VIDEOTREE: Adaptive Tree-based Video Representation for LLM Reasoning on Long Videos

Supplementary Material

In this supplementary materials, we present the following: limitations (Sec. 7), additional quantitative results (Sec. 8), additional ablation study for VIDEOTREE framework (Sec. 9), the detailed algorithm for VIDEOTREE (Sec. 10), additional implementation details (Sec. 11), additional qualitative analysis (Sec. 12).

7. Limitations

Like all LLM-based video-reasoning systems (including dense sampling) our method is limited by the ability of the captioner to accurately capture the contents of sampled frames. However, our method's modular nature means that as captioners improve, we can easily include them into the VIDEOTREE framework; similarly, we can use increasingly strong LLMs as the reasoning backbone of VIDEOTREE. While VIDEOTREE is training-free, it includes a small number of hyperparameters. In Sec. 9, we ablate these hyperparameters, showing that VIDEOTREE outperforms the uniform-sampling baseline regardless of the choice of max depth and branch width. Thus, while better hyperparameters can benefit the method, even with sub-optimal settings VIDEOTREE outperforms the uniform baseline.

8. Additional Quantitative Results

8.1. Comparison with advanced VideoLLMs on EgoSchema and NExT-QA.

In Tab. 6, we compare VIDEOTREE with advanced VideoLLMs [5, 23, 55, 62, 68, 70] on EgoSchema and NExT-QA benchmarks. Without any video-specific training, VIDEOTREE gets comparable performance on EgoSchema fullset and slightly worse results on NExT-QA results, comparison with the methods was trained on large-scale video data and massive GPU hours.

8.2. Additional evaluation benchmarks.

IntentQA Results. We report the IntentQA [24] results of VIDEOTREE and compare with existing methods. We first introduce IntentQA [24], which contains 4,303 videos and 16K multiple-choice question-answer pairs focused on reasoning about people's intent in the video. We perform a zero-shot evaluation on the test set containing 2K questions. The videos are more than 44s in average length. We compare our methods with both training-free [19, 20, 86, 89] and fine-tuned baseline [65, 82].

As shown in Tab. 7, our training-free VIDEOTREE approach achieves 66.9% zero-shot accuracy on the test set, surpassing the existing training-free approaches LLoVi [89] with 2.7% improvements and even closing the gap with finetuned method Vamos [65]. This result shows that VIDEOTREE improves performance in answering questions about intent, which is challenging since intent understanding [24] requires the model to understand the various video contexts, including the immediate communicative context, the shared experience, and the commonsense.

Method	ES	NExT-QA
InternVideo	32.1	-
Tarsier-34B	61.7	79.2
VideoChat2	60.2	61.7
VideoLLaMA 2	63.9	-
LLaVA-OV-72B	62.0	-
LongVU	67.6	-
Training-free Methods		
VIDEOTREE (ours)	61.1	75.6

Table 6. Comparison with advanced VideoLLMs on EgoSchema and NExT-QA benchmarks.

Method	Accuracy
Fine-tuned Method	
VGT	51.3
Vamos	68.5
Training-free Methods	
LangRepo	59.1
SeViLA	60.9
LLoVi	64.0
IG-VLM	64.2
VIDEOTREE (ours)	66.9

Table 7. IntentQA Results

Video-MME Short and Medium Split Results. In Tab. 8, we test VIDEOTREE on the short and medium splits of Video-MME benchmark as well. We apply the recent LLaVA-OV-7B model [23] as the frame captioner. Specifically, VIDEOTREE achieves 67.8% and 59.9% on Video-MME short and medium split, a more than 5.7% and 6.7% improvement compared to LLoVi [89] and LongVA [91].

Model	V-MME Short/Med	MLVU-Avg
LLoVi	62.1/53.2	55.1
LongVA	61.4/50.9	56.3
VideoTree	**67.8/59.9**	**60.4**

Table 8. Results on Video-MME (short and medium splits) and MLVU with LlaVA-OV-7B as captioner.

Caption Number	Avg LLM Calls	ES Subset Acc
	VideoAgent	
6.4	10.2	58.4
8.4	10.2	60.2
11.0	9.0	57.4
	VideoTree *(ours)*	
7.1	**2.3**	**61.0**
9.7	**2.5**	**61.6**
11.3	**2.8**	**62.2**

Table 9. The comparison of average LLM calls for VideoTree and VideoAgent [67] (estimated) under similar frame settings on EgoSchema subset. Results show that VideoTree achieves better results on much less LLM calls.

MLVU Results. In Tab. 8, we test VideoTree on the MLVU validation set. We again use LLaVA-OV-7B [23] as the frame captioner. Results show that VideoTree achieves a 5.3% and 4.1% gain over LLoVi [89] and LongVA [91], respectively.

9. Additional Ablation Study

In this section, we report additional ablation studies for VideoTree framework. First, we ablate the LLM calls and vision encoder size for our method. Then, we show the effect of the different hyperparameter settings for VideoTree. Finally, we analyze the effect of different VLM/LLM designs for VideoTree.

LLM Calls. In Tab. 9, we report the number of average LLM calls of VideoTree and compare with VideoAgent [67].VideoTree achieves better results on much less LLM calls under similar caption numbers (only about 30% LLM calls are needed). This is due to the adaptive and hierarchical structure of VideoTree which could extracts more keyframes faster instead of searching one frame a time. This results highlight the advantage of the hierarchical nature of VideoTree in both efficiency and effectiveness, comparing to the non-hierarchical approaches.

Visual Encoder. In Tab. 10, we study the effect of the visual encoder used in the visual clustering operation. We report the results of VideoTree on three different scales of visual encoder: OpenCLIP-B, OpenCLIP-G [14] and EVA-CLIP-8B [57] and compare to VideoAgent [67] [2]. VideoTree outperforms VideoAgent by an average of 6.9% across both encoders. Comparing different visual encoders ranging from 88M to 8B parameters, we see only a marginal drop in performance for VideoTree as the visual encoders decrease in size, indicating that our approach generalizes well to much smaller vision encoders (i.e. only a 0.2% drop when going from 8B to 88M), making the model more efficient while maintaining strong performance. Additionally, we test the performance of a self-supervised vision encoder for VideoTree. Specifically, we apply DINOv2-base [48] to VideoTree and get 64.2% on EgoSchema subset, which is 1.8% lower than the same size of CLIP model.

Visual Encoder	Params	Method	Acc.
OpenCLIP-ViT-B	88M	VideoAgent	–
		VideoTree	66.0
OpenCLIP-ViT-G	1B	VideoAgent	59.2
		VideoTree	**66.2**
EVA-CLIP-8B	8B	VideoAgent	59.4
		VideoTree	**66.2**

Table 10. Testing different visual encoder design choices in VideoTree. We also compare with VideoAgent[67] to show the effectiveness of our method.

Branch Width	ES Acc↑	#Frame↓
2	64.4	43.5
3	65.0	54.6
4	**66.2**	62.4
5	64.2	72.5
Uniform Baseline	61.2	180

Table 11. The effect of different settings for branch width of VideoTree. When the branch width is set to 4, VideoTree achieves the best performance on the EgoSchema subset. Reducing the branch width makes the model more efficient while retaining performance, outperforming all existing approaches.

Hyperparameter Analysis. In Tab. 11, we study the effect of the branch width of the tree-based representation for the VideoTree. The best performance is obtained when the branch width is set to 4. As with depth, excessive branch width reduces the VideoTree performance due to the information overwhelming to the LLM; however, even with the

[2]Note that VideoAgent only report results on OpenCLIP-ViT-G (1B) and EVA-CLIP-8B.

worst branch width setting, VIDEOTREE still outperforms the baseline.

In Tab. 12, we study the effect of the max breadth of the adaptive tree-based representation. The results indicate that even with a smaller max tree breadth, VIDEOTREE achieves good performance while using much fewer frames. Increasing the breadth generally increases performance, with the best performance when the max breadth is set to 32. However, having an excessive max breadth leads to worse results, suggesting that incorporating too much information in the adaptive tree-based representation limits the LLM reasoning ability. This links back to the intuition of having an efficient representation for the LLM reasoning over long videos.

In Tab. 13, we study the effect of the threshold on the number of highly relevant clusters, which controls the iterative process of the adaptive breadth expansion process. The best performance is obtained when the branch threshold is set to 4. Reducing the threshold improves the efficiency while retaining strong performance compared to the baseline results.

Max Breadth	ES Acc	#Frame
8	63.0	26.9
16	64.0	49.0
32	**66.2**	62.4
64	63.2	94.6
Uniform Baseline	61.2	180

Table 12. The effect of different settings for the max breadth of the first level of the tree. Results show that when the max breadth is set to 32, VIDEOTREE obtains the best performance. Reducing the max breadth improves efficiency while retaining performance.

Threshold	ES Acc	#Frame
2	63.6	13.9
3	64.4	32.2
4	**66.2**	62.4
5	64.8	79.2
Uniform Baseline	61.2	180

Table 13. The effect of different settings for the threshold on the number of highly relevant clusters. Results show that when the threshold is set to 4, VIDEOTREE obtains the best performance. Reducing the threshold improves efficiency while retaining performance.

VLM Captioner. In Tab. 14, we compares the performance of the best captioner (LaViLA for EgoSchema and CogAgent for NExT-QA) with using a LLaVA-1.6-7B [36] captioner everywhere. We observe a comparable performance on NExT-QA compared with the best captioner, while still outperforming all other existing methods in Sec. 5.1. We also observe a drop in performance on the EgoSchema subset while using LLaVA-1.6 captioner, this is likely due to a lack of egocentric data during LLaVA training, which is needed for strong performance on EgoSchema. In the future, we would like to see strong unified captioner that operate well across datasets; these would fit seamlessly into the VIDEOTREE framework, further boosting the performance. Additionally, we test the performance of the question-prompted captioner by adding the video query into the prompt of the LLaVA-1.6 captioner. The results show that on EgoSchema subset, VIDEOTREE with a question-prompted captioner achieves 63.2% accuracy, 3.0% lower than the direct captioner.

Captioner	EgoSchema Sub	NExT-QA
LLaVA-1.6-7B	59.2	73.6
Best Model	**66.2**	**75.6**

Table 14. Comparing accuracy with VIDEOTREE using the same captioner throughout (LLaVA1.6-7B) and best captioner for each benchmarks.

LLM Reasoner. We ablate the design choice of captioner and LLM for the VIDEOTREE framework in Tab. 15. With a better LLM, VIDEOTREE can achieve better performance on long video understanding tasks, indicating the potential VIDEOTREE to improve as its modules become more advanced. Notably, our GPT-3.5 variant substantially outperforms existing methods with the same LLM and standard QA prompts (VideoAgent [67] 48.8%, LLoVi [89] 51.8%), achieving 57.6% accuracy on EgoSchema subset.

Tree structure. We first note that our ablation (Tab. 5, **1.8%** improvements) highlights the importance of the tree structure to keyframe *selection*. To test its utility as a *representation* for the reasoner LLM, we conducted additional ablations, giving the reasoner a version of the tree's caption linearized in a top-down left-right traversal. On EgoSchema subset, this new version scores 64.8% while the temporal ordering one scores 66.2%; thus, while the tree is key to keyframe selection, the reasoner benefits from temporal order in video tasks.

10. Detailed Algorithm

In Algorithm 1, we present the algorithm behind VIDEOTREE.

Algorithm 1 VIDEOTREE

```
Require: Video frames $V$, query $Q$, number of clusters $k$, threshold for the number of high-relevance cluster
    $rele\_num\_thresh$, maximum number of clusters allowed $max\_breadth$, branch width $w$, visual encoder $E$, LLM
    $F_{llm}$, captioner $F_{vlm}$, cluster information $C$, relevance score $R$, tree-based video representation $Tree$
1: $k \leftarrow k\_init$
2: while $k \leq max\_breadth$ do                                   ▷ Adaptive breadth expansion
3:     $C \leftarrow$ VisualClustering$(E, V, k)$
4:     $Cap \leftarrow$ ClusterCaptioning$(F_{vlm}, V, C)$
5:     $R \leftarrow$ RelevanceScoring$(F_{llm}, C, Q, Cap)$
6:     if count$(r \in R \mid r = high) \geq rele\_num\_thresh$ then
7:         $Tree \leftarrow Tree.append(C)$                           ▷ First level of VIDEOTREE
8:         break
9:     end if
10:    $k \leftarrow k * 2$
11: end while
12: for $C_i \in C$ do                                              ▷ Relevance-guided depth expansion
13:    $\hat{C_i} \leftarrow$ DepthExpansion$(E, C_i, R_i, w)$
14:    $Tree \leftarrow Tree.append(\hat{C_i})$                       ▷ Adding hierarchy of VIDEOTREE
15: end for
16: $Cap \leftarrow$ GetCaptions$(F_{vlm}, V, Tree)$                  ▷ LLM Reasoning
17: $pred\_answer \leftarrow$ LLMReasoning$(F_{llm}, Cap, Q)$
18: return $pred\_answer$
```

Method	LLM	ES Acc
LLoVi	GPT-3.5	51.2
VideoAgent	GPT-3.5	48.8
VIDEOTREE (Ours)	GPT-3.5	**57.6**
LLoVi	GPT-4	61.2
VideoAgent	GPT-4	60.2
VIDEOTREE (Ours)	GPT-4	**66.2**

Table 15. The effect of different design choices of the LLM Reasoner for VIDEOTREE.

11. Additional Implementation Details

Additional VIDEOTREE Implementation Details. For clustering, we use the `kmeans_pytorch` library. The hyper-parameter setting for $max\_breadth$, max_depth, $branch\_width$ and $rele\_num\_thresh$ on the EgoSchema and Video-MME benchmark is 32, 3, 4 and 4 and for NExT-QA, we set the hyper-parameter as 8, 3, 2, and 3. The initial k depends on the average video length, and is set to 4 for NExT-QA and 8 for EgoSchema and VideoMME.

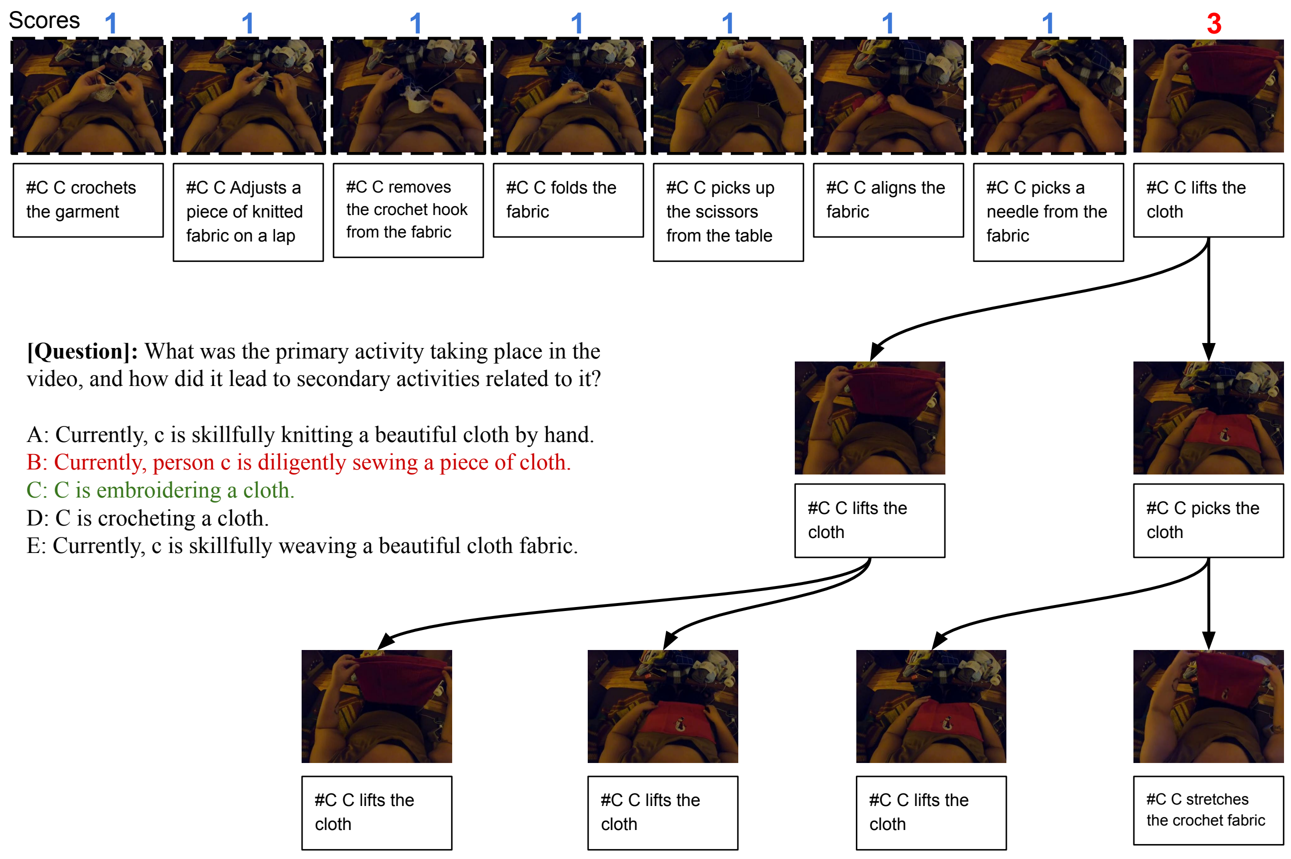
Lifelong Memory Reproduce Details. In Sec. 5.1, we report the main results of LifelongMemory [72] which is lower than the number than they reported in their paper. Here, we introduce our reproduce process in detail. For captions, since LifelongMemory authors do not provide the exact caption data/path, we directly utilize the same captioner from VIDEOTREE method and all other existing works (LaViLA) and extract the captions by 0.5FPS according to LifelongMemory paper. We then use their code to run the experiments on EgoSchema, however, the results are low and we observed a low success rate of the QA process (only about 80% success samples). We then update their output format/process code, which boost performance by about 10% and get the results in Sec. 5.1, but still lower than their paper results. Thus, for fair comparison, we directly reported the reproduced results.

Prompt Details. We provide detailed prompts for the relevance scoring prompt in Tab. 16 and LLM reasoning prompt in Tab. 17 on the EgoSchema benchmark.

Experiments Compute Resources. All experiments are conducted on 4 (or less) NVIDIA-A6000 GPU and Azure Cloud APIs (for OpenAI models). The minimal GPU memory requirement is 24GB.

12. Additional Qualitative Analysis

Additional Visualization. In Fig. 5 we show another visualization from VIDEOTREE. Here, VIDEOTREE localizes a single key activity (embroidering a cloth) taking place in the video and dynamically expands its constituent frames to answer the question correctly using a minimal number of frames. As shown in Figs. 4 and 5, the chosen keyframe distribution depends on the query: general queries yield sparser keyframes that extend to distant parts of the video. Queries with detailed actions/objects yield more concen-


Scores
1
1
1
1
1
1
1
3
#C C crochets the garment
#C C Adjusts a piece of knitted fabric on a lap
#C C removes the crochet hook from the fabric
#C C folds the fabric
#C C picks up the scissors from the table
#C C aligns the fabric
#C C picks a needle from the fabric
#C C lifts the cloth
#C C lifts the cloth
#C C picks the cloth
#C C lifts the cloth
#C C lifts the cloth
#C C lifts the cloth
#C C stretches the crochet fabric

[Question]: What was the primary activity taking place in the video, and how did it lead to secondary activities related to it?

A: Currently, c is skillfully knitting a beautiful cloth by hand.
B: Currently, person c is diligently sewing a piece of cloth.
C: C is embroidering a cloth.
D: C is crocheting a cloth.
E: Currently, c is skillfully weaving a beautiful cloth fabric.

Figure 5. Qualitative examples of VIDEOTREE keyframes and captions selection. Red options are answered wrongly with uniformly sampled frames. Green options are answered correctly by VIDEOTREE. Best viewed in color.

trated keyframes.

Failure Case. We provide the qualitative visualization of a failure case in Fig. 6. Here, we find that the failure was due to the following factors: A) The video had little scene change and multiple similar repeated actions (washing dishes). B) As a result, when VIDEOTREE expands down to more fine-grained details, the captioners give detailed description that contain some hallucinations. These captions miss the correct higher-level keyword (dish) in the selected captions. With stronger captioners, this failure case could potentially be resolved.

Human study. We conduct a human study on the accuracy of the keyframe scoring module to measure the quality of the LLM-based keyframe selection. Specifically, we ask a human annotator to judge the relevance of all 1st-level keyframes to the query, and we compare these decisions to the GPT4-based score used to evaluate this feature in VIDEOTREE. On 345 keyframes from 20 EgoSchema videos, results show that the GPT4-based keyframe scoring achieves 90.7% agreement with our human annotator, suggesting it captures human preferences well.

Table 16. **VideoTree with relevance scoring prompt on EgoSchema.**

User

You are presented with a textual description of a first view video clip, it consists of about `caption_number` frame captions sparsely sampled from the video (#C means the first person view, and #O indicates another). The ultimate goal is to answer a question related to this video, choosing the correct option out of five possible answers.

It is crucial that you imagine the visual scene as vividly as possible to enhance the accuracy of your response. After selecting your answer, rate your confidence level in this choice on a scale from 1 to 100, where 1 indicates low confidence and 100 signifies high confidence. Please provide a concise one-sentence explanation for your chosen answer. If you are uncertain about the correct option, select the one that seems closest to being correct. Meanwhile, could you provide a relevance score for each frame caption to evaluate their relevance with the query-answering process. The score is between 1,2,3, where 1 indicates low relevance and 3 signifies high relevance. Please return the relevance score in the format of a list of `caption_number` scores.

Examples: `Examples`

Description: `Captions`

Question: `Question`

Options: A: `Option-A`. B: `Option-B`. C: `Option-C`. D: `Option-D`. E: `Option-E`.

The prediction, explanation, confidence and frame relevance are (please response in the format of 'prediction:, explanation:, confidence:, frame relevance:')

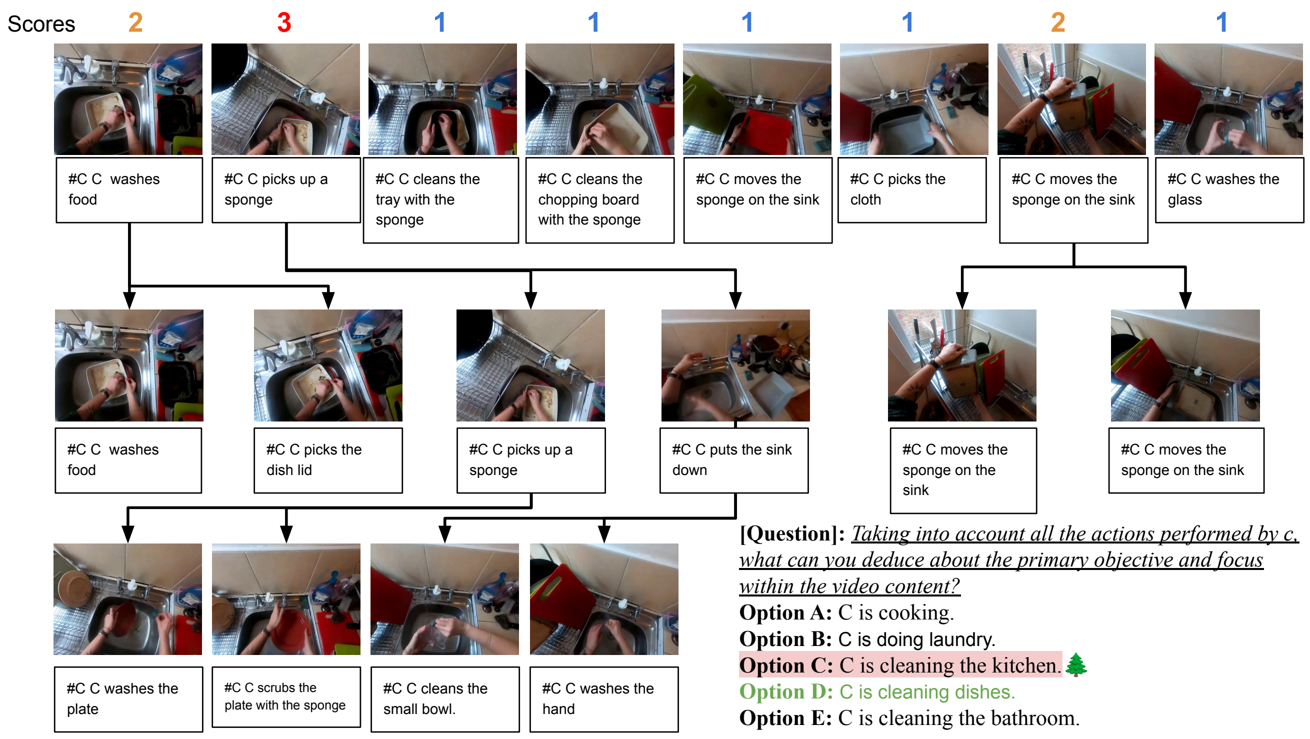
Assistant

prediction, explanation, confidence, frame relevance

Table 17. **VIDEOTREE with LLM reasoning prompt on EgoSchema.**

User
You are presented with a textual description of a first view video clip, it consists of frame captions sparsely sampled from the video (#C means the first person view, and #O indicates another). The ultimate goal is to answer a question related to this video, choosing the correct option out of five possible answers.
It is crucial that you imagine the visual scene as vividly as possible to enhance the accuracy of your response. After selecting your answer, rate your confidence level in this choice on a scale from 1 to 100, where 1 indicates low confidence and 100 signifies high confidence. Please provide a concise one-sentence explanation for your chosen answer. If you are uncertain about the correct option, select the one that seems closest to being correct.
Examples: `Examples`
Description: `Captions`
Question: `Question`
Options: A: `Option-A`. B: `Option-B`. C: `Option-C`. D: `Option-D`. E: `Option-E`.
The prediction, explanation, and confidence is (please response in the format of 'prediction:, explanation: ,confidence:')

Assistant
prediction, explanation, confidence



Scores
2
3
1
1
1
1
2
1
#C C washes food
#C C picks up a sponge
#C C cleans the tray with the sponge
#C C cleans the chopping board with the sponge
#C C moves the sponge on the sink
#C C picks the cloth
#C C moves the sponge on the sink
#C C washes the glass
#C C washes food
#C C picks the dish lid
#C C picks up a sponge
#C C puts the sink down
#C C moves the sponge on the sink
#C C moves the sponge on the sink
#C C washes the plate
#C C scrubs the plate with the sponge
#C C cleans the small bowl.
#C C washes the hand

[Question]: *Taking into account all the actions performed by c, what can you deduce about the primary objective and focus within the video content?*
Option A: C is cooking.
Option B: C is doing laundry.
Option C: C is cleaning the kitchen. 🌲
Option D: C is cleaning dishes.
Option E: C is cleaning the bathroom.

Figure 6. Failure case of VIDEOTREE.