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

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

Tsinghua University, Beijing, China | ECCV 2026

Presenter: Sunqi Fan **Homepage:** fansunqi.github.io **WeChat:** fsq159357FSQ

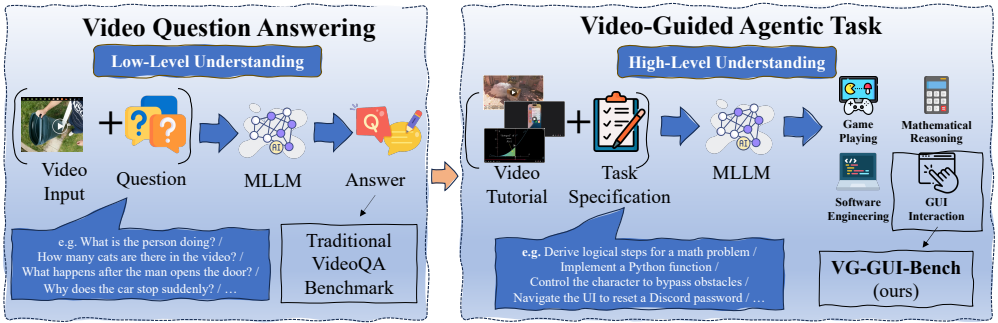
Tsinghua University

Paper

Code

Project Page

Motivation: from answering questions to learning how to act



The capability gap

VideoQA tests whether a model can **recognize and reason**. **Video-guided agentic tasks** require a model to **extract a procedure, transfer it, and execute a long-horizon task**.

Our work

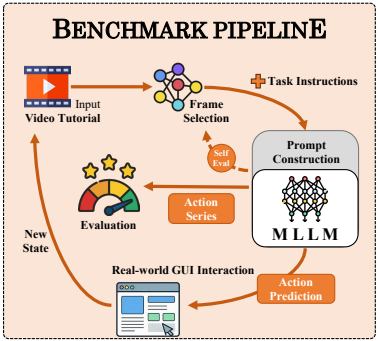
1. VG-GUI-Bench

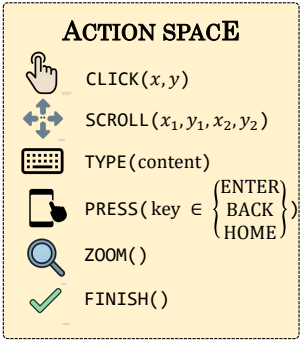
Evaluate **procedural knowledge transfer** from video tutorials to GUI tasks.

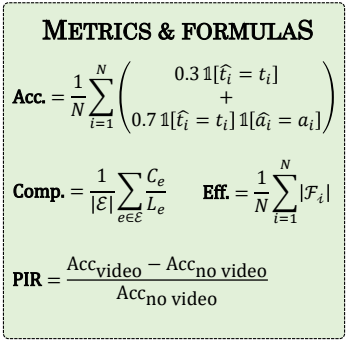
2. TASKER

Search for a **compact set of frames** using task relevance and scene dynamics.

VG-GUI-Bench: testing video in-context learning







A tutorial video is paired with a semantically related GUI episode. At every step, the agent observes selected tutorial frames, the current GUI, and its action history.

1,000
test cases

A standardized, step-wise evaluation protocol

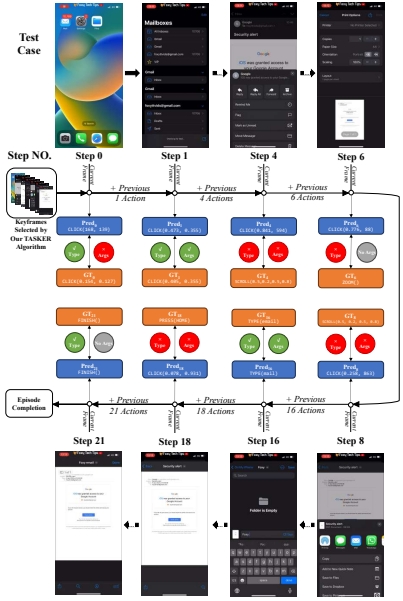
Six-action space

- `CLICK(x,y)`
- `SCROLL(x1,y1,x2,y2)`
- `TYPE(content)`
- `PRESS(key)`
- `ZOOM()`
- `FINISH()`

Four complementary metrics

- **Accuracy**: action type (0.3) + arguments (0.7)
- **Completion**: correct-step ratio per episode
- **Efficiency**: frames consumed per step
- **PIR**: relative gain from tutorial video

TASK: How To Save Emails As PDF On iPhone & iPad?



VG-GUI-Bench leaderboard and analysis

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

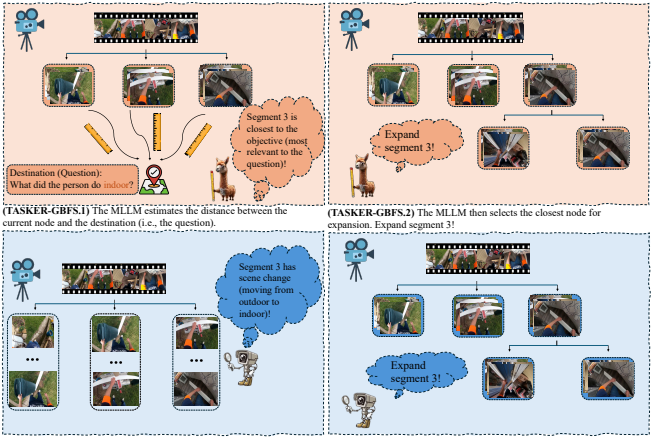
- **Best overall:** Gemini-3.1-Pro reaches 61.68%, but the benchmark remains challenging.
- **Consistent but limited gains:** all seven models improve with uniform frames, though only modestly.
- **Largest gain:** Seed-2.0-Pro improves by 3.85 points and achieves the highest PIR of 0.107.

Why is uniform sampling not enough?

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

- **Observation:** tutorials contain the complete procedure, but uniformly sampled frames yield only modest gains because key operations are sparse.
- **Implication:** we need an **adaptive and efficient keyframe selector** that focuses the model on task-relevant evidence.

TASKER turns temporal selection into graph search



(TASKER-GBFS.1) The MLLM estimates the distance between the current node and the destination (i.e., the question).

(TASKER-GBFS.2) The MLLM then selects the closest node for expansion. Expand segment 3!

(TASKER-Dijkstra.1) The MLLM identifies segment with most significant changes. Segment 3 has scene change (moving from outdoor to indoor).

(TASKER-Dijkstra.2) Segment 3 is considered the node that minimizes the movement cost function and is subsequently expanded.

Key idea

A video segment is a node whose boundary images are visible frames.

The MLLM repeatedly **scores, selects, and splits** segments until the evidence is sufficient.

Search state and expansion

State representation

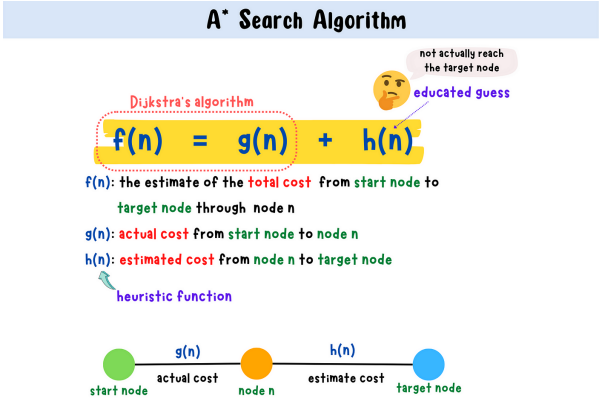
- Root node: the entire video
- Initialize with M uniform segments
- Visible frames $\mathcal{F}_v$: segment boundaries
- Open list $\mathcal{L}$: candidate segments
- Binary split for node expansion

Agentic search loop

1. Predict an answer or next action from $\mathcal{F}_v$
2. Estimate whether evidence is sufficient
3. Score candidate segments
4. Expand the top-B segments
5. Validate new frames; freeze redundant branches

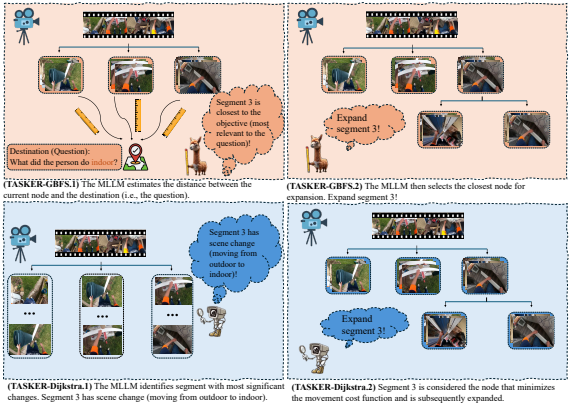
Output: prediction $\hat{y}$ + compact keyframe set $\mathcal{F}_k$

Background: A* search algorithm



Drawing inspiration from classical search algorithms.

TASKER: task-driven and scene-aware search



(TASKER-GBFS.1) The MLLM estimates the distance between the current node and the destination (i.e., the question).

(TASKER-GBFS.2) The MLLM then selects the closest node for expansion. Expand segment 3!

(TASKER-Dijkstra.1) The MLLM identifies segment with most significant changes. Segment 3 has scene change (moving from outdoor to indoor).

(TASKER-Dijkstra.2) Segment 3 is considered the node that minimizes the movement cost function and is subsequently expanded.

- **GBFS:** $f(n) = h(n)$
  Task-driven relevance to missing evidence.
- **Dijkstra:** $f(n) = g(n)$
  Scene or UI-state dynamics.
- **A*:** $f(n) = g(n) + h(n)$
  Jointly considers both criteria.

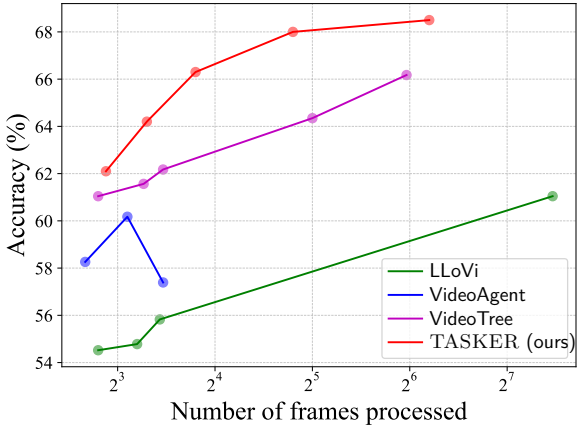
The MLLM evaluates the cost function in natural language; no task-specific training is required.

VideoQA: consistent gains across models and datasets

Method	Base	EgoSchema Full	NExT-QA Tem.	Cau.	Avg.
LVNet	GPT-4o	61.1	65.5	75.0	72.9
VideoAgent	GPT-4	54.1	64.5	72.7	71.3
VideoTree	GPT-4	61.1	70.6	76.5	75.6
TASKER	GPT-4	**63.1**	**72.3**	**78.2**	**77.4**
TASKER	GPT-4o	**63.6**	**72.9**	**79.0**	**78.1**
VideoAgent	Qwen3-VL	74.6	79.7	83.5	82.8
VideoTree	Qwen3-VL	76.7	81.9	85.1	84.3
TASKER	Qwen3-VL	**77.3**	**83.6**	**85.3**	**85.1**

TASKER consistently outperforms prior keyframe-based methods across EgoSchema and NExT-QA in a training-free, zero-shot setting.

TASKER is accurate because it spends frames selectively



At the same 66% accuracy, TASKER uses approximately **one quarter of the frames** required by VideoTree.

EgoSchema subset; identical base-model setting.

TASKER offers complementary accuracy–efficiency trade-offs

Method	Accuracy	Completion	Frames/step ↓	PIR
No video	25.32	69.03	0.00	–
Uniform sampling	39.82	70.64	10.88	0.573
Oracle keyframe	44.32	76.32	**1.00**	0.750
VideoTree	40.79	71.93	10.00	0.611
VideoAgent	39.86	71.17	5.12	0.574
TASKER-Dijkstra	40.75	**74.39**	5.88	0.609
TASKER-A*	**40.96**	71.38	8.24	**0.618**

Best transfer

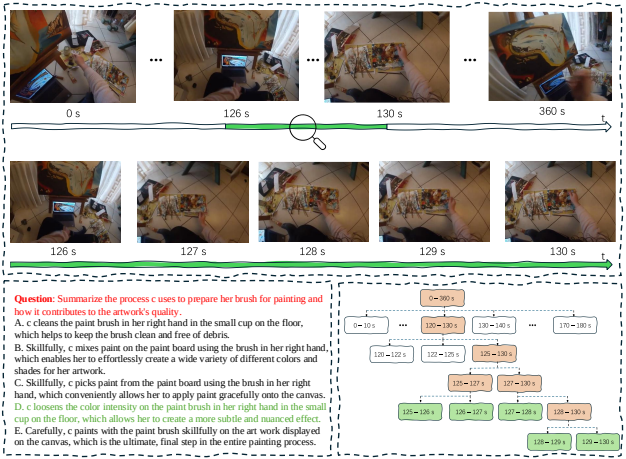
TASKER-A* gives the highest accuracy and PIR among training-free selection methods.

Complementary variants

Dijkstra reaches 74.39% completion, close to the oracle's 76.32%.

All selection methods use Qwen3-VL-235B-A22B-Instruct.

What does the search actually do?



Interpretable localization

The decisive evidence appears at **126–130 s**.

TASKER retrieves the enclosing 125–130 s segment through a sparse search while leaving most nodes untouched.

Takeaways

1. Video understanding should progress from **perception** to **procedural transfer and action**.
2. VG-GUI-BENCH provides a **1,000-case testbed** for evaluating this high-level capability.
3. TASKER unifies temporal selection across VideoQA and GUI agents through **task-driven + scene-aware search**.

Code & Data: github.com/VG-GUI-TASKER/VG-GUI-TASKER

Thank you!