Understanding HCQA's Inference-Guided Reasoning in Ego4D Video QA

Mujtaba Saqib

Hilal Aziz

Ali Hyder Bhellar

Toheel Ali Chandio

Attaullah Ansar

2025

Abstract

HCQA (Hierarchical Comprehension Question Answering) was the winning method for the Ego4D EgoSchema 2024 egocentric video QA challenge. It tackles long egocentric videos (3 min) with a three-stage pipeline: fine-grained captioning, context summarization, and inference-guided answering. Our focus here is the reasoning component (Inference-Guided Answering) – how it works, its limitations, and how we can improve it. We'll also look at recent enhancements by other researchers, especially in the reasoning pipeline, and compare them to HCQA.

1 HCQA's Reasoning Component: Structure and Process

Illustration of HCQA's pipeline: The system first generates detailed captions for video segments, then produces a summary of the whole video. Finally, an LLM (like GPT-4) is prompted with the captions, summary, and question to reason out an answer (with a self-reported confidence score). The LLM's output includes a step-by-step Reasoning and the final Answer, plus a Confidence estimate.

1.1 How Inference-Guided Answering Works in HCQA

- **Input to the LLM:** HCQA feeds the hierarchical visual information into a Large Language Model. This includes fine-grained captions (e.g., descriptions of 4-second clips) and a high-level summary of the entire video. The question and multiple-choice options are also provided to the model.
- **Chain-of-Thought Prompting:** The LLM is prompted to think step-by-step about the question before answering. This Chain-of-Thought (CoT) approach explicitly asks the model to output its reasoning process (like a short explanation) prior to giving the final answer. By guiding the model to "show its work," HCQA improves accuracy on complex visual questions.

- **In-Context Examples:** To help the model reason effectively, HCQA uses a few example QA pairs in the prompt (a form of few-shot learning). In practice, they include three high-quality examples from the EgoSchema dataset as exemplars. These examples show the model how to use the captions and summary to reason, making it easier for the LLM to follow the intended reasoning process.

- **Reflection and Self-Check:** A novel addition in HCQA is a reflection mechanism. After the model produces an answer, it is asked to assign a confidence score to its answer (e.g. 1–10). If the confidence is below a threshold (they used 5), the model is instructed to reflect and revise its answer. This means the LLM will double-check its reasoning and potentially correct any errors if it was not confident. This step is like the model "re-thinking" when it's unsure, which yielded a small performance boost in their experiments.

Example: Imagine the egocentric video shows someone washing dishes, and the question asks: "What is the person's primary activity?" The HCQA pipeline would generate detailed captions like "0:00-0:04 – Person (C) scrubs a tray; 0:08-0:12 – C rinses a plate…", then summarize "C spends the video washing and rinsing various kitchen utensils." When asked the question, the LLM is prompted to reason: "All actions involve washing kitchen items, no other major activity is seen, so the primary objective is cleaning dishes." It then answers: "C is cleaning dishes." and might output a high confidence (say 9/10). If it had low confidence, it would reconsider the summary and captions to refine its answer.

Limitations of HCQA's Reasoning Approach

While HCQA's inference-guided answering was effective, it has some **limitations** in its reasoning pipeline:

- **Caption-Question Misalignment:** The video captions and summary are generated **without explicitly considering the question**. This means some question-specific details might be missing. The model only sees whatever the captioning model described, which could **omit relevant clues** needed for that particular question. For instance, if the question asks about "what the person does after X," but the captions weren't focused on that, the reasoning might overlook it.

- **Limited Temporal Reasoning:** The approach captures local and global info, but reasoning is largely based on the text summary. The LLM might not fully understand the **temporal order or causal relations** in the video just from the summary. It can struggle with questions requiring precise event sequencing or duration (e.g., "before/after" relationships), since frame-by-frame captions "stitching" is imperfect.

- **Dependence on LLM's Correctness:** HCQA relies on GPT-4 (a very advanced but proprietary model) to do reasoning. If the LLM **misinterprets** the captions or **hallucinates** facts not in the video, the answer can be wrong. The chain-of-thought helps guide reasoning, but it's not foolproof – the model might still make logical errors or use world knowledge incorrectly if the video cues are unclear.

- **One-Pass Answering (Despite Reflection):** The reflection mechanism adds a re-check, but it only triggers when the model *itself* senses low confidence. If the model is confidently wrong (i.e., it thinks it's right but isn't), HCQA won't catch that. There's no external

verification of the reasoning – it's all internal to the LLM. This means certain mistakes (especially subtle ones where the model is overconfident) may slip through.

- **Resource and Speed Constraints:** Though not a conceptual limitation of reasoning quality, it's worth noting HCQA's process uses multiple calls to a large LLM (for summary and QA) and few-shot examples. This can be **slow and costly** (GPT-4 is expensive and rate-limited). It's a practical limitation when scaling to many questions or deploying in real-time scenarios.

In summary, HCQA's reasoning component smartly uses an LLM with structured prompts to break down the problem, but it can miss question-specific details and struggles with the inherently **long, temporal nature** of egocentric video. The method is only as good as the captions provided and the LLM's own reasoning abilities, which have room for improvement.

2 How Can We Improve the Reasoning? (Proposed Enhancements)

1. **Question-Guided Video Processing:** Make the captioning/summarization stage aware of the question. Instead of generating captions blindly, the system could focus on frames and actions relevant to the question. For example, if the question asks "why did the person do X?", the system should emphasize cause-effect events in captions. One approach is to use the question to filter or weight the captions (only keep those related to the query). This ensures the reasoning stage has all the pertinent details. Implementation idea: Use a vision model or an LLM to first read the question and then select key video segments (or objects) to describe in detail (a "question-guided captioner"). This reduces noise and highlights the evidence needed for inference.

2. **Better Temporal and Causal Modeling:** Introduce an intermediate representation that captures the *sequence of events or relationships* in the video. For instance, build a simple timeline of actions or a *scene graph* of interactions. This can help answer questions about "before/after", "when did X happen," or "who did what" more reliably. *Example:* if the video shows a person unlocking a door then entering, a timeline representation would clearly show "Unlock door → then Enter room". The reasoning module can use this to avoid confusion. *Implementation idea:* Use a small *temporal reasoning module* or even prompt the LLM to explicitly list events in order before answering. By forcing a structured understanding of the video's progression, the model can reason about temporal questions more accurately.

3. **Multi-Step or Modular Reasoning:** Instead of one big LLM handling everything, we can *split the reasoning into specialized parts.* Recent research shows benefits in having multiple "agents" or modules each handling a modality or subtask. For example:

   - A **Text agent** that reads captions and picks out key facts (especially those relevant to the question).
   - A **Visual agent** that might inspect raw video frames for specific details (e.g., recognizing an object or reading text in the video, if needed).
   - A **Knowledge agent** or **Graph agent** that looks at relationships (who is doing what to whom, and why).

These agents can then share their findings (like a team of experts), and a coordinator module (or the LLM itself acting as an organizer) combines them to answer the question. This divide-and-conquer approach means each part can reason in-depth on its specialty, potentially catching details a single model might miss. While a full multi-agent system can be complex, even a simpler version helps – for example, first use the LLM to extract relevant info from captions, then on a second pass answer the question using *only* that relevant info (filtering out distractions).

4. **Uncertainty and Self-Checking Enhancements:** HCQA's reflection is a step in the right direction, but we can expand on it. One idea is to use *self-consistency*: have the model generate multiple reasoning paths (Chain-of-Thoughts) and see if they lead to the same answer. If different "trains of thought" converge on one answer, we can be more confident in it; if they disagree, that's a sign of uncertainty. The final answer could be chosen by majority vote of these reasoning paths, which has been shown to improve reliability in QA tasks. Another improvement is to have the model explicitly check if each part of its reasoning is supported by the captions/summary (*evidence checking*). For example, after the model reasons out an answer, prompt it with: "Which caption or part of the summary supports each step of your reasoning?" This forces it to align closely with the video evidence and can catch hallucinations or unsupported claims.

5. **Fine-Tuning Specialized Models:** HCQA used a prompting approach with a general LLM. We could instead *fine-tune a model on egocentric QA data* so that it learns to reason about videos more natively. With EgoSchema and similar data, one could train a smaller open-source multimodal model to understand egocentric scenarios (first-person perspective nuances, common activities, etc.). By training on many QA examples, the model might learn recurring patterns (for example, in egocentric videos, "if hands are washing dishes in multiple shots, the person is likely cleaning up after a meal"). Fine-tuning can also integrate the reasoning steps (the model could be trained to produce a reasoning chain and answer given video captions). This *theoretical improvement* would make the reasoning process more grounded and potentially more robust.

Recent Enhancements in Egocentric Video QA Reasoning

Since HCQA's win in 2024, there have been new approaches aiming to push egocentric video QA further. These works often build on similar principles (caption + LLM reasoning) but with innovations to improve reasoning and understanding. Here are a few notable ones and how they compare to HCQA:

- **VideoMultiAgents (2025)** – This approach introduced a **multi-agent framework** for video QA. Instead of one monolithic reasoning step, it has dedicated agents for text, vision, and even scene-graph reasoning, coordinated by an organizer agent. Importantly, it uses **question-guided captioning**: the text agent generates captions focusing on parts relevant to the question. This directly addresses HCQA's issue of caption-question misalignment.

  In evaluations on EgoSchema, VideoMultiAgents achieved state-of-the-art accuracy on the *challenge's public subset* (75.4% vs the previous best ∼72% for that subset). However, on the full test set its accuracy (∼68%) was slightly below HCQA's (75%). The authors noted an interesting pattern: HCQA had a much higher full-set score than subset (75% vs ∼59%),

possibly because HCQA was tuned on the full data distribution. This suggests VideoMultiAgents' strategy excels at complex reasoning on many questions (hence the subset boost), but there's room to adapt it better to all question types.

Overall, MultiAgents showed that **dividing the reasoning by modality and using question-focused information can yield more robust reasoning**, especially for long videos with rich content.

- **VideoAgent2 with Uncertainty-Aware CoT (2025)** – This work explicitly tackled the **reasoning pipeline's reliability**. VideoAgent2 builds on an LLM-based agent (similar in spirit to HCQA, using tools and LLM planning) but adds an **uncertainty-guided chain-of-thought**. The idea is to have the LLM **assess its own uncertainty and the confidence of external tools at each reasoning step**, and use that to decide what to do next.

  For example, if the LLM isn't sure it has enough info, it will plan a targeted retrieval (e.g., "go look at frames 10s–20s for the object in question") and only finalize an answer when confidence is high. This is like an extended version of HCQA's reflection mechanism but built into a multi-step reasoning loop. The authors noted this approach *mitigates hallucinations and errors* in the agent's reasoning without needing additional training parameters.

  In practice, VideoAgent2 showed more reliable performance on long video QA benchmarks (including EgoSchema) compared to prior methods, thanks to this self-checking strategy. In simple terms, VideoAgent2's LLM "knows when it doesn't know'' and can **pause to gather more information**, leading to smarter reasoning. This addresses the scenario where HCQA's single-step LLM might confidently give a wrong answer – VideoAgent2 would detect low confidence earlier and try to find the needed evidence before answering.

- **Fine-Tuned Multimodal LLMs for EgoQA (2025)** – Another trend has been to train or fine-tune models specifically for video QA, rather than relying on GPT-4 with prompts. For instance, researchers created **QaEgo4Dv2**, a cleaned-up egocentric QA dataset, and fine-tuned open-source multimodal models like **Video-LLaVa-7B** and **Qwen-VL-7B** on it.

  The result: these tuned models set new records, **outperforming the prior state-of-the-art (HCQA and others) by a large margin** – up to **+13% accuracy** improvement on multiple-choice questions. That's a big jump, indicating that a specialized model can understand egocentric video questions better than a generic LLM guided by prompts.

  These models likely learned to handle egocentric quirks (first-person viewpoint, camera motion) and long temporal reasoning through training. The downside is the need for training data and computational cost to fine-tune. But the upside is once trained, the model can answer quickly and *more accurately on fine-grained details* than zero-shot methods.

  An error analysis from this work showed that even these models still struggle with **spatial reasoning and tiny object details** – challenges that HCQA also faced. So, there is still room to combine the best of both worlds: e.g., use fine-tuned models but also incorporate chain-of-thought prompting or multi-agent modules to further boost reasoning on those hard aspects.

Comparison of HCQA and Enhanced Approaches

The table below provides a brief comparison of HCQA's original reasoning approach with some of these enhanced approaches:

Aspect	HCQA (2024) – Inference-Guided Answering	Enhanced Approaches (2025) – What's New
Input to Reasoning	Pre-generated captions and summary not tailored to question; may miss details.	**Question-guided captions** or frame selection to ensure relevant content is included.
Reasoning Method	**Single GPT-4 CoT prompting**. One-pass with optional re-check.	**Modular or multi-agent reasoning**, iterative planning with **uncertainty-aware** adjustments.
Handling Uncertainty	**Reflection mechanism**, only triggers on low self-confidence.	**Uncertainty-guided CoT**, which pauses or adjusts based on confidence throughout.
Model Training	**Few-shot GPT-4 prompting**. No training on video data.	**Fine-tuned multimodal models** trained on egocentric QA, outperforming GPT-4.

Table 1: Comparison of HCQA and enhanced reasoning approaches.

Conclusion

HCQA's inference-guided answering component was a milestone in egocentric video QA – it showed how combining rich video captions with an LLM's reasoning (through chain-of-thought and self-reflection) can tackle complex, long videos. We broke down its method in simple terms: essentially, **"describe the video in detail, summarize it, then let the LLM think out loud to pick the answer"**. We also saw its limitations, like not always looking at the right details or fully understanding the timeline of events, and depending on the LLM's correctness. To overcome these issues, we proposed improvements such as making the video descriptions **query-aware**, adding **structured temporal reasoning**, splitting the task among **specialized sub-models**, improving the model's **self-checking abilities**, and even **training models** directly for this task. These ideas aim to make the reasoning more precise, robust, and aligned with what the question is asking. Excitingly, the research community has already begun exploring many of these directions. Multi-agent frameworks and uncertainty-guided strategies have demonstrated that more **interactive and modular reasoning pipelines** can outperform the initial HCQA approach in certain settings. Meanwhile, fine-tuned video-language models are pushing accuracy even further, showing the power of adapting models to the egocentric domain. For an undergraduate student presentation, the take-home message is: **effective video QA needs both "understanding the video" and "reasoning through the answer."** HCQA made a great step by feeding a lot of structured information into an LLM and guiding its reasoning. New methods build on this by making the understanding part smarter (focusing on what matters) and the reasoning part more human-like (stepwise, checking itself). By combining these improvements – perhaps a model that *looks at the right clues, thinks in steps, double-checks itself, and maybe has been trained on similar problems* – we can significantly improve how AI answers questions about egocentric videos. This means more accurate and trustworthy answers, which is the ultimate goal of such QA systems.

Sources

- Haoyu Zhang *et al.*, "HCQA @ Ego4D EgoSchema Challenge 2024", *arXiv preprint* (2024). – Introduces the HCQA method and its inference-guided answering with chain-of-thought and reflection.
- Noriyuki Kugo *et al.*, "VideoMultiAgents: A Multi-Agent Framework for Video Question Answering", *arXiv preprint* (2025). – Describes a multi-agent reasoning approach and notes limitations of caption-based methods (lack of query focus and temporal context). Shows improved performance on EgoSchema subset with query-focused captioning and specialized agents.
- Zhuo Zhi *et al.*, "VideoAgent2: Enhancing the LLM-Based Agent System for Long-Form Video Understanding by Uncertainty-Aware CoT", *arXiv preprint* (2025). – Introduces uncertainty-guided chain-of-thought, allowing the model to adjust its plan based on confidence, thereby reducing errors.
- Alkesh Patel *et al.*, "Advancing Egocentric Video Question Answering with Multimodal LLMs", *arXiv preprint* (2025). – Evaluates fine-tuned multimodal LLMs (Video-LLaVa, Qwen-VL) on egocentric QA. Reports new state-of-the-art results with up to 13% accuracy gain in multiple-choice QA, highlighting the benefit of domain-specific training.