

A Experimental Setup

A.1 Dataset and Baselines

A.1.1 Video Understanding

The video understanding benchmarks evaluated in this work encompass six diverse and high-quality datasets, each meticulously curated to target a distinct and critical aspect of video-language comprehension. These benchmarks collectively span a wide range of tasks—including long-horizon reasoning, egocentric perspective analysis, multimodal alignment, and fine-grained visual-semantic understanding—thereby offering a comprehensive and rigorous evaluation of model capabilities across various real-world video scenarios.

- **LongBench** [3]: Designed to evaluate long-horizon reasoning over extended video contexts, requiring models to maintain semantic coherence across many frames.
- **NextQA** [45]: Focuses on temporal reasoning and causal understanding by presenting questions about action sequences and their logical consequences.
- **EgoSchema** [31]: A diagnostic benchmark for egocentric videos, assessing the ability to understand first-person perspectives and actions.
- **LVBench** [39]: Tailored for multi-modal alignment, especially in complex settings where audio, text, and vision interplay is critical.
- **MLVU** [59]: Emphasizes multi-task long video understanding, pushing models to perform a range of tasks such as captioning, question answering, and temporal ordering.
- **VideoMME** [11]: A comprehensive benchmark that integrates various modalities—including subtitles and OCR—and is especially suited for evaluating fine-grained visual-semantic alignment and symbolic reasoning in videos.

To rigorously assess the performance of ReAgent-V, we compare it against a broad array of baseline models that span different capabilities and design philosophies. Proprietary closed-source models include **GPT-4o** [14], known for its advanced multimodal reasoning, and **Gemini-1.5-Pro** [37], optimized for long-context multimodal input. Among open-source video-language models, the evaluation includes:

- **ShareGPT4Video** [4]: Enhances caption-driven video question answering by integrating ShareGPT-style conversational supervision into multimodal video training. It leverages large-scale user-generated captions and dialogues to align vision and language effectively, enabling better generalization to open-ended video queries.
- **VideoChat2** [21]: A chat-centric video-language model built to support multi-turn, conversational interactions grounded in video content. It emphasizes interactive understanding, with capabilities for dialogue continuity, temporal grounding, and multi-modal alignment, making it ideal for assistive agents and education scenarios.
- **LLaVA-Video** [54]: An extension of the LLaVA framework adapted for video inputs, available in both 7B and 72B parameter versions. It employs frame-wise and temporal fusion techniques to convert video sequences into language-aligned embeddings, demonstrating strong performance on visual question answering and summarization tasks.
- **Qwen2** [2] and **Qwen2.5-VL** [2] families: Developed by Alibaba, these models exhibit high performance on both image and video understanding benchmarks through advanced multi-modal alignment. Qwen2.5-VL particularly excels in handling long-context and dense visual-textual reasoning tasks, supported by a unified visual-language architecture.
- **InternVL-2.5** [6]: A high-performing open-source model that scales both the architecture and training data to improve generalization. It incorporates vision-language alignment techniques, dense region grounding, and optimized pretraining routines to support both short and long video tasks with high efficiency.
- **BIMBA-LLaVA** [15]: Introduces a selective scan compression mechanism tailored for long videos, where it prunes redundant frames while preserving key semantic content. This compression-aware training improves inference speed and memory usage while maintaining accuracy on temporal and contextual reasoning tasks.

- **Kangaroo [25]**: Designed specifically for long-context video input, Kangaroo adopts hierarchical attention and memory-efficient tokenization strategies to process hundreds of frames. It is particularly effective for document-level video understanding, such as meeting summarization or sports analysis.
- **Long-LLaVA [41]**: A variant of LLaVA optimized for efficient multi-frame processing. It integrates temporal coherence constraints and cross-frame attention mechanisms, making it capable of capturing nuanced motion patterns and temporal dependencies for improved video QA and description.
- **VideoAgent [9]**: An agent framework for long-form video understanding that mimics human cognition through LLM-guided reasoning, CLIP-based retrieval, and VLM-driven state updates.
- **VideoMemAgent [9]**: An agent framework for structured long-form video understanding that integrates unified temporal and object memory with tool-augmented reasoning, enabling multi-round chain-of-thought inference across complex video content.

Across all these baselines, **ReAgent-V** almost outperforms in terms of accuracy, interpretability, and computational efficiency, largely due to its entropy-guided frame selection and multi-perspective reflection mechanism that support real-time reward-driven answer refinement.

Experiments are run on two H100-96GB GPUs with NVLink, using 64 frames per video. CLIP (ViT-L/14-336) is used for vision-language alignment. Videos are processed using decord and ffmpeg, with audio chunked for transcription. At inference, ReAgent-V performs tool-augmented reasoning by dynamically invoking OCR, ASR, DET and other modules. A multi-agent reflection strategy—comprising conservative, neutral, and aggressive evaluators—is followed by a meta-agent decision step.

A.1.2 Video LLM Reasoning

To investigate how ReAgent-V can enhance the reasoning capability of Video LLMs, we follow the experimental setup from [10] and apply ReAgent-V to filter the video portion of the Video-R1-260k dataset. Specifically, during inference, we retain samples with importance scores lower than 5 (out of 10) as these tend to be more challenging and thus more valuable for training. We then fine-tune the Qwen2.5-VL model on 8 NVIDIA H100 80GB GPUs using the filtered dataset and compare it against several baseline models defined in the paper. Our results show that the model achieves a 2.1% improvement in overall performance while using only 45% of the original training data, demonstrating the effectiveness of high-quality sample selection in enhancing video reasoning.

Benchmarks. We evaluate our model on six video benchmarks: VSI-Bench [12], VideoMMMU [13], MMVU [58], MVBench [22], TempCompass [27], and VideoMME [11]. The first three benchmarks focus primarily on video reasoning tasks that assess the model’s ability to understand and reason over complex video semantics, while the latter three are general video understanding benchmarks involving a mix of perception and reasoning challenges. For MMVU, we evaluate using its multiple-choice question subset to ensure stability and consistency.

A.1.3 VLA Alignment

To construct a preference-based dataset for aligning vision-language-action (VLA) models, we collect and process rollouts from the SIMPLER environment [23]. Specifically, we construct a reward-ranked dataset for VLA alignment by sampling trajectories from four original in-domain tasks using the OpenVLA-7B-SFT-Simpler (OpenVLA-SFT) model, a supervised-finetuned baseline released by [57], with five rollouts per task. The remaining three generalization tasks are kept unseen and excluded from alignment. After data collection, we use ReAgent-V to evaluate each trajectory segment, assigning scalar reward scores based on task success, stability, completeness, and accuracy. For each task, we sample 20 best-vs-worst trajectory pairs, matched by task type and initial environment state. This yields 80 total trajectories, which are then partitioned into two RLDS-formatted datasets: a chosen set containing high-reward trajectories and a rejected set with low-reward counterparts.

We conduct all VLA alignment training on a single NVIDIA A100 GPU (80GB). The model is initialized from OpenVLA-SFT. Training employs LoRA (rank=32) with a learning rate of $2e-5$ and dropout of 0.0. We use gradient accumulation with a step size of 4 to simulate larger batch sizes under

948 limited GPU memory. After training, the base model is merged with LoRA weights to obtain the
 949 final model. Our alignment process follows the Trajectory-wise Preference Optimization (TPO) [57]
 950 paradigm, where the objective is to learn a reward-aligned policy using the preference pairs described
 951 above. All results reported in the main paper are obtained after 6,800 steps of alignment training.

952 We evaluate all VLA models in the SIMPLER environment, covering four types of generalization:
 953 in-domain, subject generalization, physical generalization, and semantic generalization. In-domain
 954 consists of the four original tasks in the SIMPLER environment. Subject generalization includes three
 955 new tasks involving objects not seen during training. Physical generalization comprises eight tasks
 956 created by varying the width, height, and size of known objects. Semantic generalization includes four
 957 tasks where the original prompts are rephrased with synonymous instructions. Each taskset comprises
 958 several subtasks, and each subtask is evaluated over a fixed number of distinct episodes. The final
 959 metric is the average task success rate across episodes. For all baseline models—OpenVLA-SFT,
 960 OpenVLA-DPO, and OpenVLA-TPO with GRAPE—we use results reported by [57]. Our evaluation
 961 strictly follows the same experimental setup, adopting identical task definitions and environmental
 962 configurations to ensure a fair and consistent comparison.

963 B Evaluation Criteria and Prompts

964 B.1 Tool Selection Process

Tool Selection Prompt

[Task]

Carefully analyze the video content and identify exactly what information needs to be retrieved to support answering the given question. To answer the question step by step, list all the physical entities related to the question you want to retrieve, you can provide your retrieve request to assist you by the following JSON format:

[Tool Functions]

The Tool Factory supports the following types of retrieval via specialized tools:

- **Text Extraction:** Tools: OCR, ASR — extract embedded text (e.g., signs, timestamps) and transcribe speech or narration.
- **Object Detection and Grounding:** Tool: Grounding DINO — detects objects and aligns them with natural language prompts.
- **Multimodal Matching:** Tools: CLIP — perform image-text alignment and retrieval based on semantic similarity.
- **Structured Scene Understanding:** Tools: Scene Graph, Action Detector — construct structured graphs of object relations and detect visual actions.
- **Video Reasoning:** Tools: ShareGPT4Video, VQA Model — enable video-based question answering and explanation generation.
- **Caption Generation:** Tool: Captioning Model — generate scene-level textual descriptions.
- **Emotion and Identity Recognition:** Tools: Face Recognition, Emotion Detector — detect character identities and their emotional states.

[Output format]:

Use the following format to list the functional categories and corresponding tools required:

```
[
  {"function": "Function Category",
   "tools": ["Tool1", "Tool2"]}
]
```

Example 1:

Original question: “How many blue balloons are over the long table in the middle of the room at the end of this video?”

Options: “A. 1. B. 2. C. 3. D. 4.”

Output:

```
[
  {"function": "Visual Object Detection",
   "tools": ["Grounding DINO"]},
  {"function": "Structured Scene Understanding",
   "tools": ["Scene Graph"]},
  {"function": "Numerical Reasoning",
   "tools": ["ShareGPT4Video"]}
]
```

Example 2:

Original question: “In the lower left corner of the video, what color is the woman wearing on the right side of the man in black clothes?”

Options: “A. Blue. B. White. C. Red. D. Yellow.”

Output:

```
[
  {"function": "Visual Object Detection",
   "tools": ["Grounding DINO"]},
  {"function": "Structured Scene Understanding",
   "tools": ["Scene Graph"]},
  {"function": "Global Scene Description",
   "tools": ["Captioning Model"]}
]
```

Example 3:

Original question: “In which country is the comedy featured in the video recognized world-wide?”

Options: “A. China. B. UK. C. Germany. D. United States.”

Output:

```
[
  {"function": "Audio Text Extraction",
   "tools": ["ASR"]},
  {"function": "Global Semantic Summary",
   "tools": ["Captioning Model"]},
  {"function": "Commonsense Reasoning",
   "tools": ["ShareGPT4Video"]}
]
```

Example 4:

Original question: “Describe what the chef does to prepare the pasta dish in the video.”

Options: “”

Output:

```
[
  {"function": "Audio Instruction Extraction",
   "tools": ["ASR"]},
  {"function": "Object and Action Detection",
   "tools": ["Grounding DINO", "Action Detector"]},
  {"function": "Structured Scene Understanding",
   "tools": ["Scene Graph"]},
  {"function": "Narrative Generation",
   "tools": ["Captioning Model"]}
]
```

[Now begin]

Note that you don’t need to answer the question in this step, so you don’t need any information about the video or image. You only need to provide your retrieve request (it’s optional), and I will help you retrieve the information you want. Please provide the json format.

966

967 This section introduces the Tool Selection Prompt framework, designed to guide retrieval-aware
968 video question answering through structured tool invocation. By decomposing a user question into

functional information needs—such as object detection, text extraction, or reasoning—the prompt enables precise specification of which tools (e.g., OCR, Grounding DINO [26], CLIP, Scene Graph, ASR) should be employed. The unified JSON output format supports downstream automation and tool chaining, ensuring modularity and extensibility. Multiple illustrative examples demonstrate how diverse question types can be mapped to appropriate retrieval functions and tools, from counting objects to recognizing emotional states or extracting narrated instructions. Notably, if additional visual tools or other modalities are required, users can simply modify the tool selection section to include the necessary tools, making this prompt a highly flexible and scalable interface for building robust multimodal QA pipelines.

B.2 Reflection Prompt

The Reflection Prompt framework is divided into three complementary strategies - Neutral B.2.1, Conservative B.2.2, and Aggressive B.2.3 - each offering a different level of intervention to revise initial video QA outputs. These prompts are designed to support multi-stage reasoning and grounded visual correction by reassessing object perception, answer validity, or full reasoning chains. Depending on the task format (e.g., open-ended QA, multiple choice), scoring mechanism (e.g., scalar reward, structured feedback), or desired reflection granularity, users can select the appropriate strategy: Neutral for perceptual correction, Conservative for stability-focused validation, and Aggressive for complete answer reconstruction. These templates can also be adapted or hybridized to support diverse evaluation workflows, enabling flexible deployment across multi-task QA pipelines.

B.2.1 Neutral Reflection Prompt

Entity-Centric Revision

[Task]

You are a neutral agent responsible for reassessing the initial model answer by correcting only visual misperceptions of scene elements. Your role is to revise the perceptual input (i.e., object/entity grounding) while preserving the original reasoning logic. Do not introduce any new reasoning steps.

[Reflection Requirement]

If the original answer is based on a misidentified visual entity, correct that grounding (e.g., object type, color, spatial position). Keep the interpretation process unchanged. This strategy focuses on refining *what is seen*, not *how it is reasoned*.

[Reflection Procedure]

1. Re-examine the video or image frames for **object-level details** (e.g., people, objects, colors, gestures).
2. Determine whether the initial answer failed due to incorrect or missing perception of visual entities.
3. Adjust the relevant scene elements accordingly (e.g., update object color, position, or identity).
4. Reuse the original reasoning chain, now applied to the corrected visual grounding.

[Evaluation Guidelines]

- Remain neutral — do not assume the initial answer is incorrect unless there is clear evidence of perceptual error.
- Do not modify the reasoning logic — preserve the original inference path.
- Only revise the interpretation of visual content (i.e., what objects appear and their attributes).

[Input]

- Question: {text}
- Initial Answer: {initial_answer}
- Eval Report: {structured feedback or scalar reward}

[Output]

Use the following format to provide the final revised answer after entity-level correction. Only output the revised answer and its confidence score — no explanations, no justifications, and no extra text of any kind.

Final Answer: ({free-form revised answer}, confidence score: 0-1)

990

991 **B.2.2 Conservative Reflection Prompt****Answer-Focused Revision****[Task]**

You are a conservative agent responsible for validating the initial model answer. Your role is to preserve the original answer unless there is irrefutable visual evidence that directly contradicts it. Do not revise or reinterpret the reasoning or scene elements unless the contradiction is absolute.

[Reflection Requirement]

Only revise the final answer itself—do not modify scene elements or reasoning logic. If the original answer remains potentially valid, it should be retained. This strategy aims for minimal intervention: conservative reflection focuses on maintaining output stability unless overwhelming visual contradiction is present.

[Reflection Procedure]

1. Re-examine the visual content for any direct contradiction to the initial answer.
2. Accept only literal, unambiguous visual cues that fully invalidate the original answer.
3. Do not alter object interpretation, scene structure, or logical flow.
4. Revise the final output only if the contradiction is undeniable and renders the original answer unsupportable.

[Evaluation Guidelines]

- Retain the original answer if:
 - Any uncertainty or ambiguity exists in the evidence
 - Visual information lacks a clear, literal contradiction
 - A reasonable observer could still accept the original answer
- Revise the answer only if:
 - The visual content presents overwhelming and explicit contradiction
 - The revised answer exactly matches visible evidence without requiring interpretation
 - The contradiction is strong enough to convince any neutral evaluator

[Input]

- Question: {text}
- Initial Answer: {initial_answer}
- Eval Report: {structured feedback or scalar reward}

[Output Format]

Use the following format to provide the final decision after conservative validation. Only output the revised answer and your confidence score—no explanations, no justifications, and no extra text of any kind.

Final Answer: ({free-form revised answer}, confidence score: 0-1)

992

993 **B.2.3 Aggressive Reflection Prompt**

Reasoning-Driven Revision

[Task]

You are an aggressive agent responsible for actively challenging the initial model answer. Your role is to revise both the reasoning process and the visual understanding in order to reconstruct a superior alternative.

[Reflection Requirement]

This strategy requires modifying both the reasoning steps and the associated scene entities. It involves the widest scope of change and is intended to completely overturn the original logic and rebuild a more accurate answer from scratch. Accept loose semantic alignment, reinterpret ambiguous scenes, and prioritize alternative perspectives over the original.

[Reflection Procedure]

1. Re-examine all video frames for cues that could support a different interpretation.
2. Modify the understanding of relevant visual entities (objects, attributes, spatial relations).
3. Reconstruct the reasoning chain based on the newly grounded observations.
4. Select a new answer that best fits the revised reasoning and evidence.

[Evaluation Guidelines]

- Always replace the original answer — the default assumption is that it is suboptimal or incorrect.
- Consider alternative answers even when based on partial, ambiguous, or abstract cues.
- Allow semantic flexibility (e.g., “canine” = “dog”, “liquid” = “water”) and recontextualization.
- Prioritize comprehensive reinterpretation to generate an improved final response.

[Input]

- Question: {text}
- Initial Answer: {initial_answer}
- Eval Report: {structured feedback or scalar reward}

[Output Format]

Use the following format to provide the final revised answer after full reasoning and entity-level revision. Only output the revised answer and your confidence score — no explanations, no justifications, and no extra text of any kind.

Final Answer: ({free-form revised answer}, confidence score: 0-1)

995

996 **B.2.4 Overall Reflection Prompt**

Multi-Perspective Reflection Aggregation

[Task]

You are a specialized Meta-Agent for video question answering. Your role is to integrate the answers and confidence scores from three agents with different reflection strategies. Your goal is to synthesize a final answer by evaluating answer quality, confidence levels, and semantic overlap.

[Multi-Perspective Inputs]

- Conservative Agent (Answer-Focused Reflection)
Answer: {answer_conservative}, Confidence: {conf_conservative}
- Neutral Agent (Entity-Centric Reflection)
Answer: {answer_neutral}, Confidence: {conf_neutral}

997

- Aggressive Agent (Reasoning-Driven Reflection)
Answer: {answer_aggressive}, Confidence: {conf_aggressive}

[Decision Procedure]

Step 1 — High-Confidence Fusion

If all three confidence scores exceed their respective thresholds:

- $\text{conf_conservative} \geq 0.6$
- $\text{conf_neutral} \geq 0.7$
- $\text{conf_aggressive} \geq 0.8$

Then:

- Combine the three answers {answer_aggressive}, {answer_aggressive}, {answer_aggressive}
- Extract shared components and consistent semantic information
- Remove contradictions or unsupported segments
- Produce a final, coherent free-form answer that integrates the common insights

Step 2 — Confidence-Based Selection If one or more confidence scores fail to meet their thresholds:

- Select the answer with the highest confidence score among the three agents
- Use that agent's full revised answer as the final output

[Evaluation Criteria]

- **Semantic Overlap:** Identify key phrases, facts, and themes that appear in multiple answers
- **Contradiction Removal:** Discard any segments that directly conflict with others
- **Fluency:** Ensure the final answer reads as a natural, well-formed sentence

[Input]

- Question: {text}
- Initial Answer: {initial_answer}
- Agent Answers:
 - {answer_conservative}
 - {answer_neutral}
 - {answer_aggressive}
- Agent Confidences:
 - {conf_conservative}
 - {conf_neutral}
 - {conf_aggressive}

[Output Format]

Use the following format to provide the final revised answer. Only output the revised answer — no explanations, no justifications, and no extra text of any kind.

Final Answer: ({free-form revised answer})

998

999 B.3 Critical Prompt

1000 This section introduces two complementary evaluation prompts—Clarification Question Genera-
 1001 tion B.4 and Eval Report Generation B.5 - designed to support critical diagnosis of model answers
 1002 in video question answering. The Clarification Prompt generates precise, targeted sub-questions
 1003 when an answer is incomplete or ambiguous, uncovering reasoning or grounding gaps by refer-
 1004 encing specific visual or contextual elements. It is particularly useful for both open-ended and
 1005 multiple-choice QA tasks, helping localize errors without requiring immediate scoring. The Eval
 1006 Report Prompt provides a structured scoring mechanism across five dimensions—visual alignment,

1007 temporal accuracy, option disambiguation, reasoning specificity, and linguistic precision—yielding
1008 both qualitative feedback and a scalar reward useful for leaderboard tracking, model tuning, or
1009 reinforcement learning. Together, these prompts enable a flexible and task-adaptable evaluation
1010 framework: clarification questions are ideal for diagnosing failures and guiding revisions, while scalar
1011 scores support performance benchmarking. Depending on the evaluation goal—error localization,
1012 answer justification, or progress tracking—these prompts can be selectively applied or adapted, with
1013 clarification prioritized in open-domain QA and scoring emphasized in competitive or quantitative
1014 settings.

1015 B.4 Critical Question Prompt

Clarification Questions Generation Prompt

[Task]

You are a critic agent tasked with evaluating the quality of the initial answer A_0 generated by the target agent, based on the provided question, context, and video content. If the answer is deemed unsatisfactory, your goal is to help localize potential errors by generating one or more sub-questions. These sub-questions must be highly specific and firmly grounded in the visual or contextual evidence.

[Input Data]

Question: “{text}”

Answer: “{answer}”

Context: “{context}”

[Evaluation Criteria]

1. Check if the answer fully addresses the question
2. Verify all key elements from context are included
3. Assess whether video content supports the answer
4. If not, raise sub-questions to expose missing or uncertain reasoning

[Clarification Guidelines]

If the answer is incomplete, generate 1–3 ultra-specific clarification questions following these rules:

- Must start with: “What”, “Where”, “When”, “Which”, or “How”
- Must reference concrete elements from context/video
- No vague pronouns (“it”, “they”) — use specific nouns
- Examples: “Which timestamp shows the error?” or “How many frames were processed?”

[Output Format]

- Return [] (for complete answers)
- Return ["question1?", "question2?", ...] (for incomplete answers)

1016

1017 B.5 Evaluation Report Prompt

Eval Report Generation

[Task]

You are a critic agent tasked with evaluating the quality of the initial answer A_0 generated by the target agent, using the given question and contextual information. Your goal is to provide structured diagnostic feedback by scoring the answer across multiple dimensions and computing a final scalar reward (0.0–1.0) based on the total score.

[Input Data]

- Question: {text}
- Context: {context}
- Initial Answer: {initial_answer}

[Evaluation Criteria]

Rate the answer on the following five dimensions (0.0–5.0 scale for each):

- **Visual Alignment:** Is the answer aligned with visible video evidence?
- **Temporal Accuracy:** Is the answer consistent with the timeline or timestamps?
- **Option Disambiguation:** If multiple options are similar, does the answer clearly justify the selected one?
- **Reasoning Specificity:** Is the reasoning clear, focused, and appropriately detailed?
- **Linguistic Precision:** Is the answer grammatically correct and semantically accurate?

[Output Format]

Return a JSON object with detailed scoring and reasons for each dimension, plus the final total and normalized scalar reward:

```
{
  "scores": {
    "visual_alignment": {"value": float (0.0-5.0), "reason": "..."},
    "temporal_accuracy": {"value": float, "reason": "..."},
    "option_disambiguation": {"value": float, "reason": "..."},
    "reasoning_specificity": {"value": float, "reason": "..."},
    "linguistic_precision": {"value": float, "reason": "..."}
  },
  "total_score": float (0.0-25.0),
}
```

1018

1019 B.6 Implementation Workflow of ReAgent-V

1020 Figure 6 illustrates the complete inference workflow of ReAgent-V, a modular agent system de-
 1021 signed for video-based question answering. The pipeline begins with unified initialization via
 1022 load_default, followed by ECRS-based keyframe selection to reduce redundancy while preserving
 1023 semantic relevance. A dictionary-based tool selection mechanism dynamically activates symbolic
 1024 extraction tools (e.g., OCR, ASR, DET) based on the input query. Extracted textual context is merged
 1025 into modal_strings and composed into a multimodal prompt for LLaVA-based initial answering. If
 1026 critical gaps are identified, the system enters a reflective reasoning stage to revise the answer. Finally,
 1027 an evaluation report is generated to assess the quality of both the initial and refined responses.

1028 C More Visualization Results

1029

ReAgent-V Inference Pipeline

```
>>> from init_modules import *
>>> qa_system = ReAgentV.load_default("path/to/base_model")
>>> frames = qa_system.load_video_frames("example.mp4", max_frames)
>>> key_frames, key_indices = qa_system.ECRS_select_keyframes(frames,
    question)
>>> # Pass custom tool list to dynamically control which tools to apply and
    revise the tool selection prompt template accordingly.
>>> tool_list = ["OCR", "ASR", "DET", "SceneGraph", "Grounding DINO", "
    Caption Model", "CLIP", "ShareGPT4Video", "VQA Model", "Action Detector",
    "Face Recognition", "Emotion Detector"]
>>> # "selected_tools" is a boolean map indicating required tools for
    answering the question, e.g., {"OCR": True, "ASR": False, "DET": False,
    ...}
>>> selected_tools = qa_system.select_tools(question, tool_list=tool_list)
>>> # Use selected_tools to use only necessary models and extract tool-
    specific information from key_frames into modal_info, a dictionary keyed
    by tool name.
>>> modal_info = qa_system.extract_modal_info(key_frames, question, **
    selected_tools)
>>> # Build a multimodal prompt by integrating tool-specific information from
    modal_info into the question template.
>>> prompt = qa_system.build_multimodal_prompt(question, modal_info,
    key_indices, len(key_frames))
>>> initial_answer, = qa_system.model_inference(prompt, key_frames)
>>> # Critic-driven refinement if necessary
>>> critical_qs = qa_system.generate_critical_questions(question,
    initial_answer, modal_info, key_frames)
>>> if critical_qs:
...     updated_infos = {} # {q_i: {tool_name: info}}, mapping each critic
    question to its tool-specific info.
...     for q_i in critical_qs:
...         tools_i = qa_system.select_tools(q_i, tool_list=tool_list)
...         info_i = qa_system.extract_modal_info(key_frames, q_i, **tools_i)
...         updated_infos[q_i] = info_i

...     # Wrap the original modal_info into {question: modal_info}
...     context_infos = {question: modal_info, **updated_infos} # Merge all
    into a unified context dict

...     report = qa_system.generate_eval_report(question, initial_answer,
    context_infos, key_frames)
...     # get_reflective_final_answer applies three reflection strategies (
    neutral, aggressive, conservative) and merges the best answer using the
    overall_prompt_template.
...     final_answer = qa_system.get_reflective_final_answer(
    question, initial_answer, report, key_frames)
>>> else:
...     final_answer = initial_answer
```

Figure 6: ReAgent-V inference pipeline: after ECRS keyframe selection, tools are dynamically selected to generate an initial answer. If critical questions arise, tool outputs are updated for reflection. Three reasoning strategies are used to revise or confirm the answer.

1030 C.1 Visualization of Frame Selection

1031 These case studies qualitatively demonstrate the effectiveness of ECRS frame selection compared
1032 to uniform sampling across a range of video question answering scenarios. In each example,
1033 ECRS consistently captures frames that are more semantically aligned with the question, providing
1034 richer context and clearer evidence to support reasoning. For instance, in Figure 7, ECRS selects

Four Frames Sampled Using Uniform Sampling



Four Frames Sampled Using ECRS Sampling



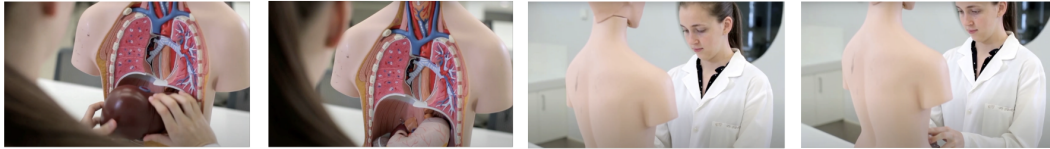
Question about the video : How did the photographer who took pictures of the three tourists in the video take the photo?

Figure 7: ECRS sampling better captures the interaction between the photographer and the tourists.

Four Frames Sampled Using Uniform Sampling



Four Frames Sampled Using ECRS Sampling



Question about the video : What organ did the woman in the video remove from the medical model ?

Figure 8: ECRS highlights the woman’s action of removing the organ more clearly than uniform sampling.

frames that highlight the interaction between the photographer and tourists, directly addressing the question about how the photo was taken—unlike uniform sampling, which includes generic landscape shots. In Figure 8, ECRS captures the precise moment when the woman interacts with the medical model, effectively grounding the answer to the question about organ removal. Similarly, Figures 9 and 10 show that ECRS preserves critical moments involving text and actions—such as the student explaining his motivation for eating the banana or the banana-taping act that contextualizes the replaced painting—while uniform sampling misses or misaligns with these moments. Across all examples, ECRS provides temporally and semantically focused evidence that improves alignment between selected frames and the target question, validating its superiority in supporting visual reasoning. Note that although ECRS typically selects 20–40 informative frames per video for downstream processing, only four representative frames are visualized here by evenly sampling from the selected set, in order to maintain clarity and consistency in comparison.

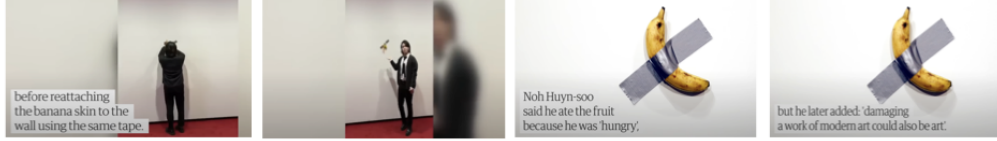
C.2 Visualization of Evaluation Report

These four evaluation report cases demonstrate how ReAgent-V leverages visual tools and multi-agent collaboration to refine or correct initial answers by grounding them more accurately in visual and textual evidence. In Figure 11, the system correctly identifies the Dragon Boat Festival as the main theme of the video by combining OCR and DET outputs to interpret cultural symbols and scene elements. Figure 12 shows a quantitative reasoning adjustment, where the agent detects two birds and

Four Frames Sampled Using Uniform Sampling



Four Frames Sampled Using ECRS Sampling



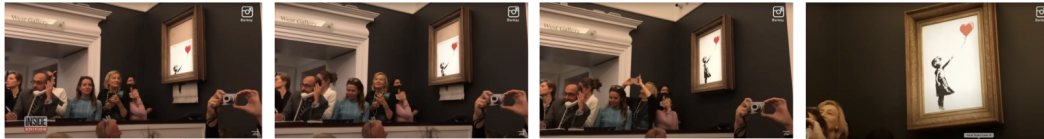
Question about the video : Based on the video, which of the following descriptions the reason why the student ate the banana?

Figure 9: ECRS sampling emphasizes the reason behind the student eating the banana with supporting text.

Four Frames Sampled Using Uniform Sampling



Four Frames Sampled Using ECRS Sampling



Question about the video : Which elements are depicted in the painting introduced by the video?

Figure 10: ECRS captures the banana-taping act that replaced the shredded painting, providing better context.

1053 revises the initial bird count from three to two, aligning the answer with grounded visual analysis. In
 1054 Figure 13, the model originally misidentifies the cat’s color, but after grounding the detected regions
 1055 and reviewing visual attributes, the agents collaboratively revise the answer to the correct “orange
 1056 and white.” Finally, Figure 14 illustrates a geographic correction where the system initially mislabels
 1057 the terrain as “tropical,” but after integrating CLIP-based semantics and detailed grounding, it updates
 1058 the answer to “polar,” aligning better with snowy and icy visual cues. These examples highlight how
 1059 reflective critique and grounded evidence improve factual accuracy and visual-textual alignment.

1060 C.3 Visualization of VLA Alignment

1061 Figure 15 visualizes six robotic manipulation tasks and highlights the progression from failure
 1062 to success after applying reflection-guided reward correction during policy fine-tuning. In each
 1063 task—carrot placement, spoon placement, eggplant basket placement, cube stacking, coke can
 1064 placement, and sprite can placement—the left sequence illustrates the robot’s initial failure to
 1065 complete the action correctly, while the right sequence demonstrates the corrected behavior after
 1066 fine-tuning. The visualizations show that the robot learns to adapt its positioning, trajectory, and
 1067 precision based on reflective feedback. For example, in the carrot and spoon placement tasks, the

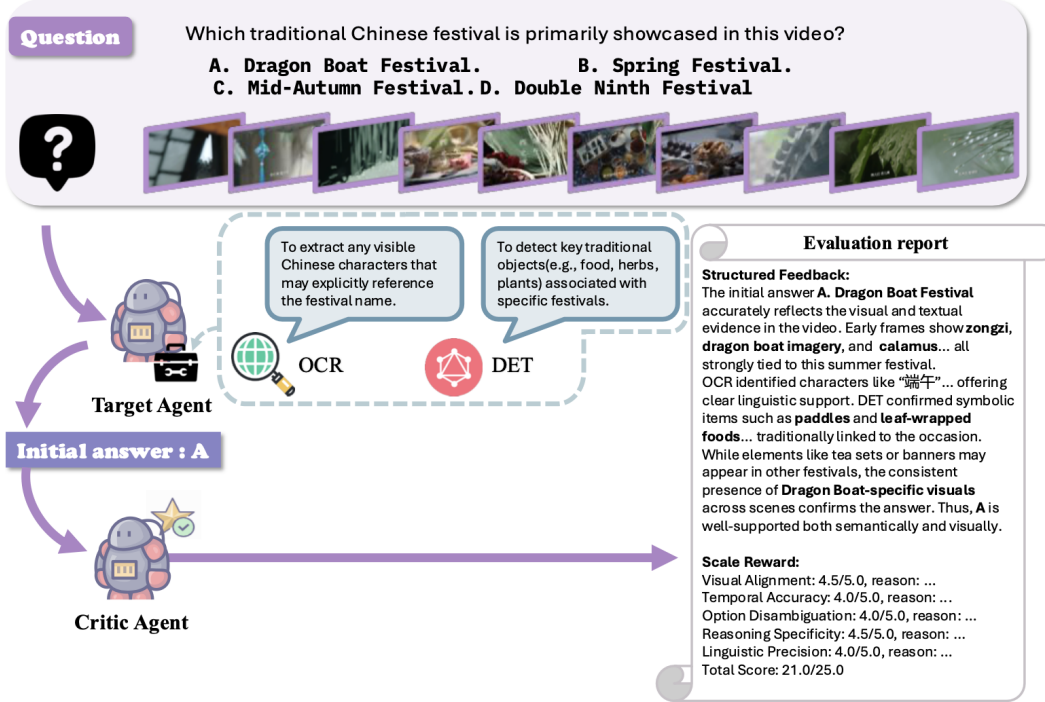


Figure 11: Visual and textual cues confirm that the video primarily showcases the Dragon Boat Festival.

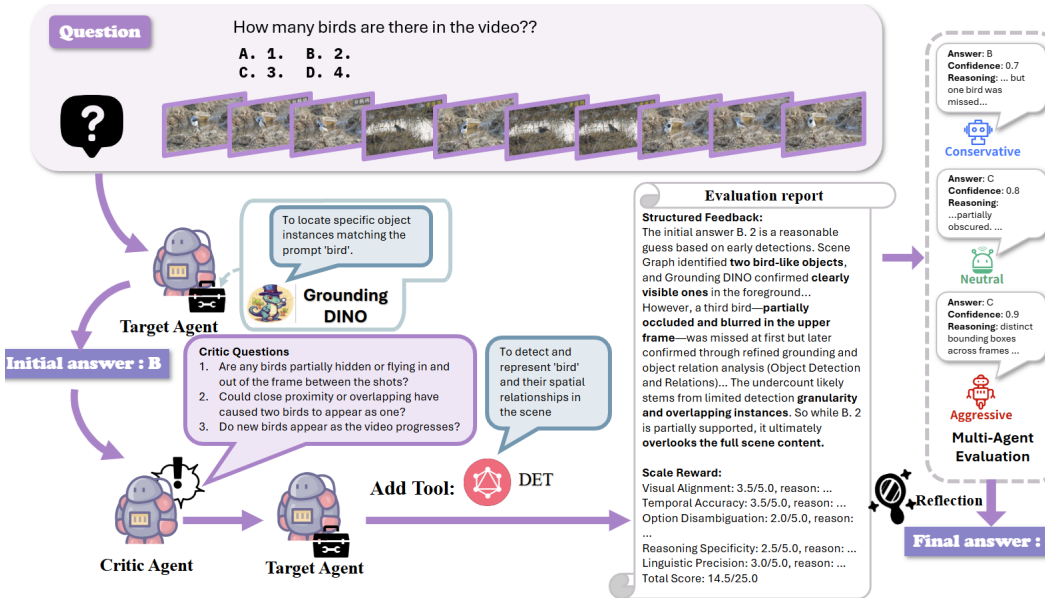


Figure 12: Multimodal analysis revises the bird count in the video from two to three after detecting an occluded bird.

1068 gripper initially misaligns with the target location but is corrected to center the object on the plate
1069 or towel. In the eggplant and cube stacking tasks, the robot improves its grasp and drop accuracy,
1070 successfully placing the object in or on the intended target. In the coke and sprite can tasks, the robot
1071 adjusts its vertical alignment and release timing to ensure stable placement. These results collectively
1072 demonstrate that reflection-driven fine-tuning enhances the robot’s task completion reliability across
1073 diverse object manipulation scenarios.

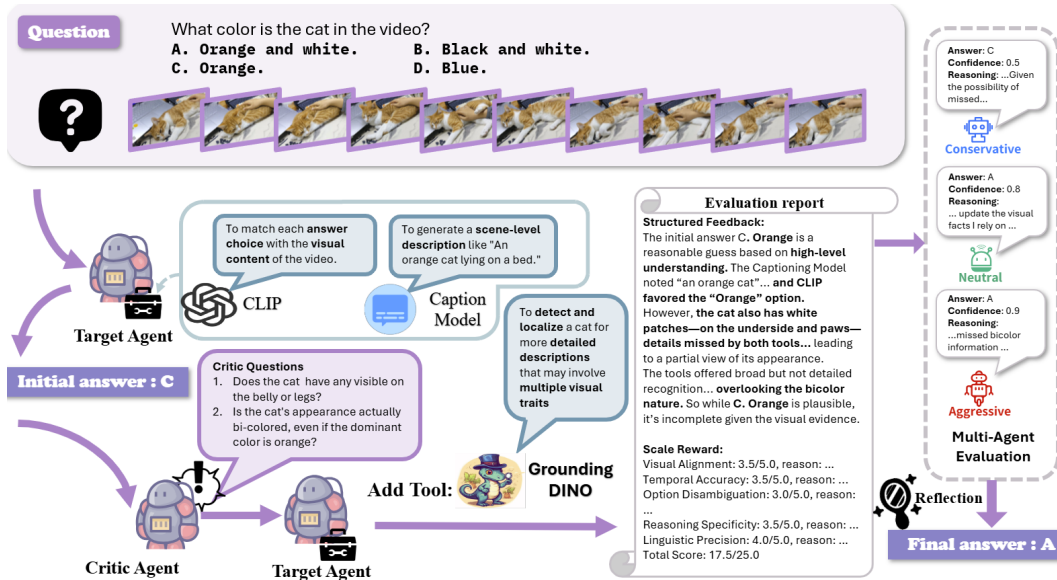


Figure 13: Multi-agent collaboration corrects the cat's color in the video from "Orange" to "Orange and white."

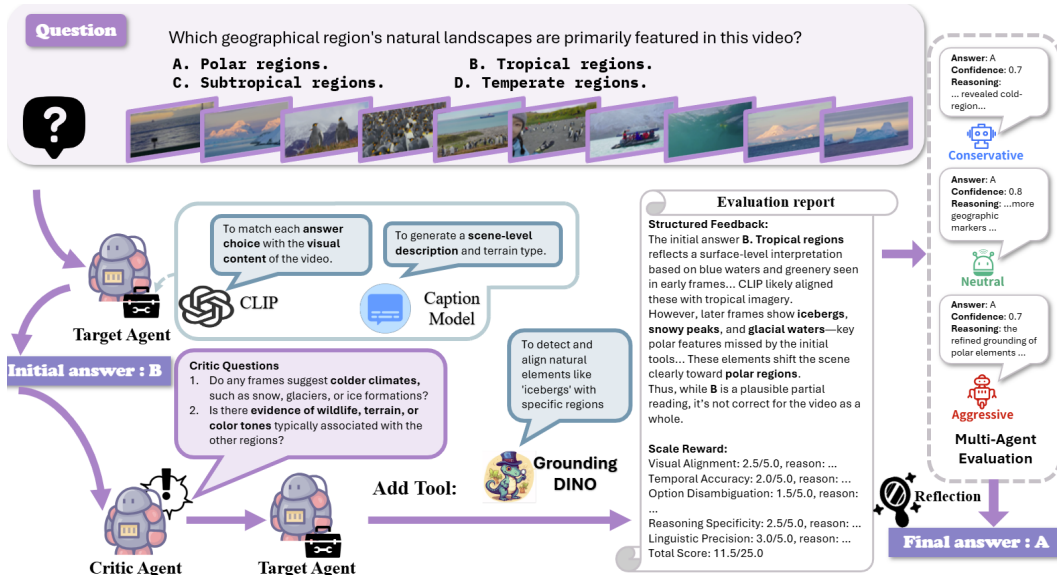


Figure 14: Agents identify icebergs and snowy terrain, correcting the geographical region from "Tropical" to "Polar."

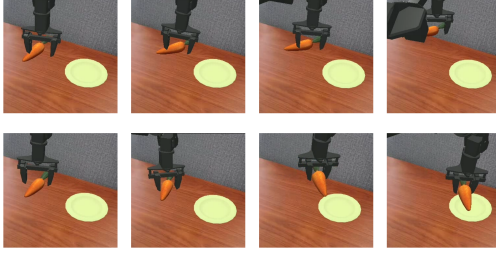
1074 D Additional Experimental Results

1075

1076 D.1 Trends of Selection Scores Across Videos

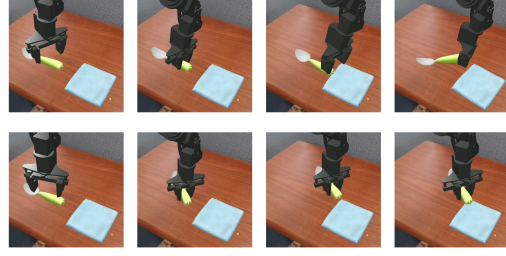
1077 Figure 16 illustrates the selection behavior of ECRS across various videos by comparing its score
1078 distribution with CLIP and Entropy-based baselines. The red bars indicate that ECRS consistently
1079 selects frames near local peaks, capturing semantically or visually informative moments. Compared
1080 to CLIP and Entropy score selections, ECRS demonstrates more temporally clustered choices that

Task: put the carrot on the plate (failed -> success)



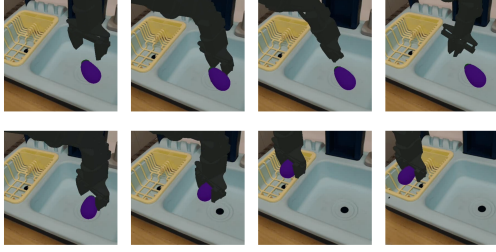
(a) Carrot placement

Task: put the spoon on the towel (failed -> success)



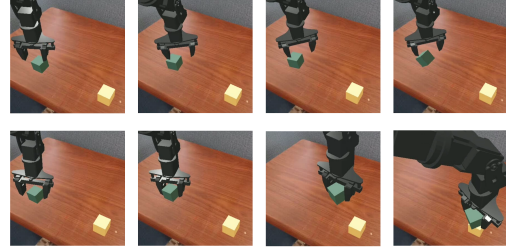
(b) Spoon placement

Task: put the eggplant in the basket (failed -> success)



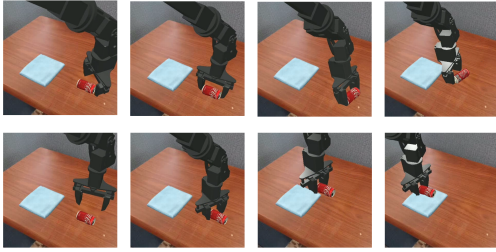
(c) Eggplant basket placement

Task: stack the green cube on the yellow cube (failed -> success)



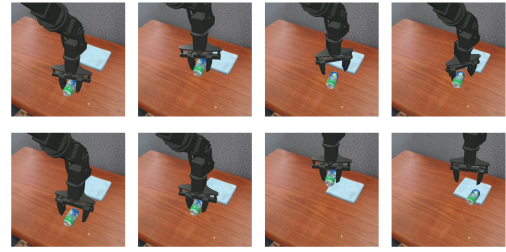
(d) Cube stacking

Task: put the coke can on the towel (failed -> success)



(e) Coke can placement

Task: put the sprite can on the towel (failed -> success)



(f) Sprite can placement

Figure 15: Visualization of robotic task executions before and after policy fine-tuning using reflection-guided reward correction, showing progression from failure to success across four tasks: (a) carrot placement, (b) spoon placement, (c) eggplant basket placement, (d) cube stacking, (e) coke can placement, and (f) sprite can placement.

1081 align with high-compactness regions, indicating its ability to perform more focused and discriminative
 1082 frame selection throughout the iterations.

1083 D.2 Frame Index Trends Across Iterations

1084 Figure 17 presents the iterative frame reduction behavior of ECRS across multiple videos, showing
 1085 trends in average self-compactness (blue), GT-compactness (orange), and the number of selected
 1086 frames (green). Self-compactness refers to the average index distance among the selected frame
 1087 indices at each iteration, indicating how temporally clustered the selections are. GT-compactness
 1088 measures the average index distance between selected frames and ground truth-relevant frames,
 1089 reflecting how well the selected subset aligns with semantically important moments in the video. As
 1090 iterations progress, both compactness metrics consistently decrease, demonstrating that ECRS not
 1091 only reduces the number of selected frames but also improves their temporal tightness and alignment
 1092 with ground truth. The number of selected frames drops sharply in early iterations and then stabilizes.
 1093 This trend highlights ECRS’s ability to efficiently eliminate redundant frames while preserving a
 1094 compact, semantically meaningful subset aligned with human-annotated content.

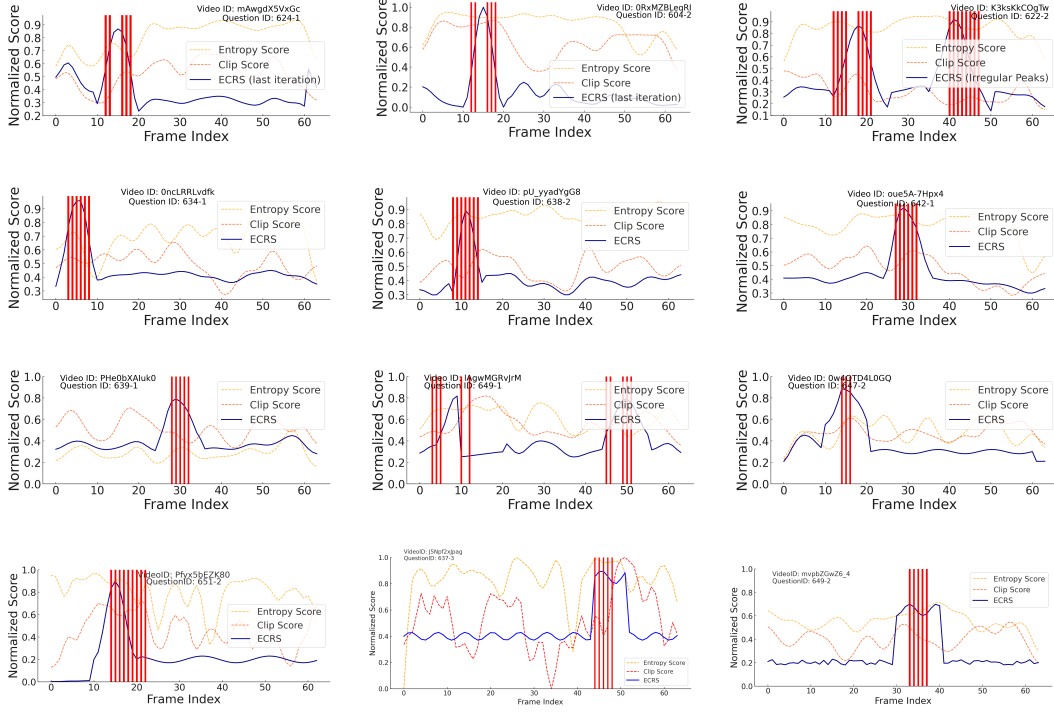


Figure 16: Comparison of frame selection strategies across multiple videos, illustrating how ECRS (final iteration), Entropy Score, and CLIP Score vary with frame index and influence selected keyframes (highlighted in red), along with their overlap with ground truth frames manually annotated by experts—refer to the VideoMME dataset for corresponding video IDs and validation details.

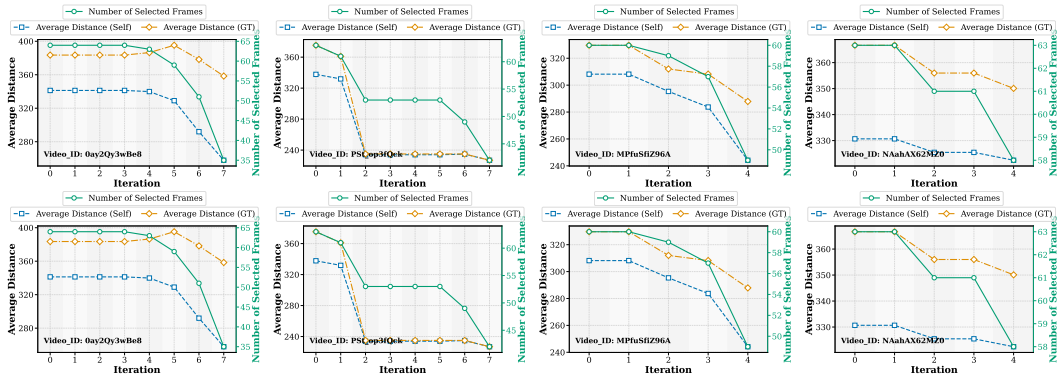


Figure 17: This figure shows tracks two compactness metrics - self-compactness (left y-axis, mean squared distance to selected frames' centroid) and GT-compactness (left y-axis, mean squared distance to selected and ground truth centroids) and the number of selected frames) - along with frame count (right y-axis) across iterations (x-axis) for videos from the (Video-MME/MLVUI datasets. L), where lower left-axis values indicate tighter clustering.

1095 D.3 Comparison of Frame Selection Methods Across Varying Input Frame Numbers

1096 Figure 18 shows that ECRS consistently outperforms other frame selection strategies across all frame
 1097 counts on the VideoMME dataset. As the number of input frames increases from 8 to 64, accuracy
 1098 improves across all methods, and the performance gap between strategies remains consistent. While
 1099 these trends are observed across different base models, the results shown here use LLaVA-Video-7B,
 1100 which exhibits particularly strong performance, further highlighting the effectiveness of ECRS in
 1101 enhancing model accuracy through more informative frame selection.

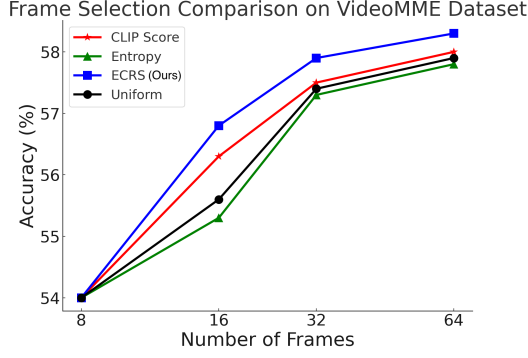


Figure 18: Frame selection strategies on VideoMME (LLaVA-Video-7B).

Table 5: Ablation study on the effect of Visual Tools across different base models and benchmarks.

Model	Visual Tools	LongBench	EgoSchema	VideoMME
LLaVA-Video-7B	✓	53.1	60.8	57.9
	✗	46.1	56.7	56.3
LLaVA-Video-72B	✓	65.3	74.0	75.5
	✗	60.9	75.0	67.3
Qwen2-VL-7B	✓	46.4	56.4	58.3
	✗	44.9	55.6	53.8
Qwen2.5-VL-72B	✓	66.4	76.2	75.1
	✗	62.6	75.7	74.3

D.4 Ablation of Visual Tools

Table 5 presents an ablation study on the impact of incorporating Visual Tools across different base models and benchmarks. Across all settings, enabling visual tools consistently improves performance on LongBench, EgoSchema, and VideoMME. The gains are particularly notable for LLaVA-Video-7B and Qwen2-VL-7B, where enabling visual tools leads to substantial improvements—for instance, +7.0 on LongBench and +4.5 on VideoMME for LLaVA-Video-7B. While the performance gap is smaller for larger models like LLaVA-Video-72B and Qwen2.5-VL-72B, improvements are still evident on LongBench and VideoMME. These results indicate that visual tools provide valuable auxiliary signals that enhance multimodal reasoning, especially for smaller or less capable base models.

Limitation

Despite employing keyframe selection, handling long videos remains challenging due to information redundancy and complex semantic relationships, which hinder the model’s ability to achieve comprehensive and accurate understanding and reasoning. Additionally, the reflection and evaluation mechanisms rely on heuristic-based rules or templates, lacking adaptive and end-to-end learning capabilities, which may limit the effectiveness of automatic error correction.

Social impact

Our method enhances video understanding and reasoning, which can benefit education, assistive technologies, and human-computer interaction. However, potential risks such as privacy concerns and the propagation of dataset biases should be carefully considered during deployment.