Env-QA: A Video Question Answering Benchmark for Comprehensive Understanding of Dynamic Environments - Supplementary Material

Difei Gao[1,2], Ruiping Wang[1,2,3], Ziyi Bai[1,2], Xilin Chen[1,2]
[1]Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, 100190, China
[2]University of Chinese Academy of Sciences, Beijing, 100049, China
[3]Beijing Academy of Artificial Intelligence, Beijing, 100084, China
{difei.gao, ziyi.bai}@vipl.ict.ac.cn, {wangruiping, xlchen}@ict.ac.cn

Overview

In the supplementary material, we provide discussion with embodied AI tasks (in Sec. A), display more details of Env-QA dataset statistics (in Sec. B), dataset construction method (in Sec. C), proposed method Temporal Segmentation and Event Attention network (TSEA) (in Sec. D), and the experiment results (in Sec. E).

A. Discussion with Embodied AI tasks

Why Env-QA and Embodied AI tasks evaluate different capabilities? The planning task, the core of embodied AI, e.g., IQA [1], ALFRED [8], is usually formulated as Markov Decision Process. That is, the action prediction only needs to restore the *current environment state* from historical observations. On the contrary, Env-QA requires the understanding of whole *trajectory of environment state changes*. For example, for counting questions in IQA, the model only needs to know the number of objects in a relatively stable environment (≈current, because the number won't change). But for Env-QA, the model should also know the changes of the number (e.g. when one object is broken). This difference allows Env-QA to additionally evaluate some crucial abilities that previous datasets don't cover, e.g. temporal reasoning with a long span.

Can one take the ALFRED trajectories and pose questions on top of them to get Env-QA? The videos recorded in ALFRED [8], can be a great supplement for Env-QA, but hard to be an alternative. ALFRED is oriented to realistic tasks, so the actions usually have a non-negligible bias which does not affect the evaluation of planning but has plagued VQA evaluation for a long time. Thus, the videos with well-controlled content in Env-QA are indispensable.

B. Dataset Statistics

In this section, we provide more statistics of the videos and question-answer pairs in Env-QA dataset.

B.1. Events in Videos

Interaction events with the environment are the core content of videos in Env-QA. The events in Env-QA are usually highly compositional, e.g., "move apple to plate" is composed of an action "move" and two objects participating in the action, "apple" and "plate". This compositionality makes Env-QA involve a large number of different events, a total of 2,402 unique events. The model needs to truly understand the details of the video to recognize the event correctly and perform further complex reasoning.

Event Distribution. In Figure 1, we show the distribution of top-50 frequent events in Env-QA dataset. Some pairs of related events are marked with the same color, e.g., "clean cloth" and "make cloth dirty". It can be seen that the most frequent events are mainly relatively simple actions, involving only one action and one object. This is because compositional events have more variants, e.g., "move apple to plate", "move apple to sink", so the number of each specific event is relatively small. The most frequent events usually involve objects that appear in multiple types of scenes, e.g., sink, bowl. It also can be seen that the event distribution in Env-QA is relatively uniform in general. Especially for some pairs of the related events, the frequencies of events are almost the same. Note that there are some exceptions that the frequencies of some related events are less even, e.g., "turn on the stove burner", and "turn off the stove burner" (the event "turn off the stove burner" has a total of 128 occurrences, and is not displayed in Figure 1 that only shows top-50 frequent events). This is because these events often appear in comprehensive task type of videos (human

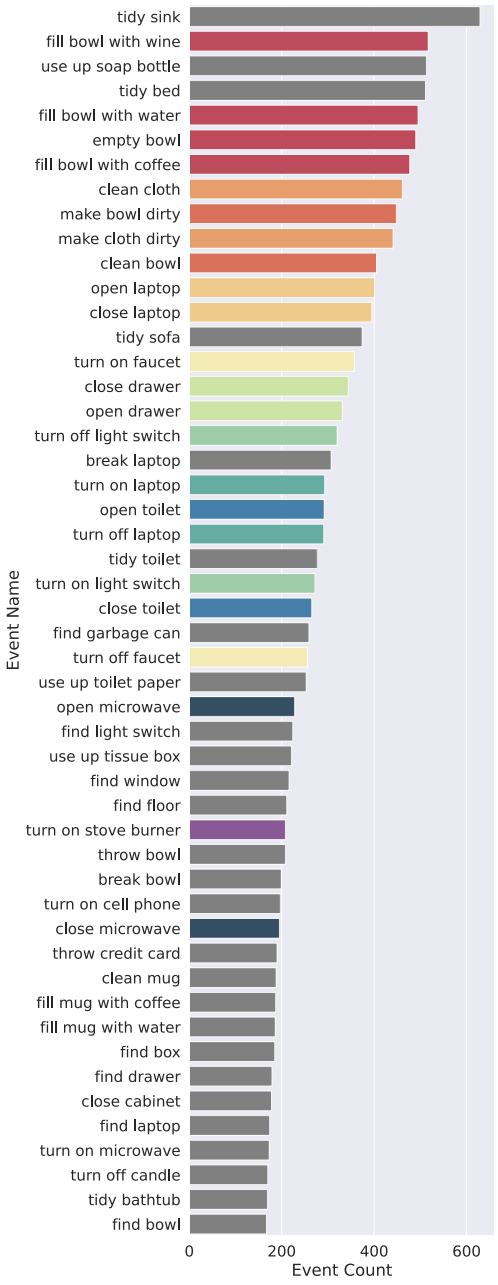


Figure 1. Distribution of the top-50 frequent events in Env-QA dataset. Some pairs of related events are marked with the same color, e.g., "clean cloth" and "make cloth dirty". The event distribution in Env-QA is relatively uniform in general. Especially for some pairs of the related events, the frequencies of events are almost the same. It demonstrates our dataset construction method well controls the sample distribution.

life tasks). To avoid the content bias that some objects always are off (e.g., stove, faucet) at the end of videos, we intentionally keep some objects on to ensure that the distribution of **state** is even.

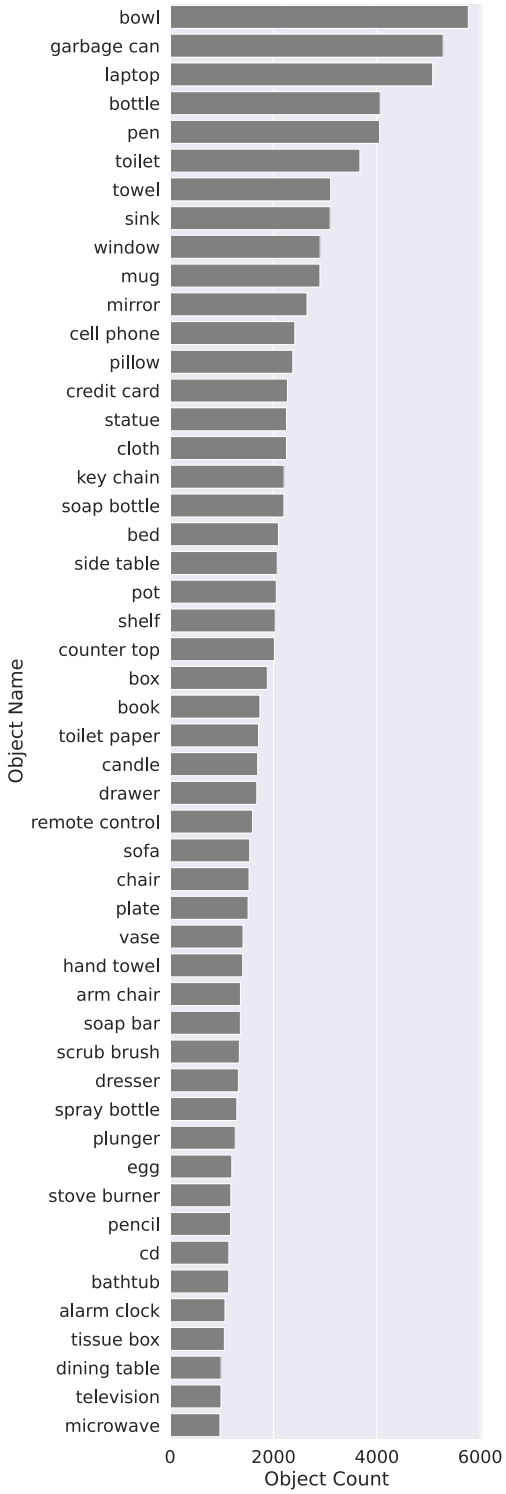


Figure 2. Distribution of the top-50 frequent objects involved in events of Env-QA dataset.

Object Distribution. In Figure 2, we display the distribution of top-50 frequent objects involved in the events. In

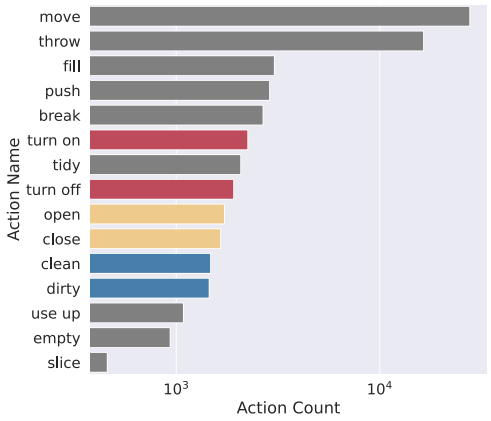


Figure 3. Distribution of the actions involved in events of Env-QA dataset. Some pairs of related actions are marked with the same color, e.g., "turn on" and "turn off".

general, the distribution of the manipulated objects is relatively even. The most frequent objects mainly appear in multiple types of environments (e.g., bowl, garbage can, laptop, and bottle) or support multiple types of actions (e.g., bowls can be moved, thrown, broken, filled with liquid, etc.).

Action Distribution. In Figure 3, we display the distribution of actions involved in the events. The frequency of the action is closely related to the number of objects supporting that action. For example, "move" and "throw" are the two most frequent actions because most interactable objects support these operations. It also can be seen that the frequencies of some related actions are quite similar, indicating that our designed instruction generation algorithm well controls the distribution of samples. Besides, we can see that the frequency of "turn on" is slightly higher than "turn off". This is because to let the state of objects be evenly distributed in the comprehensive tasks, we intentionally require annotators not to turn off some objects after using them, as mentioned above.

B.2. Questions and Answers

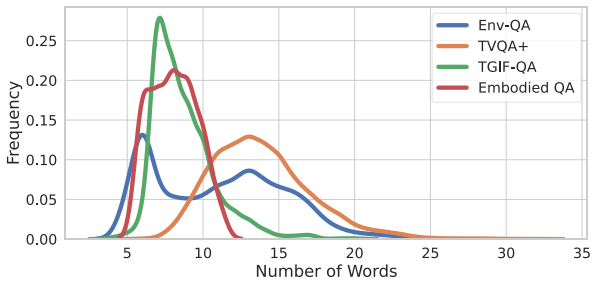


Figure 4. Question length distributions of Env-QA dataset and other related datasets.

Question Length Distribution. In Figure 4, we display the question length distributions of Env-QA and other related datasets. It can be seen that the Env-QA covers a wider range of question length compared to other datasets. The reason is that Env-QA evaluates the capabilities of the model from easy to difficult by introducing our semi-automatic dataset construction method. The shorter questions are for evaluating some simple abilities, like recognizing environment attributes, and longer questions are for evaluating long-time state tracking or multi-event temporal reasoning. The questions in TVQA+ are relatively longer than the questions in Env-QA because TVQA+ questions usually involve plots and subtitles as well as the descriptions of these content, and thus are relatively longer. Compared to Embodied QA, it can be seen that Env-QA contains more diverse questions to evaluate the visual understanding of environments.

Figure 5. Word cloud of answer words in Env-QA.

Answer Word Cloud. Due to the compositionality of events in videos, the answers about these events are also highly compositional. Env-QA involves a total of 3,705 unique answers. In Figure 5, we display the word cloud of the words in answers. It can be seen that these words are mainly about actions, objects, states, attributes, etc.

C. Dataset Construction

In this section, we provide more details of how we construct the dataset.

C.1. More Examples of Instructions

Auto-generated instructions are crucial for Env-QA dataset to control the distribution of video content. In Figure 6, we display one example for each video type in Env-QA. For each type of video, we design a sampling strategy to select events from all legal events in an environment. For random-type videos, the sampler prefers to select more diverse types of actions. For object-centric type videos, the sampler prefers to select diverse types of actions for the given objects. For action-centric type videos, the sampler selects some specific types of actions with selected objects. For comprehensive task videos, we pre-define several types of tasks, including washing objects, heating objects, boiling

Figure 6. Example instruction of each video type in Env-QA.

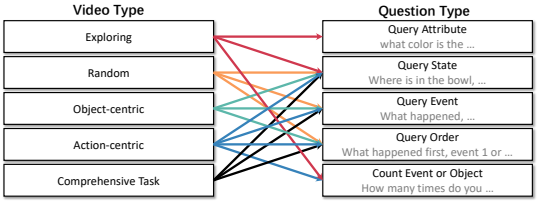


Figure 7. Correspondence between the video types and question types.

objects, and preparing foods. For each type of task, we propose several routine templates to accomplish the task. For example, to heat an object, one routine template is "Step 1: open <Object1>; Step 2: move <Object2> to <Object3>; Step 3: move <Object3> to <Object1>; ...", where the blanks in the template will be filled with satisfying objects, e.g., fill <Object1> with microwave, <Object2> with lettuce, and <Object3> with plate. This method allows us to generate a large number of different comprehensive task type of videos.

C.2. Correspondence Between Video Types and Question Types

For question generation, our designed algorithm outputs questions according to the video type and instructions of the video. In Figure 7, we show the correspondence between video types and question types. It can be seen that every type of video in Env-QA serves to evaluate some specific types of abilities.

C.3. Annotation Platform

The AI-THOR [3] API allows the researchers to control the agent to manipulate in the simulator by python code. It is a very efficient way to train models on it. However, to let annotators manipulate in the simulator, a well designed interaction method based on mouse and keyboard will be more convenient. Great thanks to the open-source Unity code of AI2-THOR, we build a web interface of AI2-THOR based on it to collect videos in Env-QA.

In Figure 8, we show the screenshot of our video collection interface. The simulator first randomly initializes the positions of the objects in the environment. Then, given a list of events, annotators need to adjust the states of the objects to make sure that all the events can be correctly performed. For example, if the given events require to close the fridge, the annotator should first open the fridge before recording. After adjusting the objects' state, the annotators are asked to perform specific actions step by step according to the given event list. We also add a check box for each event to mark the event which the annotator cannot complete, e.g., some objects are hard to be found in the given objects' position initialization. The annotation platform automatically sends the frames, corresponding instance segmentation maps, depth images, and environment metadata recorded by the annotator's AI2-THOR web simulator to our server. Some examples of collected images and corresponding annotations are shown in Figure 9.

D. Temporal Segmentation and Event Attention Networks

We illustrate more details of our proposed method, Temporal Segmentation and Event Attention networks (TSEA) in this section.

D.1. Event-Level Video Feature Extraction Module

Here, we provide more details about how to obtain the object features o_{ti}. The object features of each frame are obtained by feeding the raw frame into Faster R-CNN [7], then Temporal CNN [4], and finally appending the object name features and bounding box features. Specifically, we fine-tune the Faster R-CNN pre-trained on COCO dataset [5] with images selected from training split in Env-QA, where the object category and bounding box annotations are derived from the instance segmentation map of frames. Note that, due to the dimension of object features extracted by the original Faster R-CNN is high for further QA model processing long-time videos, we add an FC layer to reduce the object feature dimension to 300 while fine-tuning. The temporal CNN is the same as the one in [4], which uses layer normalization and CNNs for sequence modeling. For the object name feature, the predicted object label is first converted into natural language words. Then, we use GloVe [6] to encode each word of object name,

Figure 8. Screenshot of our web interface for video collection.

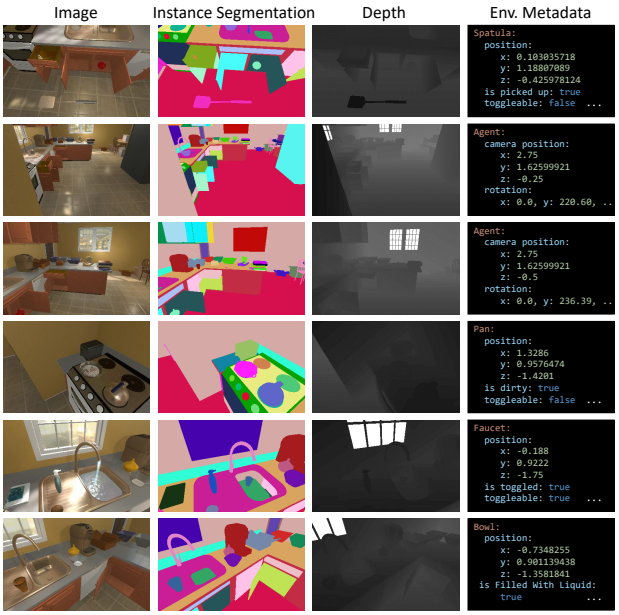


Figure 9. Some examples of collected images and corresponding annotations. Our annotation platform automatically records various types of annotations, including instance segmentation map, depth image, and environment metadata. The environment metadata saves the positions and states of the agent and objects.

then concatenate the GloVe embedding as the object name features. For bounding box features, we use the normalized bounding box coordinates, which are normalized into $[1, -1]$ as the bounding box features.

D.2. Multi-step Temporal Attention Mechanism

This section provides more details of the multi-step temporal attention mechanism for locating the key events. To extract text features, the module first uses GloVe with GRU to get the features of each word $\boldsymbol{w}_l$, where $l \in \{1, ..., L\}$ and L indicates the number of words in the question. Besides, because a question often involves multiple events, this module performs a two-step self-attention mechanism [2] to obtain the features of multiple parts of the question, $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$. Specifically, the $\boldsymbol{q}_1$ is obtained as follows:

$$\beta_l = \underset{l}{\text{Softmax}}(\boldsymbol{W}_1(\boldsymbol{w}_l \odot (\boldsymbol{W}_2 \text{ReLU}(\boldsymbol{W}_3 \boldsymbol{q})))) \quad (1)$$

$$\boldsymbol{q}_1 = \sum_{l=1}^{L} \beta_l \boldsymbol{w}_l, \quad (2)$$

where $\boldsymbol{W}_1$, $\boldsymbol{W}_2$, and $\boldsymbol{W}_3$ are trainable parameters, $\odot$ indicates element-wise multiplication. The module of calculating the $\boldsymbol{q}_2$ is the same, but with different parameters.

Then, we perform a soft attention mechanism to locate the related events based on the $\boldsymbol{q}_1$, $\boldsymbol{q}_2$ and event features $\boldsymbol{e}_m$, where the features of attended events are denoted as $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$. The attended event feature $\boldsymbol{h}_1$ is obtained as follows:

$$a_m = \underset{m}{\text{Softmax}}(\boldsymbol{W}_4(\boldsymbol{W}_5 \boldsymbol{q}_1 \odot \boldsymbol{W}_6 \boldsymbol{e}_m)) \quad (3)$$

$$\boldsymbol{h}_1 = \sum_{m=1}^{M} a_m \boldsymbol{e}_m, \quad (4)$$

where $\boldsymbol{W}_4$, $\boldsymbol{W}_5$, and $\boldsymbol{W}_6$ are trainable parameters, $\boldsymbol{W}_5$ and $\boldsymbol{W}_6$ aim to embed $\boldsymbol{q}_1$ and $\boldsymbol{e}_m$ into the same dimension. The module of calculating the $\boldsymbol{h}_2$ is the same, but with different parameters. Finally, we concatenate the $\boldsymbol{h}_1$, $\boldsymbol{h}_2$, and full question feature $\boldsymbol{q}$, and perform the attention mechanism to locate the final events most related to answer the question, formulated as:

$$a'_m = \underset{m}{\text{Softmax}}(\boldsymbol{W}_7(\boldsymbol{W}_8[\boldsymbol{h}_1; \boldsymbol{h}_2; \boldsymbol{q}] \odot \boldsymbol{W}_9 \boldsymbol{e}_m)) \quad (5)$$

$$\boldsymbol{h}_v = \sum_{m=1}^{M} a'_m \boldsymbol{e}_m, \quad (6)$$

where $\boldsymbol{W}_7$, $\boldsymbol{W}_8$, and $\boldsymbol{W}_9$ are trainable parameters, and $[;]$ indicates concatenating operation.

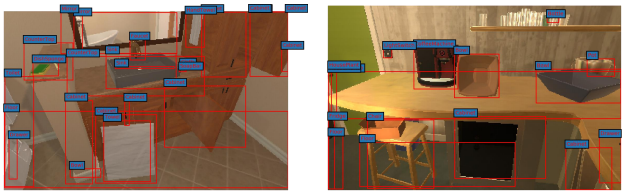


Figure 10. Example predictions from Faster R-CNN fine-tuned on Env-QA. Each bounding box is labeled with an object class.

E. Experiments

E.1. Implementation Details

In this section, we provide more details of the baseline methods. For the methods that use the question features, we use GloVe word embedding with 300 dimensions to encode the words in a question. For all models that use object features, we use our fine-tuned Faster R-CNN (as illustrated in Sec. D.1) to extract the top-30 object proposals. In Figure 10, we display the detection results of our fine-tuned Faster R-CNN.

E.2. Qualitative Results

In Figure 11 and Figure 12, we show more qualitative results of the TSEA, including the predicted role-value answers, temporal segmentation results, and event attention results. We display some key segments in the videos outputted by our temporal segmentation method and show five key frames for each event. It can be seen that our segmentation algorithm can effectively divide events according to the content of the video. When the object is far away from the camera or is picked up (observe the object from an uncommon view), the object could be misclassified, and the corresponding temporal segmentation result might make some mistakes, e.g., Q1, Q3, and Q4 in Figure 11. For question answering, it can be seen that TSEA can attend on the key events asked by the questions when the visual appearance differences between events are relatively obvious, e.g., Q1, Q2 in Figure 11. For tracking the state of an object for a long-time, the model needs to distinguish many similar events, e.g., Q3 in Figure 11, and remember the state changes caused by these events. This is still very challenging for TSEA. The query order questions are also difficult, e.g., Q5 in Figure 12, because they require to correctly locate two events and perform a multi-event temporal reasoning.

References

[1] Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. Iqa: Visual question answering in interactive environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1

[2] Ronghang Hu, Anna Rohrbach, Trevor Darrell, and Kate Saenko. Language-conditioned graph networks for relational reasoning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 10294–10303, 2019. 5

[3] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017. 4

[4] Jie Lei, Licheng Yu, Tamara Berg, and Mohit Bansal. Tvqa+: Spatio-temporal grounding for video question answering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8211–8225, 2020. 4

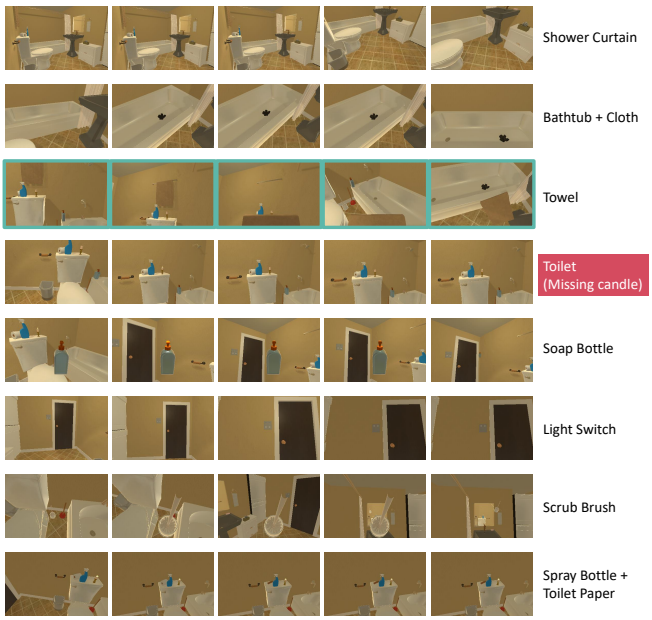
[5] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014. 4

[6] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. 4

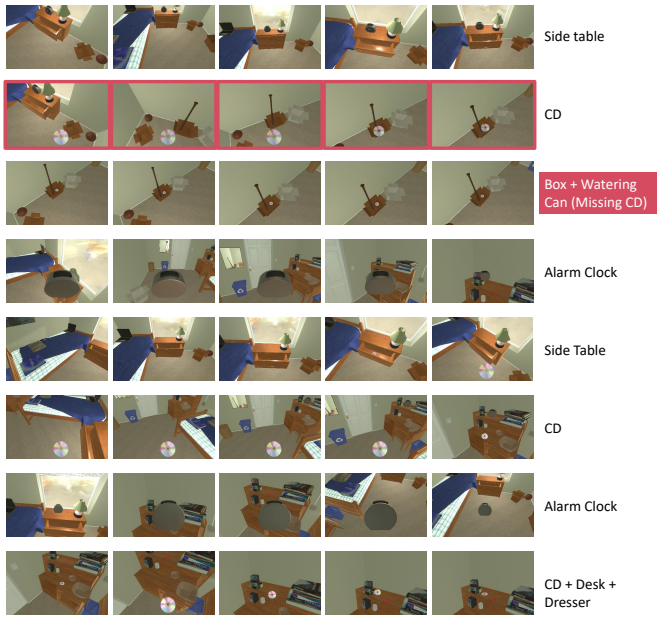
[7] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances In Neural Information Processing Systems (NeurIPS)*, pages 91–99, 2015. 4

[8] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10740–10749, 2020. 1

Q1: What place is the wine bottle moved to, after moving pot to counter top and before putting egg near garbage can?
GT: in the sink
Predicted role-value: Prep. → in, Object1 → sink



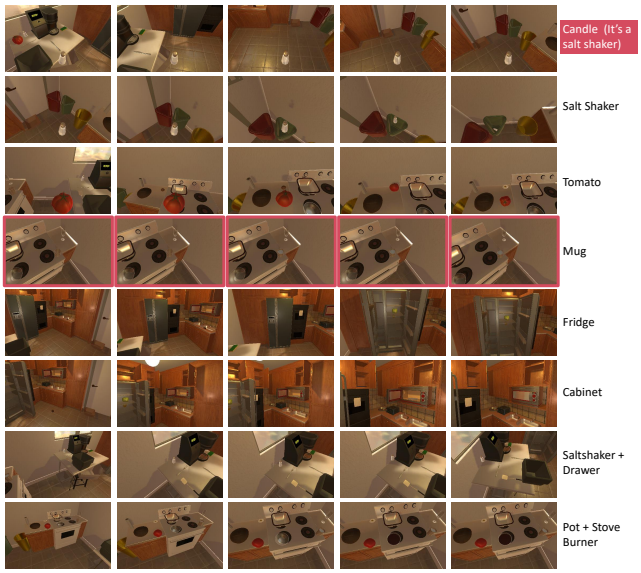
Q2: What happened, after making cloth dirty and before turning on candle?
GT: putting towel near bathtub **Predicted Role-Value:**
Action → put, Object1 → towel, Prep. → near, Object2 → bathtub



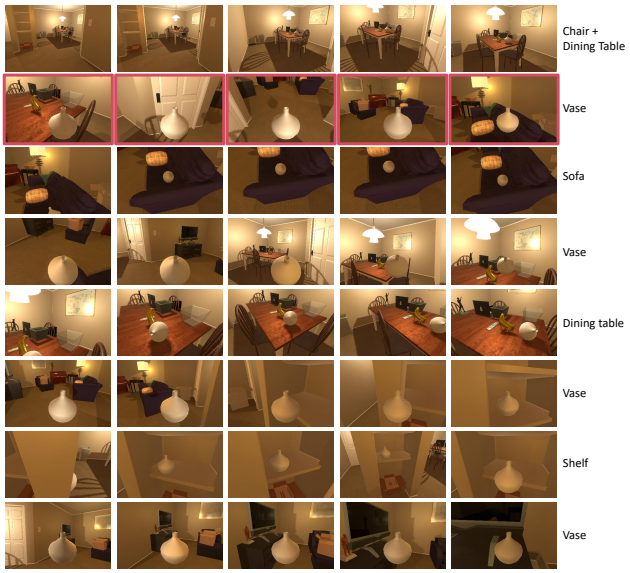
Q3: Where is the cd, at the beginning of the video?
GT: in the drawer
Predicted Role-Value: Prep. → in, Object1 → box

Q4: What object is in the garbage can, before breaking laptop?
GT: watch
Predicted Role-Value: Object1 → watch

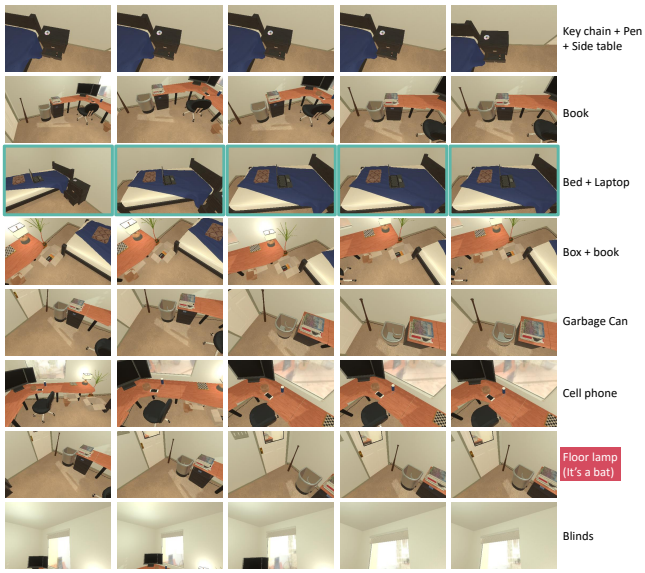
Figure 11. Example predictions from TSEA. We display key events in videos outputted by our temporal segmentation algorithm, questions, ground truth answers (denoted as **GT**), predicated role-value answers, and final attended events (marked with bounding boxes). In each row of one video sample, we display five key frames and objects that appear in each frame of the event. The class names shown with red background (e.g. Mug, Toilet) indicates that there are some mistakes in temporal segmentation.



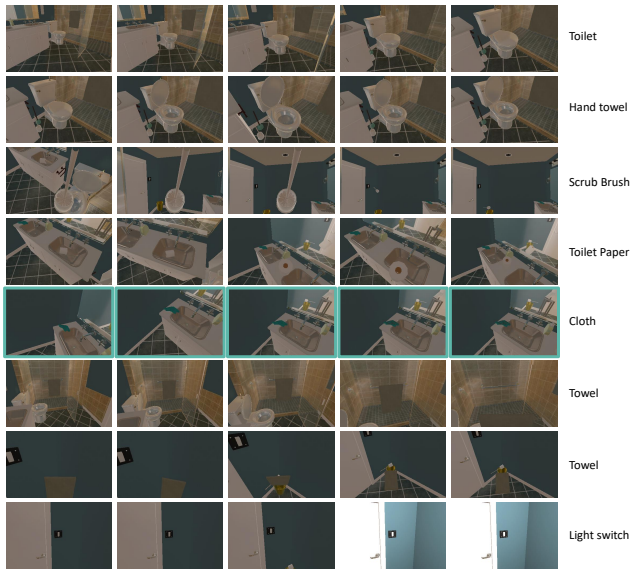
Q5: Which thing happened later, opening fridge or filling mug with water?
GT: opening fridge
Predicted Role-Value: Action → **open**, Object1 → **fridge**



Q6: How many objects were moved to garbage can, before moving vase to dining table?
GT: 0
Predicted Role-Value: Number → **1**



Q7: Where is the laptop?
GT: on the bed
Predicted Role-Value: Prep. → **on**, Object1 → **bed**



Q7: Is the cloth clean or dirty, at the beginning of the video?
GT: clean
Predicted Role-Value: Adj. → **dirty**

Figure 12. Example predictions from TSEA. We display key events in videos outputted by our temporal segmentation algorithm, questions, ground truth answers (denoted as **GT**), predicated role-value answers, and final attended events (marked with bounding boxes). In each row of one video sample, we display five key frames and the objects that appear in each frame of the event. The class names shown with red background (e.g. Candle, Floor lamp) indicates that there are some mistakes in temporal segmentation.