

# Video ReCap: Recursive Captioning of Hour-Long Videos

## Supplementary Material

Our supplementary materials contain Section **S1**: Additional Implementation Details, Section **S2**: Ego4D-HCap Data Collection Process, Section **S3**: Ego4D-HCap Dataset Analysis, Section **S4**: Additional Quantitative Results, and Section **S5**: Qualitative Results.

### S1. Additional Implementation Details

Figure **S1** Shows the schematic diagram of the proposed Video ReCap model.

**Video Encoder.** We employ the TimeSformer model [2] as our video encoder. This model, consisting of 12 transformer layers, is pretrained using a contrastive objective [11]. The input to the encoder comprises 4 RGB frames of size  $224 \times 224$ . To process the video, we divide it into 4-second clips and extract features for each clip using the pretrained video encoder. For clip caption, we utilize the dense spatiotemporal features. This allows our model to capture fine-grained details. However, we only use the CLS features for segment description and video summary, allowing efficient computation.

**Video-Language Alignment.** We utilize a pretrained language model DistilBERT [10] as our Video-Language (VL) Alignment module. It is a 6-layer transformer encoder model, where we freeze the self-attention blocks and insert a trainable cross-attention module inside each layer. It takes video features output by the video encoder and captions generated at the previous hierarchy as inputs. Note that there are no text inputs for clip captions. For segment description, we extract clip captions at each 4 seconds of the segment, and for video summary, we extract segment descriptions at each 3 minutes of the video and pass them to the VL alignment module along with corresponding video features.

**Text Decoder.** We leverage a pretrained GPT2 [9] as our text decoder. It is a 12-layer transformer model, and we insert a gated cross-attention block inside each transformer layer. We train only the cross-attention modules and freeze the rest of the model. Each cross-attention block contains a cross-attention layer and a feed-forward layer, followed by a tanh gating [4]. The tanh-gating is initialized with an initial value of zero so that the model’s output is the same as the pre-trained LLM at the beginning. As the training progresses, the model gradually learns to attend to the video-text embedding output by the VL-alignment module.

**Training the Video ReCap Model.** We follow a three-stage training pipeline for the Video ReCap model. First, we train our model 5 epoch using a batch size of 128 using clip caption data, which only uses video features. Afterward, we

employ the trained model from the first stage to extract clip captions within the videos at 4-second intervals. Then, during the second stage, we train the model for 10 epochs using a batch size of 32 using segment description samples, which take as input both video features and text features (clip captions). Finally, in the third stage, we extract segment descriptions every three minutes of the video using the trained model of the second stage and further train the model for 10 epochs using a batch size of 32 using video summary data. We use AdamW optimizer with optimizer [5] with  $(\beta_1, \beta_2) = (0.9, 0.999)$  and weight decay 0.01. We use a learning rate of  $3^{-5}$  and a cosine scheduling strategy.

**Training the Video ReCap-U Model.** Training a unified model that shares all parameters across three hierarchies is more challenging. We employ a similar three-stage approach with some additional tricks. In particular, the first-stage training is identical to the Video ReCap model. However, during the second stage, we train the Video ReCap-U model using both clip captions and segment description samples to prevent catastrophic forgetting of clip captions. One particular challenge is that the clip captions and segment description data are quite different. While clip captions use dense spatiotemporal features, segment descriptions utilize CLS features. Moreover, segment descriptions use video and text features as inputs, while clip captions only use video features. To overcome this challenge, we employ an alternate batching pipeline, where we sample a batch of clip captions and segment descriptions alternatively during the training. Since we have a lot more clip caption data ( $\sim 4M$ ) compared to segment descriptions (100K including manually annotated and LLM-generated pseudo annotations), we randomly sample 100K clip captions and only used those during the second stage of training. Finally, we train the model during the third stage using samples from all three hierarchies using a similar alternate batching approach. Since we have only  $\sim 20K$  (including manually annotated and LLM-generated pseudo annotations) samples for video summaries, we randomly sample 20K clip captions and 20K segment descriptions and used those along with video summaries during the third stage of training. This strategy prevents catastrophic forgetting of the model. It allows the training of the Video ReCap-U model, which shares all parameters across hierarchies. For Video ReCap-U, We use the same learning rate, batch size, training epoch, optimizer, and scheduler for the Video ReCap (See the previous paragraph).

**Inference.** During inference, we uniformly sample 4 frames from the corresponding clip and extract spatiotemporal features using the video encoder to use as inputs to

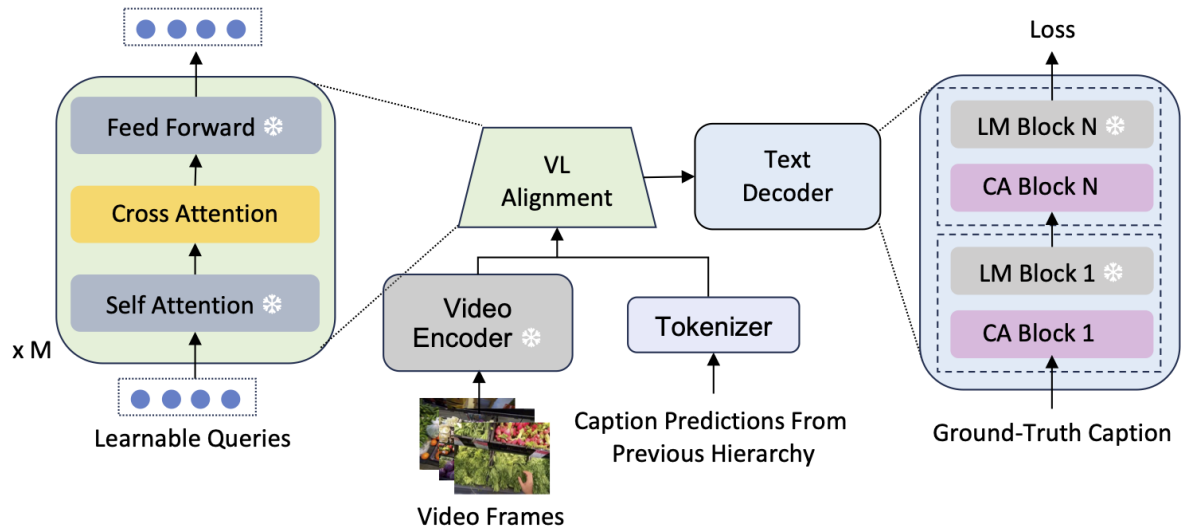


Figure S1. Model Architecture.

generate clip captions. For segment description, we extract CLS features and clip captions every 4 seconds of the segment and use them as inputs to generate segment descriptions. Lastly, we extract segment descriptions at each 3 minutes of the video and use them along with pre-extracted CLS features to generate video summaries. Note that clip boundaries are not given during the inference of segment descriptions, and segment boundaries are not given during the inference of video summaries.

We will release our code, data, and pretrained models.

## S2. Ego4D-HCap Data Collection Process

The Ego4D-HCap dataset was collected over the span of 2 months, from April 2023 to May 2023 and from September 2023 to October 2023. We recruited 91 specialized annotators through CloudResearch<sup>1</sup>, a participant-sourcing company. All annotators are based in the United States and are compensated at a rate of 9 dollars per hour, which is above the national minimum wage.

We utilized Qualtrics and Google Drive to build our data collection interface. Our interface began with an introduction to our project, guidelines for summarizing the videos, and examples of good summaries. It then asked the annotators for their ConnectID and provided them a link to the documents of videos assigned to them. Each document would contain 10-25 videos for the annotators to summarize, along with a prompt and a GIF summarizing the events of each video. The last interfaces contain text boxes for the annotators to put the text summaries for each video and the annotator's experience with the data collection interface. We used the latter to improve upon the interface so that the

quality of the annotated summaries ultimately became better. Figure S2 shows our data collection interface.

### S2.1. Guidelines for Annotators

**Overview.** In this project, we aim to develop a model that can automatically summarize long videos. Our model generates text captions for each video describing what happens every 3 minutes. We need your help to summarize those captions into a summary for the entire video. The total length of a video can be between 10 and 100 minutes.

#### Captions.

1. You are given a list of captions for each video.
2. Each caption describes what is happening every 3 minutes.
3. C refers to a person in the provided captions.
4. The captions are generated using a machine learning model, so sometimes, they can be out of order or inaccurate. In that case, you can exclude the events or details that do not make sense in the summary or refer to the GIF provided under the captions.
5. The captions may also use different terms to refer to the same thing. If only technical terms are used, then use them in your summary. Otherwise, we prefer you to use generic terms.

#### GIFs.

1. Since the videos are very long, we do not provide the full video. Instead, you are also given a GIF for each video.
2. GIFs created by sparsely sampled frames from the video, which is intended to help you better understand the overall contents of the video along with the captions.

#### Summaries.

1. The summary should be one paragraph long. Try to maintain a compression factor of 5, i.e., for every five

<sup>1</sup><https://www.cloudresearch.com>

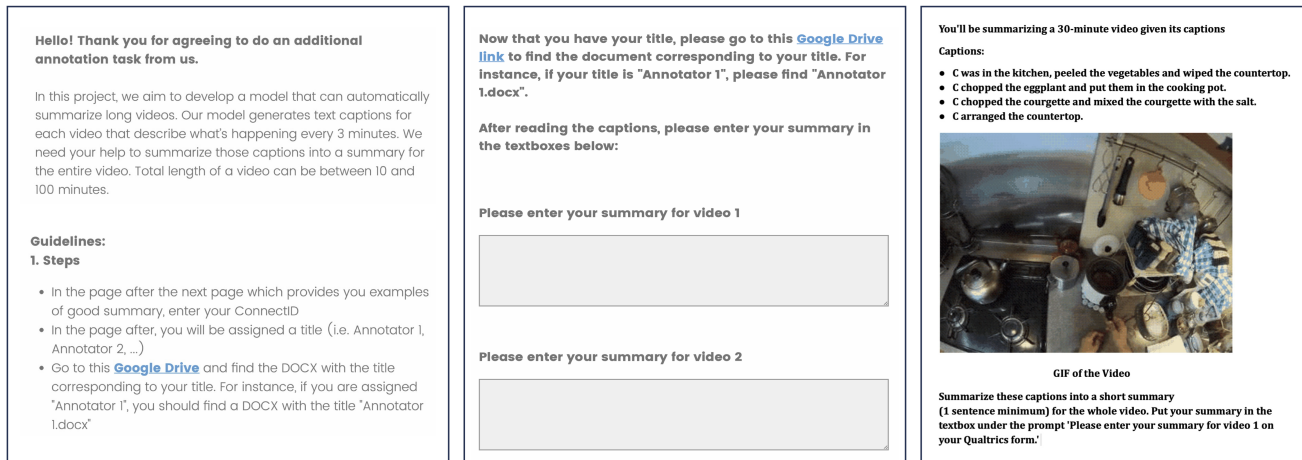


Figure S2. Data Collection Interface.

captions, you should summarize it in 1 sentence. However, each summary should be at least one sentence.

2. The summary should cover the setting, characters, and events that take place in the order of the video.
3. Avoid using X, Y or other letters to refer to characters other than C. Instead, use woman and man. Refer to examples of good summaries on the next page.
4. The summary should not have an interpretation of the characters' personalities or qualities.
5. The summary should be logically coherent, unambiguous, and understandable.
6. The summary should be grammatically correct.
7. Repetition of actions should have an underlying purpose/pattern.

## S2.2. Quality Control

To control the quality of the annotations, we pre-selected annotators before moving them forward with the official annotation task and manually reviewed the annotations. Before the official annotation task, we paid 171 annotators to complete a preliminary annotation task and selected from this pool annotators who provided desirable annotation quality. We minimized the chances of getting low-quality annotations by pre-selecting high-quality annotators and familiarizing them with an interface similar to the actual annotation task.

Another quality control method we utilized was to review the annotations ourselves manually. For each annotator, we randomly sampled half of the annotations they provided. We assessed their quality based on whether they followed the expectations outlined in Section S2.1. If less than half of the sampled annotations are of low quality, we would provide annotator feedback and ask them to redo their annotations. If the annotations were of better quality, we would replace them with the initial annotation. Otherwise, we

would discard both versions and assign them to other annotators.

## S2.3. De-identification Process

Due to the nature of the dataset and our task, our dataset has already been de-identified. Since all of our videos are sourced from Ego4D, they have undergone sensitive object detection, false positive removal, fast negative correction, and image blurring [3]. They were not modified during the dataset collection process, so the videos remain de-identified. Our annotators are also anonymized, as we recruited, managed, and corresponded with annotators on CloudResearch. Aside from their ConnectID, which we used to revise annotations, we did not collect any of the annotators' personal information.

## S3. Ego4D-HCap Dataset Analysis

**Scenarios.** Ego4D-HCap dataset comprises videos capturing diverse scenarios of various contexts, such as household settings, outdoor environments, workplaces, leisure activities, and more, totaling 127 distinct scenarios. The distribution of the most common 50 scenarios is illustrated in Figure S3. The inclusion of this extensive array of scenarios, depicting various locations and a wide spectrum of human activities, is imperative for assessing the robustness and generalizability of a model designed for the hierarchical video captioning task.

**Caption Lengths.** The distribution of caption lengths for three hierarchical levels in the Ego4D-HCap dataset is illustrated in Figure S4. Notably, clip captions are generally shorter, averaging 7.74 words per caption. In comparison, segment descriptions display a medium length, averaging 15.79 words, while video summaries are the longest, with an average of 25.59 words. Additionally, it is observed that

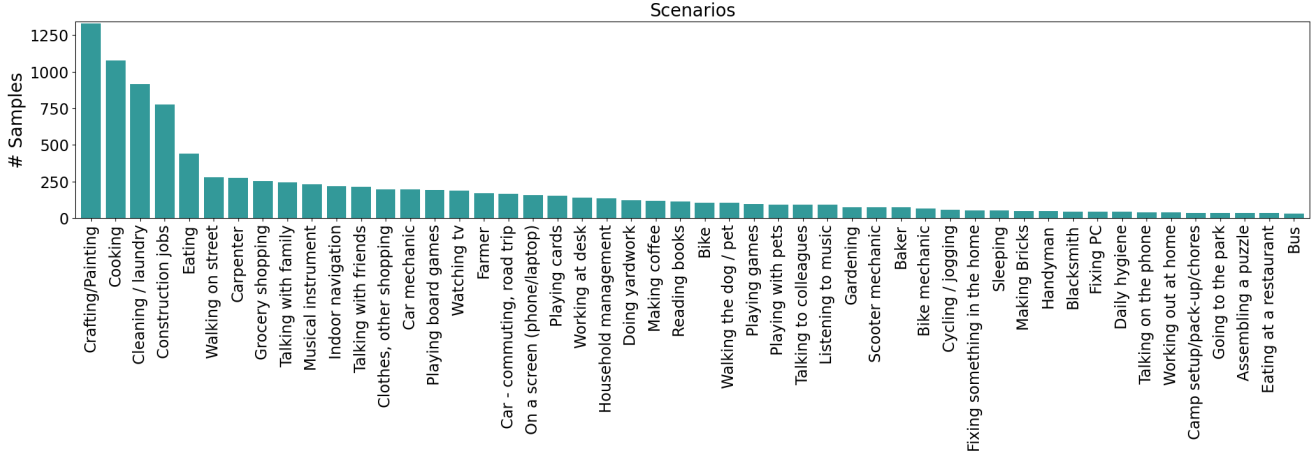


Figure S3. Distribution of the most common 50 scenarios in Ego4D-HCap dataset.

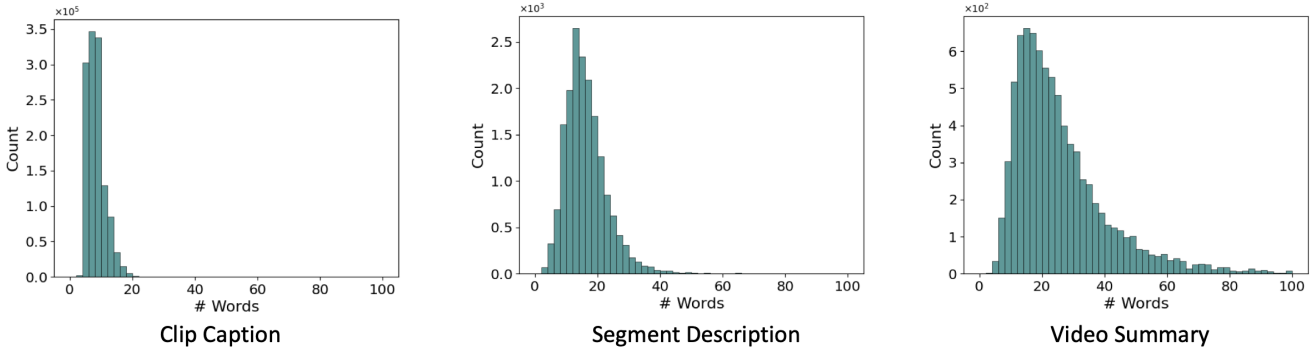


Figure S4. Distribution of the lengths of three hierarchical captions of the Ego4D-HCap dataset.

the maximum length for a clip caption is 43 words, segment descriptions can extend up to 73 words, and video summaries may reach a maximum length of 172 words.

### S3.1. Example Video Summaries.

Figure S5 Shows examples of annotated video summaries of the Ego4D-HCap dataset. We observe that video summaries are of various lengths and capture diverse scenarios, places, and activities. Typically, each video is annotated with multiple summaries. However, the figure shows only one summary per video for clarity and conciseness.

## S4. Additional Quantitative Results

**Backbone Design.** In this section, we ablate various aspects of our Video-Language Backbone design. First, we validate the effectiveness of a Language Model-based (LM) [10] Video-Language Alignment module rather than a standard Transformer resampler used in prior works [1, 11]. Table S1 shows that an LM-based Alignment module performs significantly better than the standard transformer-based resampler in all three hierarchies. Second, we in-

ject trainable cross-attention layers [1, 11] in the text decoder to incorporate video features. In contrast, several prior works [6, 8] inject video features only in the input layer while freezing the whole text decoder. Table S1 shows that using trainable cross-attention layers in the textual decoder performs significantly better than using video features in the input layer alone across all three hierarchical levels.

## S5. Qualitative Results

### S5.1. Qualitative Results on Ego4D-HCap

In Figure S6, we present three instances of hierarchical captions generated by our model. It is evident that clip captions mostly describe atomic actions and objects, such as ‘C closes the tap’ (Figure S6 (a)) and ‘C pushes the trolley’ (Figure S6 (b)). In contrast, segment descriptions focus on intermediate concepts within the video spanning longer durations, i.e., ‘C was in the kitchen, washed utensils’ (Figure S6 (a)), and ‘C arranged the tent and interacted with a woman’ (Figure S6 (c)). Moreover, video summaries aim to encapsulate the overarching content and events of the video. For example, ‘C went to the supermarket. C picked up fruits



LM Alignment	Trainable CA	Clip Caption			Segment Description			Video Summary		
		C	R	M	C	R	M	C	R	M
✗	✓	92.56	47.64	28.03	39.41	38.62	17.71	23.04	28.33	13.72
✓	✗	73.88	43.17	21.67	32.16	31.67	13.33	12.16	21.06	8.22
✓	✓	<b>98.35</b>	<b>48.77</b>	<b>28.28</b>	<b>41.74</b>	<b>39.04</b>	<b>18.21</b>	<b>28.06</b>	<b>32.27</b>	<b>14.26</b>

Table S1. **Architecture Ablation.** An LM-based [10] Video Language Alignment module provides significant performance gains compared to the transformer-based resampler used in prior works [1, 11]. Adding trainable cross-attention layers inside the text decoder performs much better than freezing the decoder.

vegetables, and interacted with other people. C bought groceries and paid at the cashier’ (Figure S6 (b)).

We also notice that while generating clip captions and segment descriptions is relatively more straightforward, generating video summaries is more challenging. For instance, while the generated video summaries of Figure S6 (a) and Figure S6 (b) are of good quality, the video summary of Figure S6 (c) could be further improved. The video summary of Figure S6 (c) fails to capture some important events of the video and includes repeated words and phrases. These challenges highlight the complexity of summarizing content in long-range videos. We anticipate that future advancements and the use of our released data will contribute to the development of more effective methods and models for this demanding task.

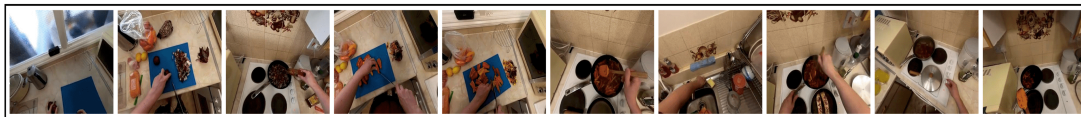
## S5.2. Qualitative Results on EgoSchema

Figure S7 illustrates the qualitative outcomes of our long-range video question answering experiment on the EgoSchema [7] dataset. The approach, detailed in ??, involves the generation of hierarchical captions utilizing the Video ReCap model for videos. Subsequently, these captions are presented to ChatGPT along with questions and answer choices as prompts, enabling the model to select the correct answer. In Figure S7 (a) and Figure S7 (b), it is evident that ChatGPT tends to choose incorrect answers when provided solely with clip captions. However, the model consistently makes correct choices in both scenarios when supplemented with video summaries. This highlights the efficacy of our generated hierarchical captions in enhancing the performance of long-range video question answering tasks. Nevertheless, in certain instances, as depicted in Figure S7 (c), our approach encounters challenges and fails to identify the correct answer.



50 min

In a garden, C trims plants with a machine as another person collects the plant leaves with a rake. They also clean with a leaf blower and remove weeds from the garden by disposing of them in a dustbin. Later, they wash their hands with water from a hosepipe and organize their tools.



44 min

In a kitchen, C retrieves an item from the fridge and chops onions and cabbage. They also wash and chop peppers, removing the seeds. Then they remove lemon seeds and cut tomatoes. C fries fish slices in a pan and smears spices on meat. They stir food with a wooden spoon, pour juice into a glass cup, and serve the food.



62 min

In a bakery, C poured flour and rolled and cut dough. C removed bread and donuts from an oven, kneaded more dough, and cut and packaged chocolate. C then baked more bread, cleaned the oven, and made more donuts.



41 min

C rode a bus to the supermarket. C purchased vegetables, and other groceries. C paid for the items at the cashier. C then walked home and went into a room upstairs.



65 min

C is outside on a construction site. He is doing various duties around the construction site. He is measuring and cutting wood, marking the wall, sanding it and also nailing it as well. He uses various tools and then operates a forklift.

Figure S5. **Examples of annotated video summaries of the Ego4D-HCap dataset.** Due to space limitation and conciseness, we show one frame for each 5 minutes of the video..



Figure S6. **Qualitative Results on Ego4D-HCap**. Generally, clip captions depict atomic actions and objects; segment descriptions focus on intermediate concepts, and video summaries encapsulate the overall content and goals of the videos. While generating clip captions and segment descriptions are often relatively easier tasks, developing a good video summary is often challenging. Our models perform well on video summaries (a) and (b), but the generated video summary (c) could be further improved.




Video	
Question	Taking into account all the actions performed by c, what can you deduce about the primary objective and focus within the video content?
Clip Caption	C wipes the bowl. C pours water in the sink. ... C picks the soap bottle. C opens the tap. ... C rinses the chopping board. C closes the tap.
Video Summary	C was in the kitchen, washed the utensils and arranged them in the shelf.
Input: Clip Captions (Baseline)	C is cleaning the kitchen. ✗
Input: Hierarchical Captions (Ours)	C is cleaning dishes. ✓
(a)	
Video	
Question	Describe the overall sequence of events in the video, paying special attention to how the character interacts with objects and maintains cleanliness throughout their activities in the kitchen.
Clip Caption	C pours the tomato in the bowl. C rinses knife. ... C wipes the counter top. C picks the plate of food. ... C eats the food. C touches the laptop.
Video Summary	C was in the house. C cooks food, ate food and operates laptop.
Input: Clip Captions (Baseline)	The main character skillfully prepares a meal, later consumes and enjoys it wholeheartedly. ✗
Input: Hierarchical Captions (Ours)	The character prepares a meal, eats it, and then uses their laptop. ✓
(b)	
Video	
Question	Summarize the main process that c carries out during the video and highlight similarities and differences observed at various stages.
Clip Caption	c rubs the vessel with the sponge. C picks the pottery ... c puts vessel on the table. C rolls the clay mold in his hands ... c rolls the clay on his finger. C rotates the pottery.
Video Summary	C was in the house, molded the clay vase and placed the clay vase on the table.
Input: Clip Captions (Baseline)	Person c diligently repairs a damaged pottery piece skillfully. ✗
Input: Hierarchical Captions (Ours)	C creates a pottery from scratch. ✗
Correct Answer	C cleans a pottery with a sponge, then decorates it with clay. ✓
(c)	

Figure S7. **Qualitative Results on EgoSchema.** The baseline method that uses only short-range clip captions as input fails in examples (a) and (b), where our approach succeeds by utilizing hierarchical captions (i.e., clip captions and video summaries). Both models fail in Example (c).



## References

- [1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. [4](#), [5](#)
- [2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, page 4, 2021. [1](#)
- [3] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022. [3](#)
- [4] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. [1](#)
- [5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. [1](#)
- [6] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. [4](#)
- [7] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *arXiv preprint arXiv:2308.09126*, 2023. [5](#)
- [8] Ron Mokady, Amir Hertz, and Amit H Bermano. Clip-cap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021. [4](#)
- [9] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. [1](#)
- [10] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. arxiv 2019. *arXiv preprint arXiv:1910.01108*, 2019. [1](#), [4](#), [5](#)
- [11] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6586–6597, 2023. [1](#), [4](#), [5](#)