**This appendix is organized as follows:**
7. Offline and Online Experiment Details.
8. Prompt templates for LTA and VPA.
9. Ablations related to visual history representation.
10. Overview of existing VCLM and Socratic Models
11. Activity scripts for online evaluation.
12. Model Error Analysis in Online Evaluation.
13. Study Activity Visualization.
14. Participant Data Collection Practices

# 7. Offline and Online Experiment Details

## 7.1. Offline Benchmark Tasks

We evaluate VCLM and Socratic models on two existing video-based forecasting benchmarks – Long Term Action Anticipation (LTA) [15] and Visual Planning for Assistance (VPA) [33] using offline datasets – Ego4D [15] and CrossTask [52] respectively (Sec. 4). Here, we provide a detailed overview of the datasets and our experimental setup for each of these tasks.

**Ego4D-LTA** [15]: Ego4D consists of 3,670 hours of video footage of everyday activities, with 53 different scenarios. Out of this, we use the LTA (forecasting) subset, which entails 116 hours. This subset contains 1723 clips that cover an action space of 115 verbs and 478 nouns. We use the standard train and validation splits proposed by Ego4D [15] for our evaluation. In the LTA task, given 8 video segments from a video clip as input, the models must predict the 20 future actions in the form of verb, noun, and verb + noun, in correct order. Edit distance between the predicted sequence of actions and the ground truth action sequence in the video clip is used as a metric for evaluation following Ego4D [15].

**CrossTask-VPA** [52]: CrossTask consists of 2.7K instructional videos for 18 different tasks from multiple domains, covering 374 hours of footage. Some of the action classes were shared among different tasks, with a total of 118 actions. Each video consists of an average 7.6 action steps. We follow [33] to construct a train split with 1,564 videos and a test split with 752 videos. We extract multiple test samples from each test video for VPA – specifically, given an annotated video consisting of K steps, we generate $K-Z$ samples, leaving at least $Z = 3, 4$ steps to predict in the future. This leads to a dataset of 4123 test samples for our evaluation. Our VPA task definition also follows [33] – given an untrimmed video and a goal of the activity/task in the video described in natural language as input, the models must predict the up to 4 future actions in the form of verb+noun, in correct order. Evaluation compares the predicted action sequence with the ground truth actions in the video using mIoU, per step accuracy, and success rate metrics (Sec. 4).

## 7.2. Model Modifications for Online Evaluation

**Goal-Conditioned Summarization.** Sec. 5.2 provides an overview of modifications for our multimodal LLMs to enable online evaluation. The online settings entail noisy stream of redundant video frames leading to long narration history. To handle such long narration histories in a robust manner, one of the biggest changes we make in these models is goal-conditioned summarization. This greatly reduces the number of tokens in the input and allows the language model to attend to a longer narration histories more robustly while still leveraging few-shot examples. The summarization is performed by LLama2-13B Chat using the following prompt:

```
A person is currently attempting to
[goal]. Their task is in progress and
their goal is not yet complete. The
following are low level narrations of
their actions.

[narration history]

Please summarize these into a smaller
set of high-level narrations. Focus on
narrations that are relevant to the
goal and do not include irrelevant
narrations in your high-level summary.
Begin every high-level narration with
the text, 'A person ':
1. A Person
```

The output from the LLM is parsed by only keeping lines starting with numbers. The helps remove any conversation or filler language in the response. This summarized history often contains 5-20 high-level narrations and is used by the LLM to perform prediction. In cases where the raw narration history exceeds the LLM context window, narrations are uniformly subsampled by the smallest integer factor to fit within the LLM context window. Few-shot examples are also summarized offline.

**Goal-Generation for Few-shot Examples.** We evaluate the utility of goal conditioning in online experiments, akin to our offline experiments (Sec. 9.5). To that end, our pilot studies show that goal-conditioned prediction performs much better than prediction without goals in the online setting. Specifically, we find that goals help the LLM identify which parts of noisy input video stream and narrations are relevant to completing the activity. In order to ensure our few-shot examples to the multimodal LLMs, which are obtained from Ego4D, are appropriately goal conditioned for online experiments, we need to annotate these with goal information. Since, goal information is not available in Ego4D, we again resort to LLMs for obtaining pseudo goal labels for these videos. Specifically, we use Llama2-70B chat with the following prompt to generate goals for Ego4D LTA training set videos:

```
The user took these physical actions:
[Narration History]

What are the top 3 goals of the user?

Respond only in JSON that satisfies the
    Response type:
type ResponseList = [Response_1,
    Response_2, ..., Response_3]
type Response = {
```

```
user_goal: str;
confidence: float;
explanation: str;
}
Provide {user_goal} in the format of 'They
    wanted to {user_goal}', the
    {confidence} of the goal given the
    context (on a scale from 0 to 1), and a
    terse {explanation} of the given goal
    and its confidence.
```

where [Narration History] is the full narration history for the clip generated by LaViLa. We parse the output text as a JSON and select the goal with the highest confidence.

**Performance Comparison Between Offline and Online Models.** We also evaluate our online-modified models on the VPA task to ensure our modifications do not drastically alter performance. Table 5 shows the performance difference between online and offline models is relatively minimal for $Z = 1$ despite the online models replacing the explicit segmentation model with uniform segmentation for faster inference. Note that in the online setting, the models provide only the next action to the user, wait for the user to execute that action, and then replan, which is a prediction horizon of $Z = 1$. Prior work has shown that poor segmentation can reduce performance on the VPA task by up to 50% [33]. Our online modified models see a maximum 14% drop in performance, which indicates our online modifications (clustering and summarization) help mitigate the performance drop from our simplified segmentation.

| Model | Z=1 | | Z=3 | | | Z=4 | |
|---|---|---|---|---|---|---|---|
| | mAcc | SR | mAcc | mIOU | SR | mAcc | mIOU |
| Socratic Online | 22.5 | 2.3 | 17.9 | 29.8 | 1.1 | 17.8 | 34.7 |
| VCLM Online | 23.3 | 4.3 | 18.5 | 33.2 | 1.8 | 18.9 | 41.3 |
| Socratic Offline | 22.8 | 5.6 | 22.2 | 35.6 | 3.0 | 21.2 | 37.4 |
| VCLM Offline | 27.2 | 6.9 | 25.2 | 41.7 | 4.3 | 25.5 | 45.5 |

Table 5. **Comparison between online and offline models on the VPA task.** Note that the online models suggest the next step to the user, wait for the user to execute that step, and then replan ($Z = 1$).

### 7.3. Online System Inference and Hardware

The remote server for online inference utilizes 5 NVIDIA Tesla V100 GPUs with 32GB of VRAM each. The LaViLa narrator model occupies the first GPU. The narrator model runs asynchronously on batches of frames sent by the local machine every 2 seconds and saves clustered narrations to a cache. The total communication latency for a batch of frames from the Aria glasses to the local machine and then to the remote server was less than 400ms. The Llama2 13B model was distributed across the remaining 4 GPUs using the Huggingface transformers and accelerate libraries. When the user triggers assistance, the narration thread is paused and the set of narrations in the cache is used for summarization and prediction. The total latency from the time the user requests assistance to the time the assistance is relayed to them over the earbuds ranges from 10-25 seconds, depending on the length of the history. After assistance is communicated to the user, the narration thread resumes.

## 8. Prompt Templates for LTA and VPA

Detailed prompt templates for our offline benchmark tasks LTA and VPA as shown in figures 5 and 6. The prompt for LTA (Fig. 5) consists of *examples* text narration sequences pertaining to the full video from 8 videos of the training set and the *visual history* of 8 segments from the current video. The narrations are from the LaViLa narration model [49]. Likewise, the prompt for VPA (Fig. 6) includes *examples* of full action sequences consisting of ground truth (GT) action labels for 8 videos from the training set and the *visual history* of the current video, which entails actions predicted following previous work [33] noted as [predicted action].

```
"Task description"
#Prompt example *8 from training set:
   1. [narration]
   2. [narration]
   ...
   N. [narration]


#Visual history from current video:
   T-8. [narration]
   ...
   T-1. [narration]
   T.
```

Figure 5. **Prompt template for Ego4D LTA**. We set $N$ to be the total number of actions in the video and $T$ to be the starting action index that we want to predict in the current video.

```
"Task description"
#Prompt example *8 from training set:
   Goal: [CrossTask Task Title]
   1. [GT action]
   2. [GT action]
   ...
   N. [GT action]


#Visual history from current video:
   Goal: [CrossTask Task Title]
   1. [predicted action]
   ...
   T-1. [predicted action]
   T.
```

Figure 6. **Prompt template for CrossTask VPA**. We use the video's task title from CrossTask as goal description for VPA and append it in the front of the action sequence in our prompts for VPA. We use predicted actions following previous work [33] to construct the visual history of the current video. $N$ and $T$ follow the same design as in LTA.

| Model | Supervised | $Z=1$ | | $Z=3$ | | | $Z=4$ | |
|---|---|---|---|---|---|---|---|---|
| | Samples | mAcc | SR | mAcc | mIOU | SR | mAcc | mIOU |
| Random | | 0.9 | 0.0 | 0.9 | 1.5 | 0.0 | 0.9 | 1.9 |
| Socratic 7B | 8 | 22.3 | 4.3 | 21.0 | 33.3 | 2.6 | 20.8 | 36.2 |
| VCLM 7B | 8 | 27.9 | 6.8 | 24.8 | 41.7 | 4.1 | 24.0 | 45.0 |
| Socratic 13B | 8 | 22.8 | 5.6 | 22.2 | 35.6 | 3.0 | 21.2 | 37.4 |
| VCLM 13B | 8 | 27.2 | 6.9 | 25.2 | 41.7 | 4.3 | 25.5 | 45.5 |
| Socratic 70B | 8 | 28.1 | **9.1** | 26.6 | **43.6** | 5.5 | 25.5 | 45.7 |
| VCLM 70B | 8 | 28.1 | 8.9 | **26.9** | 43.4 | **6.1** | **26.8** | **46.9** |

Table 6. **Varying LLM size in visual planning (VPA) on CrossTask.** Mean accuracy, mean IoU, and Success Rate (SR) percentages are shown for short $Z = 1$ and medium $Z = 3, 4$ horizons. We use predicted actions from a finetuned segmentation model following previous work [33] to construct the visual history of the current video.

# 9. Ablations on Visual History Representation

## 9.1. Evaluation of the benefit from implicit representation of visual information for smaller LLMs across different LLM sizes.

Tables 6 above shows that mAcc gap in VPA task for 7B models with and without visual conditioning at Z = 1,3,4 is 5.6%, 3.8%, and 3.2% respectively. The mAcc gap for 13B models at Z = 1,3,4 is 4.4%, 3%, and 4.3% and for 70B models is 0%, .3%, and 1.3% respectively as in Table 3. Implicit visual representation aids smaller LLMs across model sizes.

## 9.2. Model Performance Across LLM Size

Table 7 shows how performance on the Long-Term Action Anticipation task varies with the size of LLM used in the approach. Note the scaling laws observed in 13 and 70 billion parameters are consistent when looking at approaches that uses smaller LLMs with 7 billion parameters.

| Model | Model Size | ED@Z=20 | | |
|---|---|---|---|---|
| | | Verb | Noun | Action |
| AntGPT [48] | 7B/13B | .756 | .725 | - |
| Palm* [16] | 7B | .732 | .812 | .958 |
| Socratic 7B | 7B | .731 | .786 | .951 |
| VCLM 7B | 7B | .792 | .765 | .958 |
| Socratic 13B | 13B | .731 | .732 | .929 |
| VCLM 13B | 13B | .740 | .751 | .932 |
| Socratic 70B | 70B | **.726** | **.712** | **.928** |
| VCLM 70B | 70B | .739 | .731 | .931 |

Table 7. **Varying LLM size in Long-term action anticipation on Ego4D.** Edit distance values for forecasting horizon of $Z = 20$ actions is shown on v1 validation set.

## 9.3. Task-relevant information from visual history

Different aspects of the visual history can be extracted and represented in text for VCLMs and Socratic models. It is unclear what aspects should be extracted to enable efficient forecasting in such models. To that end, we compare different modes of task-relevant information for Socratic multimodal LLMs on the Ego4D LTA task. Each of these modes of information can be obtained from different pre-trained vision-language models. Specifically, we consider information on objects, actions, and narrations describing activities in the video as the three relevant information modes.

| Information Type | VLM | ED@(Z=20)↓ | | |
|---|---|---|---|---|
| | | Verb | Noun | Actions |
| Narrations Only | LaViLa | **.731** | .787 | **.951** |
| Narrations + Objects | LaViLa + Detic | .734 | **.776** | .952 |
| Narrations + Actions | LaViLa + LaViLa Dual Encoder | .732 | .812 | .958 |

Table 8. **Comparison of different information types/modes that can represent a video's history using Socratic models on LTA.** Edit distance values for forecasting horizon of $Z = 20$ actions is shown on v1 validation set.

We obtain object descriptions using Detic [50] with the Ego4 LTA noun set as a custom vocabulary, recognized actions using the LaViLa dual encoder with the Ego4D LTA closed-set of actions, and open-set narrations using the LaViLa narration model [49]. We test three settings i.e., combinations of these information modes: only narrations, narrations and objects, narrations and actions. The only narrations setting uses the same prompts as our Socratic model described in Sec 3. The narrations and objects setting prepends a list of recognized objects from the input video before the narrations in the visual history. The narrations and actions setting follows the same prompting structure as Palm [16]. All three settings use the same retrieval-based prompting approach as described in Sec 4.2. Action and object prompts are generated on the LTA train set.

Table 8 shows the results for these three settings on Ego4D LTA. All models use Llama2-7B as the LLM. As seen in the table, neither adding object nor action information from the visual history noticeably improves performance on the LTA task. Following this result, we determine that object and closed-set action information is a subset of open-set narration information when it comes to visual history representation. Consequently, we use only narrations to represent visual history for both VCLM and Socratic models in all our offline and online experiments (Sec. 4, 5).

## 9.4. Comparison of narrators for visual history

Video history might be sufficiently represented using open-set narrations that describe the activity in the video (Tab. 8) for video-based forecasting tasks. To determine an appropriate video narration model for our multimodal LLMs in forecasting tasks, we compare two SOTA video narrators – LaViLa [49] and the Blip-2 [21]. We perform this comparison using the Llama2-7B Socratic models on LTA (Table 9). Following Palm [16], we feed the median frame from a video segment along with the prompt "A person is " to the Blip-2 model for generating a narration describing the video segment. In contrast, the LaViLa narration model uses 4 evenly spaced frames. We parse the output of the narrator model by replacing references to the participant with "A person" to ensure consistent structure.

| Narrator | Input Frames | Narrator LM | ED@(Z=20)↓ | | |
|---|---|---|---|---|---|
| | | | Verb | Noun | Actions |
| LaViLa | 4 | GPT2-XL (1.5B) | **.731** | **.787** | **.951** |
| Blip-2 | 1 | OPT 2.7B | .758 | .883 | .978 |

Table 9. **Comparison of narrators using Llama2-7B Socratic models on LTA.** Narrator LM represents the LLM backbone for each narrator model. Edit distance values for forecasting horizon of $Z = 20$ actions is shown on v1 validation set.

Note that, LaViLa narrator uses GPT2-XL (1.5 B parameters) as its LLM backbone while Blip-2 uses OPT 2.7B. LaViLa is also explicitly trained on Ego4D to narrate video clips [49].

As seen in Table 9, despite its smaller language model backbone, LaViLa narrator significantly outperforms Blip-2 for capturing relevant visual history for forecasting. This is likely due to LaViLa's narration-specific and egocentric training data, as well as consumption of 4 frames from the input video rather than just 1. Based on this analysis, we use LaViLa narrator for open-set narration generation of visual history for all our experiments unless otherwise specified.

### 9.5. Medium history-medium horizon forecasting in VPA without goal

Our offline benchmark tasks – VPA and LTA, cover the spectrum of medium to long forecasting based on medium to long visual history respectively. However, unlike LTA, the VPA task [33] also uses the goal of the activity in the video, in addition to the visual history, for forecasting future actions. To understand the performance of multimodal LLMs on medium history, medium horizon forecasting problems without the availability of goal information, we conduct an ablation on VPA. Specifically, we evaluate the best performing multimodal LLM – VCLM 70B on VPA with and without goal information (Table 10). We simply remove the goal information from the VPA prompt (Fig. 6) for this analysis.

The results show that the information regarding the goal enables the VCLMs to make better mid-horizon forecasting predictions while slightly decreasing the accuracy of short horizon predictions. Thus, the performance of multimodal LLMs may overall drop in medium history, medium horizon forecasting problems when goal information is not available. Since the availability of goal information leads to improved performance and since such information may be easy to obtain in user-in-the-loop settings, we frame our online evaluation using VPA's task definition i.e., with inclusion of goal.

### 9.6. Text-based history representation for VCLMs on long-history tasks

Since current VCLMs may be capable of encoding only limited visual history, we also provide text-based representation of history as input to VCLMs for our long history-based forecasting tasks e.g., LTA. We conduct an ablation experiment to determine the contribution of such additional text-based history representation – when used along with visual embeddings in VCLMs. Specifically, we want to determine how VCLMs without text history perform in long-history tasks. We run this ablation with a Llama2-70B chat VCLM on LTA using the following prompt:

| Model | Goal | $Z = 1$ | | $Z = 3$ | | | $Z = 4$ | |
|---|---|---|---|---|---|---|---|---|
| | | mAcc | SR | mAcc | mIOU | SR | mAcc | mIOU |
| VCLM 70B | No | **28.6** | 8.1 | 26.5 | 41.8 | 5.5 | 26.3 | 45.0 |
| VCLM 70B | Yes | 28.1 | **8.9** | **26.9** | **43.4** | **6.1** | **26.8** | **46.9** |

Table 10. **Short and medium horizon visual planning (VPA) on CrossTask w/wo the goal.** Mean accuracy, mean IoU, and Success Rate (SR) percentages are shown for short $Z = 1$ and medium $Z = 3, 4$ horizons. LaViLA is used as the narration model while Internvideo is used as the video encoder for VCLM as in LTA.

| Model | Visual History | Visual Encoder | ED@(Z=20)↓ | | |
|---|---|---|---|---|---|
| | | | Verb | Noun | Action |
| VCLM 70B | V | Internvideo | 1.000 | 1.000 | 1.000 |
| VCLM 70B | V+T | Internvideo+LaViLa | 0.739 | 0.731 | 0.931 |

Table 11. **Long-term action anticipation on Ego4D w/wo text history.** Edit distance values for forecasting horizon of $Z = 20$ actions is shown on v1 validation set.

```
Predict the next 20 actions in the form of
    (verb,noun)
```

The result in Table 11 shows that without a text-based history representation, the VCLM model fails to output meaningful predictions. Text-based history representation both provide a template for generation and help ground the VCLM to information not captured in its 8 input frames. We show examples of generated text with and without text history below.

**With text history:**

```
//Generated text:
16. A person drops a garlic peel in a bowl
17. A person presses a garlic clove with
    the knife
18. A person presses a garlic clove with a
    knife
19. A person drops the garlic clove in the
    bowl
```

**Without text history:**

```
//Generated text:
Pairs e.g. (drive, road) 1. Put CDs on top
    of magazines . 2. Paste pictures
    butterflies and birds near flowers . 3.
    Glue colored paper sequins on the bench
    in front of the boat .
```

```
//Generated text:
```

```
pairs. 1. tv, eyes 2. jako the bird is
    used.
```

## 10. Overview of existing VCLM and Socratic Models

Tables 12 and 13 show how our implemented Socratic and VCLM models compare to other models within each approach. Socratic models differ in what kinds of text the VLM's generate (actions, narrations, objects, etc), which VLMs are used, whether prediction is conditioned on an inferred goal, how prediction is performed, either by directly generating subsequent actions, or using chain of thought reasoning, and what LLM is used. We select our implemented socratic model by testing actions, narrations, and objects on the Ego4D LTA task (table 8). We found that adding actions or objects, provided by the LaViLa encoder [49] and Detic [50] respectively, did not improve performance over open-set narrations provided by the LaViLa narrator model [49]. Accordingly, our representative Socratic model implementation uses only narrations as a text-based representation of visual history.

VCLM models attach a pretrained vision encoder to a pretrained LLM by mapping outputs from the vision encoder into the token embedding space of the LLM. VCLM models primarily differ by what pretrained image or video encoder is used, how aggregation across image level features is done if an image encoder is used, whether the model explicitly aligns video representations with text representations, whether the entire model is fine-tuned on an instruction dataset, and what LLM-backbone is used. Some VCLMs for video use a pretrained video encoder that samples frames from the video and processes them together. In contrast, other video VCLMs use a pretrained image encoder on a set of sampled frames and aggregate image-level features with a separate aggregation module. This aggregation module could be a form of attention, concatenation, pooling, or convolution. Prior work has indicated no clear advantage between image or video level features [24]. VCLM models also employ two different methods for training, alignment training and instruction tuning. In alignment training, aggregation and projection layers are trained to better map inputs from the vision encoder space to the LLM token space using a set of video-text pairs and a contrastive loss function. In instruction tuning, the entire LLM backbone is finetuned using a multimodal instruction dataset. While not every VCLM uses both of these training methods, multiple prior works have found that alignment training followed by instruction tuning is generally more performative [24, 31]. Given these insights, we select AnyMAL [31] as a representative VCLM method for our benchmark tasks. AnyMAL uses InternVideo [42] as a video encoder, followed by an attention-based projector (perceiver resampler [1]) and is trained with both alignment training and instruction tuning. Furthermore, AnyMAL is trained on the HowTo100M dataset [30]. This dataset features videos of people performing daily tasks and is thus more relevant to our use case than VCLMs trained with other video datasets. Finally, AnyMAL also features 13-billion and 70-billion parameter versions, allowing us to test scaling laws for multimodal LLM prediction. The InternVideo endocder samples 8 frames from input videos. These frames, once encoded, use 256 tokens of the 2048 token context window of the Llama 2 models.

## 11. Activity Scripts for Online Evaluation

The following are the scripts participants used in the online study. Users complete steps up to the "**Get Assistance**" mark in any order they deem reasonable. The vision-based assistant takes over from there to guide users in completing the activities. The remaining steps are the steps we expect users to execute to successfully complete the activity.

### 11.1. Prepare a Latte

1. Get a cup and put it in the espresso machine
2. Pull 2x espresso shot using the espresso machine
3. Pour milk into a metal pitcher
   **Get Assistance**
4. Froth milk using the steam wand
5. Pour milk into espresso cup

### 11.2. Make a Caprese Salad

1. Cut the tomato into slices
2. Cut the fresh mozzarella into slices
3. Tear the basil leaves
4. Arrange the tomato on the plate
   **Get Assistance**
5. Arrange the mozzarella slices on the plate
6. Sprinkle the torn basil on top
7. Drizzle olive oil on top

### 11.3. Make a BLT Sandwich

1. Cut three slices of tomato
2. Pull off a leaf of lettuce
3. Take two slices of bread
4. Put mayonnaise on the bottom piece of bread
5. Put lettuce on the bottom piece of bread
   **Get Assistance**
6. Put tomato slices on top of the lettuce
7. Put bacon on top of the tomato slices
8. Put the top piece of bread on

| Method | Text | VLMs | Goal-Conditioned | Prediction | LLM |
|---|---|---|---|---|---|
| Palm [16] | Actions, Narrations | EgoVLP, Blip2 | No | Direct | GPT-Neo-1.3B |
| Ant-GPT [48] | Actions | CLIP | Yes | Chain-of-thought | Llama2 7B/13B |
| VideoChat-Text [22] | Actions, Objects, Audio Transcript | InternVideo, InternImage, Whisper | No | Direct | Vicuna 13B |
| Socratic (Ours) | Narration | LaViLa-Narrator | Yes | Direct | Llama2 13B/70B |

Table 12. **A comparison of Socratic models for prediction.** Models are compared by type of text, VLMs used, whether they are goal-conditioned, how they perform prediction, and which language models they use.

| Method | Encoder | Aggregation | Alignment Training | Instruction Tuning | LLM |
|---|---|---|---|---|---|
| Video-LLaVA [24] | LangugeBind | N/A | Yes | Yes | Vicuna 7B |
| LLaVA-NeXT [26] | CLIP | Concatenation | Yes | Yes | Vicuna 7B/13B |
| Video-ChatGPT [29] | CLIP | AveragePooling | No | Yes | Vicuna 7B |
| VideoChat-Embed [22] | BLIP2 | Q-Attention | Yes | Yes | Vicuna 13B |
| Video-Llama [47] | BLIP2 | Q-Attention | Yes | No | Llama2 7B/13B |
| Video-Llama 2 [8] | CLIP | Spatial-Temporal Conv | Yes | Yes | Mistral-Instruct 7B |
| TimeChat [36] | EVA-CLIP | Q-Attention | No | Yes | Llama2 7B |
| AnyMAL (Ours) [31] | InternVideo | N/A | Yes | Yes | Llama2 13B/70B |

Table 13. **A comparison of Visually-Conditioned Language Models capable of processing video.** Models are compared by vision encoder, how image level features are aggregated (if needed), whether the model is explicitly trained to align video features to text, and whether the model is instruction tuned.

| Method | Task | Redundant | Infeasible | Irrelevant |
|---|---|---|---|---|
| VCLM | BLT | 7 | 4 | 2 |
| | Caprese | 17 | 3 | 1 |
| | Latte | 8 | 6 | 1 |
| | Total | 32 | 13 | 4 |
| Socratic | BLT | 16 | 4 | 3 |
| | Caprese | 12 | 0 | 1 |
| | Latte | 5 | 12 | 3 |
| | Total | 33 | 16 | 7 |

Table 14. **A breakdown of cases where participants skipped assistant instructions**. Skips were categorized as redundant actions the participant had already completed, infeasible actions that could not be completed in the current task, and irrelevant actions which had no bearing on the current task. The numbers represent total numer of skips across all participants.

## 12. Model Error Analysis in Online Evaluation

Table 14 shows a detailed breakdown of cases when participants skipped assistant instructions by method and activity. Recall that participants could skip instructions that were already completed (redundant), were infeasible in the current activity setting, or were irrelevant to the activity at hand. Both the Socratic and VCLM approaches have a similar total amount of skipped instructions and a similar distribution across skip reasons. While redundant skips were the most common type of skip by far, (63% of all skips) the Socratic approach suggested a lot of infeasible actions for the latte activity specifically. Many of these actions were relevant to other latte settings (grinding coffee beans) or actions that would have been completed prior to the start of the activity episode (like setting up the espresso machine). Interestingly, the VCLM approach made significantly fewer infeasible suggestions for the latte task, which may be a product of its direct visual conditioning. However, this lower skip rate did not translate to a higher activity completion rate.

## 13. Study Activity Visualization

Figure 7 visualizes the most successful episode from each of the 3 cooking activities in the online study. Unfortunately, the BLT sandwich activity had no successful episodes so the closest episode is visualized. Figure 8 shows a planning mistake in the latte activity. In the top row, the assistant suggested that the participant add milk before steaming it. The bottom row shows the correct sequence of actions. Grounding mistakes result in skipped actions and were not executed.

## 14. Participant Data Collection Practices

We obtained internal approval to collect egocentric videos from volunteer study participants in our office. We ensured that egocentric data contained no identifiable participant information. The data was stored in a private drive which only the study administrators had access to. The faces of other people who appear in participant videos were also blurred to protect their identities. We have no plans to release the full data beyond what is currently available for visualization purposes.

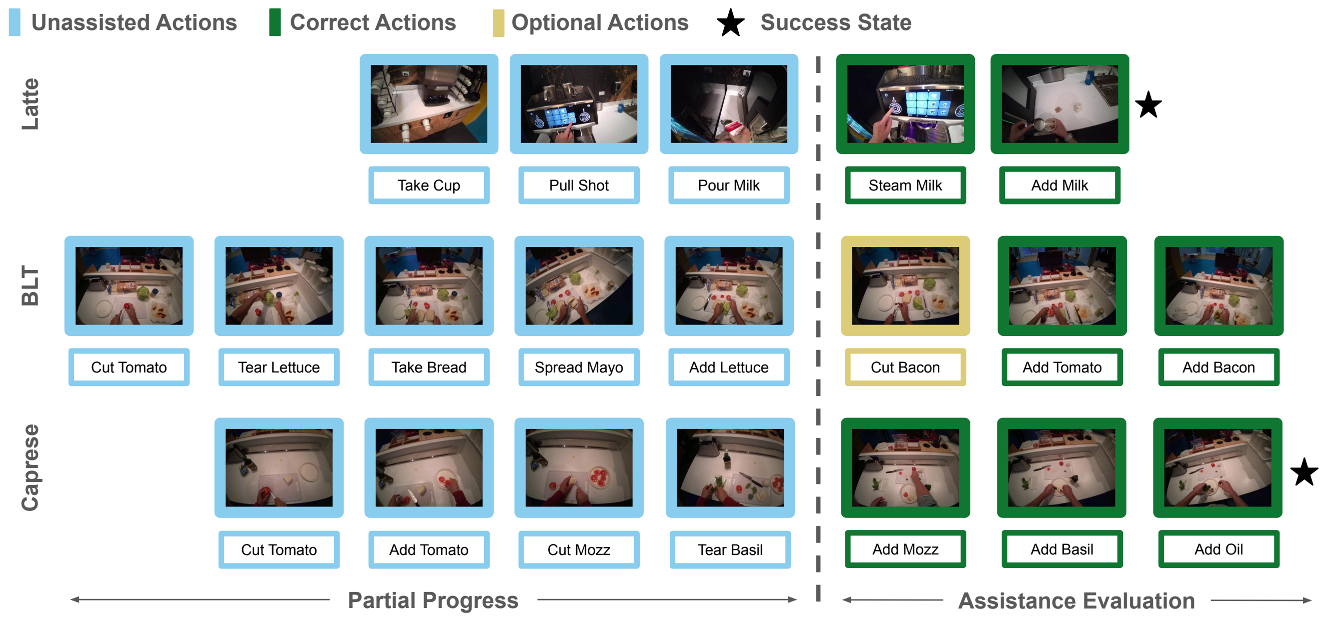

Figure 7. **A visualization of the most successful episodes from each cooking activity.** From top to bottom the activities are: make an espresso latte, make a BLT sandwich, make a caprese salad. No participant successfully accomplished the BLT activity in our study.

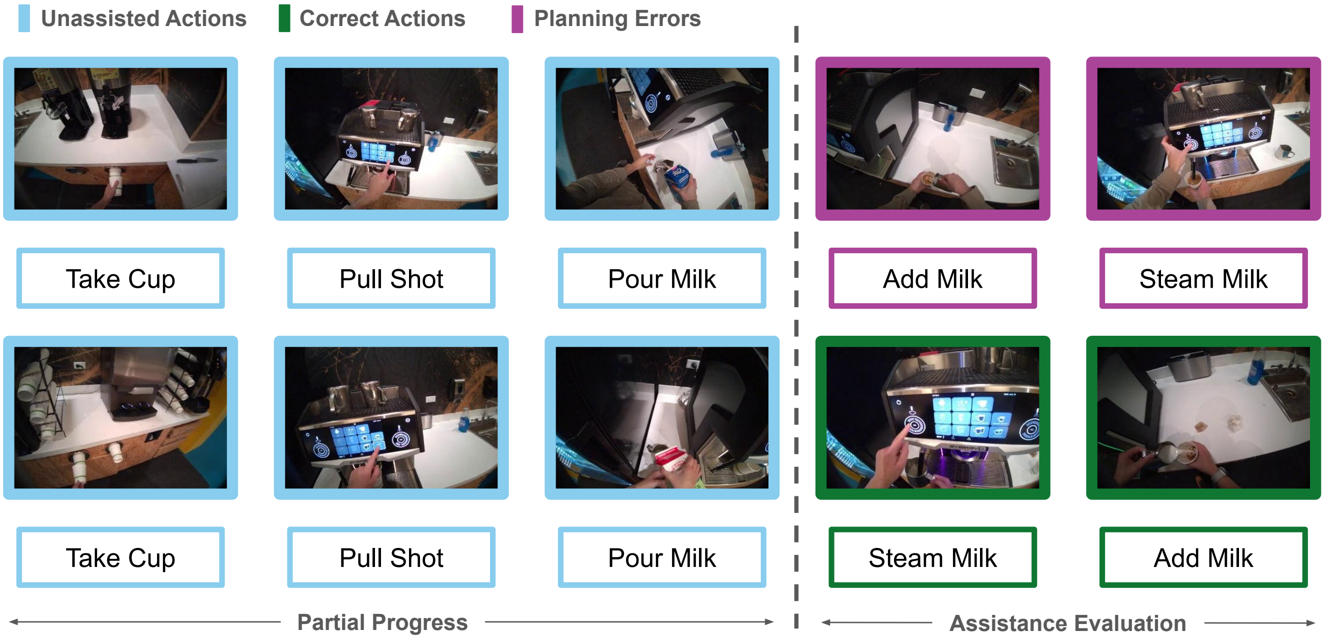

Figure 8. **Visualization of a planning mistake in the latte activity**. Top row shows a planning mistake during the latte activity. Bottom row shows the correct sequence of actions for completing the activity. Optional actions are omitted from the sequence.