Building a Mind Palace: Structuring Environment-Grounded Semantic Graphs for Effective Long Video Analysis with LLMs

Supplementary Material

In this appendix, we report additional details on VideoMindPalace, the Video MindPalace Benchmark (VMB), additional results and visualizations. In section 7, we include more visualizations of both queries and qualitative results across different kinds of questions in VMB. We then give more information on the graph construction for VideoMindPalace and how we prompt VideoMindPalace to obtain the answers for the questions in VMB in section 8. Next, in section 9, we present additional ablations for VideoMindPalace. Finally, we further detail VMB in section 10.

7. More qualitative results

Video graph representation visualization In the figure 4, we present a graph-based visualization of a 30-second video using our Video MindPalace framework, structured into three conceptual layers. The first layer maps humans and detected objects as nodes, with edges illustrating their interactions and movement over time and space. The second layer defines key activity zones, connecting them through edges that reflect their three-dimensional spatial configuration. The third layer outlines the broader scene structure, where node represent individual room.

Video reasoning visualization In the figure 5, we showcase additional examples of VideoMindPalace's reasoning capabilities across various question types. For each question type, we provide a representative example. To illustrate how VideoMindPalace effectively answer these questions, we utilize GPT-4 to identify specific segments of the graph containing the necessary information for accurate inference using the following prompt: ``You are an expert in analyzing semantic graphs generated from video content. In the following graph, nodes capture spatial concepts (e.g., objects, activity zones, rooms), and edges signify spatiotemporal, layout relationships and human-object interaction. Given a query, your task is to identify specific segments of the graph that provide sufficient information to answer the query accurately. Input: 1. Constructed graph: [...], 2. Query and options: [...].''

8. More graph construction details

8.1. Prompt design of zero-shot video QA

We employ the following prompts to extract critical information from the constructed graph for zero-shot video question answering tasks: 1. For Multiple-Choice Video QA: ``You are an expert at reasoning with semantic graphs to analyze video content. In the following graph, nodes capture spatial concepts (e.g., objects, activity zones, rooms), and edges signify spatiotemporal, layout relationships and human-object interaction. Your task is to answer a query based on the provided graph by selecting the correct answer from given options. Input: 1. Constructed graph: [...], 2. Query and options: [...]. Task: Analyze the graph and determine which option correctly answers the query. Provide only the correct answer (e.g.,``A'') without additional explanations.''. 2. For open-ended video QA: ``You are an expert in analyzing video content to summarize human actions and activities. In the following graph, nodes capture spatial concepts (e.g., objects, activity zones, rooms), and edges signify spatiotemporal, layout relationships and human-object interaction. Your task is to extract and describe fine-grained actions, transitions, and spatial sequences performed by a person in the video. The summary should provide a coherent and detailed account of the activity flow, highlighting both actions and their relationships to the environment. Input: Constructed graph: [...].''.

8.2. Graph construction heuristic

Layer 1 - Human and Object: we consider u and v represent two entities in the scene, such as a human or an object. Each detected object is assigned a unique identifier, denoted as ID_u, and categorized with a semantic label C_u. The spatial location of an object in frame t is represented by its bound-

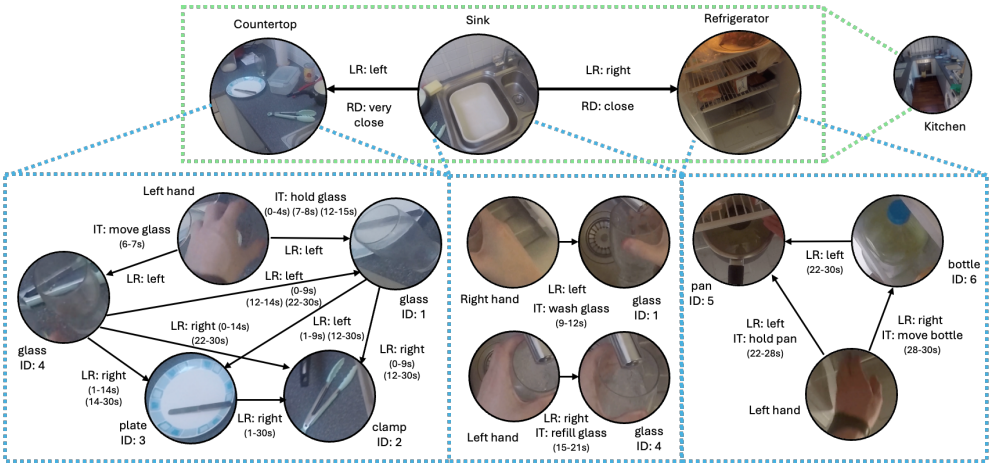

Figure 4. We visualize the graph representation of a 30-second video using our Video MindPalace, which organizes information across three semantic layers. Layer 1 captures the human and detected objects as nodes, with edges representing their spatiotemporal interactions. Layer 2 abstracts the scene into activity zones, where nodes represent distinct areas of activity and edges encode their spatial relationships in 3D space.

ing box $B_u(t) = (x_{\min}, y_{\min}, x_{\max}, y_{\max})$. Between the two nodes N_u and N_v, three types of edges are defined, each capturing a specific relationship. The first type, the Interaction Type Edge (I_{uv}), encodes the predicted interaction type between u and v, as inferred by a captioning model (e.g., *"cutting"*, *"holding"*, or *"placing"*). For each interaction, we concatenate the center frame with the start and end frames into a single composite frame and feed it to captioner to prediction. The second type, the Temporal Window Edge (T_{uv}), specifies the duration of the interaction as $T_{uv} = (t_{\text{start}}, t_{\text{end}})$, where $t_{\text{start}} = \frac{f_{\text{start}}}{\text{FPS}}$ and $t_{\text{end}} = \frac{f_{\text{end}}}{\text{FPS}}$. Here, f_{start} and f_{end} represent the start and end frames of the interaction, respectively, as predicted by a human-object interaction tracker used in [12]. The third type, the Spatial Relationship Edge (S_{uv}), captures the relative position of u with respect to v. Using the bounding box centers at t_{start}, the relative horizontal position is determined based on the x_{center} coordinates, where $x_{\text{center}} = \frac{x_{\min}+x_{\max}}{2}$. If $x_{\text{center},u} > x_{\text{center},v}$, u is labeled as *"right"*; otherwise, u is labeled as *"left"*. Additionally, we track objects manipulated by humans and update their spatial relation with other nodes whenever an object is moved. Together, these edges encode the semantic, spatial, and temporal relationships between the nodes, forming a comprehensive graph.

Layer 2 - Activity Zones: To estimate layout relations, we first consider two activity zones, which is clustered using method introduced at 3.2, A and B, derived from the video data. Using the centroid positions (x_A, y_A, z_A) and (x_B, y_B, z_B) of each zone, we define their relative layout based on their spatial arrangement. For instance, B is labeled as "left of" A if $x_B > x_A$ and the differences along y and z axes are within a predefined threshold of 0.5. Similarly, "right of", "in front of" and "behind" relationships are determined by comparing coordinate displacements along x, y, and z axes. These directional descriptors are attached as attributes to the edge connecting the zones. To capture relative distances, we compute the Euclidean distance between the centroids of A and B as $d_{AB} = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2}$. Additionally, we normalize the distance for scale invariance as $d_{AB}^{\text{normalized}} = \frac{d_{AB} - d_{\min}}{d_{\max} - d_{\min}}$, where $d_{\min}$ and $d_{\max}$ are the minimum and maximum distances across all zone pairs. we categorize the distances into five levels based on thresholds: $0.1, 0.2, 0.4, 0.7$. Distances falling below these thresholds are labeled as *very close* (0–3 steps), *close* (4–6 steps), *moderate* (7–10 steps), *far* (11–15 steps), or *very far* (16+ steps). This categorization allows us to qualitatively assess physical proximity between zones. This combined representation, with directional relationships and normalized distances, enables reasoning about both the spatial layout and physical proximity of the two activity zones.

Layer 3 - Scene Layout: To construct Layout Distance at Layer 3, we group multiple activity zones from Layer 2 into higher-level spatial groupings based on room-level context using captions generated by the captioning model. These captions help identify the specific room where each activity zone belongs, such as "kitchen" or "living room". We use the center frame and four uniformly sampled frames per activity zone, feeding them to the captioner for room classification, followed by majority voting for robust evaluation. The layout distance between rooms is determined by analyzing their physical layout and positioning within the environment. Consider two rooms, R_1 and R_2. Using the centroid coordinates (x_r, y_r, z_r) of each room—calculated as the average of the centers of the activity zones belonging to that room—we com-

pute the Euclidean distance between them as $d_{R_1R_2} = \sqrt{(x_{R_2} - x_{R_1})^2 + (y_{R_2} - y_{R_1})^2 + (z_{R_2} - z_{R_1})^2}$. This distance estimation follows the same approach used in Layer 2 for determining layout distances. These room-level distances contextualize the spatial flow between areas, capturing both proximity and potential movement pathways.

Lastly, as mentioned in Section 3.2, two additional hyperparameters are used: the visual threshold for feature similarity, $s^* = 0.6$, and the distance threshold for spatial proximity, $d^* = 0.5$.

9. Ablation Studies

Representation size and context window We compared token counts for representing videos of various lengths (1, 3, 10, 20, and 30 minutes). LLovi, using a sliding window approach, resulted in token counts of 5.1 17.7, 58.7, 117.7, and 175.1k, respectively, as video length increased. In contrast, our graph approach produced significantly lower token counts of 3.2, 10.1, 30.9, 53.5, and 69.4k, demonstrating a more concise representation, particularly for longer videos.

Method	1mins	3mins	10mins	20mins	30mins
LLovi	5.1	17.7	58.7	117.7	175.1
Ours	3.2	10.1	30.9	53.5	69.4

Table 4. The table compares token counts for representing videos of various lengths. LLovi uses a sliding window approach, resulting in significantly higher token counts as video length increases, whereas our graph achieves a more concise representation, particularly for longer videos.

Cluster by location vs Split by temporal window We hypothesize that clustering temporally distant yet spatially relevant frames within a layered graph enables VideoMindPalace to reduce information overload and redundancy in long video analysis. To validate this, we compare our approach with an alternative method that segments the video into shorter chunks and builds separate graphs for each chunk, setting the number of chunks equal to our number of clusters for a direct comparison. Table 5 shows that our location-based clustering method outperforms temporal window segmentation across all video lengths, with improvements especially notable in Medium and Long videos (2.7% and 3.4% higher, respectively). This indicates that clustering by location effectively reduces redundancy while preserving spatial relevance in longer videos.

Impact of different tools on reasoning performance We evaluated the impact of each pipeline component on the EgoSchema/VMB benchmark by replacing tools with weaker or stronger alternatives where applicable. The weaker and stronger LLMs (GPT-3.5 and GPT-4o) showed changes of -4.2/-4.6 and +3.8/+4.3, respectively. For cluster models (CLIP-S and CLIP-L), the changes were -1.2/-0.9 and +0.7/+1.0. Stronger tracker and captioner tools (BotSort and GPT-4o) contributed +0.5/+1.5 and +2.0/+2.6, respectively. In main paper, we use GPT4, CLIP-B, EgoSTARK as the tools. As shown below, stronger tools improved performance, with the LLM and captioner having the most significant impact.

Method	Short	Medium	Long
Split by temporal window	48.5	44.5	42.2
Cluster by location	49.1	47.2	45.6

Table 5. Comparison of performance (%) between temporal window segmentation and location-based clustering across different video lengths (Short, Medium, and Long).

Tools	LLM	Tracker	Captioner	Cluster
Weaker	-4.2/-4.6	-	-	-1.2/-0.9
Stronger	+3.8/+4.3	+0.5/+1.5	+2.3/+2.8	+0.7/+1.0

Table 6. Ablation of each component in our pipeline on the EgoSchema/VMB benchmark by replacing tools with weaker or stronger alternatives.

10. Video Mind Palace Benchmark

10.1. Query creation

To construct a robust set of VideoQA queries for the proposed Video MindPalace Benchmark, we employed a systematic pipeline combining LLM-generated questions with human verification to ensure quality and consistency. From each video, we first extract keyframes sampled at 1 frame per second (1 fps) to succinctly represent the content. For each keyframe, we generate a descriptive caption using GPT-4 and provide a detailed textual description of detected objects, including their IDs and bounding box coordinates. These inputs are fed into GPT-4 to generate diverse reasoning questions requiring spatial, temporal, and layout-aware understanding, such as: "Which object is to the left of the dining table?" (spatial), "What event happens immediately after the person enters the kitchen?" (temporal), and "How are the sofa and coffee table arranged in relation to the TV?" (layout-aware). Each question is accompanied by five answer options, including one correct answer and four distractors designed to challenge reasoning abilities. For open-ended queries, we prompt GPT-4 to generate detailed captions for each keyframe and compile these captions to summarize the actions and activities performed by the individ-

Figure 5. More qualitative results of VideoMindPalace on the VMB benchmark, showcasing examples for each question type. To demonstrate how VideoMindPalace effectively answers these questions, we leverage GPT-4 to pinpoint specific graph components that provide the necessary information for accurate responses.

ual throughout the video. To ensure accuracy, all questions and answers undergo rigorous review by human annotators, who validate correctness, correct errors, and refine phrasing for clarity and consistency. This human-in-the-loop step is critical to maintaining high-quality questions across the benchmark. The finalized questions are then tagged with metadata, such as reasoning type, video length, and associated video segments, and compiled into the Video MindPalace Benchmark. This process ensures a comprehensive and reliable set of VideoQA queries for evaluating video understanding models. To generate challenging distractors, we provide GPT-4 with keyframes and detailed tracking data (bounding boxes, labels, IDs) to identify nearby objects as potential distractors. To address tracking errors, annotators then reviews the distractors for accuracy and difficulty, ensuring at least two nearby objects are included.

10.2. Statistics

We construct our benchmark using 200 videos sourced from the EPIC-KITCHENS and Ego-4D datasets, both of which consist of long, unscripted egocentric recordings capturing participants performing daily activities in various environments. On average, the selected videos are 11 minutes in length. Specifically, 68 videos are categorized as short (less than 3 minutes), 85 videos as medium-length (3 to 10 minutes), 35 as long (10 to 30 minutes), and 12 as very long

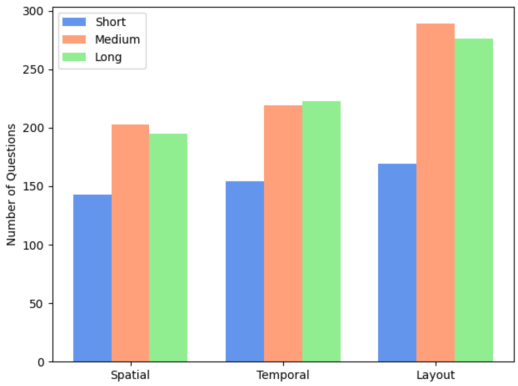

Figure 6. Query Distribution by Video length and Reasoning Categories.

(over 30 minutes). Each video length category includes between 100 and 300 questions, spanning all three types of queries, resulting in a total of approximately 1,800 questions in the benchmark. For a detailed distribution of query types by video duration and reasoning category, please refer to Fig 6.