

EgoFact: A BENCHMARK FOR MULTI-HOP MULTI-MODAL RETRIEVAL-AUGMENTED GENERATION

Anonymous authors

Paper under double-blind review

ABSTRACT

Retrieval-Augmented Generation (RAG) has emerged as a powerful approach to improve large language models (LLMs) by grounding their outputs in external knowledge. However, progress in the multimodal domain remains limited, largely due to the lack of suitable benchmarks. Existing multimodal corpora are often built by merging unimodal datasets, which rarely support queries requiring multi-hop reasoning and thus reduce most tasks to single-modality, one-hop retrieval. To address this gap, we introduce *EgoFact*, the first benchmark explicitly designed for multi-hop reasoning across visual and textual corpora. Success on *EgoFact* requires models to retrieve and integrate evidence spanning multiple modalities. We systematically evaluate existing RAG systems and uncover fundamental limitations in multimodal evidence integration and reasoning. Motivated by these findings, we propose a localization-first framework for cross-modal video reasoning that enables more precise evidence grounding and substantially improves reasoning accuracy. Extensive experiments demonstrate the effectiveness of our approach, establishing new state-of-the-art results on multimodal RAG tasks. Together, the benchmark and framework lay a foundation for advancing research in this emerging area and for building more reliable multimodal reasoning systems.

1 INTRODUCTION

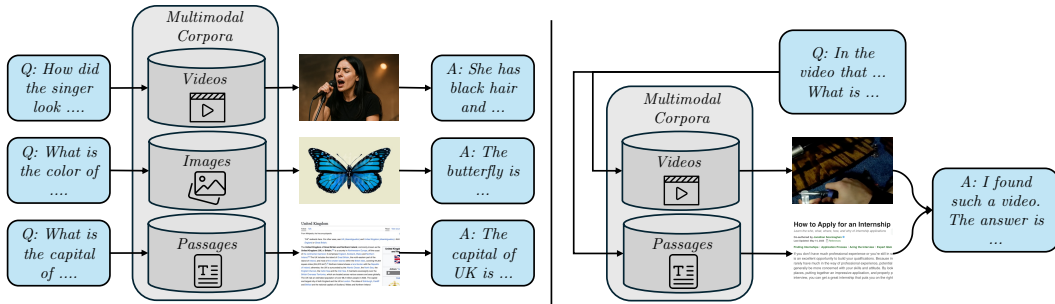


Figure 1: Illustration of multi-hop reasoning in RAG. *Left*: Existing multimodal benchmarks include multiple modalities in their corpora but can often be solved by retrieving and reasoning over a single modality. *Right*: Our proposed benchmark, *EgoFact*, features queries that require integrating evidence across modalities through multi-hop reasoning to arrive at the correct answer.

Large language models (LLMs) have transformed information access and content generation, yet they remain prone to hallucinations and struggle to reason beyond their training data (Zhang et al., 2023; Arora et al., 2023). Retrieval-augmented generation (RAG) addresses these limitations by grounding model outputs in external evidence. While RAG has achieved remarkable progress in text-based domains across diverse applications (Sarathi et al., 2024; Wang et al., 2024; 2025b; Ng et al., 2025; Wiratunga et al., 2024; Wang et al., 2025a), its extension to other modalities—particularly video—remains underexplored. Yet many real-world queries demand precisely this capability: integrating visual observations from video with complementary textual knowledge.

Consider a simple example: a cooking assistant determining “Did I already add salt?” can resolve this by grounding its reasoning purely in the video timeline. By contrast, an egocentric video showing

a person using a specific tool to tighten bolts reveals the tool’s appearance, but answering “*What safety precautions should be followed when using this tool?*” requires consulting external textual sources such as manuals or *WikiHow*. These examples highlight the spectrum of challenges: while some questions depend only on visual understanding, many require multi-hop reasoning that integrates video evidence with external knowledge.

The need for multimodal reasoning arises naturally in everyday contexts. People increasingly turn to instructional videos, short-form content, and augmented reality (AR) interfaces for information. In these settings, an AI system must not only interpret events in the video but also connect them to external knowledge. These trends underscore the importance of multimodal, multi-hop RAG for real-world applications such as education, assistive technologies, and robotics.

Despite growing interest, existing benchmarks only partially capture the challenges outlined above: the need to reason directly over raw video while integrating external textual knowledge. [Luo et al. \(2024\)](#) focus on intra-video reasoning supported by auxiliary signals such as OCR and ASR. [Jeong et al. \(2025\)](#) study retrieval over large video corpora but rely on transcripts that closely mirror video content. [Ren et al. \(2025\)](#) emphasize cross-video reasoning with LongerVideos, yet their benchmark remains unimodal and text-rich. [Yeo et al. \(2025\)](#) propose UniversalRAG with routing across modalities, but their corpora are assembled from unimodal datasets and their video tasks largely reformulated into text. Collectively, these works advance the field but either extend existing resources or treat modalities in isolation. As a result, current benchmarks fall short of capturing the multimodal, multi-hop reasoning required for authentic video understanding.

To address these limitations, we introduce *EgoFact*, a benchmark for evaluating RAG systems on multi-hop reasoning that integrates video with textual knowledge. Built on egocentric video paired with external resources like *WikiHow*, it enables rigorous assessment of how models retrieve, align, and reason across modalities. At its core, *EgoFact* features queries of two types: *single-hop*, answerable directly from video alone, and *two-hop* that link observed events in video with procedural or factual knowledge in text. To capture different reasoning demands, queries span two categories: *local* (single-event object-focused) and *temporal* (event-ordering across activities). Together, these dimensions capture realistic multimodal reasoning scenarios that existing testbeds overlook. Finally, the use of egocentric videos grounds the benchmark in everyday experiences, while *WikiHow* provides procedural, step-by-step knowledge that supports faithful evaluation of cross-modal retrieval and reasoning.

Beyond benchmark construction, evaluating existing VideoRAG methods on *EgoFact* reveals a critical weakness: *retrieval alone is not enough*. Even when the correct clips are retrieved, models often fail to leverage them effectively, producing hallucinations or inconsistent answers. Strikingly, these failures arise even on questions that are trivial for humans given the same clips, underscoring that the real bottleneck lies not in retrieval but in visual grounding. Current systems operate at a coarse granularity, treating entire clips as evidence rather than isolating the precise snippets that contain the answer.

To overcome this deficiency, we propose a *localization-first framework* that identifies and grounds reasoning in the most relevant video segments within retrieved clips. By directing attention to these informative regions, the method reduces spurious generation and improves answer accuracy. For example, rather than processing an entire 30-second cooking clip, the model isolates the 5-second snippet capturing an event (such as when milk is added), enabling more precise and reliable reasoning.

Ultimately, by ensuring that answers are anchored both in what users see and in reliable external sources, *EgoFact* lays the groundwork for building AI systems that can interact more naturally with human experiences. This work makes three key contributions: (1) *EgoFact*, the first benchmark for retrieval-augmented reasoning over egocentric, action-oriented video with both single-hop and two-hop multimodal queries; (2) a *systematic diagnosis* showing that existing video-RAG systems underperform not because of retrieval, but due to weak visual grounding; and (3) a *localization-first method* that grounds reasoning in relevant video snippets, yielding notable improvements.

2 ON MULTI-HOP MULTIMODAL RETRIEVAL

Real-world information-seeking problems are rarely solvable in a single step or from a single source; instead, they demand combining evidence scattered across heterogeneous modalities—text, images,

videos, or tables—through multiple reasoning steps. Consider a scenario where a person is navigating a new city with a wearable camera. To answer the question, “Which building is this, and when was it built?”, a system must connect visual recognition of the building with textual knowledge such as Wikipedia entries, effectively linking evidence across modalities. These challenges highlight two complementary directions for advancing retrieval-augmented generation: the need for *multi-hop reasoning*, where answers are constructed by chaining together multiple pieces of evidence, and the need for handling *rich contexts*, where relevant signals are embedded in long, noisy, and multimodal streams. Below we discuss each of these challenges.

RAG on Multiple Hops. Many real-world queries require multi-hop reasoning, where systems must chain together evidence distributed across different sources. This idea has been extensively explored in text-based settings, with benchmarks such as *HotpotQA* (Yang et al., 2018) and *MuSiQue* (Trivedi et al., 2022) explicitly designed to evaluate models’ ability to retrieve and integrate information from multiple documents. However, to the best of our knowledge, these efforts remain largely confined to *textual corpora*, leaving open the challenge of extending multi-hop reasoning to multimodal domains.

RAG on Rich Contexts. Another fundamental challenge for RAG arises in *rich contexts*, where evidence is embedded beyond textual corpora. Research has explored RAG in various modalities, including images (Hu et al., 2025), videos (Ren et al., 2025; Luo et al., 2024; Jeong et al., 2025), tabular data (Joshi et al., 2024), etc. These efforts typically focus on single-hop retrieval, where the model retrieves a single piece of evidence from a specific modality to answer a query.

To advance toward real-world applications, we introduce *EgoFact*, a dataset designed to evaluate RAG systems on their ability to perform both single and multi-hop reasoning across multimodal corpora. By integrating egocentric videos with complementary textual corpora, *EgoFact* presents unique challenges that require models to understand context, actions, and interactions from both visual and textual cues.

2.1 VIDEO CORPORA CONSTRUCTION

Egocentric recordings are particularly valuable. Unlike third-person instructional or documentary videos used in prior VideoRAG benchmarks, egocentric videos capture fine-grained, first-person perspectives of human activities. This viewpoint naturally reflects situated, real-world contexts and foregrounds objects and actions most relevant to the wearer, making it especially suitable for studying grounded multimodal reasoning. We base our selection on the *Ego4D* dataset (Grauman et al., 2022), focusing specifically on the subset annotated in *Ego4DSounds* (Chen et al., 2024), which offers reliable event-level labels to help curate videos with grounded actions. Importantly, these annotations are used only for dataset curation and are not included as part of the benchmark itself. To manage dataset scale, we further restrict the pool to videos from a set of *pre-defined scenarios* directly relevant to common real-world applications (e.g., daily household tasks, cooking, or tool usage). This selection ensures that the resulting dataset captures both the multimodal reasoning challenges and the domain constraints central to day-to-day use cases.

Each selected video is segmented into 30-second clips. After applying filtering and selection (details in C), we obtain a curated set of 127 representative clips. We then generate a caption for each clip using *GPT-4o-mini* to support later query construction. These clips constitute the backbone of the video corpora in *EgoFact*, providing a controlled yet realistic setting for evaluating cross-modal retrieval and reasoning.

2.2 QUERIES DESIGN

To evaluate multimodal reasoning under controlled conditions, *EgoFact* queries are organized along two complementary axes: the number of hops (*1-hop* vs. *2-hop*) and the grounding type (*local* vs. *temporal*). Their combination yields four query families, posed as multiple-choice selection tasks. The overall generation workflow is illustrated in fig. 2, with concrete examples provided in table 4.

Table 1: Query statistics of *EgoFact*, spanning hop count (1-hop vs. 2-hop) and grounding type (Local vs. Temporal). Each cell corresponds to one query family; in total the dataset contains 396 queries.

Grounding	1-hop	2-hop
Local	113	73
Temporal	119	91

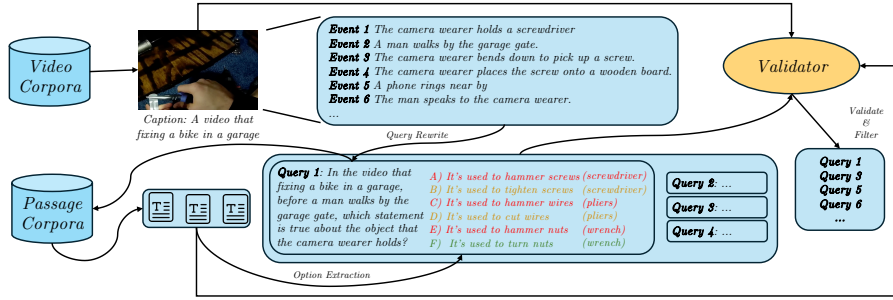


Figure 2: Overall design of the query construction in *EgoFact*. Each query follows a structured pattern: A, C, and E are counterfactuals (red), with A and C tied to distractor objects. B and D are factual statements about these distractors (orange). Only F corresponds to the correct object and fact (green). The events list (1–6) is derived from Ego4DSounds annotations, used solely for dataset curation and not included in the benchmark.

1-hop vs. 2-hop queries. In 1-hop queries, the answer is derived directly from the video by recognizing an object involved in a specific event. In 2-hop queries, the model must identify the relevant object in the video and also retrieve a factual attribute of that object from an external text passage. To structure this, multiple-choice options are grouped into three object–passage pairs: (A,B), (C,D), and (E,F). Each pair corresponds to a distinct object and its associated passage. Within each pair, option one (e.g., A, C, E) states a counter-factual claim about the object, while option two (e.g., B, D, F) states the factual one. The first two pairs (A,B) and (C,D) are distractors referring to other objects, while only the last pair (E,F) corresponds to the correct object shown in the video. Within that final pair, E gives a false property and F gives the true one. Here, each true or counter-factual option is generated by an LLM conditioned on the passage from WikiHow. To construct the textual side of the corpora, we aggregate all passages that appear in these queries, ensuring a consistent pool of evidence for retrieval. This design ensures that solving 2-hop queries requires not just object recognition but also cross-modal factual reasoning.

Local vs. temporal queries. Local queries are tied to a single event within a clip, asking about the object directly involved in that event. Temporal queries, in contrast, exploit event ordering: one event serves as a temporal anchor while another provides the target of the question. Note that temporal queries in *EgoFact* always use two consecutive events to form the anchor–target pair. For example, a temporal query may ask, “Before a woman plays with a baby, what object does she place onto the neck?”—requiring the model to resolve sequence as well as grounding.

Reliability check of queries. To ensure the reliability of the benchmark, each generated query is validated with three complementary conditions. 1) the query must be *valid*: when the ground-truth event information is provided, an LLM should be able to produce the correct answer. 2) it must be *non-trivial*: given only the query without the supporting event, the LLM should *not* be able to guess the correct answer based on common-sense knowledge. 3) the query must be *non-ambiguous*: the same query should not admit multiple plausible answers across the target document and other contextually similar documents. Queries that fail any of these checks are discarded. This triple criterion ensures that each query in *EgoFact* is answerable only when grounded in the correct evidence, remains challenging without it, and avoids ambiguity across related contexts. The detailed query construction pipeline is presented in fig. 2, and further described in appendix C.2.

Finally, we generate a total of 396 queries spanning the four query families, as shown in table 1.

3 EMPIRICAL INSIGHTS ON *EgoFact*

In this section, we assess the effectiveness of existing RAG systems on *EgoFact* by adapting a state-of-the-art video-based RAG framework and evaluating its performance under our benchmark design. Our purpose is to understand where current methods succeed and where they fail, thereby identifying the key bottlenecks for multimodal, multi-hop reasoning. To this end, we analyze both retrieval performance and how well retrieved evidence is understood.

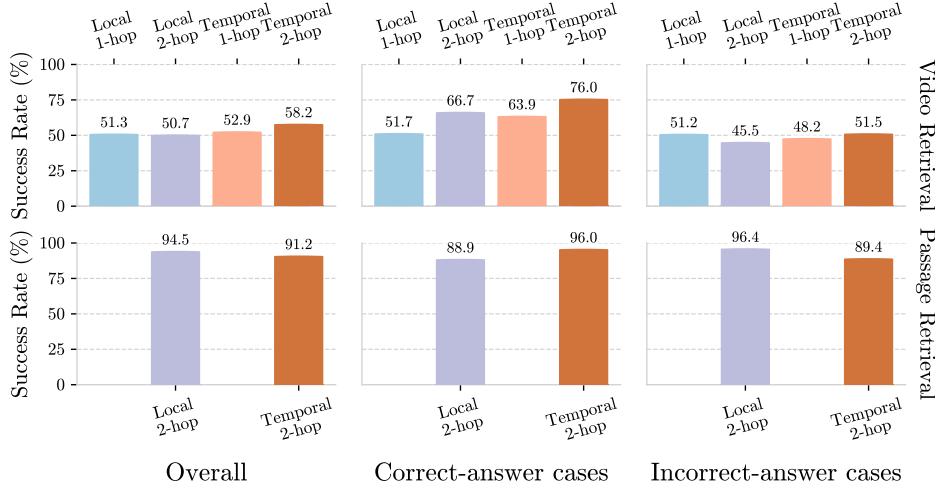


Figure 3: Retrieval performance across video and passage tasks. The first row shows *Video Retrieval* results, and the second row shows *Passage Retrieval* results, aligned with their corresponding 2-hop categories. We observe that (1) passage retrieval maintains a consistently high success rate, (2) video retrieval success rate is generally in the 50%–60% range, and (3) even in the *Incorrect-answer cases*, video retrieval still achieves around 50% success rate, suggesting that the retrieved videos are often reasonably relevant.

3.1 EVALUATION SETUP

For evaluation, we focus on VideoRAG (Ren et al., 2025), as it was found to perform better than other baselines on our benchmark. We begin by briefly describing the original framework.

Framework of Ren et al. (2025). The original framework is designed for single-hop, video-based RAG tasks. It first uses *MiniCPM* to generate video descriptions, which are stored as a video description corpus. Given a query, the system retrieves related descriptions from this corpus, then performs video retrieval through two branches: (1) a visual embedding-based branch using *ImageBind* (Girdhar et al., 2023), and (2) an entity-based branch that extracts entities from the video descriptions with *MiniCPM* and retrieves videos by entity matching. The retrieved video segments are then captioned with query-guided video captioning, again using *MiniCPM*. Finally, the pre-generated video descriptions and the new captions are concatenated with the query and passed to *MiniCPM* to produce the final answer.

We adapted this framework to fit *EgoFact*. Specifically: (1) we replace the original *MiniCPM* video captioning backbone with *GPT-4o-mini* (OpenAI et al., 2024) for improved query-guided video captioning; (2) we remove the pre-captioned video descriptions and disable the entity-based video retrieval branch, as both components were found to be ineffective for *EgoFact*; (3) given a query that requires information from passage corpora, besides retrieving video segments, we perform an additional passage retrieval step from the text corpora (as described in section 2), where we retrieve the top-6 passages per query; and (4) the final answer is generated from the concatenated video and text evidence. This workflow is illustrated in the upper part of fig. 7, while hyperparameters and prompts are provided in appendix D.2.

3.2 RETRIEVAL PERFORMANCE

We examine retrieval performance, decomposed into video retrieval and passage retrieval. For both tasks, we measure *success rate* as the proportion of queries for which the correct item is retrieved. We further categorize results into three groups: *Overall* (all queries), *Correct-answer cases* (queries where the final answer is correct), and *Incorrect-answer cases* (queries where the final answer is incorrect). This breakdown helps identify whether retrieval errors are driving overall failures. The results are summarized in fig. 3. We find some key observations as follows:

Passage retrieval is reliable. Across all 2-hop categories, passage retrieval maintains consistently high success rate (generally above 90%), with little variation between the *Correct-answer* and

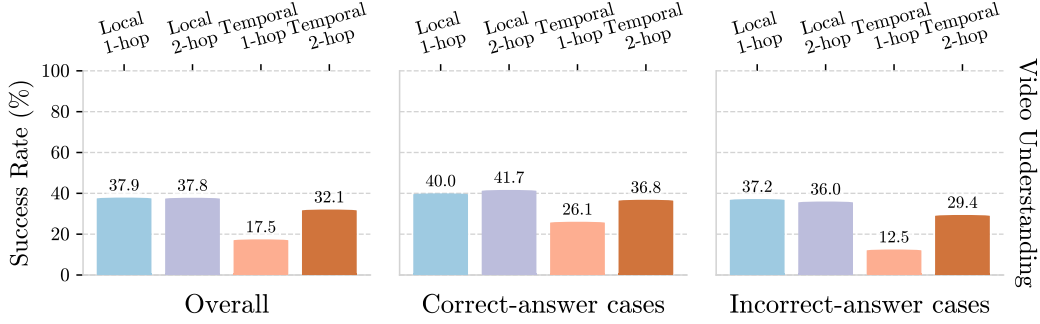


Figure 4: Video understanding performance in successful retrieval cases, with three columns aligned to the setting in fig. 3. Each bar indicates the proportion of queries for which an LLM (GPT-4o-mini in this case), given the query and the ground-truth video description, can correctly identify the key information required to answer. Overall, the proportions remain low (generally below 50%), with performance especially poor in the *Incorrect-answer cases*, where retrieved videos rarely provide sufficient information. This highlights video understanding as a critical area in need of improvement.

Incorrect-answer cases. This indicates that the passage retriever is generally effective, and retrieval errors might not be major contributor to possible failures for *Incorrect-answer cases*.

Video retrieval is challenging but not the main bottleneck. Video retrieval success rate generally falls in the 50%–60% range, indicating that identifying the correct video clip remains somewhat challenging. However, even in the *Incorrect-answer cases*, video retrieval still achieves around 50% accuracy, suggesting that at least half of the failed queries still retrieve reasonably relevant video clips. These results indicate that retrieval alone does not fully explain the failures observed.

3.3 ANALYSIS OF EVIDENCE UNDERSTANDING

We next analyze how well the model understands retrieved evidence, both video and text. The goal of this evaluation is to probe not only whether retrieval succeeds, but also whether the model can effectively interpret the retrieved material to answer the queries. Here we only consider queries where retrieval is successful. Since no ground-truth metric exists, we rely on proxies. For video understanding, we use an LLM (*GPT-4o-mini* here) as a judge: given the query and the query-conditioned caption of the ground-truth video clip, the judge is prompted to determine whether the key information needed to answer is present. For passage understanding, we directly examine the model’s answer distribution across options *A–F* (Section 2), where *A*, *C*, and *E* denote counterfactuals and *B*, *D*, and *F* denote factual statements. Therefore, a model that understands the passage should consistently favor *B*, *D*, and *F*. The results are summarized in fig. 4 and fig. 5 and we present the detailed prompts in appendix D.1. We can summarize some key observations as follows:

Passage understanding is proficient. When the correct passage is retrieved, the model consistently favors *B*, *D*, and *F* over counterfactuals in about 80% of the cases. This suggests that passage understanding is reliable, though still leaving room for improvement.

Video understanding is the main bottleneck. Overall, video understanding performance remains low (generally below 50%), with especially poor results in the *Incorrect-answer cases*, where retrieved videos rarely provide sufficient information. This highlights video understanding as the critical limitation in current systems.

In short, our analysis reveals that both passage and video retrieval are reasonably effective, and that passage understanding is proficient. *The main bottleneck lies in video understanding, where models struggle to extract the key information needed to answer queries, even when the correct video is retrieved.* This points to the need for improved visual grounding and reasoning ability in *EgoFact*.

4 TOWARDS BETTER VIDEO UNDERSTANDING

How can we improve video grounding and understanding for RAG?

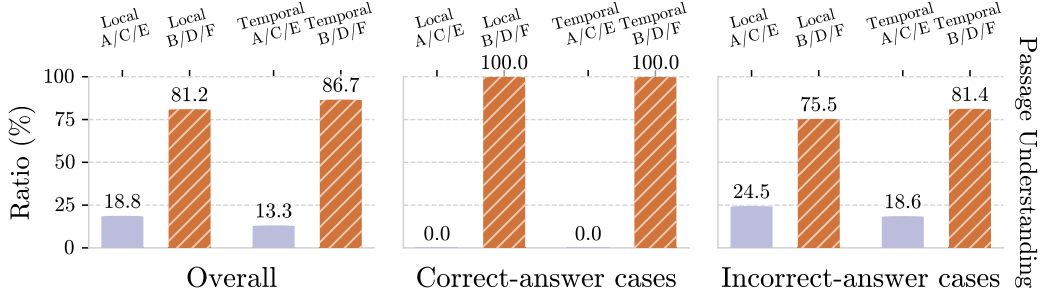


Figure 5: Textual understanding performance under successful retrieval. The three columns follow the same setting as in fig. 3. Each bar shows the distribution of answers across options A–F, where A, C, and E denote counterfactual statements and B, D, and F denote factual ones. When the correct passage is retrieved, the model consistently favors B, D, and F, suggesting that passage understanding is reliable. Note that in *Correct-answer cases*, the answer is always F by design.

This question may seem daunting at first glance, as video understanding remains a longstanding open challenge in AI. Yet our analysis offers a different perspective: by examining where existing RAG systems succeed, we identify a promising path forward.

Specifically, effective video understanding requires a model to *grasp the query*, *ground it in video content*, and *generate an textual response* (e.g., *answer or caption*); failure in any component leads to errors. Our retrieval analysis reveals a crucial insight: current models can retrieve relevant video clips reliably, even when the query depends on short, localized evidence within the clip. This indicates that existing retrieval mechanisms already handle the first two components—*grasping the query* and *grounding it in video content*—reasonably well.

These findings suggest a fresh perspective: rather than requiring models to ground and generate jointly, we can pre-localize the evidence in video and allow the model to focus exclusively on generation.

Table 2: Video understanding performance with and without localization under the same setting as fig. 4.

Method	All	Correct-answer cases	Incorrect-answer cases
VideoRAG (Ren)	30.33	41.67	28.17
+ Localization	40.95	44.05	38.89

this design does not require retraining the backbone model; rather, it integrates seamlessly as a plug-and-play module in the RAG pipeline. As we demonstrate in the following sections, this simple adjustment consistently improves video understanding and leads to more reliable multimodal reasoning.

4.1 EXPERIMENT

Baselines. We compare our approach against several representative RAG frameworks for multimodal reasoning. (1) *GraphRAG* (Edge et al., 2025): a traditional graph-based RAG model that constructs local–global graphs over the corpus. For adaptation to our setting, we follow prior practice and use *GPT-4o-mini* to generate a textual description for each video, which is then treated as a node in the graph. (2) *VideoRAG* (Jeong et al., 2025): a video-centric RAG model that retrieves relevant video clips given a query. To align with our benchmark, we also extend it with an additional passage-retrieval branch so that both modalities are considered. (3) *VideoRAG (Ren)* (Ren et al., 2025): the state-of-the-art video-based RAG framework that serves as the main baseline in our analysis. (4) *VideoRAG (Ren) + Localization*: our variant that integrates the proposed localization step into the *VideoRAG (Ren)* model, enabling the system to focus on the most query-relevant video segments before answer generation. We detail implementation and hyperparameters in appendix D.2.

Metrics. We evaluate all methods on *EgoFact* using accuracy as the main metric, defined as the proportion of queries for which the model selects the correct answer option. We report results separately for each of the four query families (Local 1-hop, Local 2-hop, Temporal 1-hop, Temporal 2-hop) as well as overall accuracy across all queries.

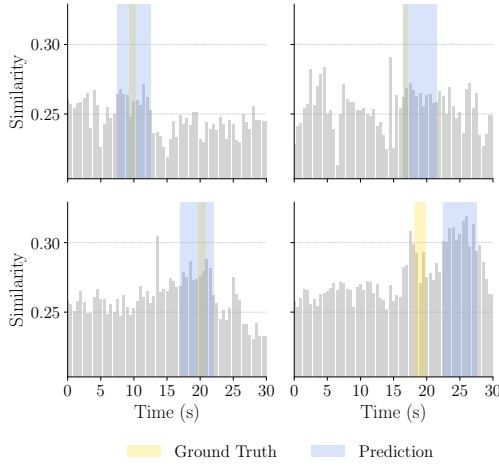


Figure 6: Demonstration of the localization process. Given a query and a retrieved video, we sample frames at regular intervals, compute query–video relevance, and select the most relevant contiguous segment as the localized snippet for reasoning. Four examples from *EgoFact* are shown: the first three are successes, while the last is a failure.

Methodology. For localization, we first sample each video at 2 frames per second. Given a target segment length of 5 seconds (about 10 frames), we slide a window of this length across the entire sequence of sampled frames. For every possible window, we compute the average query–video similarity using *MetaCLIP* (Xu et al., 2024), and select the window with the highest mean similarity as the localized segment. We then fix the downstream frame budget at 15 frames for all video-based methods, including both baselines and our approach. All the other setting remain the same with section 3. We illustrate this process in fig. 6, and provide the detailed prompts, workflow, and hyperparameters in appendix D.2.

Table 3: Comparison of RAG variants on *EgoFact*. Columns correspond to query families (1-hop vs. 2-hop, Local vs. Temporal), with the last column showing the overall average. VideoRAG (Ren) with localization consistently outperforms its counterpart, especially on 1-hop queries. Best results are in bold.

Method	Local 1-hop	Local 2-hop	Temporal 1-hop	Temporal 2-hop	Overall
GraphRAG	15.04	8.22	21.01	4.40	13.13
VideoRAG (Jeong)	10.62	12.33	29.41	9.89	16.41
VideoRAG (Ren)	25.66	24.66	30.25	27.47	27.27
+ Localization	31.86	27.40	43.70	27.47	33.59

ral scope for video reasoning. Although 2-hop queries see smaller improvements, the consistent gains confirm localization as an effective strategy for enhancing multimodal reasoning under *EgoFact*. Consistent with this, the analysis in table 2 shows that localization makes the LLM more likely to identify the key information needed to answer.

4.2 ANALYSIS

We analyze our proposed localization method by asking two natural questions. (1) *how does localization perform under different temporal ranges?*, and (2) *to what extent does localization improve over naive alternatives?* To address these questions, we design two experiments. In the first, we compare localization against a *Video Split* baseline that divides each retrieved video into fixed-length, non-overlapping segments (e.g., 5 seconds) and uses them as retrieval units. In the

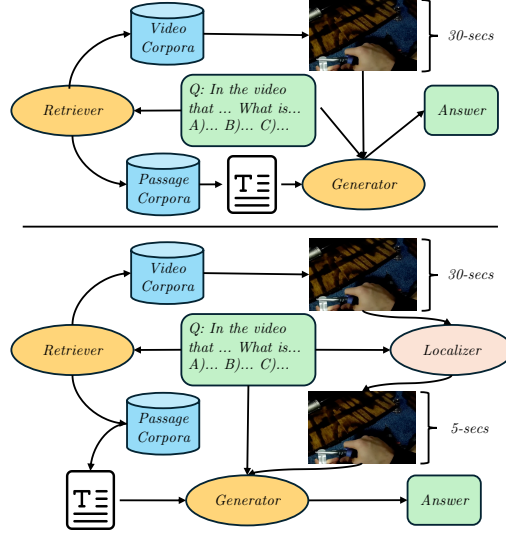


Figure 7: Comparison of workflows. *Top*: baseline pipeline of (Ren et al., 2025), where candidate video clips and passages are retrieved and fed directly into reasoning without localization. *Bottom*: our approach adds a localization module to identify the most relevant segment within each retrieved video before answer generation.

Results. The results are summarized in table 3. Incorporating localization into *VideoRAG* (Ren) consistently improves performance across three query families—*Local 1-hop*, *Local 2-hop*, and *Temporal 1-hop*—as well as the overall average. The gains are most pronounced in 1-hop settings, where queries rely more directly on video evidence. This aligns with our earlier finding that textual passages are generally easier to interpret, while localization sharpens the tempo-

second, we vary the window size used in localization to examine how temporal granularity influences performance.

Findings. The results are summarized in fig. 8. On the one hand, *Localization* consistently outperforms naive splitting, confirming that scope preservation and relevance scoring are crucial for effective video understanding. On the other hand, localization performance follows an inverted U-shape across window sizes: very short windows risk missing essential context, very long windows dilute relevance with noise, while intermediate ranges strike the best balance. Together, these results highlight both the effectiveness of localization and the importance of selecting appropriate temporal granularity.

Although both *Video Split* and *Localization* enable fine-grained retrieval, they differ in how discriminative the search space becomes. When a video is split into many short chunks, similar-looking fragments from different videos may appear nearly indistinguishable, making retrieval more noisy. In contrast, *Localization* first identifies a query-relevant video clip at the global level and then refines the evidence within that scope, which reduces confusion across videos and yields clearer grounding.

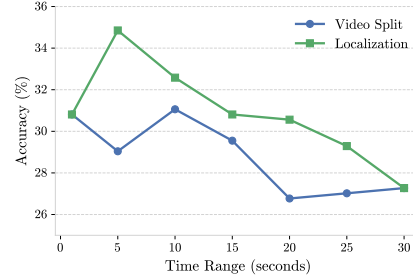


Figure 8: Comparison of *Video Split* and *Localization*. Localization outperforms naive segmentation and shows an inverted U-shaped performance curve, peaking at mid-sized windows.

5 RELATED WORK

Multimodal RAG Approaches. Prior work has explored RAG for multimodal settings, especially video, but under settings different from ours. Jeong et al. (2025) introduces video with adaptive frame selection and corpus-level retrieval over *HowTo100M* (Miech et al., 2019), but both retrieval and generation rely on auxiliary text like subtitles or transcripts. Ren et al. (2025) design the *LongerVideos* benchmark for lecture, documentary, and entertainment series, emphasizing cross-video reasoning; their framework integrates graph-based textual knowledge grounding with multimodal embeddings, operating mainly in text-rich contexts. Luo et al. (2024) focus on intra-video only retrieval: from a single target video, they extract OCR, ASR, and object-detection metadata, filtering them for relevance before feeding into an LVLM. *UniversalRAG* (Yeo et al., 2025) extends RAG to text, image, and video corpora with a modality- and granularity-aware router; however, its testbed merely concatenates existing datasets and can be addressed using simple 1-hop reasoning.

Challenges for Prior Approaches on *EgoFact*. *EgoFact* removes these scaffolds: our videos have no subtitles, ASR, or OCR, and complementary passages do not mirror video content but provide external knowledge for the second reasoning hop. This design enforces a stricter setting, where models must effectively retrieve from both the video corpus and the complementary text corpus without relying on subtitles or transcriptions. Methods optimized for video-text co-availability, or for cross-video aggregation, underperform here because (i) their retrieval modules depend on auxiliary text to bridge video and query, (ii) their generation modules assume the availability of text-aligned frames, and (iii) they are not designed for fine-grained intra-video grounding in the absence of textual aids. *EgoFact* thus complements prior benchmarks by explicitly highlighting visual grounding as the critical challenge and testing whether systems can succeed in multi-hop multimodal reasoning without relying on text that merely shadows video content.

6 CONCLUSION

As multimodal AI continues to advance, robust benchmarks are crucial for guiding progress. In this work, we introduced *EgoFact*, the first testbed designed to intertwine visual and textual knowledge, and provided a systematic diagnosis of existing RAG systems. We envision *EgoFact* as a foundation for developing and evaluating more capable multimodal RAG approaches. Looking ahead, future directions include incorporating richer modalities, supporting deeper reasoning chains, and enabling more interactive settings—paving the way for AI systems that can better understand and assist in complex real-world scenarios.

REFERENCES

- Kushal Arora, Layla El Asri, Hareesh Bahuleyan, and Jackie Chi Kit Cheung. Why exposure bias matters: An imitation learning perspective of error accumulation in language generation, 2023. URL <https://arxiv.org/abs/2204.01171>.
- Changan Chen, Puyuan Peng, Ami Baid, Zihui Xue, Wei-Ning Hsu, David Harwath, and Kristen Grauman. Action2sound: Ambient-aware generation of action sounds from egocentric videos, 2024. URL <https://arxiv.org/abs/2406.09272>.
- Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitanaky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization, 2025. URL <https://arxiv.org/abs/2404.16130>.
- Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all, 2023. URL <https://arxiv.org/abs/2305.05665>.
- Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrahm Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanov, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3,000 hours of egocentric video, 2022. URL <https://arxiv.org/abs/2110.07058>.
- Wenbo Hu, Jia-Chen Gu, Zi-Yi Dou, Mohsen Fayyaz, Pan Lu, Kai-Wei Chang, and Nanyun Peng. Mrag-bench: Vision-centric evaluation for retrieval-augmented multimodal models, 2025. URL <https://arxiv.org/abs/2410.08182>.
- Soyeong Jeong, Kangsan Kim, Jinheon Baek, and Sung Ju Hwang. Videorag: Retrieval-augmented generation over video corpus, 2025. URL <https://arxiv.org/abs/2501.05874>.
- Pankaj Joshi, Aditya Gupta, Pankaj Kumar, and Manas Sisodia. Robust multi model rag pipeline for documents containing text, table & images. In *2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, pp. 993–999, 2024. doi: 10.1109/ICAAIC60222.2024.10574972.
- Yongdong Luo, Xiwu Zheng, Xiao Yang, Guilin Li, Haojia Lin, Jinfa Huang, Jiayi Ji, Fei Chao, Jiebo Luo, and Rongrong Ji. Video-rag: Visually-aligned retrieval-augmented long video comprehension, 2024. URL <https://arxiv.org/abs/2411.13093>.
- Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2630–2640, 2019.
- Karen Ka Yan Ng, Izuki Matsuba, and Peter Chengming Zhang. Rag in health care: A novel framework for improving communication and decision-making by addressing llm limitations. *NEJM AI*, 2(1):AIra2400380, 2025. doi: 10.1056/AIra2400380. URL <https://ai.nejm.org/doi/full/10.1056/AIra2400380>.

- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL <https://arxiv.org/abs/2303.08774>.
- Xubin Ren, Lingrui Xu, Long Xia, Shuaiqiang Wang, Dawei Yin, and Chao Huang. Videorag: Retrieval-augmented generation with extreme long-context videos, 2025. URL <https://arxiv.org/abs/2502.01549>.
- Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. Raptor: Recursive abstractive processing for tree-organized retrieval, 2024. URL <https://arxiv.org/abs/2401.18059>.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition, 2022. URL <https://arxiv.org/abs/2108.00573>.

- Jingru Wang, Wen Ding, and Xiaotong Zhu. Financial analysis: Intelligent financial data analysis system based on llm-rag, 2025a. URL <https://arxiv.org/abs/2504.06279>.
- Mingqiu Wang, Izhak Shafran, Hagen Soltau, Wei Han, Yuan Cao, Dian Yu, and Laurent El Shafey. Retrieval augmented end-to-end spoken dialog models, 2024. URL <https://arxiv.org/abs/2402.01828>.
- Zora Zhiruo Wang, Akari Asai, Xinyan Velocity Yu, Frank F. Xu, Yiqing Xie, Graham Neubig, and Daniel Fried. Coderag-bench: Can retrieval augment code generation?, 2025b. URL <https://arxiv.org/abs/2406.14497>.
- Nirmalie Wiratunga, Ramitha Abeyratne, Lasal Jayawardena, Kyle Martin, Stewart Massie, Ikechukwu Nkisi-Orji, Ruvan Weerasinghe, Anne Liret, and Bruno Fleisch. Cbr-rag: Case-based reasoning for retrieval augmented generation in llms for legal question answering, 2024. URL <https://arxiv.org/abs/2404.04302>.
- Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying clip data, 2024. URL <https://arxiv.org/abs/2309.16671>.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering, 2018. URL <https://arxiv.org/abs/1809.09600>.
- Woongyeong Yeo, Kangsan Kim, Soyeong Jeong, Jinheon Baek, and Sung Ju Hwang. Universalrag: Retrieval-augmented generation over corpora of diverse modalities and granularities, 2025. URL <https://arxiv.org/abs/2504.20734>.
- Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. How language model hallucinations can snowball, 2023. URL <https://arxiv.org/abs/2305.13534>.

A THE USAGE OF LLMs

Besides the query construction process mentioned in section 2, the LLMs are used solely for rewriting purposes to ensure clarity and coherence of the text. They have not been employed for generating any original content or research ideas. All technical details, methodologies, and findings are the result of our own research and analysis.

B COMPUTATIONAL RESOURCES

All experiments were conducted on a single NVIDIA A100 GPU with 80GB memory.

C *EgoFact* DETAILS

C.1 VIDEO CORPORA CONSTRUCTION DETAILS

To constrain the volume of video data while ensuring relevance to everyday activities, we use videos from the *Ego4DSounds* subset (Chen et al., 2024), restricted to the following annotated scenarios:

- *Bike mechanic*
- *Getting car fixed*
- *Car mechanic*
- *Car - commuting, road trip*
- *Fixing something in the home*

Each selected video is further segmented into uniform 30-second windows, which serve as the atomic units of video corpora. This procedure yields an initial collection of 2,124 video segments. Then, we screenshot every 3-second to get 10 screenshots for each segment. For each screenshot, we use *GPT-4o-mini* with the prompt below to generate a concise and detailed narration.

```

**ROLE**
You are a scene understanding expert.

**INPUT**
•Scenario: a list of short strings identifying the video context.
•Screenshot: the image itself, provided as an attached image.

**OUTPUT**
Output a single paragraph that describes the scenario in detail.
```

Then we concatenate the narrations of all screenshots within a segment to form a comprehensive video description, then concatenate them with *Ego4DSounds* labels and sort to get a video timeline. The timeline is used for two purposes: (1) to generate keywords for video embedding and clustering, and (2) to generate narrations for query construction. To generate narrations, we use *OpenAI o3* with the following prompt:

```

**TASK**
Generate a narration for a video timeline.

**INPUT**
A video timeline in the format:
Scenarios: <comma-separated list of contexts>
```

```

702 Timeline a list of entries, each line in the format:
703   [<timestamp>] (<type>) <text>
704
705 Where:
706 - `<timestamp>` is the time in seconds, formatted as HH:MM:SS.ss
707 - `<type>` is either "Sound" or "Scene"
708   - Scene - a detailed visual description of screenshot at that second
709   - Sound - a label indicating a sound in the video at that second
710 - `<text>` is the description of the sound or scene
711
712 **OUTPUT**
713 Return two narrations:
714 1. A concise narration that summarizes the video content in a single
715    sentence, starts with "the video that".
716 2. A detailed narration that describes the video content in a few
717    sentences, capturing the main events, actions, and objects, starts
718    with "the video that".
719 Return only the parsable JSON object with two keys "concise" and
720    "detailed", each containing the respective narration, without any
721    additional text.
722
723 **EXAMPLE**
724 INPUT:
725 Scenarios: Bike mechanic, Fix something in the home
726 Timeline:
727 [00:01:30.00] (Scene) The scene depicts a bike mechanic working in a
728    room, where various tools are scattered across a wooden workbench.
729 [00:01:32.82] (Sound) #C C picks up a kick scooter from the floor.
730 [00:01:35.33] (Sound) #O A man walks in the door.
731 ...
732
733 OUTPUT:
734 {
735   "concise": "the video that a bike mechanic works in a room with
736     scattered tools, while a person picks up a kick scooter and
737     another person enters.",
738   "detailed": "the video that shows a bike mechanic in a room filled
739     with various tools on a wooden workbench. At one point, the
740     camera ..."
741 }
742
743 To generate keywords, we also use OpenAI o3 with the following prompt:
744
745 **TASK**
746 Generate a list of keywords from the provided video timeline.
747
748 **INPUT**
749 A video timeline in the format:
750 Scenarios: <comma-separated list of contexts>
751 Timeline a list of entries, each line in the format:
752   [<timestamp>] (<type>) <text>
753
754 Where:
755 - `<timestamp>` is the time in seconds, formatted as HH:MM:SS.ss
756 - `<type>` is either "Sound" or "Scene"
757   - Scene - a detailed visual description of screenshot at that second
758   - Sound - a label indicating a sound in the video at that second
759 - `<text>` is the description of the sound or scene
760
761 **OUTPUT**
762 Return a JSON array of keywords that capture the main objects, actions,
763    contexts, etc., each paired with a relevance score from 1 to 10. The
764    keywords should be as detailed as possible.
765 Return only the parsable JSON array without any additional text.

```

```

756 **EXAMPLE**
757 INPUT:
758 Scenarios: Bike mechanic, Fix something in the home
759 Timeline:
760 [00:01:30.00] (Scene) The scene depicts a bike mechanic working in a
761     room, where various tools are scattered across a wooden workbench.
762 [00:01:32.82] (Sound) #C C picks up a kick scooter from the floor.
763 [00:01:35.33] (Sound) #O A man walks in the door.
764 ...
765 OUTPUT:
766 {
767     "bike mechanic": 10,
768     "kick scooter": 8,
769     "tools": 6,
770     ...
771 }

```

After obtaining the narrations and keywords for each segment, we embed and weight the keywords (weight is the relevance score generated by previous step) to get video embeddings using *OpenAI-Text-Embedding-3-small*. Then we apply k -nearest neighbor (kNN) clustering in the embedding space to identify representative samples and eliminate highly redundant segments. The hyperparameters for clustering are set as: number of clusters $k = 0.2 * 2124$. After filtering, we obtain a curated set of 424 representative segments, each 30 seconds in length. Then we filter again to remove segments that have too few events (less than 2) resulting in a final set of 127 segments. These segments constitute the backbone of the video corpora in *EgoFact*, providing a controlled yet realistic setting for evaluating cross-modal retrieval and reasoning.

C.2 QUERY CONSTRUCTION DETAILS

After obtaining the video corpora, we generate queries based on the event annotations provided in *Ego4DSounds*. Our query generation pipeline consists of several stages:

C.2.1 EVENT STRUCTURE EXTRACTION

We first convert the raw event annotations into a structured format using *OpenAI o3*. Each event is represented as:

- **rewritten**: the human-readable rewritten event text
- **agent**: the person performing the action
- **action**: the verb describing the action
- **patient**: the object involved in the action
- **patient_canonical**: the canonical form of the patient for retrieval
- **patient_location**: the location of the patient
- **patient_destination**: the destination of the patient
- **instrument**: any instrument used in the action
- **event_location**: the location where the event takes place

For example, the event "#C C wearer picks the cellphone from the table" is converted to:

- **rewritten**: the camera wearer picks up the cellphone from the table
- **agent**: camera wearer
- **action**: pick
- **patient**: cellphone

- **patient_canonical:** cellphone
- **patient_location:** table
- **patient_destination:** null
- **instrument:** null
- **event_location:** null

C.2.2 QUERY TEMPLATE CONSTRUCTION

For *local 1-hop queries*, we construct questions asking about the interacting object in a specific video context: "In [video description], what is the object that [agent] [action] [location context]?"

For *temporal 1-hop queries*, we identify event pairs within the same video segment occurring within 4 seconds of each other, and construct questions with temporal relationships: "In [video description]. Before/After [reference event], what is the object that [target agent] [target action] [target location context]?"

C.2.3 DEDUPLICATION AND DISAMBIGUATION

We apply deduplication to remove semantically similar queries by checking for clips with similar visual indicators (similarity > 0.7) that contain events with similar action descriptions (similarity > 0.8) but different objects (similarity < 0.9). This ensures each query has a unique, unambiguous answer that cannot be confused with events in visually similar clips.

C.2.4 DISTRACTOR GENERATION AND QUALITY CONTROL

For each query, we use *OpenAI o3* to generate 5 distractor options plus the correct answer with the following criteria:

```

**TASK**
Given a question and answer pair for video clip understanding, generate
5 wrong
distractors and score the overall question quality from 0-10 based on:
- Reasonable distractors in context
- Visually distinguishable from correct answer
- Not synonyms or near-equivalents
- Cannot be answered without watching the video

**OUTPUT**
JSON object with "Options" (6 total), "Reasoning", and "Score"

```

We filter out queries with scores below 7 and further remove queries that can be answered by *OpenAI o3* without video context, ensuring all queries require visual understanding.

C.2.5 MULTI-HOP QUERY CONSTRUCTION

For *2-hop queries*, we extend *1-hop queries* by replacing the answer with factual statements about the answer. The process involves:

1. Passage Retrieval: For each answer option, we retrieve semantically related WikiHow passages using *OpenAI-Text-Embedding-3-small* with similarity above 0.5.

2. Statement Generation: For each option-passage pair, we generate 4 statements (3 counterfactual, 1 factual) using *OpenAI o3*:

```

**TASK**
Given a term and passage, generate exactly four statements:
1-3: Plausible but incorrect statements
4: True fact from the passage
Requirements:
- Not commonsense knowledge

```

- Use "it"/"this" to refer to term
- Self-contained sentences
- No direct passage references

```

**OUTPUT**
JSON array with exactly four strings

```

3. Validity Verification. We test each statement set with *OpenAI o3* both with and without the reference passage:

```

**TASK**
Given a multiple-choice question with options A-E, determine the correct
answer.

```

```

**INPUT**
Passage: [reference passage or empty]
Question: [multiple-choice question]

```

```
**OUTPUT**
Single letter A-E indicating correct answer
```

We ensure *validity* (correct answer identifiable with passage) and *non-triviality* (correct answer not identifiable without passage).

4. Final Assembly: We select 2 factual statements and 4 counterfactual statements from verified option-passage pairs, constructing the final 6-option multiple choice question with exactly one correct factual statement, prefixed with contextual information: "When [doing activity]. [Statement about object]".

C.3 EXAMPLE QUERIES

We provide examples of the four query types in *EgoFact* in table 4.

C.4 FULL QUERY LIST

The complete list of queries in *EgoFact* is included here for later reference.

[illegible]

Query Type	Query (Example)	Options
Local 1-hop	In a video that shows cleaning and repairing equipment on a cluttered workbench, what is the object that the camera wearer plugs into the socket?	A) power drill B) soldering iron C) desk lamp D) hot glue gun E) vacuum cleaner F) cable
Local 2-hop	In a video that shows cleaning and repairing equipment on a cluttered workbench, which statement is true about the object that the camera wearer plugs into the socket?	A) When cleaning your room as teens. Professional cleaning routines advise operating it before picking up visible rubbish... (<i>false statement about vacuum cleaner</i>) B) When cleaning your room as teens. When refreshing a bedroom, it should first be run over the floor and then moved up... (<i>true statement about vacuum cleaner</i>) C) When making a paper christmas tree. It is generally avoided for embellishing the pointed tip... (<i>false statement about hot glue gun</i>) D) When making a paper christmas tree. It is considered the most effective adhesive for securing decorations... (<i>true statement about hot glue gun</i>) E) When fixing a broken iPod. Because its connector is permanently fused to the logic board... (<i>false statement about cable</i>) F) When fixing a broken iPod. It lies hidden under a metal tray near the bottom of the device... (<i>true statement about cable</i>)
Temporal 1-hop	In a video that shows fixing a device on a living room floor. Before a woman plays with a baby, what is the object that the woman places onto the neck?	A) scarf B) headphones C) necklace D) baby bottle E) remote control F) hand
Temporal 2-hop	In a video that shows fixing a device on a living room floor. Before a woman plays with a baby, which statement is true about the object that the woman places onto the neck?	A) When childproofing a living room. Because its components are considered too large to swallow... (<i>false statement about remote control</i>) B) When childproofing a living room. Its small buttons can detach and end up under furniture... (<i>true statement about remote control</i>) C) When reusing household objects for cleaning. Spritzing the interior of it with cooking spray... (<i>false statement about baby bottle</i>) D) When reusing household objects for cleaning. Putting it into a mesh laundry bag prevents it... (<i>true statement about baby bottle</i>) E) When making cat toys out of common household items. It should remain outside the sock while you funnel catnip... (<i>false statement about hand</i>) F) When making cat toys out of common household items. It is placed inside the sock, with the fingers holding the catnip... (<i>true statement about hand</i>)

Table 4: Examples of the four query types in *EgoFact*. For 2-hop queries, only F corresponds to the correct object and fact, while other options are distractors or counter-facts.

Under review as a conference paper at ICLR 2026

[illegible]

Under review as a conference paper at ICLR 2026

[illegible]

Under review as a conference paper at ICLR 2026

[illegible]

Under review as a conference paper at ICLR 2026

[illegible]

[illegible]

D.1 VIDEO UNDERSTANDING EVALUATION DETAILS

For video understanding evaluation, we use *GPT-4o-mini* with the following prompt to evaluate the correctness of caption of each *correctly retrieved* video segment:

```
# Role
You are an information sufficiency checker.

# Goal
Given a video description and a multiple-choice question stem (without
  answer options), decide whether the description provides enough
  information to reasonably answer the question. Output ONLY `Y` or `N`.

# Decision Rules
Return `Y` if the description contains the main action or event
  mentioned in the question (the "query anchor") and provides at least
  some details about the subject, object, or context involved. The
  description does not need to include every fine-grained attribute,
  but it must mention the relevant action. Return `N` if the
  description omits the anchor event or is too vague to connect to the
  question.

# Inputs
- description: string -textual description of the video
- question_stem: string -the multiple-choice question prompt (WITHOUT
  options)

# Output
Return ONLY a single uppercase character: `Y` or `N` (no extra text,
  punctuation, or explanation).

# Examples
## Question 1
Which statement is true about the object that the camera wearer plugs
  into the socket?"

## Description 1
1. The camera wearer walks into a room with a workbench and various
  tools.
2. ...
...
8. The camera wearer takes a power drill and plugs it into the wall
  socket before continuing.

## Output 1
Y

## Question 2
In a video that repairing a car wheel and brakes using tools, what is
  the object that the camera wearer picks from the bonnet?

## Description 2
1. The camera wearer is working on a car in a garage.
2. ...
... (no info about the bonnet)
8. The camera wearer cleans dust off several tools.

## Output 2
N
```

After determining binary correctness for each retrieved segment, we calculate video understanding accuracy as the proportion of queries—within each query type—that are deemed correct according to the above evaluation.

D.2 BASELINE IMPLEMENTATION DETAILS

In this section, we provide additional implementation details for the baselines evaluated in section 4.

D.2.1 GRAPHRAG

For evaluation, we adopt the *local* configuration of *GraphRAG*. We use *GPT-4o-mini* as the chat model and *text-embedding-3-small* as the embedding model. Documents are segmented into chunks of 1200 tokens with an overlap of 100 tokens. This setup ensures that evaluation focuses on the local search capability of GraphRAG.

To perform multiple-choice QA, we slightly modify the prompt file *local_search_system_prompt.txt* to make it answer multiple-choice questions. Specifically, we append the following instructions to the original prompt:

---Role---

You are a helpful assistant responding to multiple-choice questions about data in the tables provided.

---Goal---

Generate a response of the target length and format that responds to the user's question, selecting **one letter choice (A, B, C, ...)** as the final answer, and providing a brief explanation.

- You must always output a single letter answer, even if uncertain.
- If unsure, choose the most likely/certain answer based on the data and general knowledge.
- Do not output "I don't know" or "No answer."

Points supported by data should list their data references as follows:

"This is an example sentence supported by multiple data references [Data: <dataset name> (record ids); <dataset name> (record ids)]."

Do not list more than 5 record ids in a single reference. Instead, list the top 5 most relevant record ids and add "+more" to indicate that there are more.

For example:

"Person X is the owner of Company Y and subject to many allegations of wrongdoing [Data: Sources (15, 16), Reports (1), Entities (5, 7); Relationships (23); Claims (2, 7, 34, 46, 64, +more)]."

where 15, 16, 1, 5, 7, 23, 2, 7, 34, 46, and 64 represent the id (not the index) of the relevant data record.

Do not include information where the supporting evidence for it is not provided.

---Target response length and format---

Output format should be:

****Answer: X****
<one short paragraph explanation>

---Data tables---

```

1620 {context_data}
1621
1622
1623 ---Goal---
1624
1625 Generate a response of the target length and format that responds to the
1626 user's question, selecting one multiple-choice option (letter) with
1627 explanation. Always provide a single letter answer, supported by
1628 evidence if available, or the most plausible choice if uncertain.
1629
1630 ---Target response length and format---
1631
1632 Add sections and commentary to the response as appropriate for the
1633 length and format. Style the response in markdown.
1634
1635
1636 D.2.2 VIDEO-RAG (JEONG)
1637
1638 In this variant, we use the Qwen2.5-VL-3B-Instruct model for generation. Same with VideoRAG
1639 (Ren), we set number of video clips to retrieve as 5 and number of passages to retrieve as 15. To
1640 incorporate passage retrieval, we add an additional step to retrieve relevant WikiHow articles using
1641 the same text embedding model as query embedding in the original paper. And set the number of
1642 passages to retrieve as 5. We also slightly modify the original prompt to include passages and make it
1643 answer multiple-choice questions. The prompt is as follows:
1644
1645 Reference passages: {passages_text}. Considering the given videos,
1646 explain {query}. Please only give an capitalized option (A, B, C, D,
1647 E, or F) without any additional text.
1648
1649
1650 D.2.3 VIDEO-RAG (REN) (WITH OR WITHOUT LOCALIZATION)
1651
1652 Fallback strategy Sometimes the prompt triggered safety policies in GPT-4o-mini, resulting in no
1653 answer. In such cases, we fallback to original MiniCPM model to inference locally.
1654
1655 For the main model, we use GPT-4o-mini as the chat model and text-embedding-3-small as the
1656 embedding model. We set number of video clips to retrieve as 5 where each clip is represented by 15
1657 uniformly sampled frames when generating video description. For passage retrieval, we retrieve 6
1658 relevant WikiHow articles (one passage per option) using the same text embedding model as query
1659 embedding in the original paper. We also slightly modify the original prompt to include passages.
1660 We detail the prompts we use below.
1661
1662 Our prompt for query rewriting for visual retrieval is as follows:
1663
1664 -Goal-
1665 Rewrite the question as a single short sentence starting with "A video
1666 that ...".
1667
1668 -Examples-
1669 Question: In a video where a group of hikers are crossing a narrow
1670 bridge, what is the landscape surrounding them?
1671 Output: A video that depicts hikers crossing a narrow bridge.
1672
1673 Question: In a video showing a crowded marketplace, what is the vendor
1674 placing on the display table?
1675 Output: A video that shows a crowded marketplace.
1676
1677 Question: In a video where a person is cooking in a small kitchen, which
1678 statement is true about the object that is being stirred in the pot?
1679 Output: A video that shows a person cooking in a small kitchen.
1680
1681 -Real Data-
1682 Question: {input_text}
1683 Output:

```

Our prompt for query rewriting for passage retrieval is as follows:

-Goal-

Given a multiple-choice question (MCQ), produce a JSON array of declarative search queries, one per option, that could retrieve text passages relevant to judging the options' correctness.

#####

-Examples-

#####

Question:

In a video showing a chef preparing a meal in a kitchen, which statement is true about the object that the camera wearer uses to stir soup?

- (A) When baking cakes. The object is recommended for folding whipped cream into sponge batter to avoid deflating the mixture.
- (B) When baking cakes. The object should be avoided for mixing flour because its flexible edges cannot break up clumps.
- (C) When making scrambled eggs. The object is considered best for preventing sticking and producing soft curds in a nonstick pan.
- (D) When making scrambled eggs. The object may melt when exposed to high stovetop temperatures.
- (E) When crafting clay models. The object is often used to scoop and shape soft clay before firing.
- (F) When crafting clay models. The object should not be used because it absorbs water and weakens the clay structure.

Output:

```
[
  "A passage about when baking cakes, some object is recommended for
  folding whipped cream into sponge batter.",
  "A passage about when baking cakes, some object should be avoided for
  mixing flour because it cannot break up clumps.",
  "A passage about when making scrambled eggs, some object is best for
  preventing sticking and creating soft curds.",
  "A passage about when making scrambled eggs, some object may melt when
  exposed to high heat.",
  "A passage about when crafting clay models, some object is used to
  scoop and shape soft clay.",
  "A passage about when crafting clay models, some object should not be
  used because it absorbs water and weakens the structure."
]
```

Question:

In a science lab video, which statement is true about the object that the man places on the lab bench?

- (A) When measuring chemicals for titration. The object should not be used for precise volumes since it lacks calibration marks.
- (B) When measuring chemicals for titration. The object is recommended for exact milliliter measurement to ensure accuracy.
- (C) When heating liquids over a Bunsen burner. The object is safe because borosilicate material resists cracking from sudden temperature changes.
- (D) When heating liquids over a Bunsen burner. The object should not be placed directly on the flame because it may shatter without protective gauze.
- (E) When holding biological samples. The object is commonly sterilized in an autoclave before introducing living cultures.
- (F) When holding biological samples. The object cannot be autoclaved since heat and pressure would deform the glass permanently.

Output:

```
[
  "A passage about when measuring chemicals for titration, some object
  should not be used for precise volumes.",
  "A passage about when measuring chemicals for titration, some object is
  recommended for exact milliliter measurement.",
]
```

```

1728     "A passage about when heating liquids, some object is safe because
1729         borosilicate resists cracking.",
1730     "A passage about when heating liquids, some object should not be placed
1731         directly on a flame.",
1732     "A passage about when holding biological samples, some object is
1733         commonly sterilized in an autoclave.",
1734     "A passage about when holding biological samples, some object cannot be
1735         autoclaved because heat deforms glass."
1736 ]
1737 #####
1738 -Real Data-
1739 #####
1740 Question:
1741 {input_text}
1742 #####
1743 Output:
1744
1745 Our prompt for keyword extraction for video retrieval is as follows:
1746
1747 - Goal -
1748 Given a query, extract the most relevant and concrete keywords that can
1749     help locate the answer in a video.
1750
1751 Requirements:
1752 1. Output one line of a numbered list of the extracted keywords,
1753     separated by commas and ending with a period.
1754 2. Do not include overly generic or meaningless terms such as
1755     "object".
1756
1757 #####
1758 - Examples -
1759 #####
1760 Question: Which animal does the protagonist encounter in the forest
1761     scene?
1762 #####
1763 Output:
1764 1. animal, 2. protagonist, 3. forest.
1765
1766 Question: What is the weather like during the opening scene of the film?
1767     (A) Sunny (B) Rainy (C) Snowy (D) Windy
1768 #####
1769 Output:
1770 1. weather, 2. opening scene, 3. film, 4. Sunny, 5. Rainy, 6. Snowy, 7.
1771     Windy.
1772
1773 Question: In a video showing a chef preparing a meal in a kitchen, what
1774     is the object that the camera wearer uses to stir soup?
1775 #####
1776 Output:
1777 1. chef, 2. meal, 3. kitchen, 4. camera wearer, 5. soup.
1778
1779 Question: In a video showing a chef preparing a meal in a kitchen, which
1780     statement is true about the object that the camera wearer uses to
1781     stir soup?
1782 #####
1783 Output:
1784 1. chef, 2. meal, 3. kitchen, 4. camera wearer, 5. soup.
1785
1786 #####
1787 - Real Data -
1788 #####
1789 Question: {input_text}
1790 #####

```

1782 Output:
 1783
 1784 Our prompt for generating multiple-choice answers is as follows for *1-hop* queries:
 1785
 1786 ---Role---
 1787 You are a helpful assistant responding to a multiple-choice question
 1788 with retrieved knowledge.
 1789
 1790 ---Goal---
 1791 Generate a concise response that addresses the user's question by
 1792 summarizing relevant information derived from the retrieved text and
 1793 video content. Ensure the response aligns with the specified format
 1794 and length.
 1795 Please note that there is only one choice is correct.
 1796
 1797 ---Retrieved Information From Videos---
 1798 {video_data}
 1799
 1800 ---Retrieved Text Chunks---
 1801 {chunk_data}
 1802
 1803 ---Goal---
 1804
 1805 Generate a concise response that addresses the user's question by
 1806 summarizing relevant information derived from the retrieved text and
 1807 video content. Ensure the response aligns with the specified format
 1808 and length.
 1809 Please note that there is only one choice is correct.
 1810
 1811 ---Notice---
 1812 Please provide your answer in JSON format as follows:
 1813 {{
 1814 "Answer": "The label of the answer, like A/B/C/D or 1/2/3/4 or
 1815 others, depending on the given query"
 1816 "Explanation": "Provide explanations for your choice. Use sections
 1817 and commentary as needed to ensure clarity and depth. Format the
 1818 response in Markdown."
 1819 }}
 1820 Key points:
 1821 1. Ensure that the "Answer" reflects the correct label format.
 1822 2. Structure the "Explanation" for clarity, using Markdown for any
 1823 necessary formatting.
 1824
 1825 Our prompt for generating multiple-choice answers is as follows for *2-hop* queries:
 1826
 1827 ---Role---
 1828 You are a helpful assistant responding to a multiple-choice question
 1829 with retrieved knowledge.
 1830
 1831 ---Goal---
 1832 Generate a concise response that addresses the user's question by
 1833 summarizing relevant information derived from the retrieved text,
 1834 video content, and retrieved passage from a passage corpus. Ensure
 1835 the response aligns with the specified format and length.
 Please note that there is only one choice is correct.
 ---Retrieved Information From Videos---
 {video_data}

```

1836
1837 ---Retrieved Text Chunks---
1838
1839 {chunk_data}
1840
1841 ---Retrieved Passages---
1842
1843 {passage_data}
1844
1845 ---Goal---
1846
1847 Generate a concise response that addresses the user's question by
1848 summarizing relevant information derived from the retrieved text and
1849 video content. Ensure the response aligns with the specified format
1850 and length.
1851 Please note that there is only one choice is correct.
1852
1853 ---Notice---
1854 Please provide your answer in JSON format as follows:
1855 {{
1856   "Answer": "The label of the answer, like A/B/C/D or 1/2/3/4 or
1857   others, depending on the given query"
1858   "Explanation": "Provide explanations for your choice. Use sections
1859   and commentary as needed to ensure clarity and depth. Format the
1860   response in Markdown."
1861 }}
1862 Key points:
1863 1. Ensure that the "Answer" reflects the correct label format.
1864 2. Structure the "Explanation" for clarity, using Markdown for any
1865 necessary formatting.
1866
1867 Our prompt for video captioning is as follows:
1868
1869 List only the main actions shown across the provided screenshots,
1870 presented step by step as a numbered list (1., 2., 3., ...).
1871 If any intermediate steps appear missing between screenshots, reasonably
1872 infer and include them to ensure continuity.
1873 Make each action as specific and detailed as possible, focusing on the
1874 key elements of the query -if some elements in it are not relevant
1875 or visible, you may ignore them.
1876 Avoid describing the environment, visual appearance, or any irrelevant
1877 details.
1878 Output only the ordered sequence of actions.
1879
1880 All other settings are the same as default.
1881
1882 D.2.4 ADDITIONAL DETAILS ON LOCALIZATION
1883
1884 We use the following prompt to generate text for video localization for each query:
1885
1886 -Goal-
1887 Given a query about a video, output a JSON object with two fields:
1888
1889 1. "description": the anchor event in the video, used to localize the
1890 relevant part.
1891 - If the query explicitly contains a temporal cue ("before ..." or
1892 "after ..."),
1893 then "description" must be that anchor event.
1894 Example: "before they reach the summit" → "description": "they reach
1895 the summit"
1896 Example: "after the camera wearer chops vegetables" → "description":
1897 "the camera wearer chops vegetables"
1898 - If the query does not contain such cues, then "description" must be
1899 the event itself

```

```

1890     that needs to be observed to answer the query.
1891     Example: "what does the goalkeeper do when the ball is kicked toward
1892             the goal?"
1893     →"description": "the goalkeeper kicks the ball toward the goal"
1894
1895 2. "position": where the part that must be watched lies relative to this
1896    anchor.
1897    One of: "before", "after", or "within".
1898
1899 -Examples-
1900 Question: In a video that shows children playing soccer in a park, what
1901          is the object that goalkeeper kicks toward the goal?
1902 Output: {{
1903     "description": "the goalkeeper kicks something toward the goal",
1904     "position": "within"
1905 }}
1906
1907 Question: In a video that shows children playing soccer in a park, which
1908          statement is true about the object that goalkeeper kicks toward the
1909          goal?
1910 Output: {{
1911     "description": "the goalkeeper kicks something toward the goal",
1912     "position": "within"
1913 }}
1914
1915 Question: In a video that shows a chef preparing food in a restaurant
1916          kitchen, after the camera wearer chops vegetables, what is the
1917          object that the camera wearer place into the pan?
1918 Output: {{
1919     "description": "the camera wearer chops vegetables",
1920     "position": "after"
1921 }}
1922
1923 Question: In a video that shows a chef preparing food in a restaurant
1924          kitchen, after the camera wearer chops vegetables, which statement
1925          is true about the camera wearer place into the pan?
1926 Output: {{
1927     "description": "the camera wearer chops vegetables",
1928     "position": "after"
1929 }}
1930
1931 Question: In a video that shows a group of people hiking up a mountain
1932          trail, before people reach the summit, what is the object that one
1933          of the hikers adjust on their backpack?
1934 Output: {{
1935     "description": "people reach the summit",
1936     "position": "before"
1937 }}
1938
1939 -Real Data-
1940 Question: {input_text}
1941
1942 Output a parseable JSON object with "description" and "position" fields,
1943         without any additional text.

```

1944 If the *position* turns to be *before* or *after*, which is a temporal cue, we shift the localization window
1945 by 1 second accordingly.
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997