Bridging VideoQA and Video-Guided Agentic Tasks via Generalized Keyframe Extraction

Sunqi Fan, Qingle Liu, Runqi Yin, Meng-Hao Guo, and Shuojin Yang✉

Tsinghua University, Beijing, China
{fansq20, lql24, yrq24}@mails.tsinghua.edu.cn,
{gmh, yangshuojin}@tsinghua.edu.cn
Project Page: https://vg-gui-tasker.github.io/

Abstract. Video understanding is a fundamental capability for multimodal intelligence, and recent Multimodal Large Language Models (MLLMs) have achieved remarkable performance on Video Question Answering (VideoQA) benchmarks. However, existing benchmarks primarily evaluate whether models can perceive shallow visual cues, while rarely examining whether MLLMs can learn deeper knowledge or procedural skills from video tutorials and generalize them to downstream long-horizon agentic tasks. To address this gap, we introduce **VG-GUI-Bench** (Video-Guided GUI Benchmark), a new benchmark designed to evaluate whether MLLM-based GUI agents can follow video tutorials to complete corresponding GUI interactive tasks. Furthermore, we observe that the performance of models on both VideoQA and video-guided agentic tasks critically depends on effective keyframe extraction. Based on this observation, we propose **TASKER** (Task-driven And Scene-aware Keyframe searchER), a keyframe extraction algorithm that jointly considers task relevance and scene dynamics to identify informative frames. Experimental results demonstrate that TASKER achieves significant performance improvements on both VideoQA and video-guided agentic task benchmarks, outperforming the best baseline by 2.0% on the EgoSchema fullset and 1.8% on the NExT-QA dataset, respectively. These results further highlight the potential of generalized keyframe extraction methods for video understanding tasks. Our code and data are available at https://github.com/VG-GUI-TASKER/VG-GUI-TASKER.

Keywords: Video Understanding· GUI Agent· Keyframe Extraction

1 Introduction

Multimodal large language models have recently substantially advanced video understanding, achieving strong results on popular VideoQA benchmarks [10, 53]. Despite this progress, the current evaluation paradigm remains largely centered on shallow perceptual cues, such as recognizing objects, attributes, and short-term actions. As a consequence, existing benchmarks provide limited insight into a more fundamental question: can MLLMs learn deeper knowledge or

✉ Corresponding author.

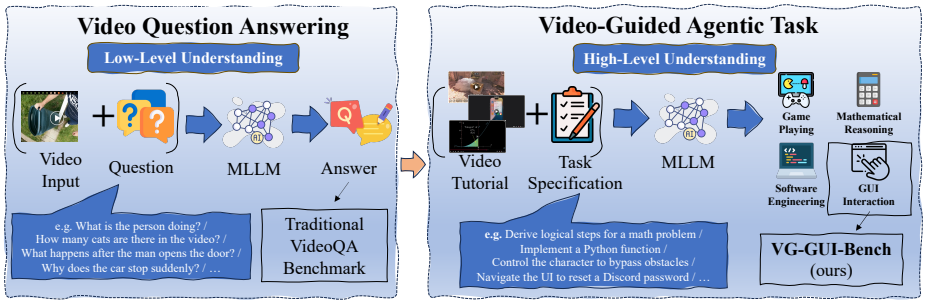

Fig. 1: Demonstration of the 2 progressive levels. This work aims to advance video understanding from the VideoQA paradigm (low-level understanding) toward the Video-Guided Agentic Task paradigm (high-level understanding).

procedural skills from videos and generalize them to solve new tasks that require long-horizon agentic capabilities? This limitation becomes particularly evident in real-world learning scenarios. Video tutorials are widely used to teach complex procedures, ranging from software operations to device configuration and daily-life skills. Solving such tasks requires more than recognizing visual events; it involves understanding step-by-step procedures, abstracting key operations, and transferring them to new environments. In other words, models must be able to extract actionable knowledge from videos and apply it to downstream tasks, which is a capability that can be viewed as a form of Video In-Context Learning.

To better understand this capability gap, we characterize video understanding along two progressive levels, forming a natural hierarchy from perception to action, as in Figure 1:

- **Low-Level: VideoQA.** The foundational task of video understanding, where models must identify temporally relevant moments, comprehend visual evidence, and perform question-conditioned reasoning to extract factual information from videos.
- **High-Level: Video-Guided Agentic Tasks.** A more advanced setting where models are expected to learn procedural knowledge from video demonstrations and transfer it to long-horizon decision making and execution. For example, a model may watch a tutorial on *"how to change a Discord account password"* and translate the demonstrated procedure into step-by-step actions within a structured Graphical User Interface (GUI) environment.

To further evaluate MLLMs' high-level video understanding ability, we introduce VG-GUI-Bench (Video-Guided GUI Benchmark), a benchmark designed to evaluate whether MLLM-based GUI agents can follow video tutorials to complete semantically related tasks. In VG-GUI-Bench, each tutorial video is paired with a corresponding GUI task that requires similar procedural knowledge. This pairing enables systematic evaluation of whether models can extract

procedural steps from videos and generalize them to new interactive tasks, providing a practical testbed for studying video in-context learning.

Beyond benchmark design, our study reveals a shared bottleneck across both VideoQA and video-guided agentic tasks: model performance critically depends on how the model identifies and attends to task-relevant temporal content within videos. Long videos often contain redundant or irrelevant segments, while key procedural evidence may appear only briefly. As a result, naive frame sampling strategies either miss critical moments or introduce excessive redundancy, leading to degraded reasoning and inefficient computation. For example, on the NExT-QA benchmark, using GPT-4o with a frame selection strategy improves accuracy by approximately 15% compared to GPT-4o with uniform sampling, with the same number of frames.

To tackle this challenge, we propose TASKER (Task-driven And Scene-aware Keyframe searchER), a generalized keyframe extraction method inspired by traditional graph search algorithms. TASKER combines task-driven relevance estimation with scene-aware temporal dynamics to select a compact yet informative set of frames that preserves essential procedural evidence while discarding redundant content. This design allows TASKER to operate effectively across both VideoQA and video-guided agentic tasks, providing a unified mechanism for temporal information selection.

Extensive experiments demonstrate that TASKER achieves both high accuracy and strong frame efficiency across VideoQA and video-guided agentic task benchmarks, consistently outperforming prior temporal selection and video-agent methods such as VideoTree and VideoAgent. In our experiments, TASKER achieves 63.1% accuracy on EgoSchema fullset [29] (surpassing the best baseline by 2.0%) and 77.4% average accuracy on NExT-QA [55] (surpassing the best baseline by 1.8%). In Figure 4, we compare the frame efficiency of TASKER with two baselines, highlighting its ability to effectively identify key information. These results highlight the effectiveness of generalized keyframe extraction as a mechanism for bridging VideoQA and video-guided agentic tasks.

Our contributions can be summarized as follows:

- We identify a key limitation of existing video benchmarks and propose a two-level taxonomy that connects VideoQA with video-guided agentic tasks, highlighting the role of video in-context learning.
- We introduce VG-GUI-Bench, a new benchmark that pairs video tutorials with GUI agent tasks to evaluate procedural knowledge transfer from videos.
- We propose TASKER, a task-driven and scene-aware keyframe extraction algorithm that improves both accuracy and frame efficiency across VideoQA and video-guided agentic tasks.

2 Related Work

Video Question Answering VideoQA is a core subtask of video understanding that evaluates a model's ability to reason over temporal visual content in

response to language queries [31, 55, 67]. Recent benchmarks increasingly emphasize long-form videos and complex reasoning [10, 11, 29, 32, 53, 61, 68]. Early approaches relied on CNN-based visual encoders and lightweight fusion modules [3, 13, 14, 20, 43, 44]. With the emergence of LLMs, recent methods typically integrate pretrained visual encoders with projection layers and large language models for reasoning and generation [22, 23, 37, 40, 48, 52, 64, 65]. Beyond end-to-end Video-LLMs, agent-based frameworks [42, 54] further enhance reasoning through prompting, memory, tool use, and planning [7–9, 12, 18, 63].

Keyframe Extraction Efficiently identifying informative frames is crucial to understanding long videos. Existing methods adopt diverse strategies, including learned frame selection [33], attention-based segment extraction [58], uniform sampling with image-grid modeling [59], and recursive agent-guided selection [47]. VideoTree [51] constructs a static hierarchical tree via feature clustering and performs LLM-guided search over the tree. **Compared with VideoTree's precomputing features for all frames, we selectively extract information during the search process, enabling question-aware tree construction with reduced computational overhead. Our primary difference from VideoAgent [47] lies in that our frame selection strategy explicitly accounts for intra-video scene transitions as well as the intrinsic structural organization of the video.**

Video-Guided Tasks Video-guided tasks require models to acquire a task or skill by reasoning over instructional videos [15, 16, 50, 57, 60]. For example, Mobile-Agent-V enables agents to acquire GUI operation knowledge from instructional videos and apply it to new tasks [45]. On the data and evaluation side, the community has contributed a range of datasets [17, 27, 28, 41, 66] and benchmarks [5, 6, 24–26]. **Compared with TongUI [62] and Watch-and-Learn [39], which focus on converting videos to learnable trajectories, our TASKER method works at the frame selection level as a training-free module for both VideoQA and downstream agentic tasks.**

3 Method

In the method section, we first present the details of VG-GUI-Bench, followed by the TASKER algorithm. The preliminaries on classical graph search algorithms are provided in **Appendix A**.

3.1 VG-GUI-Bench

Despite growing interest in video-guided GUI agents, there remains a lack of high-quality benchmarks for evaluating MLLMs on long-horizon GUI task execution from video tutorials. To fill this gap, we construct VG-GUI-Bench (Video-Guided GUI Benchmark), a dedicated benchmark for systematically assessing MLLM performance on video-guided, long-horizon GUI tasks.

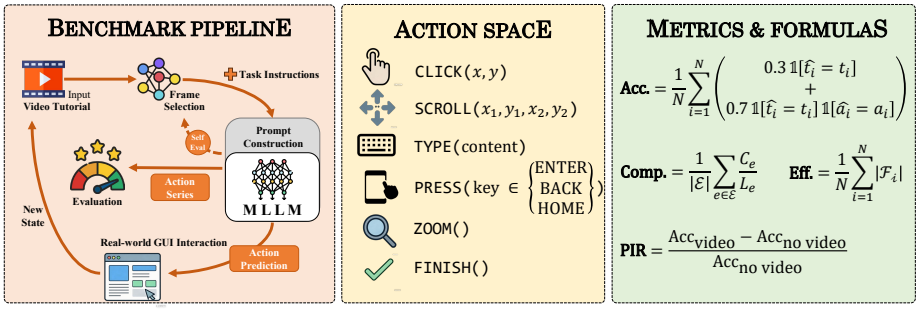

Fig. 2: Overview of the VG-GUI-Bench benchmark, including benchmark pipeline, action space, metrics and formulas.

Data Source We build upon the high-quality dataset provided by MONDAY [17], from which we obtain input tutorial videos, ground-truth action sequences, and keyframe screenshots as evaluation references. We further design task-specific prompts to guide the model in generating predicted actions at each step. On average, each episode contains 10.71 steps, making VG-GUI-Bench a genuinely long-horizon benchmark that requires sustained reasoning over extended action sequences. In total, VG-GUI-Bench has 1,000 test cases.

Action Space Previous works often adopt inconsistent and ad-hoc action naming conventions, lacking a unified standard and a scientifically grounded taxonomy. To address this, we define a standardized action space comprising the following six action types:

- `CLICK(x, y)`: Perform a tap at the coordinate (x, y).
- `SCROLL(x1, y1, x2, y2)`: Perform a swipe or drag gesture from (x_1, y_1) to (x_2, y_2), covering directional interactions that require two coordinate pairs.
- `TYPE(content)`: Input a text string specified by the `content` argument.
- `PRESS(key)`: Press a system-level key, where `key` $\in$ {`BACK`, `HOME`, `ENTER`}. These keys carry distinct semantic meanings on mobile platforms and cannot be substituted by other operations.
- `ZOOM()`: Perform a pinch-to-zoom gesture. This action takes no arguments.
- `FINISH()`: Indicate that the task has been completed. This action takes no arguments.

Evaluation Pipeline As illustrated in Figure 2, the evaluation pipeline of VG-GUI-Bench operates as follows. The input tutorial video is first processed by a frame selection module, which extracts relevant keyframes from the video. The selected frames, together with the task instruction, are then used to construct a prompt that is fed into the MLLM to predict the next GUI action. The predicted action is executed in the GUI environment, producing a new interaction state. Based on this state, the model continues predicting subsequent actions step by step. The predicted actions are finally evaluated against the ground-truth action

sequence by VG-GUI-Bench. This process repeats iteratively until the episode terminates.

Evaluation Metrics We propose four complementary metrics to provide a comprehensive assessment:

- **Accuracy** measures the correctness of individual action predictions. A prediction receives a score of 0.3 if the action type is correct, and an additional 0.7 if the action arguments are also correct. For argument-free actions (`ZOOM`, `FINISH`), a correct type prediction yields the full score. In addition to the overall accuracy, we also report the type accuracy and score breakdown for each individual action category to provide finer-grained insights into model behavior.

$$\text{Acc.} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left(\hat{t}_i = t_i\right)\left(0.3 + 0.7 \times \mathbb{1}\left(\hat{a}_i = a_i\right)\right) \tag{1}$$

  where N denotes the total number of steps, t_i and $\hat{t}_i$ are the ground-truth and predicted action types, and a_i and $\hat{a}_i$ are the ground-truth and predicted action arguments.
- **Completion** measures the proportion of correctly executed steps within each episode, averaged across all episodes:

$$\text{Comp.} = \frac{1}{|\mathcal{E}|}\sum_{e\in\mathcal{E}} \frac{C_e}{L_e} \tag{2}$$

  where $\mathcal{E}$ denotes the set of all episodes, C_e is the number of correctly executed steps in episode e, and L_e is the total number of steps in episode e.
- **Efficiency** measures the average number of input frames consumed per prediction step, reflecting the computational cost of the frame selection strategy:

$$\text{Eff.} = \frac{1}{N}\sum_{i=1}^{N} |\mathcal{F}_i| \tag{3}$$

  where $|\mathcal{F}_i|$ denotes the number of frames provided to the MLLM at step i.
- **Performance Improvement Rate (PIR)** measures how much the model can learn from the video:

$$\text{PIR} = \frac{\text{Acc}_{\text{video}} - \text{Acc}_{\text{no video}}}{\text{Acc}_{\text{no video}}} \tag{4}$$

  where $\text{Acc}_{\text{no video}}$ and $\text{Acc}_{\text{video}}$ denote the accuracy without and with video tutorial input respectively.

3.2 TASKER Algorithm

In this section, we first define the search objective, nodes, cost function, and termination conditions in our TASKER algorithm, providing a comprehensive

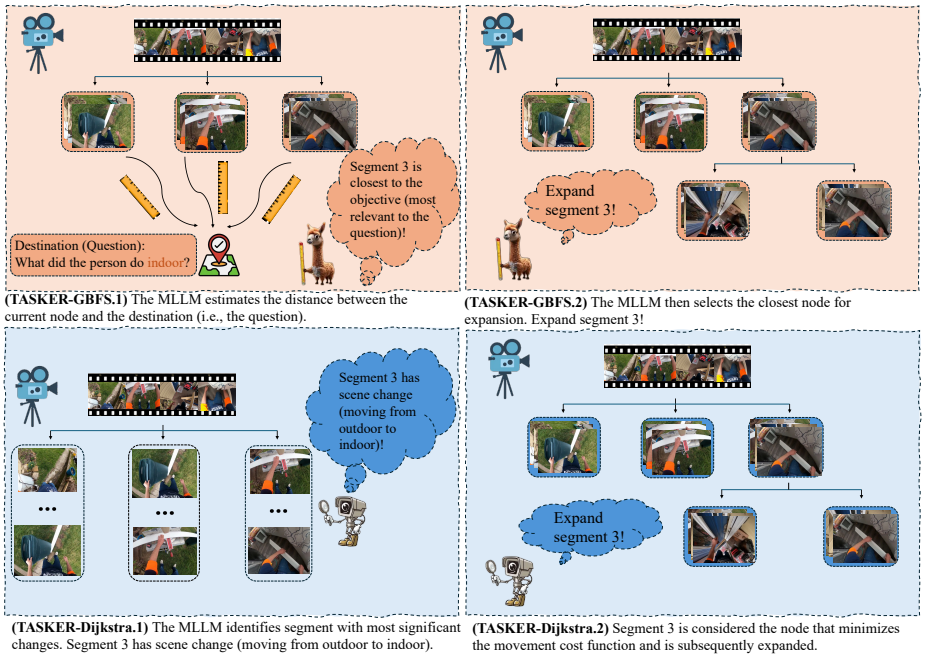

Fig. 3: Illustration of TASKER's cost function evaluation and node expansion steps. TASKER-GBFS variant evaluates distance based on question relevance. TASKER-DIJKSTRA variant evaluates distance based on scene dynamics.

overview of the tree-structured keyframe search process. We also explain how the algorithm utilizes the retrieved information to answer questions. The key steps of TASKER (leveraging MLLMs to evaluate the cost function and node expansion) are illustrated in Figure 3.

Search Objective The search objective is to identify a sufficient set of keyframes whose combined information is sufficient to answer the question or complete the agentic task. The ultimate goal is to answer the question.

Nodes Expansion In TASKER algorithm, we divide the video into multiple video segments, with each video segment representing a node. The initial node $\mathcal{N}_0$ is the entire video, which is first uniformly split into M segments, where M is a tunable hyper-parameter. These video segments are then put into an open list $\mathcal{L}$. The next node to be expanded, or the next video segment to be processed, is selected based on the cost function $f(n)$ we define. The expansion process means further subdividing the selected video segment. In this work, we perform a binary split on the segment for node expansion.

Answer Prediction We define the first and last frames of all current video segments as **Visible Frames** $\mathcal{F}_v$. They are connected to each other, meaning that the last frame of one video segment is the first frame of the next. We can fully utilize the information in the visible frames, while the information in the

other frames is temporarily inaccessible. For the visible frames, we can employ two approaches: directly inputting the frames into the MLLM or first generating captions and then performing reasoning in the textual modality. In either way, we predict an answer or action based on the information from the visible frames.
Cost Function Leveraging the evaluation capability of the MLLMs, we design different cost functions based on different basic search algorithms.

- **TASKER-GBFS: Task-Driven Cost Function** In Greedy Best-First Search (GBFS), the cost function $h(n)$ represents the distance from the current node to the destination. Accordingly, we let the MLLM evaluate the current visible frame's information and identify what visual information is missing for answering the question. The missing information can be seen as the distance between the current node and the destination. GBFS algorithm selects the node with the smallest $h(n)$ to expand. In our adaptation, the MLLM attempts to identify where the missing visual information is likely located between which two specific invisible frames, determining which video segments should be expanded. This leads to a variant of the TASKER algorithm named TASKER-GBFS.
- **TASKER-Dijkstra: Scene-Aware Cost Function** In Dijkstra's algorithm, the cost function $g(n)$ represents the cost of reaching the current node from the start point. In our adaptation, the MLLM evaluates the visible frames to identify the video segment with the most significant scene change, such as transitions in scenes, figures, or activities between the first and last frames of a segment. Segments with larger scene changes are considered more informative and thus prioritized for further exploration, corresponding to nodes with smaller $g(n)$. Notably, in classical Dijkstra's algorithm, the cost function does not depend on the destination; similarly, in TASKER-Dijkstra, the MLLM is not driven by the task or question, but instead selects keyframes based on scene changes and the intrinsic structure of the video, making the search purely scene-aware.
- **TASKER-A*: Task Driven & Scene Aware Cost Function** For the A* Algorithm, the cost function is the sum of the heuristic evaluation function and the movement cost function, i.e., $f(n) = h(n) + g(n)$, which means that A* Algorithm takes into account both the distance from the current node to the destination and the distance from the start point to the current node. Correspondingly, in our TASKER-A* variant, the MLLM must simultaneously consider two factors: (1) which video segment is likely to contain the missing information, and (2) which video segment exhibits the most significant scene change. Only video segments that satisfy both are prioritized for expansion.
- **TASKER-BFS: Naive Cost Function** We also propose a naive algorithm variant, TASKER-BFS which does not rely on a MLLM to evaluate the cost function. Instead, it performs a breadth-first expansion, continually splitting all the existing video segments (in the case of no pruning). Like BFS, TASKER-BFS advances in a wave-like manner, steadily progressing. This variant is suitable for situations where a MLLM cannot be accessed,

Algorithm 1 Task-driven And Scene-aware Keyframe searchER (TASKER)

Require: Video v, question q, MLLM F, confidence threshold C, max iteration T, uniform sampling size M, beam size B, frozen frame set $\mathcal{S}_{\text{frozen}}$
Ensure: Answer $\hat{y}$, keyframes $\{\mathcal{F}_k\}$

$\mathcal{N}_0 \leftarrow v$
$\mathcal{N}_1, \mathcal{N}_2, ..., \mathcal{N}_M \leftarrow \text{UniformSegment}(\mathcal{N}_0, M)$
$\mathcal{L} \leftarrow \{\mathcal{N}_0, \mathcal{N}_1, ..., \mathcal{N}_M\}$
$\mathcal{S}_{\text{frozen}} \leftarrow \emptyset$
$t \leftarrow 1$
while $t \leq T$ **do**
  $\{\mathcal{F}_v\} \leftarrow \text{ExtractVisibleFrames}(\mathcal{L})$
  $\hat{y} \leftarrow \text{PredictAnswer}(F, \{\mathcal{F}_v\}, q)$
  $c_1, c_2 \leftarrow \text{EvaluateConfidence}(F, \hat{y}, \{\mathcal{F}_v\}, q)$
  if $c_1 \geq C$ and $c_2 \geq C$ **then**
    break
  else
    $\{p\} \leftarrow \text{EvaluateCostFunction}(F, \{\mathcal{F}_v\}, \mathcal{L} \setminus \mathcal{S}_{\text{frozen}})$
    $\mathcal{L} \leftarrow \text{SelectAndExpandNodes}(\mathcal{L}, \{p\}, B)$
    $\mathcal{L}, \mathcal{S}_{\text{frozen}} \leftarrow \text{ValidateNewFrame}(\mathcal{L}, \mathcal{S}_{\text{frozen}}, F, q)$
  end if
  $t \leftarrow t + 1$
end while
$\{\mathcal{F}_k\} \leftarrow \{\mathcal{F}_v\}$
return $\hat{y}$, $\{\mathcal{F}_k\}$

or where the overhead introduced by the MLLM is less of a concern, with a greater emphasis on ensuring no information is overlooked.

We do not intend to introduce TASKER-DFS variant, as its depth-first expansion focuses on a single initial segment. Without strict termination conditions, it is prone to falling into local optima.

Termination Condition and Confidence Estimation Traditional search algorithms rely on deterministic termination conditions, typically defined by whether the search objective has been reached. In contrast, for keyframe search in VideoQA and Video-Guided Agentic Tasks, such conditions are difficult to define, as it is unclear whether sufficient information has been collected. To address this issue, we leverage the reflection and self-evaluation capabilities of MLLMs to estimate the confidence of the predicted answer and determine whether to terminate the search. TASKER stops when the model produces a sufficiently confident prediction. Specifically, we combine two confidence estimation methods through a voting mechanism, as described below.

– **Self-Evaluation and Self-Reflection** MLLMs can be instructed to self-evaluate their responses, reflecting on potential shortcomings in their responses [35,38]. Therefore, after generating an answer, we input the question, information of visible frames, and the MLLM's previous reasoning chain and

predicted answer back into the model. The MLLM then assesses the accuracy and reliability of its previous answer and output a confidence score (c_1).

– **Temporal Summarization** The captions of the sampled frames are discrete. To integrate the sampled frames along the temporal dimension, we instruct the MLLM to summarize their captions to form a cohesive overview of the video. We use few-shot examples [2] to generate a more accurate and detailed video summary. Then we prompt the MLLM to predict the answer and output a confidence score (c_2) based on the summary. The advantage of this approach is to consider the sampled frames in a complete temporal context rather than in isolation.

We employ a voting mechanism to ensemble the above two methods. The search process only terminates when both methods independently determine that they have sufficient confidence ($c_1 \geq C$ and $c_2 \geq C$, C is the threshold).

Additionally, after each expansion, we apply a **frame validation** step: each newly revealed frame is first checked for visual redundancy against existing visible frames, and then assessed by the MLLM for task relevance. Redundant or irrelevant frames are discarded (with nearby relevant replacements searched when possible), and segments that yield only redundant frames are added to a frozen set $\mathcal{S}_{\text{frozen}}$ to avoid repeated exploration. We summarize TASKER algorithm in Algorithm 1.

4 Experiments

Our experiments are conducted on two VideoQA benchmarks, EgoSchema [29] and NExT-QA [55], as well as the proposed video-guided agentic task benchmark, VG-GUI-Bench. Further details of EgoSchema and NExT-QA benchmarks can be found in **Appendix B**.

4.1 Main Results

VideoQA Results We compare the performance of TASKER with various related approaches on LLM-driven VideoQA using TASKER-A* variant. Most of the baselines are mentioned in the related work and **Appendix A**. Implementation details are provided in **Appendix B**. Prompts we use are listed in **Appendix C**. Table 1 demonstrates that TASKER significantly outperforms all these baselines. Specifically, TASKER (with GPT-4 as base LLM) achieves 63.1% accuracy on EgoSchema fullset (surpassing the best baseline by 2.0%) and 77.4% accuracy on NExT-QA (surpassing the best baseline by 1.8%). Results based on the open-source MLLM (Qwen3-VL-235B-A22B-Instruct) also show that TASKER consistently outperforms VideoTree [51] and VideoAgent [47].

Moreover, TASKER operates in a training-free, zero-shot setting, while it still outperforms training-based methods such as LVNet [33] and Vamos [46]. Meanwhile, TASKER processes only visible frames, for instance, achieving the reported performance requires only about 15% of the total frames. In contrast, methods like LangRepo [19] and LifelongMemory [49] process all frames without selection.

Table 1: Comparison between TASKER and other methods. We highlight the gain of our method over VideoTree [51] in blue.

Model	(M)LLM	EgoSchema		NExT-QA			
		Sub.	Full	Tem.	Cau.	Des.	Avg.
Based on Open-source Captioners and Open-source LLMs							
MVU [34]	Mistral-13B	60.3	37.6	55.4	48.1	64.1	55.2
LangRepo [19]	Mixtral-8x7B	66.2	41.2	51.4	64.4	69.1	60.9
Video-LLA+INTP [36]	Vicuna-7B v1.5	-	38.6	58.6	61.9	72.2	62.7
Based on Proprietary MLLMs							
IG-VLM [21]	GPT-4V	59.8	-	63.6	69.8	74.7	68.6
LVNet [33]	GPT-4o	68.2	61.1	65.5	75.0	81.5	72.9
Based on Open-source Captioners and Proprietary LLMs							
ProViQ [4]	GPT-3.5	57.1	-	-	-	-	64.6
MoReVQA [30]	PaLM-2	-	51.7	64.6	70.2	-	69.2
Vamos [46]	GPT-4	51.2	48.3	-	-	-	-
LLoVi [63]	GPT-4	61.2	-	61.0	69.5	75.6	67.7
VideoAgent [47]	GPT-4	60.2	54.1	64.5	72.7	81.1	71.3
VideoAgent [9]	GPT-4	62.8	60.2	-	-	-	-
LifelongMemory [49]	GPT-4	64.1	58.6	-	-	-	-
VideoTree [51]	GPT-4	66.2	61.1	70.6	76.5	83.9	75.6
TASKER (Ours)	GPT-4	**68.0** (1.8 ↑)	**63.1** (2.0 ↑)	**72.3** (1.7 ↑)	**78.2** (1.7 ↑)	**85.4** (1.5 ↑)	**77.4** (1.8 ↑)
TASKER (Ours)	GPT-4o	**68.6** (2.4 ↑)	**63.6** (2.5 ↑)	**72.9** (2.3 ↑)	**79.0** (2.5 ↑)	**86.1** (2.2 ↑)	**78.1** (2.5 ↑)
Based on Open-source MLLMs							
VideoAgent [47]	Qwen3-VL	75.8	74.6	79.7	83.5	86.5	82.8
VideoTree [51]	Qwen3-VL	77.2	76.7	81.9	85.1	86.3	84.3
TASKER (ours)	Qwen3-VL	**79.4** (2.2 ↑)	**77.3** (0.6 ↑)	**83.6** (1.7 ↑)	**85.3** (0.2 ↑)	**87.5** (1.2 ↑)	**85.1** (0.8 ↑)

Additionally, as shown in Figure 4, we compare TASKER's frame efficiency with other keyframe extraction methods in the same condition on EgoSchema [29] subset. **At the same accuracy level (66%), TASKER uses only about 1/4 of the frames required by VideoTree.** Moreover, VideoTree clusters features of all frames during preprocessing, whereas TASKER only has access to visible frames and does not utilize information from the rest. The results show that TASKER utilizes frames more efficiently than LLoVi [63], VideoAgent [47] and VideoTree [51], demonstrating its superior ability to identify key information.

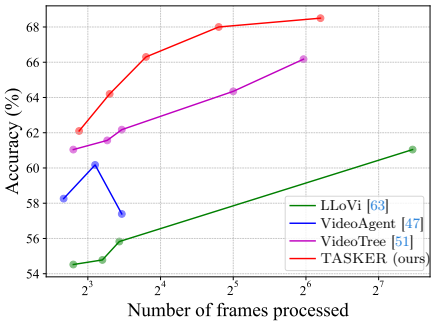

Fig. 4: Demonstration of TASKER's high frame efficiency. When processing the same number of video frames with the same (M)LLM, TASKER achieves higher QA accuracy.

Video-Guided Agentic Task Results In Table 2, we provide a VG-GUI-BENCH leaderboard with 7 frontier models. Gemini-3.1-Pro consistently achieves the best overall accuracy, regardless of whether no video input is used or 10 frames are uniformly sampled from the video. Across most models, incorporating 10 uniformly sampled frames consistently improves performance over the

Table 2: VG-GUI-Bench leaderboard.

Rank	Model	Mode	Acc.(%)	Type Acc.(%)	Comp.(%)	PIR
1	Gemini-3.1-Pro	No Video	58.51	74.58	78.53	–
		10 Uniform Frames	61.68	76.25	78.61	0.054
2	GPT-5-mini	No Video	55.76	71.93	75.97	–
		10 Uniform Frames	58.40	75.27	78.96	0.047
3	Kimi-K2.5	No Video	57.22	72.91	76.31	–
		10 Uniform Frames	58.22	74.78	78.72	0.017
4	Claude-Sonnet-4.6	No Video	44.35	67.81	72.24	–
		10 Uniform Frames	45.80	67.71	72.35	0.033
5	Seed-2.0-Pro	No Video	35.93	70.66	74.75	–
		10 Uniform Frames	39.78	75.96	79.36	0.107
6	Qwen3-VL-235B-A22B	No Video	25.90	66.63	69.73	–
		10 Uniform Frames	26.88	67.42	71.69	0.038
7	Gemini-3.1-Flash	No Video	24.73	67.03	70.54	–
		10 Uniform Frames	26.02	69.19	72.60	0.052

Table 3: TAKSER and Baselines' Results on VG-GUI-Bench. We report Acc., Type Acc., per-action scores, Comp., Eff., and PIR. For the keyframe selection methods, best and second-best results are in red and blue. For all methods, we use Qwen3-VL-235B-A22B-Instruct [1] as the base LLM.

Method	Overall(%)		Per-Action Score(%)						Comp.(%)	Eff. ↓	PIR
	Acc.	Type Acc.	CLICK	SCROLL	TYPE	PRESS	ZOOM	FINISH			
Reference Baselines											
No Video	25.32	65.85	26.89	26.18	3.66	29.23	0	0	69.03	0	-
All Keyframes	37.21	65.75	49.48	2.38	25.81	5.26	0	0	72.01	13.23	0.470
Uniform Sampling	39.82	66.34	52.90	5.25	15.71	6.50	0	0	70.64	10.88	0.573
Oracle Keyframes	44.32	73.31	60.34	1.92	0	0	0	0	76.32	1	0.750
Keyframe Selection Methods											
VideoTree [51]	40.79	67.52	53.41	7.83	14.75	6.50	0	0	71.93	10.00	0.611
VideoAgent [47]	39.86	67.03	51.61	10.70	13.81	5.26	0	0	71.17	5.12	0.574
TASKER-BFS	40.01	69.97	51.87	9.66	17.16	5.26	0	0	73.77	6.10	0.580
TASKER-GBFS	40.26	69.97	52.10	9.98	16.75	5.26	0	0	73.75	5.81	0.590
TASKER-Dijkstra	40.75	71.05	52.84	9.66	17.48	5.26	0	0	74.39	5.88	0.609
TASKER-A*	40.96	67.71	53.68	8.71	13.81	6.50	0	0	71.38	8.24	0.618

no-video setting, demonstrating the benefit of temporal visual information for GUI understanding. Notably, Seed-2.0-Pro exhibits the largest gain in overall accuracy, improving from 35.93% to 39.78%.

Our TASKER method's results on VG-GUI-Bench are presented in Table 3. We consider the following four baselines as references:

- **No Video**: The model performs the GUI agent task without taking the video tutorial as input. Since it cannot refer to the tutorial, this baseline yields the weakest performance.
- **All Keyframes**: We provide the set of tutorial frames corresponding to all moments where action behaviors occur. With access to these references, the model can closely follow the demonstrated procedure and achieves relatively strong results.

– **Uniform Sampling**: We uniformly sample frames from the full video. Although this does not guarantee coverage of key instructional moments, the stable global context allows it to outperform the *All Keyframes* baseline.
– **Oracle Keyframe**: We provide the specific tutorial frame corresponding to the current GUI action with Set-of-Mark [57] annotations, allowing the model to "peek at the answer". While this enables high scores by copying visual targets, it encourages over-reliance on visual imitation, causing complete failure on operations like `TYPE` and `PRESS`.

We also compare against representative keyframe selection methods, VideoTree and VideoAgent. Overall, our TASKER framework demonstrates excellent performance. Specifically, TASKER-A* achieves the highest Overall Acc. (40.96), PIR (0.618), and `CLICK` score (53.68), outperforming the strong VideoTree baseline. Furthermore, compared to other dynamic selection methods, TASKER effectively deduplicates redundant frames, yielding superior accuracy with higher efficiency (fewer frames per step). Beyond accuracy, our approach attains high task completion rates, with TASKER-Dijkstra (74.39) closely approaching the *Oracle Keyframe* upper bound. Finally, the varied search strategies demonstrate the framework's internal diversity, robustness, and strong extensibility.

In addition to the VG-GUI-Bench benchmark, we also collected accompanying videos for OSWorld [56] and evaluated whether the TASKER method can improve the capabilities of GUI agents on OSWorld. Detailed results are provided in **Appendix G**.

4.2 Ablation Studies

Basic Search Algorithms In Table 3, we have already compared various variants of the TASKER algorithm and observed that TASKER-A* achieves the best performance.

In Table 4, we further investigate performance and frame efficiency of several TASKER algorithm variants with different base search algorithms on the EgoSchema subset. The frame efficiency is measured by the number of visible frames. Specifically, on EgoSchema dataset, all videos are three minutes long, and we set the initial frame rate fps = 1, which means the initial overall frame number is 180. We observe that TASKER-A* achieves the highest accuracy. TASKER-BFS ranks last in accuracy and has the lowest frame efficiency, as TASKER-BFS algorithm exhaustively explores all branches, leading to higher exploration costs. TASKER-GBFS slightly outperforms TASKER-Dijkstra on both metrics, while TASKER-A* combines the strengths of them, significantly improving accuracy with only a slight compromise in frame efficiency. This demonstrates that efficient keyframe localization requires both the heuristic search function and the movement cost function.

Table 4: Ablation on basic search algorithms. We highlight the improvement of TASKER-A* over the naive TASKER-BFS, emphasizing the role of cost-function evaluation.

Algorithm	Accuracy	# Visible
TASKER-BFS	64.7	31.2
TASKER-GBFS	67.0	**27.3**
TASKER-Dijkstra	66.8	27.6
TASKER-A*	**68.0** (**3.3** ↑)	27.9

Termination Condition In Table 5, we conduct ablation experiments on the termination condition of the search process, using TASKER-A* on the EgoSchema subset.

The results show that self-evaluation & self-reflection and temporal summarization assess information sufficiency from different perspectives. When combined, they enhance the reliability of confidence estimation, leading to improved algorithm performance.

Table 5: Ablation on termination condition

Method	Accuracy	# Visible Frames
Self-Evaluation	67.4	**27.4**
Summarization	67.3	28.2
Vote	**68.0**	27.9

Base LLM We also conducted an ablation study on the base LLM of the TASKER algorithm in Table 6.

We find that GPT-4o achieves the best performance as the base LLM. In contrast, reasoning models such as o3-mini and Deepseek-R1 perform slightly worse than GPT-4o, likely due to the relatively straightforward nature of visual reasoning in our tasks.

Table 6: Ablation on different base LLMs

Base LLM	Accuracy	# Visible Frames
GPT-4	68.0	27.9
GPT-4o	**68.6**	**26.7**
o3-mini	67.3	28.3
Deepseek-R1	67.6	26.9
LLaMA-3.3-70B	65.2	27.4

5 Analysis

5.1 Comparison with Video-LLMs

Compared with end-to-end Video-LLMs, the keyframe-based training-free method (e.g., TASKER) offers advantages in terms of training cost, inference efficiency, and interpretability. A detailed comparison with representative Video-LLMs, including performance and resource requirements, is provided in **Appendix C**.

5.2 Visualization

Figure 5 presents a visualized case study. In this 3-minute video, the key information for answering the question is located between 126s and 130s. Our TASKER algorithm precisely identifies this critical video segment by searching along the video tree and expanding relevant nodes. And it retrieves all frames within the 125s–130s range as keyframes, successfully answering the question. In the video tree, we mark the nodes traversed by the key search path in yellow and mark the final leaf node obtained from the key search path in green. The nodes outside the key search path are barely expanded.

6 Conclusion

In this work, we take a step toward deeper video understanding by connecting low-level VideoQA with higher-level video-guided agentic tasks. Specifically, we

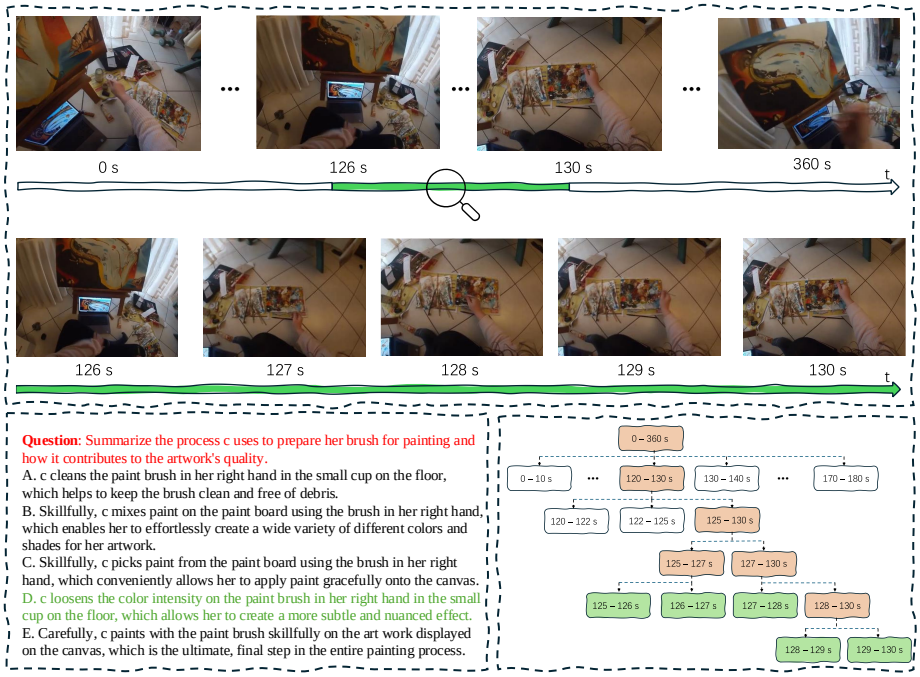

Fig. 5: Visualization of tree-search and nodes expansion process of TASKER method solving a VideoQA case from EgoSchema [29].

introduce VG-GUI-Bench, a benchmark that pairs tutorial videos with corresponding GUI interaction episodes to evaluate whether MLLM-based agents can extract procedural knowledge from videos and transfer it to long-horizon decision making. Building on the shared bottleneck of temporal content selection across both settings, we propose TASKER, a task-driven and scene-aware keyframe search algorithm that formulates keyframe extraction as a generalized graph-search problem, with MLLMs used to evaluate cost functions and termination confidence. Extensive experiments on VideoQA benchmarks demonstrate that TASKER consistently improves accuracy while using substantially fewer frames than prior keyframe-based baselines, highlighting a practical performance–efficiency trade-off in a training-free regime.

7 Acknowledgement

This work was supported by Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (No. JYB2025XDXM101) and the National Natural Science Foundation of China (project No. 62495061, 62220106003).

References

1. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang, F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., Zhu, K.: Qwen3-VL technical report (2025), https://arxiv.org/abs/2511.21631 12
2. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual (2020), https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html 10
3. Carreira, J., Zisserman, A.: Quo vadis, action recognition? A new model and the kinetics dataset. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 4724–4733. IEEE (2017). https://doi.org/10.1109/CVPR.2017.502 4
4. Choudhury, R., Niinuma, K., Kitani, K.M., Jeni, L.A.: Video question answering with procedural programs. In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G. (eds.) Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XXXVIII. Lecture Notes in Computer Science, vol. 15096, pp. 315–332. Springer (2024). https://doi.org/10.1007/978-3-031-72920-1_18 11
5. Dong, Y., Tian, S., Liu, S., Ding, S., Zang, Y., Dong, X., Cao, Y., Wang, J., Liu, Z.: Demo-ICL: In-context learning for procedural video knowledge acquisition (2026), https://arxiv.org/abs/2602.08439 4
6. Dou, S., Zhang, M., Yin, Z., Huang, C., Shen, Y., Wang, J., Chen, J., Ni, Y., Ye, J., Zhang, C., Xie, H., Hu, J., Wang, S., Wang, W., Xiao, Y., Liu, Y., Xu, Z., Guo, Z., Zhou, P., Gui, T., Wu, Z., Qiu, X., Zhang, Q., Huang, X., Jiang, Y.G., Wang, D., Yao, S.: CL-bench: A benchmark for context learning (2026), https://arxiv.org/abs/2602.03587 4
7. Fan, S., Cui, J., Guo, M., Yang, S.: Tool-augmented spatiotemporal reasoning for streamlining video question answering task. In: Belgrave, D., Zhang, C., Montoya, L.N., Lin, H., Pascanu, R., Koniusz, P., Ghassemi, M., Chen, N., Ruíz, I.V.M., Loaiza-Bonilla, A. (eds.) Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diago, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025 (2025), http://papers.nips.cc/paper_files/paper/2025/hash/bbdd9525c5ecd3a03fee43d8fd64bd83-Abstract-Conference.html 4
8. Fan, S., Guo, M.H., Yang, S.: Agentic keyframe search for video question answering (2025), https://arxiv.org/abs/2503.16032 4

9. Fan, Y., Ma, X., Wu, R., Du, Y., Li, J., Gao, Z., Li, Q.: VideoAgent: A memory-augmented multimodal agent for video understanding. In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G. (eds.) Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XXII. Lecture Notes in Computer Science, vol. 15080, pp. 75–92. Springer (2024). https://doi.org/10.1007/978-3-031-72670-5_5 4, 11
10. Fu, C., Dai, Y., Luo, Y., Li, L., Ren, S., Zhang, R., Wang, Z., Zhou, C., Shen, Y., Zhang, M., Chen, P., Li, Y., Lin, S., Zhao, S., Li, K., Xu, T., Zheng, X., Chen, E., Shan, C., He, R., Sun, X.: Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025. pp. 24108–24118. IEEE (2025). https://doi.org/10.1109/CVPR52734.2025.02245 1, 4
11. Guo, M., Xu, J., Zhang, Y., Song, J., Peng, H., Deng, Y., Dong, X., Nakayama, K., Geng, Z., Wang, C., Ni, B., Yang, G., Rao, Y., Peng, H., Hu, H., Wetzstein, G., Hu, S.: Rbench: Graduate-level multi-disciplinary benchmarks for LLM & MLLM complex reasoning evaluation. In: Singh, A., Fazel, M., Hsu, D., Lacoste-Julien, S., Berkenkamp, F., Maharaj, T., Wagstaff, K., Zhu, J. (eds.) Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025. Proceedings of Machine Learning Research, vol. 267. PMLR (2025), https://proceedings.mlr.press/v267/guo25r.html 4
12. Gupta, T., Kembhavi, A.: Visual programming: Compositional visual reasoning without training. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023. pp. 14953–14962. IEEE (2023). https://doi.org/10.1109/CVPR52729.2023.01436 4
13. Hara, K., Kataoka, H., Satoh, Y.: Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 6546–6555. IEEE (2018). https://doi.org/10.1109/CVPR.2018.00685 4
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 770–778. IEEE (2016). https://doi.org/10.1109/CVPR.2016.90 4
15. Hu, J., Cheng, Z., Si, C., Li, W., Gong, S.: Cos: Chain-of-shot prompting for long video understanding (2025), https://arxiv.org/abs/2502.06428 4
16. Hu, S., Lin, K.Q., Shou, M.Z.: Showui-π: Flow-based generative models as gui dexterous hands (2025), https://arxiv.org/abs/2512.24965 4
17. Jang, Y., Song, Y., Sohn, S., Logeswaran, L., Luo, T., Kim, D., Bae, K., Lee, H.: Scalable video-to-dataset generation for cross-platform mobile agents. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025. pp. 8604–8614. IEEE (2025). https://doi.org/10.1109/CVPR52734.2025.00804 4, 5
18. Jeoung, S., Huybrechts, G., Ganesh, B., Galstyan, A., Bodapati, S.: Adaptive video understanding agent: Enhancing efficiency with dynamic frame sampling and feedback-driven reasoning (2024), https://arxiv.org/abs/2410.20252 4
19. Kahatapitiya, K., Ranasinghe, K., Park, J., Ryoo, M.S.: Language repository for long video understanding. In: Che, W., Nabende, J., Shutova, E., Pilehvar, M.T. (eds.) Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025. pp. 5627–5646. Findings of ACL, Associa-

tion for Computational Linguistics (2025). https://doi.org/10.18653/V1/2025.FINDINGS-ACL.294 10, 11
20. Karacan, L., Sarıgül, M.: Full-frame video stabilization via spatiotemporal transformers. Computational Visual Media **11**(3), 655–667 (2025). https://doi.org/10.26599/CVM.2025.9450416 4
21. Kim, W., Choi, C., Lee, W., Rhee, W.: An image grid can be worth a video: Zero-shot video question answering using a vlm (2024), https://arxiv.org/abs/2403.18406 11
22. Li, R., Wang, X., Zhang, Y., Wang, Z., Yeung-Levy, S.: Temporal preference optimization for long-form video understanding (2025), https://arxiv.org/abs/2501.13919 4
23. Lin, B., Ye, Y., Zhu, B., Cui, J., Ning, M., Jin, P., Yuan, L.: Video-llava: Learning united visual representation by alignment before projection. In: Al-Onaizan, Y., Bansal, M., Chen, Y. (eds.) Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024. pp. 5971–5984. Association for Computational Linguistics (2024). https://doi.org/10.18653/V1/2024.EMNLP-MAIN.342 4
24. Lin, J., Yu, Z., Karlsson, B.F.: Switch: Benchmarking modeling and handling of tangible interfaces in long-horizon embodied scenarios (2026), https://arxiv.org/abs/2511.17649 4
25. Lin, K.Q., Hu, S., Li, L., Yang, Z., Wang, L., Torr, P., Shou, M.Z.: Computer-use agents as judges for generative user interface (2025), https://arxiv.org/abs/2511.15567 4
26. Lin, K.Q., Li, L., Gao, D., Wu, Q., Yan, M., Yang, Z., Wang, L., Shou, M.Z.: VideoGUI: A benchmark for GUI automation from instructional videos. In: Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J.M., Zhang, C. (eds.) Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024 (2024), http://papers.nips.cc/paper_files/paper/2024/hash/804e757b7d7043c26701c3a313032101-Abstract-Datasets_and_Benchmarks_Track.html 4
27. Liu, G., Zhao, P., Liu, L., Chen, Z., Chai, Y., Ren, S., Wang, H., He, S., Meng, W.: Learnact: Few-shot mobile gui agent with a unified demonstration benchmark (2025), https://arxiv.org/abs/2504.13805 4
28. Lu, D., Xu, Y., Wang, J., Wu, H., Wang, X., Wang, Z., Yang, J., Su, H., Chen, J., Chen, J., Mao, Y., Zhou, J., Lin, J., Hui, B., Yu, T.: Videoagenttrek: Computer use pretraining from unlabeled videos (2025), https://arxiv.org/abs/2510.19488 4
29. Mangalam, K., Akshulakov, R., Malik, J.: Egoschema: A diagnostic benchmark for very long-form video language understanding. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023 (2023), http://papers.nips.cc/paper_files/paper/2023/hash/90ce332aff156b910b002ce4e6880dec-Abstract-Datasets_and_Benchmarks.html 3, 4, 10, 11, 15
30. Min, J., Buch, S., Nagrani, A., Cho, M., Schmid, C.: Morevqa: Exploring modular reasoning models for video question answering. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024. pp. 13235–13245. IEEE (2024). https://doi.org/10.1109/CVPR52733.2024.01257 11

31. Nguyen, T., Bin, Y., Xiao, J., Qu, L., Li, Y., Wu, J.Z., Nguyen, C., Ng, S., Luu, A.T.: Video-language understanding: A survey from model architecture, model training, and data perspectives. In: Ku, L., Martins, A., Srikumar, V. (eds.) Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024. Findings of ACL, vol. ACL 2024, pp. 3636–3657. Association for Computational Linguistics (2024). https://doi.org/10.18653/V1/2024.FINDINGS-ACL.217 4
32. Ning, M., Zhu, B., Xie, Y., Lin, B., Cui, J., Yuan, L., Chen, D., Yuan, L.: Video-bench: A comprehensive benchmark and toolkit for evaluating video-based large language models. Computational Visual Media **12**(1), 71–84 (2026). https://doi.org/10.26599/CVM.2025.9450516 4
33. Park, J., Ranasinghe, K., Kahatapitiya, K., Ryu, W., Kim, D., Ryoo, M.S.: Too many frames, not all useful: Efficient strategies for long-form video QA. In: Demberg, V., Inui, K., Marquez, L. (eds.) Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2026 - Volume 1: Long Papers, Rabat, Morocco, March 24-29, 2026. pp. 3569–3588. Association for Computational Linguistics (2026). https://doi.org/10.18653/V1/2026.EACL-LONG.164 4, 10, 11
34. Ranasinghe, K., Li, X., Kahatapitiya, K., Ryoo, M.S.: Understanding long videos with multimodal language models. In: The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025 (2025), https://openreview.net/forum?id=0xKi02I29I 11
35. Ren, J., Zhao, Y., Vu, T., Liu, P.J., Lakshminarayanan, B.: Self-evaluation improves selective generation in large language models. In: Proceedings of Machine Learning Research. vol. 239, pp. 49–64. PMLR (2023), https://proceedings.mlr.press/v239/ren23a.html 9
36. Shang, Y., Xu, B., Kang, W., Cai, M., Li, Y., Wen, Z., Dong, Z., Keutzer, K., Lee, Y.J., Yan, Y.: Interpolating video-llms: Toward longer-sequence lmms in a training-free manner (2024), https://arxiv.org/abs/2409.12963 11
37. Shen, X., Xiong, Y., Zhao, C., Wu, L., Chen, J., Zhu, C., Liu, Z., Xiao, F., Varadarajan, B., Bordes, F., Liu, Z., Xu, H., Kim, H.J., Soran, B., Krishnamoorthi, R., Elhoseiny, M., Chandra, V.: LongVU: Spatiotemporal adaptive compression for long video-language understanding. In: Proceedings of the 42nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 267, pp. 54582–54599. PMLR (2025), https://proceedings.mlr.press/v267/shen25j.html 4
38. Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., Yao, S.: Reflexion: language agents with verbal reinforcement learning. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023 (2023), http://papers.nips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html 9
39. Song, C.H., Song, Y., Goyal, P., Su, Y., Riva, O., Palangi, H., Pfister, T.: Watch and learn: Learning to use computers from online videos (2026), https://arxiv.org/abs/2510.04673 4
40. Song, E., Chai, W., Wang, G., Zhang, Y., Zhou, H., Wu, F., Chi, H., Guo, X., Ye, T., Zhang, Y., Lu, Y., Hwang, J.N., Wang, G.: Moviechat: From dense token to sparse memory for long video understanding (2024), https://arxiv.org/abs/2307.16449 4

41. Sun, Y., Zhao, S., Yu, T., Wen, H., Va, S., Xu, M., Li, Y., Zhang, C.: GUI-Xplore: Empowering generalizable GUI agents with one exploration. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025. pp. 19477–19486. IEEE (2025). https://doi.org/10.1109/CVPR52734.2025.01814 4
42. Tang, Y., Bi, J., Xu, S., Song, L., Liang, S., Wang, T., Zhang, D., An, J., Lin, J., Zhu, R., Vosoughi, A., Huang, C., Zhang, Z., Liu, P., Feng, M., Zheng, F., Zhang, J., Luo, P., Luo, J., Xu, C.: Video understanding with large language models: A survey (2024), https://arxiv.org/abs/2312.17432 4
43. Tran, D., Bourdev, L.D., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015. pp. 4489–4497. IEEE (2015). https://doi.org/10.1109/ICCV.2015.510 4
44. Wang, J.W., Shen, L.Y.: Spatiotemporal fusion transformer for video demoiréing. Computational Visual Media **11**(4), 849–869 (2025). https://doi.org/10.26599/CVM.2025.9450502 4
45. Wang, J., Xu, H., Zhang, X., Yan, M., Zhang, J., Huang, F., Sang, J.: Mobile-agent-v: A video-guided approach for effortless and efficient operational knowledge injection in mobile automation (2025), https://arxiv.org/abs/2502.17110 4
46. Wang, S., Zhao, Q., Do, M.Q., Agarwal, N., Lee, K., Sun, C.: Vamos: Versatile action models for video understanding. In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G. (eds.) Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XII. Lecture Notes in Computer Science, vol. 15070, pp. 142–160. Springer (2024). https://doi.org/10.1007/978-3-031-73254-6_9 10, 11
47. Wang, X., Zhang, Y., Zohar, O., Yeung-Levy, S.: Videoagent: Long-form video understanding with large language model as agent. In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G. (eds.) Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXX. Lecture Notes in Computer Science, vol. 15138, pp. 58–76. Springer (2024). https://doi.org/10.1007/978-3-031-72989-8_4 4, 10, 11, 12
48. Wang, Y., Li, K., Li, Y., He, Y., Huang, B., Zhao, Z., Zhang, H., Xu, J., Liu, Y., Wang, Z., Xing, S., Chen, G., Pan, J., Yu, J., Wang, Y., Wang, L., Qiao, Y.: InternVideo: General video foundation models via generative and discriminative learning (2022), https://arxiv.org/abs/2212.03191 4
49. Wang, Y., Yang, Y., Ren, M.: Lifelongmemory: Leveraging llms for answering queries in long-form egocentric videos (2024), https://arxiv.org/abs/2312.05269 10, 11
50. Wang, Z., Chen, B., Yue, Z., Wang, Y., Qiao, Y., Wang, L., Wang, Y.: Videochat-a1: Thinking with long videos by chain-of-shot reasoning. In: Koenig, S., Jenkins, C., Taylor, M.E. (eds.) Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026. pp. 10467–10475. AAAI Press (2026). https://doi.org/10.1609/AAAI.V40I13.38018 4
51. Wang, Z., Yu, S., Stengel-Eskin, E., Yoon, J., Cheng, F., Bertasius, G., Bansal, M.: VideoTree: Adaptive tree-based video representation for LLM reasoning on long videos. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition,

CVPR 2025, Nashville, TN, USA, June 11-15, 2025. pp. 3272–3283. IEEE (2025). https://doi.org/10.1109/CVPR52734.2025.00311 4, 10, 11, 12
52. Weng, Y., Han, M., He, H., Chang, X., Zhuang, B.: LongVLM: Efficient long video understanding via large language models. In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G. (eds.) Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XXXIII. Lecture Notes in Computer Science, vol. 15091, pp. 453–470. Springer (2024). https://doi.org/10.1007/978-3-031-73414-4_26 4
53. Wu, H., Li, D., Chen, B., Li, J.: Longvideobench: A benchmark for long-context interleaved video-language understanding. In: Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J.M., Zhang, C. (eds.) Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024 (2024), http://papers.nips.cc/paper_files/paper/2024/hash/329ad516cf7a6ac306f29882e9c77558-Abstract-Datasets_and_Benchmarks_Track.html 1, 4
54. Xiao, J., Huang, N., Qin, H., Li, D., Li, Y., Zhu, F., Tao, Z., Yu, J., Lin, L., Chua, T., Yao, A.: Videoqa in the era of llms: An empirical study. Int. J. Comput. Vis. **133**(7), 3970–3993 (2025). https://doi.org/10.1007/S11263-025-02385-8 4
55. Xiao, J., Shang, X., Yao, A., Chua, T.: NExT-QA: Next phase of question-answering to explaining temporal actions. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. pp. 9777–9786. IEEE (2021). https://doi.org/10.1109/CVPR46437.2021.00965 3, 4, 10
56. Xie, T., Zhang, D., Chen, J., Li, X., Zhao, S., Cao, R., Hua, T.J., Cheng, Z., Shin, D., Lei, F., Liu, Y., Xu, Y., Zhou, S., Savarese, S., Xiong, C., Zhong, V., Yu, T.: OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In: Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J.M., Zhang, C. (eds.) Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024 (2024), http://papers.nips.cc/paper_files/paper/2024/hash/5d413e48f84dc61244b6be550f1cd8f5-Abstract-Datasets_and_Benchmarks_Track.html 13
57. Yang, J., Zhang, H., Li, F., Zou, X., Li, C., Gao, J.: Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v (2023), https://arxiv.org/abs/2310.11441 4, 13
58. Yang, Z., Chen, G., Li, X., Wang, W., Yang, Y.: Doraemongpt: Toward understanding dynamic scenes with large language models (exemplified as A video agent). In: Salakhutdinov, R., Kolter, Z., Heller, K.A., Weller, A., Oliver, N., Scarlett, J., Berkenkamp, F. (eds.) Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024. Proceedings of Machine Learning Research, vol. 235, pp. 55976–55997. PMLR (2024), https://proceedings.mlr.press/v235/yang24d.html 4
59. Ye, J., Wang, Z., Sun, H., Chandrasegaran, K., Durante, Z., Eyzaguirre, C., Bisk, Y., Niebles, J.C., Adeli, E., Fei-Fei, L., Wu, J., Li, M.: Re-thinking temporal search for long-form video understanding. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025. pp. 8579–8591. IEEE (2025). https://doi.org/10.1109/CVPR52734.2025.00802 4
60. Yu, S., Ling, Y., Fang, C., Zhou, Q., Zhao, Y., Chen, C., Zhu, S., Chen, Z.: LLM-guided scenario-based gui testing (2025), https://arxiv.org/abs/2506.05079 4

61. Zhang, B., Guo, Y., Yang, R., Zhang, Z., Xie, J., Suo, J.: Darkvision: A benchmark and study for low-light image/video analysis. Computational Visual Media **12**(3), 615–642 (2026). https://doi.org/10.26599/CVM.2025.9450461 4
62. Zhang, B., Shang, Z., Gao, Z., Zhang, W., Xie, R., Ma, X., Yuan, T., Wu, X., Zhu, S., Li, Q.: Tongui: Internet-scale trajectories from multimodal web tutorials for generalized GUI agents. In: Koenig, S., Jenkins, C., Taylor, M.E. (eds.) Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026. pp. 12367–12375. AAAI Press (2026). https://doi.org/10.1609/AAAI.V40I15.38229 4
63. Zhang, C., Lu, T., Islam, M.M., Wang, Z., Yu, S., Bansal, M., Bertasius, G.: A simple LLM framework for long-range video question-answering. In: Al-Onaizan, Y., Bansal, M., Chen, Y. (eds.) Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024. pp. 21715–21737. Association for Computational Linguistics (2024). https://doi.org/10.18653/V1/2024.EMNLP-MAIN.1209 4, 11
64. Zhang, H., Li, X., Bing, L.: Video-llama: An instruction-tuned audio-visual language model for video understanding. In: Feng, Y., Lefever, E. (eds.) Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023 - System Demonstrations, Singapore, December 6-10, 2023. pp. 543–553. Association for Computational Linguistics (2023). https://doi.org/10.18653/V1/2023.EMNLP-DEMO.49 4
65. Zhang, Y., Ni, B., Chen, X.S., Zhang, H.R., Rao, Y., Peng, H., Lu, Q., Hu, H., Guo, M.H., Hu, S.M.: Bee: A high-quality corpus and full-stack suite to unlock advanced fully open mllms (2026), https://arxiv.org/abs/2510.13795 4
66. Zhang, Y., Guo, X., Goh, Y., Hu, J., Chen, Z., Wang, X., Gao, D., Shou, M.Z.: Showui-aloha: Human-taught gui agent (2026), https://arxiv.org/abs/2601.07181 4
67. Zhong, Y., Ji, W., Xiao, J., Li, Y., Deng, W., Chua, T.: Video question answering: Datasets, algorithms and challenges. In: Goldberg, Y., Kozareva, Z., Zhang, Y. (eds.) Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022. pp. 6439–6455. Association for Computational Linguistics (2022). https://doi.org/10.18653/V1/2022.EMNLP-MAIN.432 4
68. Zhou, J., Shu, Y., Zhao, B., Wu, B., Liang, Z., Xiao, S., Qin, M., Yang, X., Xiong, Y., Zhang, B., Huang, T., Liu, Z.: MLVU: benchmarking multi-task long video understanding. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025. pp. 13691–13701. IEEE (2025). https://doi.org/10.1109/CVPR52734.2025.01278 4