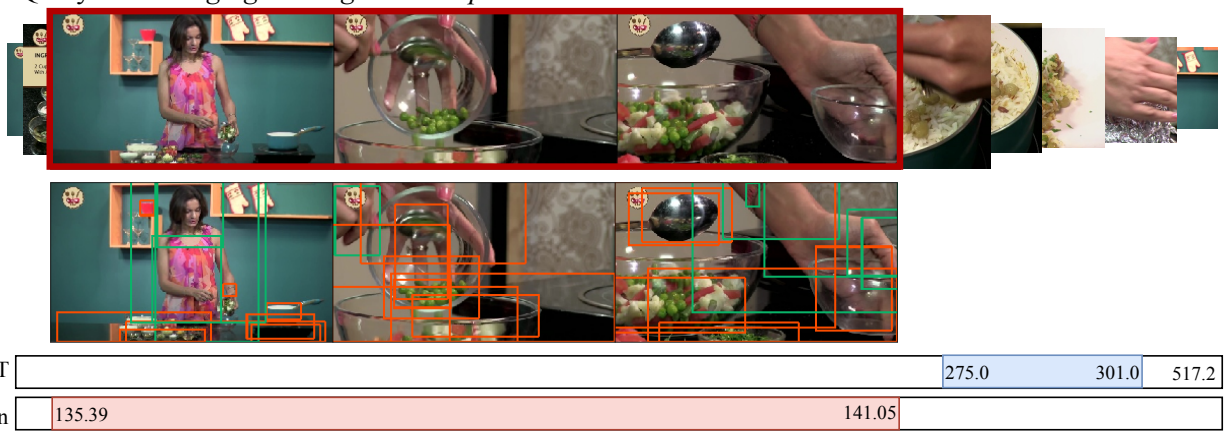

Figure 14. Failure case 2.

Figure 15. Failure cases of our method in the YouCookII dataset.

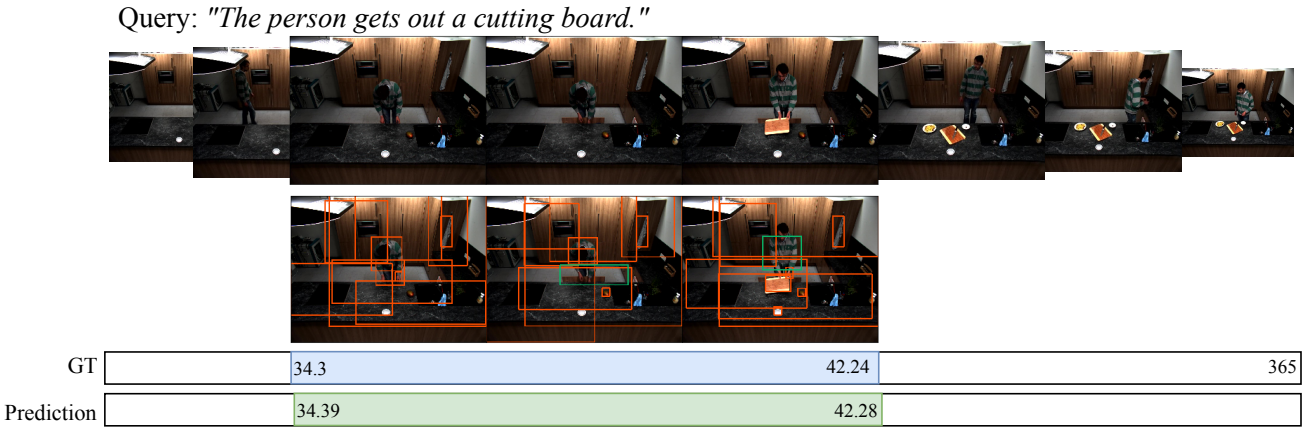

Figure 16. Success example 1.

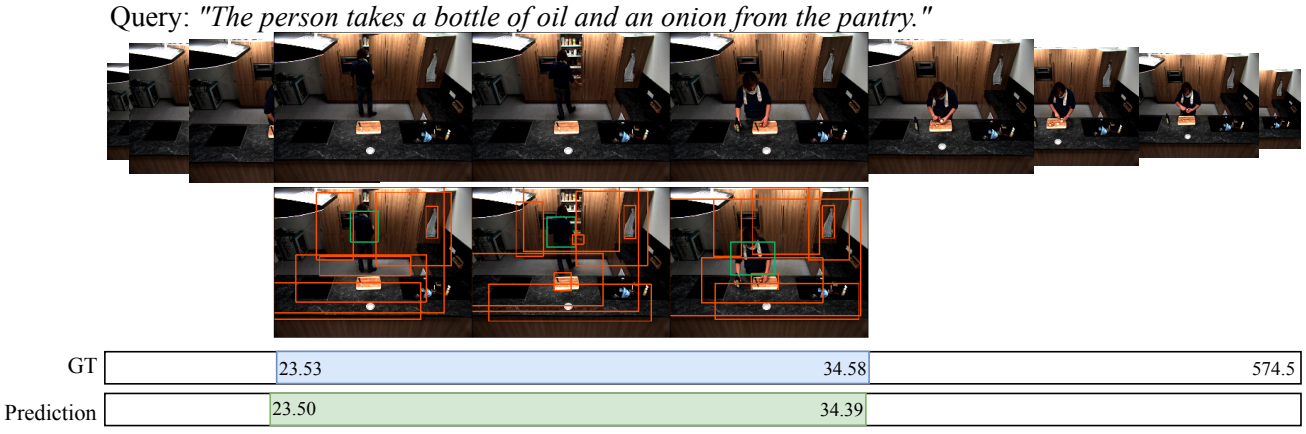

Figure 17. Success example 2.

Figure 18. Success examples of our method in the TACoS dataset.

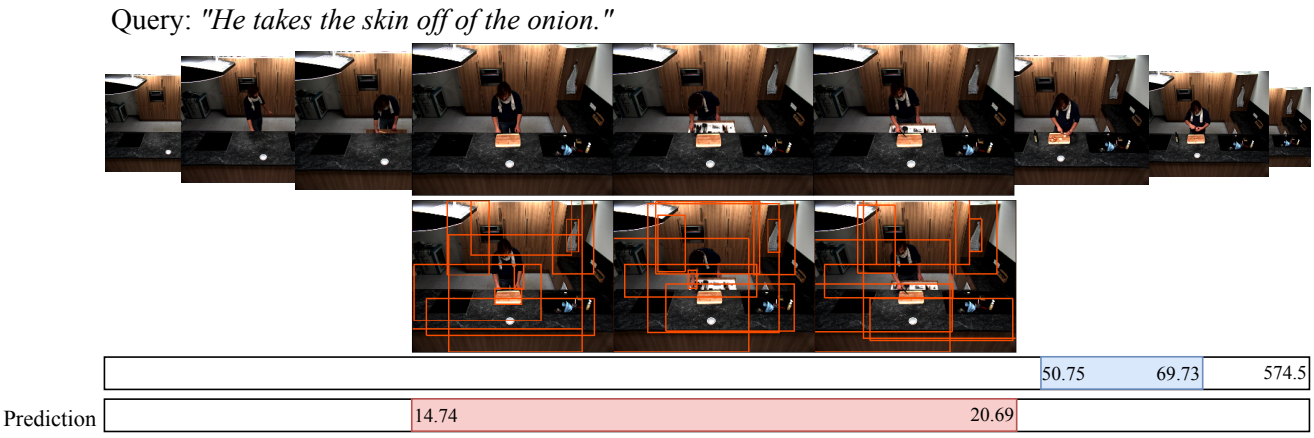

Figure 19. Failure case 1.

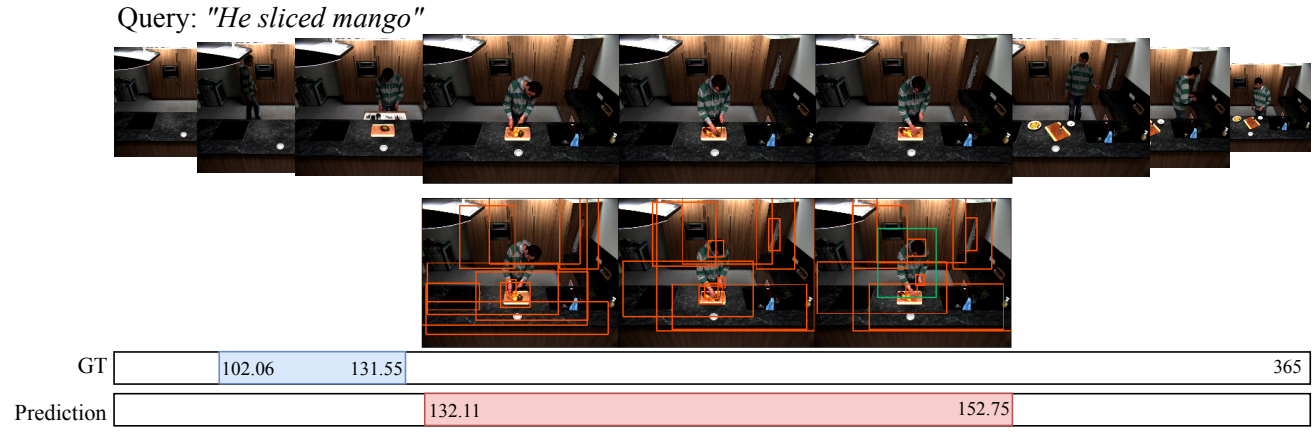

Figure 20. Failure case 2.

Figure 21. Failure cases of our method in the TACoS dataset.