Graph-to-Frame RAG: Visual-Space Knowledge Fusion for Training-Free and Auditable Video Reasoning

Supplementary Material

Table 1. **Robustness to graph corruption on Qwen2.5-VL-7B.** We corrupt the retrieved subgraph before rendering the reasoning frame. Accuracy (%) is reported; the default row uses our standard G2F-RAG without corruption. G2F-RAG shows graceful degradation and remains above the no-RAG baseline even at ϵ=40%.

Dataset	Default	ϵ=10%	ϵ=20%	ϵ=40%	Baseline
MLVU	**73.4**	72.8	71.6	69.3	68.8
WildVideo	**55.4**	54.9	53.8	51.6	51.3
VideoMME	**70.6**	70.0	68.9	66.7	65.1

1. Study of G2F-RAG's Robustness

Real-world pipelines inevitably encounter noisy or partially incorrect graphs. We therefore evaluate the robustness of G2F-RAG under controlled graph corruption and report the natural error rate of our offline graph construction. Our goal is to verify that the graph remains auxiliary and that the LMM prioritizes video evidence; when noise is detected or the selected subgraph exceeds budget, the hierarchical mechanism routes to the video-only path or retries with a tighter subgraph, preserving easy-case performance.

Protocol. We evaluate Qwen2.5-VL-7B + G2F-RAG on MLVU, WildVideo, and VideoMME (w/o sub). For each test sample, we first obtain the retrieved subgraph and then inject corruption at three budgets $\epsilon \in 10\%, 20\%, 40\%$ while keeping retrieval fixed. Corruption types include node drop (remove entities/events), edge rewire (shuffle causal/topological links), and attribute perturbation (short text edits on roles/affordances or concept snippets). The rendering agent turns the corrupted subgraph into the same single reasoning frame appended at the end. Routing and fallback are enabled unless otherwise noted.

Findings. Table 1 shows a smooth accuracy decline as corruption grows. With ϵ=10%, the average drop is within 0.6% across datasets. At ϵ=20%, the loss remains modest ($\sim$1.1–1.8%), and even at ϵ=40% G2F-RAG stays competitive, remaining above the no-RAG baseline (MLVU 68.8%, WildVideo 51.3%, VideoMME 65.1%). We attribute the resilience to three design choices: visual-space delivery that limits text-load amplification, a compact single-frame budget that bounds noise exposure, and the hierarchical routing with fallback that defers to video-only answers when uncertainty is high or the subgraph exceeds budget. In diagnostic runs with routing disabled, the ϵ=40% setting drops further (e.g., MLVU 67.3% and VideoMME 65.9%), confirming that difficulty-aware routing is an essential guardrail.

Table 2. **Observed graph issues in practice and their impact (Qwen2.5-VL-7B + G2F-RAG).** Rates are measured over 300 query instances; ΔAcc is the average accuracy change when the issue occurs, with routing and fallback enabled. The small deltas indicate that graphs function as auxiliary signals and that the system prefers video evidence when uncertainty is detected.

Issue Type	Rate (%)	ΔAcc (%)
Empty or underspecified subgraph	3.7	-0.4
Wrong entity link (merge/split error)	4.3	-0.7
Causal inconsistency (edge rewire)	2.8	-0.9
Missing affordance or concept snippet	5.1	-0.6
Any issue (total)	9.6	-0.7

Natural error rate of the offline graph. We further audit the uncorrupted pipeline by sampling 300 queries across 120 videos and manually labeling subgraph issues at the point of rendering. We record the frequency of empty/underspecified subgraphs, wrong entity links, causal inconsistencies, and missing affordances. We also measure the impact on accuracy relative to the default configuration when routing and fallback are enabled.

Discussion. Table 2 shows that graph imperfections are infrequent and have limited impact under our default settings. When the agent detects low confidence or inconsistency, it either tightens the subgraph or falls back to the video-only path, which bounds the effect of noise. Qualitatively, we also observe that contradictory edges are often ignored by the LMM when the visual stream provides strong evidence, consistent with our "video evidence primacy" prompting. Together with the corruption study, these results demonstrate that the proposed graph-to-frame fusion is robust: the diagram is a compact, auditable hint rather than a brittle dependency, and the hierarchical mechanism maintains performance even when the auxiliary graph is imperfect.

2. Effect of Different API Versions

To examine sensitivity to the choice of agent backends, we replace the offline graph-construction and the on-

Table 3. **Effect of agent API backends (Qwen2.5-VL-7B + G2F-RAG).** Each row replaces both the offline graph-construction agent and the online routing–retrieval agent as a pair. Scores are accuracy (%) with three significant digits. Small but consistent gains appear with stronger agent backends, most notably on the knowledge-intensive WildVideo.

Agent Pair (Offline / Online)	MLVU	WildVideo	VideoMME
GPT-4o / GPT-4o-mini	73.4	55.4	70.6
Gemini 2.5 Pro / Gemini 2.5 Flash	73.8	56.1	71.0
GPT-5 / GPT-5-mini	**74.5**	**56.8**	**71.5**

line routing–retrieval agents with stronger API models while keeping the LMM backbone, prompts, decoding, and fusion policy unchanged. The default uses GPT-4o for graph construction (API `gpt-4o-2024-05-13`) and GPT-4o-mini for routing and subgraph extraction (API `gpt-4o-mini-2024-07-18`). We evaluate two stronger pairs: Gemini 2.5 Pro for graph construction with Gemini 2.5 Flash for routing, and GPT-5 for graph construction with GPT-5-mini for routing (APIs `gemini-2.5-pro`, `gemini-2.5-flash`, `gpt-5-2025-08-07`, `gpt-5-mini-2025-08-07`). All experiments use Qwen2.5-VL-7B with `End-1` single-frame fusion and the same retrieval budget.

The results in Table 3 show that G2F-RAG is robust across API providers and benefits moderately from stronger agent backends. Gains are most visible on WildVideo, where improved routing precision and cleaner subgraphs translate into about 0.7–1.4% above the default, while MLVU and VideoMME (w/o sub) improve by roughly 0.4–1.1%. The effect size is modest, which indicates that performance is primarily governed by the visual-space fusion with the frozen backbone rather than by any single API choice. This aligns with our design: the graph remains auxiliary, the reasoning frame is compact, and difficulty-aware routing protects easy cases. In practice, the choice among these backends can be guided by cost and latency constraints, with G2F-RAG delivering consistent accuracy regardless of provider.

3. Prompts

Graph-Construction Agent. Figure 1 presents the prompt designed for the graph-construction agent to build a problem-agnostic, once-for-all knowledge graph for the given video that can be reused across questions.

Orchestration Agent. Figure 2 presents the prompt designed for the orchestration agent to decide whether to answer directly with the base LMM or to invoke graph-to-frame fusion, and produce a minimal plan for retrieval and rendering.

Retrieval Agent. Figure 3 presents the prompt designed for the retrieval agent to select a minimal sufficient subgraph that supports answering the question and can be rendered as a single reasoning frame.

Video LMMs. Figure 4 presents the prompt designed for the LMM to answer the question.

Graph-Construction Agent (Offline DHKG) Prompt

You are a video world-modeling agent. Build a problem-agnostic, once-for-all knowledge graph for the given video that can be reused across questions. Output a single valid JSON object in English. Do not include explanations outside JSON.

Goal. Produce a dual-view graph that captures (i) event causality and intent (why/how) and (ii) scene objects, affordances, and spatial layout (what/where). Connect the two views with dense cross-links. Avoid timestamps; when evidence is needed, you may reference frame indices only.

General rules
1) Be comprehensive yet precise; avoid hallucination. If uncertain, omit the field instead of guessing.
2) Every node must contain a "class" field among {"entity","action","event","object","location","concept_knowledge"}.
3) Use short, stable ids (e.g., "E1", "A3", "O2", "L1"); maintain uniqueness and reuse ids consistently across sections.
4) Prefer grounded facts; when external world knowledge is used, mark "source": "external" and provide "citation" text. Keep external knowledge minimal and generic.
5) Add "confidence" in [0,1] and optional "frame_idx_evidence":[...] for verifiability (indices only, no timestamps).

Required JSON keys (skeleton). { "video_id": ..., "DHKG": { "eventCausalGraph": { "entities": [...], "actions": [...], "events": [...], "edges": [...] }, "sceneAffordanceGraph": { "objects": [...], "locations": [...], "concept_knowledge": [...] }, "crossLinks": [...] } }

Event-Causal Graph (ECG)
1) entities: people/agents with roles from visual evidence (e.g., "chef", "customer"). Fields: id, name, role, attributes, class="entity", confidence, frame_idx_evidence.
2) actions: fine-grained acts with intent, preconditions, postconditions. Fields: id, verb, subject, objects, intent, preconditions, postconditions, class="action", confidence, frame_idx_evidence.
3) events: higher-level groupings of actions (e.g., "prepare drink"). Fields: id, title, actions_included, summary, class = "event".
4) edges: typed relations among {entities,actions,events}. Allowed types include causal_link, enables, inhibits, precedes, uses, produces. Each edge: src_id, dst_id, relation, rationale (1–2 short phrases).
Scene-Affordance Graph (SAG)
1) objects: visually grounded items with functional uses. Fields: id, name, attributes, physical_properties{is_static, is_movable}, affordances[...], class = "object".
2) locations: functional areas and connectivity. Fields: id, name, contains[object_ids], connectivity[{to, traversal_type, qualitative_distance}], class = "location".
3) concept_knowledge: concise world knowledge relevant to objects/events (e.g., "a kettle heats water"). Fields: id, type, statement, source{"video", "external"}, citation (if external), class = "concept_knowledge".
Cross-links (ECG $\leftrightarrow$ SAG)
Each link ties narrative nodes to the environment. Use relation in {"occurs_in", "involves", "uses_object", "at_location", "supported_by_concept"}. Include src_id, dst_id, relation, rationale.

Quality and safety checks before returning JSON.
1) All ids referenced by edges and crosslinks must exist.
2) No free text outside JSON; no timestamps; frame indices only if evidence is provided.
3) Roles, intents, affordances, and causal links must be consistent with visible cues or clearly marked as external with citation.
4) Keep names concise and canonical; deduplicate near-duplicates.
5) Ensure the JSON parses.

Now build the DHKG for the video.

Figure 1. Prompt for the offline graph-construction agent (DHKG).

Orchestration Agent (Online Routing) Prompt

You are a decision controller for difficulty-aware routing in G2F-RAG. Given a user question q, a video V, and its offline DHKG $\mathcal{G}$, decide whether to answer directly with the base LMM or to invoke graph-to-frame fusion, and produce a minimal plan for retrieval and rendering. Output a single valid JSON object in English. Do not include explanations outside JSON.

Objective. Protect easy cases and activate fusion only when beneficial. No timestamps; if evidence is referenced, use frame indices only.

Decision rule. Estimate proxy utilities in [0,1] for two strategies: direct video reasoning (Base) and graph-to-frame fusion (G2F). Compute $\Delta U = \widehat{U}\_\mathrm{G2F} - \widehat{U}\_\mathrm{Base}$ and compare to a threshold $\tau \in [0, 1]$. Return "hard" if $\Delta U \geq \tau$, otherwise "easy". Provide a confidence score in [0,1]. If confidence is low or the planned selection exceeds budget, set "fallback": true and choose the easy path or reduce the plan conservatively.

Internal factors to consider (do not output): locality of visual evidence, cross-scene dependency, triggers for external knowledge, estimated reasoning steps, ambiguity/noise.

Plan requirements. When routing is "hard", produce a compact plan that:
1) selects a minimal sufficient subgraph from $\mathcal{G}$ linking question anchors (entities, actions, objects, locations) to candidate answers via causal or spatial links;
2) respects strict budgets and keeps visual tokens small;
3) specifies one reasoning frame appended at the end with a minimal diagram style and low text density;
4) lists required anchors by id and forbids speculative content.

Required JSON keys (skeleton). { "route_decision": "easy" or "hard", "scores": { "U_G2F": float, "U_Base": float, "DeltaU": float, "tau": float }, "confidence": float, "fallback": true|false, "plan": { "anchors": {"must_include_ids":[...]}, "selection_policy": "minimal_sufficient", "budgets": {"max_nodes": int, "max_edges": int, "max_visual_tokens": int}, "include_types": ["entity", "action", "event", "object", "location", "concept_knowledge"], "style": {"frame_count":1, "placement": "end", "diagram": "minimal", "text_density": "low"} } }

Quality checks before returning JSON.
1) All ids referenced in "anchors" must exist in $\mathcal{G}$; no invented nodes.
2) The plan must not exceed budgets; otherwise set "fallback": true and shrink the plan.
3) No free text outside JSON; no timestamps.

Return only the final JSON object containing the decision, scores, confidence, fallback flag, and the compact plan.

Figure 2. Prompt for the online orchestration agent (difficulty-aware routing and planning).

Retrieval Agent (Subgraph Selection) Prompt

You are a subgraph selection agent in G2F-RAG. Given a user question q, the offline DHKG $\mathcal{G}$, and an orchestration plan, select a minimal sufficient subgraph that supports answering the question and can be rendered as a single reasoning frame. Output one valid JSON object in English. Do not include explanations outside JSON.

Inputs. { "question": q, "DHKG": ..., "plan": { "anchors": {"must_include_ids": [...]}, "selection_policy": "minimal_sufficient", "budgets": {"max_nodes": K_n, "max_edges": K_e, "max_visual_tokens": K_vt}, "include_types": ["entity", "action", "event", "object", "location", "concept_knowledge"], "style": {"frame_count": 1,"placement": "end", "diagram": "minimal", "text_density": "low"} } }

Selection rules.
1) Use only ids that exist in $\mathcal{G}$; do not invent nodes or edges.
2) Cover all anchors and connect them to candidate answers using the smallest set of edges that preserves causal and spatial dependencies; prefer ECG chains cross-linked to SAG objects/locations.
3) Keep the subgraph connected; remove dangling or redundant nodes.
4) Include concept_knowledge only if needed to resolve affordance/function/commonsense ambiguity.
5) Enforce budgets strictly. If exceeded, prune by importance: shortest causal paths, required cross-links, high-confidence nodes first; drop low-utility peripherals.
6) No timestamps; frame indices may be included only as evidence.
7) Keep labels concise to control visual tokens.

Required JSON keys (skeleton). { "selected_subgraph": { "nodes": [{"id": "string", "class": "entity|action|event|object|location|concept_knowledge", "label": "short_name", "confidence": 0..1, "frame_idx_evidence": [int, ...]} ...], "edges": [{"src": "id", "dst": "id", "relation": "causal_link|uses_object|occurs_in|enables|precedes", "note": "short_phrase"} ...], "coverage": { "question_anchors_covered": ["id", ...], "paths": [["id", "id",...], ...] }, "stats": {"num_nodes": int, "num_edges": int} }, "render_spec": { "diagram": "minimal", "layout": "topology_plus_causality", "labels": {"max_chars": 18, "abbrev": true}, "icons": true, "text_density": "low", "frame_count": 1, "placement": "end", "legend":[{"id": "id", "short_label": "string"} , ...] }, "confidence": 0..1, "checks": { "within_budget": true|false, "ids_exist": true|false, "no_timestamps": true }, "fallback_reduced": true|false }

Quality checks before returning JSON.
1) All ids in nodes, edges, coverage, and legend must exist in $\mathcal{G}$.
2) The subgraph must be connected and within budgets; if not, set "fallback_reduced": true and prune while preserving anchors and at least one causal path.
3) Keep text minimal and unambiguous to control visual tokens.

Return only the final JSON object containing the selected subgraph, rendering specification, confidence, and checks.

Figure 3. Prompt for the retrieval agent (minimal sufficient subgraph selection).

LMM Inference Prompt

You are a video question answering model. The input is a video V where the last frame is a compact reasoning frame that visualizes retrieved knowledge. Answer the question primarily from the original video frames; treat the reasoning frame as auxiliary context to bridge missing links. In any conflict, follow the visual evidence in the video. Provide a concise, factual answer.

Video: `<LOCAL_VIDEO_PATH>`
Question: `<USER_QUESTION>`

Figure 4. Prompt for the final video LMM. The reasoning frame is auxiliary; the original video frames are primary.