VideoAgent: Long-form Video Understanding with Large Language Model as Agent

Xiaohan Wang*, Yuhui Zhang*, Orr Zohar, and Serena Yeung-Levy

Stanford University
{xhanwang,yuhuiz,orrzohar,syyeung}@stanford.edu

This document provides more details of our approach and additional experimental results, organized as follows:

A Run-time Analysis of CLIP

While CLIP in *VideoAgent* may see up to all the frames within the video and VLM only captions a few frames, using CLIP is computationally efficient and negligible compared to using an LLM or VLM for several reasons. Firstly, CLIP's feature computation involves just a single feed-forward process. Secondly, CLIP employs an image-text late interaction architecture, enabling the caching and reusing of image frame features across different text queries.

Consider the scenario where the computation of CLIP features requires x seconds per image and text, while VLM captioning necessitates y seconds per image, and LLM computation consumes z seconds per round. For a video comprising N frames, suppose the *VideoAgent* selectively processes n frames out of N across t rounds of iterations. In this context, the time to compute CLIP features for images and texts amounts to Nx and nx seconds respectively. The generation of VLM captions requires ny seconds, and LLM operations total tz seconds. Consequently, the proportion of time dedicated to computing CLIP features relative to the overall computation time is approximated by $\frac{Nx+nx}{Nx+nx+ny+tz}$.

In practice, utilizing OpenCLIP ViT-G as CLIP, CogAgent as VLM, GPT-4 as LLM, with an A6000 GPU and the EgoSchema dataset, we find that $N = 180$, $n = 8.4$, $x = 0.02$, $y = 20$, $z = 10$, and $t = 3$. Therefore, the formula simplifies to $\frac{180\times0.02+8.4\times0.02}{180\times0.02+8.4\times0.02+8.4\times20+3\times10}$, which evaluates to 1.9%. This demonstrates that, under these conditions, the computation of CLIP features represents a small fraction, specifically 1.9%, of the total computational effort.

It should also be noted that the estimation above represents an upper bound. In practice, our segment-level retrieval approach necessitates computing features only within specified segments, rather than across all N frames, which further enhances efficiency.

* Equal contribution. Project page: https://wxh1996.github.io/VideoAgent-Website/.

B Additional Implementation Details

Details of CogAgent. For the experiments on NExT-QA [57], we utilize CogAgent [70] as the captioner, which is a frame-based captioning model. CogAgent has 18B parameters with input image resolution 1120 × 1120.

Details of LaViLa. For the experiments on EgoSchema [29], we utilize LaViLa [70] as the captioner, which is a clip-based captioning model. LaViLA takes input clip with resolution at 4 × 336 × 336. Following [69], to ensure zero-shot evaluation, we utilize the LaViLa model retrained on the ego4D data, filtering out the overlapped videos with EgoSchema.

Details of CLIP. We employ the EVA-CLIP-8B-plus [44] model for frame retrieval, a state-of-the-art CLIP model that includes a vision encoder with 7.5 billion parameters and a text encoder with 0.7 billion parameters. This model processes images at a resolution of 448 × 448 and produces output features with a dimensionality of 1280.

C Prompts for GPT-4

The specific prompts utilized for GPT-4 to predict answer, self reflect, and find missing information are shown in Figures 1, 2, and 3, respectively.

````
Given a video that has 180 frames, the frames are decoded at 1 fps. Given the following descriptions of the sampled frames in the
↪  video:
{'frame 1': '#C C rolls the dough on the table with both hands.', 'frame 28': '#C C puts the dough in the dough roller', 'frame
↪  45': '#C C walks to the doughs on the work table.', 'frame 76': '#C C picks dough from the baking tray', 'frame 90': '#C C
↪  picks up dough', 'frame 111': '#C C picks dough from the tray of doughs with both hands. ', 'frame 135': '#C C throws the
↪  dough on the dough roller', 'frame 171': '#C C places the dough on the baking tray', 'frame 180': '#C C rolls the dough on the
↪  baking table with his hands.'}
Please answer the following question:
```
How would you briefly describe the sequential order of the process that c performed on the dough, from initial handling to final
↪  placement on the tray?
0. Initially, c first carefully rolled the dough on the flour, then smoothly rolled it on the table, and finally, gently placed it
↪  in the awaiting tray.
1. C first placed the dough in the tray, then rolled it on the table, and finally rolled it on the flour.
2. Initially, c carefully rolled the dough on the table, then skillfully placed it in the tray, and ultimately, gently rolled it
↪  on the flour.
3. Initially, c first carefully placed the dough in the tray, then gently rolled it on the flour, and ultimately smoothly rolled
↪  it on the table.
4. C first rolled the dough on the table, then rolled it on the flour, and finally placed it in the tray.
```
Please think step-by-step and write the best answer index in Json format {'final_answer': 'xxx'}. Note that only one answer is
↪  returned for the question.
````

Fig. 1: The prompt to predict answer. We instantiate the prompt with a real video and question (highlighted in orange).

```
Please assess the confidence level in the decision-making process.
The provided information is as as follows,
Given a video that has 180 frames, the frames are decoded at 1 fps. Given the following descriptions of the sampled frames in the
↪ video:
{'frame 1': '#C C rolls the dough on the table with both hands.', 'frame 28': '#C C puts the dough in the dough roller', 'frame
↪ 45': '#C C walks to the doughs on the work table.', 'frame 76': '#C C picks dough from the baking tray', 'frame 90': '#C C
↪ picks up dough', 'frame 111': '#C C picks dough from the tray of doughs with both hands. ', 'frame 135': '#C C throws the
↪ dough on the dough roller', 'frame 171': '#C C places the dough on the baking tray', 'frame 180': '#C C rolls the dough on the
↪ baking table with his hands.'}
Please answer the following question:
```
How would you briefly describe the sequential order of the process that c performed on the dough, from initial handling to final
↪ placement on the tray?
0. Initially, c first carefully rolled the dough on the flour, then smoothly rolled it on the table, and finally, gently placed it
↪ in the awaiting tray.
1. C first placed the dough in the tray, then rolled it on the table, and finally rolled it on the flour.
2. Initially, c carefully rolled the dough on the table, then skillfully placed it in the tray, and ultimately, gently rolled it
↪ on the flour.
3. Initially, c first carefully placed the dough in the tray, then gently rolled it on the flour, and ultimately smoothly rolled
↪ it on the table.
4. C first rolled the dough on the table, then rolled it on the flour, and finally placed it in the tray.
```
Please think step-by-step and write the best answer index in Json format {'final_answer': 'xxx'}. Note that only one answer is
↪ returned for the question.
The decision making process is as follows,
Based on the sequence of sampled frames provided, the following sequential order of the process can be described:

- C rolls the dough on the table with both hands. (frame 1 & frame 180)
- C puts the dough in the dough roller. (frame 28 & frame 135)
- C walks to the doughs on the work table. (frame 45, context not clear enough to make this a major step)
- C picks dough from the baking tray. (frame 76)
- C picks up dough. (frame 90, likely a continuation of picking it from the tray based on the next frame)
- C picks dough from the tray of doughs with both hands. (frame 111, this confirms the action started at frame 76)
- C places the dough on the baking tray. (frame 171)

From the above sequence, the most accurate description is that C first rolls the dough on the table, then puts it through the
↪ dough roller, and finally places it on the baking tray. Thus, the correct sequential order, represented by the best answer,
↪ should be:

4. C first rolled the dough on the table, then rolled it on the flour, and finally placed it in the tray.

Now to provide the answer in the requested JSON format:

```
{'final_answer': '4'}
```
Criteria for Evaluation:
Insufficient Information (Confidence Level: 1): If information is too lacking for a reasonable conclusion.
Partial Information (Confidence Level: 2): If information partially supports an informed guess.
Sufficient Information (Confidence Level: 3): If information fully supports a well-informed decision.
Assessment Focus:
Evaluate based on the relevance, completeness, and clarity of the provided information in relation to the decision-making context.
Please generate the confidence with JSON format {'confidence': 'xxx'}
```

Fig. 2: The prompt to self reflect. We instantiate the prompt with a real video and question (highlighted in orange).

```
Given a video that has 180 frames, the frames are decoded at 1 fps. Given the following descriptions of five uniformly sampled
↪  frames in the video:
{'frame 1': '#C C rolls the dough on the table with both hands.', 'frame 45': '#C C walks to the doughs on the work table.',
↪  'frame 90': '#C C picks up dough', 'frame 135': '#C C throws the dough on the dough roller', 'frame 180': '#C C rolls the
↪  dough on the baking table with his hands.'}
To answer the following question:
```
How would you briefly describe the sequential order of the process that c performed on the dough, from initial handling to final
↪ placement on the tray?
0. Initially, c first carefully rolled the dough on the flour, then smoothly rolled it on the table, and finally, gently placed it
↪ in the awaiting tray.
1. C first placed the dough in the tray, then rolled it on the table, and finally rolled it on the flour.
2. Initially, c carefully rolled the dough on the table, then skillfully placed it in the tray, and ultimately, gently rolled it
↪ on the flour.
3. Initially, c first carefully placed the dough in the tray, then gently rolled it on the flour, and ultimately smoothly rolled
↪ it on the table.
4. C first rolled the dough on the table, then rolled it on the flour, and finally placed it in the tray.
```
However, the information in the initial five frames is not suffient.
Objective:
Our goal is to identify additional frames that contain crucial information necessary for answering the question. These frames
↪  should not only address the query directly but should also complement the insights gleaned from the descriptions of the
↪  initial five frames.
To achieve this, we will:
1. Divide the video into four segments based on the intervals between the initial five frames.
2. Determine which segments are likely to contain frames that are most relevant to the question. These frames should capture key
↪  visual elements, such as objects, humans, interactions, actions, and scenes, that are supportive to answer the question.
For each frame identified as potentially relevant, provide a concise description focusing on essential visual elements. Use a
↪  single sentence per frame. If the specifics of a segment's visual content are uncertain based on the current information, use
↪  placeholders for specific actions or objects, but ensure the description still conveys the segment's relevance to the query.
Select multiple frames from one segment if necessary to gather comprehensive insights.
```
{'frame_descriptions': [{'segment_id': '1/2/3/4', 'duration': 'xxx - xxx', 'description': 'frame of xxx'}, {'segment_id':
↪ '1/2/3/4', 'duration': 'xxx - xxx', 'description': 'frame of xxx'}, {'segment_id': '1/2/3/4', 'duration': 'xxx - xxx',
↪ 'description': 'frame of xxx'}]}
```

**Fig. 3: The prompt to find missing information.** We instantiate the prompt with a real video and question (highlighted in orange).
```