UNDERSTANDING LONG VIDEOS THROUGH GRAPH REPRESENTATION

Mayssa Hajj Hassan
American University of Beirut

Dr. Ammar Mohanna
American University of Beirut

ABSTRACT

Understanding and reasoning over long videos remains challenging for large video language models (LVLMs), which struggle to process hours of content while preserving discourse structure across topics, scenes, and speakers. Naive chunk-based Retrieval-Augmented Generation (RAG) often breaks temporal coherence, ignores conversational roles, and retrieves redundant or off-topic segments, limiting its usefulness for high-level understanding and summarization. We introduce GraphCast, a graph-based retrieval and reasoning framework for long video understanding. GraphCast first converts the full transcript into a semantic discourse graph by clustering the episode into topics, assigning speaker roles, and mapping each topic to a schema-driven graph (narrative, descriptive, informative, instructional, or argumentative), then linking topic-level subgraphs. It then enriches this transcript-derived graph with visual grounding by aligning video clips to existing nodes and edges, updating the graph and attaching multi-modal attributes such as visual descriptions and reference clips. On top of this unified semantic–visual graph, GraphCast performs graph-based retrieval and structured reasoning to support tasks like long-form QA and reconstructing concise video summaries, e.g., a five-minute video that faithfully condenses a two-hour podcast. While our proof-of-concept focuses on conversational podcasts, the framework is designed to generalize to arbitrary long-form videos. Our code is publicly available at `https://github.com/MayssaHH/Graphcast`

1 INTRODUCTION

Multi-modal large language models (MLLMs) have demonstrated impressive visual understanding and reasoning capabilities, paving the way for progress in video-centric tasks such as question answering, captioning, and story understanding (Li et al., 2023; OpenAI, 2023). Recently, large video language models (LVLMs) have extended these capabilities to richer temporal inputs, enabling models to process dynamic visual content and spoken language together in a unified framework. Long-form videos—including lectures, documentaries, streaming content, and conversational podcasts—are especially important in real-world applications, as they concentrate most of the information people consume daily and often contain complex narratives and evolving topics that span hours.

However, processing and reasoning over such long-context videos remains a substantial challenge for existing LVLMs. Representing long videos as frame sequences or dense clips quickly explodes the number of visual tokens, often far beyond the model's context window. Even when transcripts are available, a two-hour video can easily produce tens of thousands of tokens from the audio alone, making it difficult to retain global discourse structure and fine-grained dependencies across segments. To handle long inputs, current systems often resort to sparse frame sampling or token compression, but these strategies inevitably discard visual details and blur temporal structure, limiting high-level understanding and fine-grained reasoning.

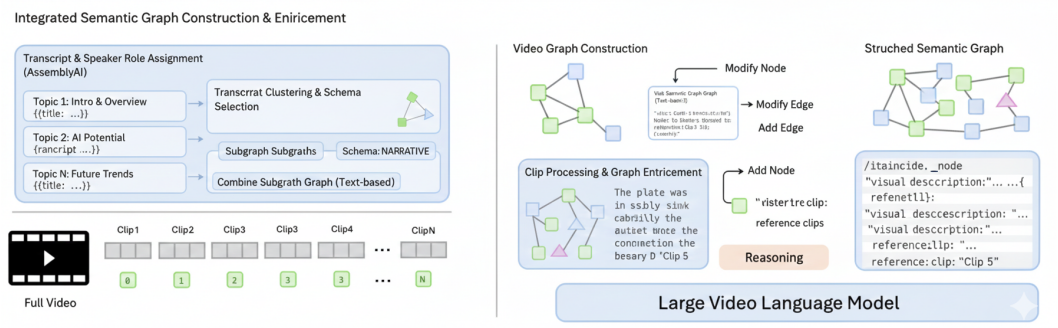

Figure 1: Overview of **GraphCast**. The framework first constructs a semantic discourse graph from the full transcript (topic clustering, schema-driven subgraphs, and speaker roles), then enriches it with visual information from video clips, and finally uses the unified semantic–visual graph for retrieval, reasoning, and video reconstruction.

Retrieval-Augmented Generation (RAG) has emerged as a promising paradigm for scaling language models to long contexts by retrieving only a subset of relevant evidence for each query (Lewis et al., 2020; Izacard & Grave, 2021). Extending RAG to videos, recent works segment long videos into shorter clips or textual chunks and retrieve over this collection to support question answering and other downstream tasks. While effective to some extent, naïve chunk-based video RAG typically suffers from three key limitations. First, segmenting a video into short clips or transcript chunks tends to break temporal coherence and scatter related events across multiple independent documents, making it difficult to track entities and arguments over time. Second, purely chunk-based retrieval ignores higher-level discourse structure—such as topic shifts, speaker roles, and rhetorical functions (e.g., explanation vs. instruction)—which are particularly salient in conversational content like podcasts. Third, retrieval noise and redundancy easily emerge when many semantically similar chunks are retrieved, overwhelming LVLMs and leading to shallow or inconsistent answers.

Another line of work tackles long videos through agent-based pipelines, where an LLM orchestrates a suite of tools for transcript generation, video analysis, planning, and multi-step reasoning. While these systems can achieve strong performance, they frequently depend on proprietary, closed-source models and complex tool-chains, which introduce high computational and monetary costs and limit reproducibility and deployment flexibility. Moreover, they typically operate on flat sets of clips or text segments without an explicit, reusable structural representation of the video.

In this work, we explore a different perspective: representing long videos through semantic–visual discourse graphs that explicitly model topics, speakers, and rhetorical relations, then using this structure to drive retrieval, reasoning, and summarization. We introduce **GraphCast**, a graph-based retrieval–reasoning framework designed for long video understanding. GraphCast starts from the full transcript of a video, augmented with speaker roles, and clusters it into topic-level segments. Each topic is then mapped to a schema-driven graph according to its dominant discourse mode—narrative, descriptive, informative, instructional, or argumentative—yielding a topic-specific subgraph whose nodes capture key units of content (e.g., claims, events, definitions) and whose edges encode semantic and discourse relations within the topic. Topic subgraphs are linked according to their transitions and shared elements, forming a global semantic discourse graph that summarizes the entire video at the transcript level. An overview of the GraphCast framework is shown in Figure 1.

To incorporate visual grounding, GraphCast does not rebuild a graph from video clips directly. Instead, it uses the semantic discourse graph as a base and enriches it with video information. The video is segmented into clips, and each clip is aligned to existing nodes and edges in the graph. This alignment allows the system to modify node and edge content, add new nodes or edges when needed, and attach multi-modal attributes such as visual descriptions and reference clip indices. The result is a unified semantic–visual graph where each node and relation can be traced back to one or more concrete video segments, and where discourse-level structure and audiovisual evidence are tightly coupled.

On top of this representation, GraphCast supports graph-based retrieval and structured reasoning for long-form question answering and video summarization. Given a query or a summarization objective (e.g., "produce a five-minute video that captures the key ideas of this two-hour podcast"), the system can identify salient subgraphs, retrieve the corresponding clips, and assemble them into a concise yet coherent video that preserves the original conversational flow across speakers and topics. While our proof-of-concept experiments focus on conversational podcasts—an especially challenging form of long video with multiple speakers and loosely structured topic shifts—the framework is designed to generalize to arbitrary long-form videos where both transcript and visual streams are available.

Our main contributions are as follows:

- We propose **GraphCast**, a transcript-first, schema-driven graph representation for long videos that encodes topic structure, speaker roles, and discourse relations before grounding them in visual evidence.
- We introduce a multimodal graph enrichment step that aligns video clips to an existing semantic discourse graph, enabling node/edge updates and the attachment of visual attributes and reference clips, which supports graph-based retrieval, reasoning, and reconstruction of concise video summaries.
- We present a podcast-based proof of concept for long-form question answering and video summarization, and discuss how GraphCast can be integrated with off-the-shelf LVLMs and extended to general long-form video understanding.

2 Related Work

2.1 Large Video Language Models

Recent multi-modal large language models (MLLMs) and large video language models (LVLMs) extend language models to images and videos by coupling visual encoders with powerful LLM backbones. Early works such as Flamingo and BLIP-2 demonstrated strong few-shot and zero-shot performance on image–text tasks by combining frozen vision encoders with lightweight bridging modules into large language models (Li et al., 2023; OpenAI, 2023). LVLMs like MiniGPT-4, Video-ChatGPT, and Video-LLaVA further adapt this paradigm to videos, typically by sampling frames or short clips and feeding concatenated visual tokens and text into the LLM backbone. While these models achieve impressive results on short clips and benchmarks with limited temporal extent, their architectures are fundamentally constrained by context length and computational budget when scaling to hour-long videos. Long-form videos—including lectures, documentaries, streaming content, and conversational podcasts—are especially important in real-world applications, as they concentrate much of the information people consume daily and often contain complex narratives and evolving topics that span hours.

To better handle longer sequences, several works revisit temporal modeling and memory mechanisms for long videos, for example by using hierarchical temporal attention, memory tokens, or document-like video representations. Other methods explore token reduction and compression strategies to fit more content into limited context windows. Despite these advances, most LVLMs still process long videos as relatively flat sequences of frames, clips, or captions, without an explicit representation of discourse structure or topic transitions. In contrast, GraphCast builds a structured semantic discourse graph from the full transcript first, then enriches it with visual information, and uses this graph as a reusable backbone for retrieval, reasoning, and summarization.

2.2 Agent-based Long Video Understanding

Another line of work leverages agent-style pipelines where an LLM orchestrates tools for perception, retrieval, and reasoning over long videos. Systems such as VideoAgent and DrVideo convert long videos into text-like artifacts (e.g., frame captions, clip summaries, or long "video documents"), then use multi-stage loops of planning, retrieval, and refinement to progressively locate question-relevant evidence (Wang et al., 2024; Ma et al., 2025). Related approaches similarly transfer long-document reasoning abilities of LLMs to video by working primarily in language space, often without training new LVLMs (Zhang et al., 2024). These systems can achieve strong performance on long-video

QA benchmarks, but they typically operate on flat sets of snippets (captions, frames, or chunks) and re-run complex tool-chains per query.

Moreover, agent-based pipelines frequently depend on heavyweight proprietary models and multi-step orchestration, which can make them expensive and hard to reproduce. They rarely maintain an explicit, query-agnostic structural representation of the video that can be reused across tasks. GraphCast shares the spirit of being training-free and model-agnostic, but instead of repeatedly regenerating structure per query, it constructs a persistent semantic–visual graph once per video from transcripts and clips, which can then support multiple downstream tasks such as QA and video reconstruction.

2.3 Video Retrieval-Augmented Generation and Graph-based Reasoning

Retrieval-Augmented Generation (RAG) has become a standard approach for extending language models to long or external knowledge sources by retrieving a subset of relevant documents for each query (Lewis et al., 2020; Izacard & Grave, 2021). Video RAG methods adapt this idea by chunking long videos into clips or caption segments, indexing them, and retrieving top-ranked segments at inference time to support question answering and related tasks. Recent frameworks explore more sophisticated retrieval strategies, including document-style video representations, frame selection, or multi-stage filtering to reduce the number of visual tokens.

More closely related to our work are methods that introduce graph structures into video understanding. Several approaches construct spatio-temporal graphs over entities or frames and perform reasoning over these graphs to answer compositional or long-range queries. Vgent, for example, represents long videos as clip-level graphs with edges defined by semantic and temporal relations, and then performs graph-based retrieval followed by structured reasoning to reduce noise and aggregate evidence across clips (Shen et al., 2025). Other recent works similarly combine graph-based retrieval or reasoning with long-video RAG, focusing on better selection of frames or clips under tight context budgets.

GraphCast differs in two key aspects. First, it is transcript-first and schema-driven: instead of building a graph directly from clips, it clusters the full transcript into topic segments, assigns discourse schemas (narrative, descriptive, informative, instructional, argumentative), and constructs topic-level subgraphs that encode rhetorical relations and speaker roles before visual information is considered. Second, visual grounding is formulated as graph enrichment, where clips align to existing nodes and edges, potentially modifying them or adding new ones while attaching multi-modal attributes (visual descriptions, reference clips). This design makes the graph itself the primary long-term memory of the video, with retrieval and reasoning operating over a structured representation that already captures discourse and topic transitions—crucial for podcast-style content and for reconstructing concise yet coherent video summaries.

2.4 Discourse Structure, Podcasts, and Summarization

Discourse parsing and rhetorical structure theory have a long history in NLP, and recent LLM-based systems have revisited discourse-aware representations for long documents, meetings, and dialogues. Podcasts and long conversational videos share many of the same challenges: multiple speakers, loosely structured topic shifts, and interleaved narrative, explanatory, and argumentative segments. Prior work on podcast or meeting summarization typically approaches the problem from a text-only perspective, using clustering, topic detection, or hierarchical summarization without explicit graph representations, and without tying summaries back to specific audiovisual segments.

GraphCast explicitly tailors its semantic discourse graph to this setting by (i) modeling speaker roles at the node level, (ii) organizing content into topic-specific subgraphs with discourse schema, and (iii) maintaining explicit connections between nodes/edges and reference clips. This enables not only long-form QA but also video-level summarization in which a short summary video is reconstructed by selecting and ordering clips that correspond to the most salient subgraphs—for example, generating a five-minute highlight reel that faithfully captures the key ideas of a two-hour podcast—bridging the gap between text-based discourse modeling and multi-modal long-video understanding.

3 METHODOLOGY

Given a long video v with audio and visual streams, GraphCast builds a semantic–visual discourse graph that can be reused across tasks such as long-form QA and video summarization. The framework has two stages: (i)transcript-first semantic graph construction, and (ii) multi-modal enrichment with video clips, followed by graph-based retrieval and reasoning.

3.1 TRANSCRIPT AND TOPIC-CENTRIC REPRESENTATION

We first convert the audio of the video into a speaker-attributed transcript using an automatic speech recognition (ASR) system with diarization (we used AssemblyAI for this task). This yields a sequence

$$T = \{(u_i, s_i, t_i^{\text{start}}, t_i^{\text{end}})\}_{i=1}^{L}, \tag{1}$$

where each utterance u_i is associated with a speaker label s_i and a time span $[t_i^{\text{start}}, t_i^{\text{end}}]$.

GraphCast then clusters the transcript into topic-level segments:

$$\mathcal{C} = \{C_1, \ldots, C_M\}, \tag{2}$$

where each C_j is a contiguous or near-contiguous block of utterances that form a coherent topic. For each topic, we derive (i) a short topic title summarizing the segment, and (ii) a concatenated topic transcript that starts from the earliest utterance of the segment and includes all utterances assigned to that topic. At this stage, we obtain a structured JSON-like representation:

```
{
  "topic_1": {
    "title": "title of the first topic",
    "transcript": "u_1 ... u_k"
  },
  "topic_2": {
    "title": "title of the second topic",
    "transcript": "..."
  }
}
```

In our podcast proof-of-concept, this topic clustering often aligns with natural shifts in the conversation (e.g., moving from high-level trends to concrete case studies), but the formulation is general and applies to other long-form videos such as lectures or documentaries.

3.2 SCHEMA-DRIVEN SEMANTIC DISCOURSE GRAPH

For each topic C_j, GraphCast predicts a dominant discourse schema from a small set

$$\mathcal{S} = \{\text{narrative, descriptive, informative, instructional, argumentative}\}. \tag{3}$$

This schema acts as a coarse prior on the structure of the topic: narrative segments emphasize events and temporal progression, argumentative segments emphasize claims and support/attack relations, instructional segments emphasize steps and preconditions, and so on.

Within each topic, we further decompose the transcript into elementary discourse units (e.g., sentences or clause-level segments) and instantiate them as nodes. For topic C_j, we obtain a set of nodes

$$\mathcal{N}_j = \{n_1^{(j)}, \ldots, n_{K_j}^{(j)}\}. \tag{4}$$

Each node stores at least:

- the textual content,
- the speaker label,
- the time span,
- the topic identifier and schema,
- optional metadata such as keywords.

We then add edges between nodes to encode relations determined by the schema. In practice, GraphCast uses a unified set of connection types:

- **Narrative schema:** `ACTION_RELATION`, `SPATIAL_RELATION`, `TEMPORAL_RELATION`;
- **Descriptive schema:** `HAS`, `IS`;
- **Informative schema:** `CONCEPT_TO_CONCEPT`, `IS_DEFINITION`, `IS_EXAMPLE`, `IS_EXPLANATION`;
- **Instructional schema:** `SEQUENTIAL_RELATION`, `CONDITIONAL_RELATION`;
- **Argumentative schema:** `SUPPORTING_RELATION`, `COUNTER_SUPPORTING_RELATION`, `CONCLUSION_RELATION`.

These connection types can be instantiated using a combination of heuristics (e.g., adjacency, shared entities) and LLM-based relation classification.

These can be instantiated using a combination of heuristics (e.g., adjacency, shared entities) and LLM-based relation classification. For topic C_j, we obtain a topic-level graph

$$G_j^{\text{text}} = (\mathcal{N}_j, \mathcal{E}_j), \tag{5}$$

where $\mathcal{E}_j$ contains schema-dependent edges between nodes.

Finally, GraphCast links topic-level graphs into a global semantic discourse graph:

$$G^{\text{text}} = (\mathcal{N}, \mathcal{E}), \tag{6}$$

where $\mathcal{N} = \bigcup_j \mathcal{N}_j$ and $\mathcal{E}$ includes both intra-topic edges $\mathcal{E}_j$ and inter-topic edges encoding transitions, shared entities, or recurring themes (e.g., a key concept revisited in multiple segments). This graph compactly summarizes the transcript-level structure of the entire video, with explicit topics, speakers, and rhetorical relations.

3.3 Multimodal Graph Enrichment with Video Clips

The second stage grounds this semantic discourse graph in the visual stream. We segment the video into fixed-length or content-aware clips:

$$\mathcal{V} = \{c_1, \ldots, c_K\}, \tag{7}$$

where each clip c_k has a time span $[t_k^{\text{start}}, t_k^{\text{end}}]$. For each clip, we compute a visual description using frame sampling and a vision–language model (e.g., an LVLM or captioning model), and optionally extract additional signals such as scene changes or on-screen text.

Instead of constructing a new graph from clips, GraphCast treats the existing semantic discourse graph G^{text} as a base graph and performs graph enrichment as follows:

1. **Alignment.** For each clip c_k, we identify all nodes $n \in \mathcal{N}$ whose time spans overlap with the clip or whose utterances are spoken during the clip. This defines candidate alignments between clips and nodes/edges.
2. **Node and edge updates.** For aligned nodes, we allow the graph to be updated based on visual evidence. For example, if a node's content mentions "the compute super cycle" while the clip shows a graph of model parameters, the visual description can refine the node's representation or add attributes indicating that a figure or slide is present.
3. **Node and edge additions.** If the clip reveals salient visual information not captured by the transcript (e.g., on-screen text, diagrams, or interactions without speech), we can insert new nodes and link them via edges to the most related existing nodes.

Each node thus becomes a multi-modal node with attributes such as:

```
{
  "id": "node_1",
  "content": "Compute super cycle",
  "speaker": "Host",
  "topic": "Introduction and overview of the podcast",
  "schema": "informative",
  "timespan": [t_start, t_end],
  "visual_description": "The host stands in front of a slide showing a
      curve of compute growth over time.",
  "reference_clips": [3, 4]
}
```

Analogously, edges can store textual relation descriptions and references to supporting clips. The enriched graph is denoted

$$G^{\text{mv}} = (\mathcal{N}^{\text{mv}}, \mathcal{E}^{\text{mv}}), \tag{8}$$

emphasizing that nodes and edges now carry multi-modal attributes.

3.4 Graph-based Retrieval, Reasoning, and Video Reconstruction

Once G^{mv} is constructed, GraphCast uses it as a unified memory for retrieval, reasoning, and video reconstruction.

Query-aware subgraph retrieval. Given a text query q (e.g., a question or a summarization instruction), we score nodes and edges according to their semantic similarity to q, their topic relevance, and schema-informed priors (e.g., argumentative nodes may be prioritized for "why" questions). We then extract a query-centric subgraph G_q by:

- selecting top-k nodes ranked by a relevance score,
- expanding to their neighbors within a few hops to recover local context,
- pruning overly redundant or low-information nodes.

Structured reasoning over graphs. The subgraph G_q is serialized into a structured prompt (e.g., as topic-ordered node lists with relation annotations and attached visual descriptions) and fed to an LVLM or LLM. Instead of directly operating on raw clips or flat chunks of text, the model sees a compressed, structured view of the video that already encodes discourse and topic transitions. This supports:

- long-form question answering, where the model aggregates evidence across topics and speakers;
- explanation-style outputs that respect the original narrative or argumentative structure.

Graph-driven video summarization. For video reconstruction tasks, such as generating a five-minute highlight reel of a two-hour podcast, GraphCast operates at the graph level:

1. rank nodes and subgraphs by importance (e.g., based on centrality, schema type, or task-specific scoring);
2. select a subset of nodes and edges that covers the main topics while maintaining continuity;
3. retrieve the associated reference clips from $\mathcal{V}$;
4. order clips according to the structure of the selected subgraph (e.g., topic order plus within-topic narrative order) and pass them to a downstream video editing or stitching component.

Because clip selection is guided by the semantic discourse graph, the resulting summary tends to preserve both the high-level topic flow and the speaker dynamics of the original video, rather than being a disjoint collection of visually salient moments. While our experiments focus on conversational podcasts, the same mechanism applies to other genres where transcripts and visual streams jointly define the narrative.

REFERENCES

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open-domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the ACL (EACL)*, 2021. arXiv:2007.01282.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, 2020. arXiv:2005.11401.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning (ICML)*, 2023. arXiv:2301.12597.

Ziyu Ma, Chenhui Gou, Hengcan Shi, et al. Drvideo: Document retrieval based long video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. arXiv:2406.12846.

OpenAI. GPT-4V(ision) system card. Technical report, OpenAI, 2023. Technical report, September 2023.

Xiaoqian Shen, Wenxuan Zhang, Jun Chen, and Mohamed Elhoseiny. Vgent: Graph-based retrieval-reasoning-augmented generation for long video understanding. In *Advances in Neural Information Processing Systems*, 2025. arXiv:2510.14032.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *European Conference on Computer Vision (ECCV)*, 2024. arXiv:2403.10517.

Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. A simple LLM framework for long-range video question-answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024. arXiv:2312.17235.