ADAPTIVE RETRIEVAL AND VERIFICATION: A SHARED AGENTIC CONTROLLER FOR CYBERSECURITY RAG AND GRAPHRAG

Anonymous authors
Paper under double-blind review

ABSTRACT

Cybersecurity question answering often requires connecting evidence across heterogeneous sources and reasoning over multiple entities and relationships. Conventional Retrieval-Augmented Generation (RAG) retrieves semantically similar evidence as a flat context, limiting its ability to support cross-domain and multi-hop questions. We propose an agentic, multi-layer cybersecurity GraphRAG framework that integrates heterogeneous repositories through explicit bridge relationships and combines semantic retrieval with intent-aware graph traversal and re-ranking. A shared, bounded controller is instantiated over both flat RAG and GraphRAG to plan evidence requirements, assess retrieval sufficiency, perform targeted retries, and verify generated claims before returning a complete answer, partial answer, or abstention. We evaluate Basic RAG, Agentic RAG, GraphRAG, and Agentic GraphRAG on a 40-question cybersecurity benchmark. Agentic GraphRAG achieves the highest overall answer-quality score of $4.04/5$, compared with 3.82 for GraphRAG, 3.05 for Agentic RAG, and 3.00 for Basic RAG. Relative to GraphRAG, it provides numerical improvements of 5.8% in overall answer quality, 20.5% in pooled identifier recall, and 30.9% on multi-hop questions. These gains increase mean latency from 4.23 to 30.96 seconds. The results indicate that agentic control is most effective when supported by structured, relational evidence, while revealing a substantial quality–efficiency trade-off.

1 INTRODUCTION

Cybersecurity investigations require connecting vulnerabilities, affected products, attack techniques, malware, threat actors, and defensive controls across independently maintained sources such as MITRE ATT&CK (Strom et al., 2018), the National Vulnerability Database (National Institute of Standards and Technology, 2024), Common Weakness Enumeration (MITRE Corporation, 2024), and the Known Exploited Vulnerabilities catalog (Cybersecurity and Infrastructure Security Agency, 2024). Their heterogeneous schemas, identifiers, and relation types make cross-source and multi-hop reasoning difficult. RAG grounds language models in external knowledge (Lewis et al., 2020; Gao et al., 2023b), but conventional RAG retrieves semantically similar passages as a flat context and does not explicitly preserve relationships among cybersecurity entities. GraphRAG uses knowledge graphs to support relational retrieval and reasoning (Hogan et al., 2021; Edge et al., 2024); however, one-shot RAG and GraphRAG cannot adapt when their initial evidence is incomplete. Existing agentic RAG methods introduce iterative retrieval and verification, but primarily address general-domain question answering or technical-literature analysis (Zhang et al., 2025; Singh, 2026). Applying the same agentic controller to both flat and graph retrieval over heterogeneous cybersecurity knowledge remains underexplored. We propose an agentic, multi-layer cybersecurity GraphRAG framework. Repository-specific layers preserve source-specific entity and relation schemas, while explicit bridge relationships connect vulnerabilities, software, attack techniques, malware, atomic procedures, and threat-intelligence reports. A hybrid retriever combines semantic search, graph traversal, bridge-aware expansion, and re-ranking to construct traceable reasoning subgraphs. A shared, bounded controller operates over either flat RAG or GraphRAG: it plans the required evidence, assesses retrieval sufficiency, performs targeted retries, and verifies generated claims before returning a complete answer, partial answer, or abstention.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We evaluate Basic RAG, Agentic RAG, GraphRAG, and Agentic GraphRAG on a manually verified 40-question cybersecurity benchmark spanning exact lookup, cross-domain integration, attack relationships, metadata retrieval, and multi-hop inference. This four-system design characterizes the observed contributions of graph-structured retrieval and agentic control within the evaluated cybersecurity setting.

Our main contributions are:

- A multi-layer cybersecurity knowledge graph with provenance-aware cross-source bridges, together with a hybrid GraphRAG retriever combining identifier resolution, semantic seeding, relation-aware traversal, and graph-based re-ranking.
- A shared, bounded controller for adaptive retrieval, evidence-sufficiency assessment, claim verification, revision, and abstention over both flat RAG and GraphRAG.
- A four-system evaluation that characterizes the interaction between graph-structured retrieval and bounded agentic control across cybersecurity question types.

2 Related Work

Prior work extends one-shot RAG through query reformulation (Ma et al., 2023), interleaved retrieval and reasoning (Trivedi et al., 2023), active retrieval during generation (Jiang et al., 2023), reasoning-guided actions (Yao et al., 2022), and complexity-aware strategy selection (Jeong et al., 2024). Other methods assess retrieval or generation quality: Self-RAG uses self-reflection (Asai et al., 2023), Corrective RAG triggers corrective retrieval (Yan et al., 2024), and Chain-of-Verification and RARR verify or revise generated claims (Dhuliawala et al., 2023; Gao et al., 2023a). Our controller combines planning, evidence-sufficiency assessment, bounded retrieval, and claim verification in one workflow instantiated over both flat RAG and GraphRAG. Recent agentic RAG systems pursue complementary objectives. TeaRAG improves the token efficiency of multi-round retrieval through graph-based evidence compression and optimized reasoning (Zhang et al., 2025). TechRAG combines multimodal retrieval, graph-guided expansion, evidence gating, and specialized agents for technical-literature analysis (Singh, 2026). In contrast, our framework uses a shared, bounded controller over both retrieval architectures and operates on an explicit multi-layer cybersecurity knowledge graph connecting vulnerabilities, ATT&CK techniques, software, malware, atomic procedures, and threat-intelligence reports. We do not claim novelty for the individual control operations. Our contribution is their integration within a common controller and their four-system evaluation-Basic RAG, Agentic RAG, GraphRAG, and Agentic GraphRAG-over heterogeneous cybersecurity knowledge. This design characterizes how graph-structured retrieval and adaptive control interact in cross-source and multi-hop cybersecurity question answering.

3 Multi-Layer GraphRAG and Shared Agentic Controller

3.1 Framework Overview

Our framework combines a persistent multi-layer cybersecurity knowledge graph, a hybrid GraphRAG retriever, and a shared bounded controller. The knowledge graph integrates heterogeneous repositories through cross-layer bridge relations, while the GraphRAG retriever combines semantic seeding, graph traversal, and re-ranking to construct query-specific evidence subgraphs.

As shown in Figure 1, we evaluate four configurations. Basic RAG and GraphRAG perform one retrieval followed by single-pass generation. Agentic RAG and Agentic GraphRAG use the same controller for evidence planning, retrieval-sufficiency assessment, query reformulation, answer generation, and claim verification. They differ only in their retriever adapters: Agentic RAG can reformulate a semantic query, whereas Agentic GraphRAG can additionally request a maximum traversal depth and evaluate retrieved nodes, edges, and paths. The controller may issue at most three retrieval calls before returning a complete answer, a partial answer, or an abstention.

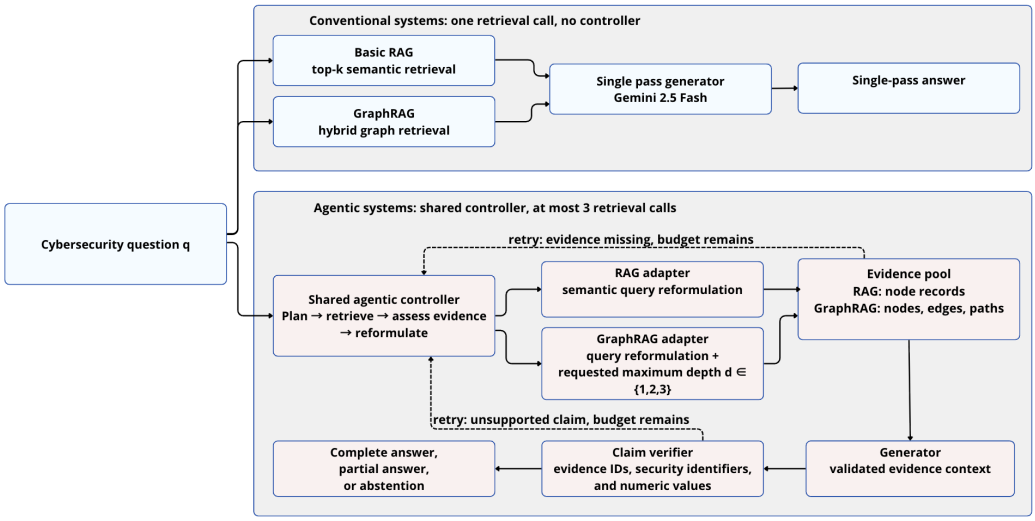

Figure 1: Overview of the four evaluated systems. Basic RAG and GraphRAG perform one-shot retrieval and generation. Agentic RAG and Agentic GraphRAG share the same bounded controller but use architecture-specific retriever adapters. Dashed arrows indicate evidence and claim-triggered retries within a budget of at most three retrieval calls.

3.2 Knowledge Graph Construction

Cybersecurity evidence is distributed across repositories that describe complementary aspects of an attack lifecycle. MITRE ATT&CK captures adversarial tactics, techniques, software, groups, and mitigations; the National Vulnerability Database (NVD) records vulnerabilities, affected products, weaknesses, and severity information; Atomic Red Team provides executable adversary-emulation tests and behavioral observables; and Cyber Threat Intelligence (CTI) reports describe real-world campaigns and threat activity. We integrate these sources into a unified heterogeneous knowledge graph while preserving each repository's source-specific entity types, identifiers, attributes, and relation meanings. The resulting structure is modeled as a directed, typed graph $G = (V, E)$. Each node $v \in V$ is assigned an entity type $\tau(v) \in \mathcal{T}$, such as *technique*, *cve*, *software*, *group*, *mitigation*, *atomic_test*, or *observable*. Each edge $e = (u, r, v) \in E$ is associated with a relation type $r \in \mathcal{R}$, such as *uses*, *mitigates*, *associated_technique*, or *subtechnique_of*. Each node also carries a layer label

$$\ell(v) \in \{\text{ATTACK}, \text{VULN}, \text{ATOMIC}, \text{REPORT}\},$$

which identifies its source repository and distinguishes intra-layer from cross-layer relations.

Detailed diagrams of the multi-layer knowledge graph and hybrid retrieval pipeline are provided in Appendix A. Each repository is parsed into a semantic layer and normalized to a common schema. Entities are unified using canonical identifiers, while duplicate relations are removed according to their $(\text{source}, \text{relation}, \text{target})$ triples. Selected nodes are encoded using a pretrained sentence encoder, yielding $\mathbf{e}_v \in \mathbb{R}^d$. The graph and vector representations form the persistent knowledge base used during retrieval.

3.3 Bridge Construction

To enable traversal across repositories, we introduce *bridge edges* connecting semantically related entities in different semantic layers. Formally, a bridge edge is defined as $(u, r, v) \in E$ satisfying $\ell(u) \neq \ell(v)$, where $\ell(\cdot)$ denotes the repository layer of a node. Bridge edges are constructed through cross-layer entity resolution. Between the vulnerability and ATT&CK layers, software and product entities referenced by a CVE are matched to ATT&CK software using the lexical similarity function $\text{sim}_{\text{str}}(\cdot, \cdot)$. Entity pairs whose similarity exceeds a threshold τ_b are connected through a

related_software relation. When matched ATT&CK software is associated with one or more techniques, the framework adds an inferred *associated_technique* bridge between the CVE and those techniques. This bridge represents an indirect association through matched software rather than confirmed exploitation provenance. A similar entity-resolution procedure connects malware and tool mentions extracted from CTI reports to their corresponding ATT&CK software entities. Automatically generated bridges are supplemented with a small set of manually verified links. These relations enable retrieval paths to cross repository boundaries and aggregate vulnerability, technique, software, and threat-intelligence evidence.

3.4 HYBRID RETRIEVAL

Given a question q, the retriever initializes a seed set $S(q)$. An exact seed is a graph node whose canonical identifier occurs in the question; when no identifier is detected, semantic seeds are the k nodes with the highest embedding similarity to the question. As illustrated in Figure 3, the proposed pipeline consists of query analysis, seed initialization, weighted graph traversal, graph-aware reranking, and reasoning-subgraph construction.

The pipeline first analyzes q to identify its reasoning intent and whether it spans multiple cybersecurity domains. The detected intent determines the traversal configuration

$$C(q) = (\mathcal{R}_{\text{allow}}, d_{\max}),$$

where $\mathcal{R}_{\text{allow}}$ specifies the relation types that may be followed and $d_{\max}$ defines the maximum traversal depth. This configuration limits unnecessary expansion for direct lookup questions while allowing deeper traversal for multi-hop queries.

Seed initialization depends on whether the question contains resolvable cybersecurity identifiers. Let $I(q)$ denote the identifiers extracted from q, and let

$$V_I(q) = \{v \in V : \text{id}(v) \in I(q)\}$$

denote the corresponding graph nodes. The seed set is defined as

$$S(q) = \begin{cases} V_I(q), & V_I(q) \neq \emptyset, \\ \arg\,\text{top-k}_{v \in V}\, \text{sim}(\mathbf{e}_q, \mathbf{e}_v), & \text{otherwise.} \end{cases} \tag{1}$$

where $\mathbf{e}_q$ is the question embedding and $\text{sim}(\cdot,\cdot)$ denotes cosine similarity. This rule prioritizes exact identifier matches while retaining semantic retrieval for descriptive natural-language questions. Starting from $S(q)$, the framework performs bounded weighted breadth-first traversal. A node v reached from node u through relation r at depth d receives the propagated score

$$s(v) = s(u)\, w(r)\, \lambda\, \beta(u,v)\, \iota(r,q). \tag{2}$$

where $w(r)$ is a relation-specific weight, $\lambda \in (0,1)$ applies a constant penalty at each traversal hop, so that a path of length d receives an overall depth factor of λ^d, $\beta(u,v) > 1$ rewards traversal across bridge edges satisfying $\ell(u) \neq \ell(v)$, and $\iota(r,q)$ measures alignment between relation r and the detected query intent. Low-scoring candidates are pruned, while an evidence-preserving expansion step retains direct attributes of exact-identifier seeds, such as the severity metadata associated with a CVE. The remaining candidates are re-ranked using

$$s'(v) = s(v) \cdot \psi_{\text{intent}}(v) \cdot \psi_{\text{bridge}}(v) \cdot \psi_{\text{kw}}(v) \cdot \psi_{\text{sem}}(v) \cdot \psi_{\text{topic}}(v), \tag{3}$$

which combines intent alignment, bridge-path participation, lexical overlap, semantic similarity, and topic relevance. The highest-ranked candidates are then assembled into a bounded reasoning subgraph $G_q = (V_q, E_q) \subseteq G$ using type-aware selection, entity-diversity constraints, and connectivity pruning. The node budget is adapted to the estimated complexity of the question. The next subsection formalizes the reasoning paths preserved within G_q.

3.5 MULTI-HOP REASONING

The retrieval procedure returns a structured reasoning subgraph rather than an unordered collection of relevant nodes. For every node retained in the final subgraph, the framework preserves at least one traversal path connecting that node to a seed entity. A reasoning path is defined as

$$p = (v_0, r_1, v_1, \ldots, r_k, v_k), \qquad v_0 \in S(q), \tag{4}$$

where each relation r_i connects v_{i-1} to v_i, and v_k is an evidence node retained in the query-specific subgraph. Because candidate scores are propagated along traversed edges according to equation 2, every node retained in G_q is associated with at least one preserved path from a seed.

Let $\mathcal{P}_q = \{p_1, \ldots, p_m\}$ denote the set of m seed-anchored paths retained after traversal, re-ranking, and pruning. For each path p_i, let $V(p_i)$ and $E(p_i)$ denote its node and edge sets, respectively. The reasoning subgraph is then constructed as

$$V_q = \bigcup_{p_i \in \mathcal{P}_q} V(p_i), \qquad E_q = \bigcup_{p_i \in \mathcal{P}_q} E(p_i), \qquad G_q = (V_q, E_q), \tag{5}$$

subject to the node-budget and connectivity constraints. This construction preserves the relationships among retrieved facts instead of presenting them as isolated evidence items. Bridge edges allow retained paths to cross semantic layers. For example, a path initialized from a vulnerability may traverse associated software or an ATT&CK technique and reach a mitigation, threat report, or other connected entity. Such evidence may not be recovered through retrieval restricted to a single repository or through independent similarity-based lookup.

In this framework, multi-hop reasoning is implemented through relation-aware retrieval rather than a separate symbolic inference module. The graph topology determines the available cross-layer transitions, while the retrieval configuration determines the relevant relation types and traversal depth. The resulting subgraph provides the generator with both the retrieved evidence and the explicit relations connecting it. In contrast, conventional vector-based RAG retrieves semantically similar evidence items without explicitly representing their connecting paths, leaving the generator to infer relationships during generation. GraphRAG instead constructs a connected, query-specific evidence subgraph before generation, providing structured context and explicit retrieval provenance for the answer-generation stage.

3.6 Answer Generation

For GraphRAG, the reasoning subgraph G_q is serialized into a structured context containing seed entities, multi-hop evidence, and the relational paths connecting them. Basic RAG instead provides a flat collection of semantically retrieved evidence. In the non-agentic comparison, both systems use the same Gemini 2.5 Flash generator and generation settings, thereby isolating the effect of the retrieval architecture.

The agentic configurations use the same underlying RAG and GraphRAG retrievers, augmented by the shared controller described in Section 3.7. The controller assesses evidence before generation and verifies the resulting claims afterward, triggering additional retrieval, revision, or abstention when support is insufficient.

3.7 Shared Agentic Controller

We wrap both one-shot retrievers with the same bounded controller, using architecture-specific adapters rather than separate agents. For each question, the controller maintains a retrieval plan, request history, accumulated evidence, and a budget of at most $B = 3$ retrieval calls. State is reset between questions, and reference answers, gold identifiers, and evaluation scores are never available to the controller. The RAG adapter invokes the Basic RAG top-k cosine retriever ($k = 12$) and allows only semantic-query reformulation. The GraphRAG adapter invokes the hybrid graph retriever, represents its output as node, edge, and path evidence, and permits query reformulation and traversal depths in $\{1, 2, 3\}$. Seed resolution, intent detection, relation selection, graph expansion, and re-ranking remain functions of the underlying retriever. Each agentic system begins with the original question and default retrieval settings, making its first attempt identical to its one-shot counterpart. The controller first identifies the facts and relationships required by the question. After each retrieval, it classifies them as supported, partially supported, or missing and issues a novel targeted request when evidence is incomplete and budget remains. The generator receives only evidence associated with supported or partially supported requirements. Its draft is then decomposed

into claims and verified against the accumulated evidence. Unsupported material claims trigger further retrieval when possible; otherwise, they are revised or removed. The controller returns a complete answer, a supported partial answer, or an abstention when no supported claim remains. Evidence references are validated programmatically: cited identifiers must exist in the retrieved pool, supported claims require valid evidence, and security identifiers and material numerical values must occur in the question or retrieved context. Relational claims require evidence explicitly stating the relationship rather than connectivity or co-occurrence alone. The process terminates when all claims are supported, the budget is exhausted, no novel request is available, or no supported evidence remains. Full pseudocode and recorded trace fields are provided in Appendix A.1.

4 Experimental Setup

4.1 Datasets

The experimental knowledge base integrates four complementary cybersecurity repositories: MITRE ATT&CK (Strom et al., 2018), the National Vulnerability Database (NVD) National Institute of Standards and Technology (2024), Atomic Red Team Red Canary (2026), and structured Cyber Threat Intelligence (CTI) reports Trend Micro (2022); Google Cloud (2026); ESET (2026); Cybersecurity and Infrastructure Security Agency (CISA) (2021a;b); Kaspersky (2026). Together, these sources provide information on adversarial tactics and techniques, vulnerabilities and affected products, executable attack procedures, behavioral observables, malware, and incident-level threat activity. Each repository is represented as a semantic layer in the heterogeneous knowledge graph described in the Knowledge Graph Construction Section 3.2. After parsing, normalization, entity deduplication, and bridge construction, the resulting knowledge graph contains 38,950 nodes, 88,215 edges, 38 node types, and 72 relation types.

4.2 Benchmark Construction

We constructed a benchmark of 40 cybersecurity questions to evaluate evidence retrieval and multi-hop question answering over the integrated knowledge graph. The questions range from direct entity lookup to cross-layer reasoning requiring evidence from multiple cybersecurity repositories. Each question belongs to one of five semantic categories: *Exact Lookup*, involving the direct retrieval of an entity or attribute (e.g., "What is the CVSS severity score of CVE-2023-23397?"); *Attack Relationships*, covering relations among ATT&CK tactics, techniques, software, threat groups, and mitigations (e.g., "Which ATT&CK techniques are used by APT29?"); *Cross-Domain*, requiring evidence integration across semantic layers (e.g., "Which ATT&CK techniques are associated with CVE-2023-23397?"); *Mixed Metadata*, aggregating attributes from different entity types (e.g., "Which products and vendors are affected by CVE-2023-23397?"); and *Multi-Hop Reasoning*, requiring traversal across multiple relations or layers (e.g., "Which mitigations apply to the attack techniques used to exploit CVE-2023-23397?"). Questions were also labeled according to three reasoning difficulty levels: *easy*, *medium*, and *hard*, reflecting progressively more complex retrieval and reasoning requirements. An *easy* question (e.g., "What does T1059 represent?") requires an exact entity lookup within a single semantic layer. A *medium* question (e.g., "Which ATT&CK techniques are linked to CVE-2026-25760?") requires cross-layer retrieval between vulnerabilities and ATT&CK techniques. A *hard* question (e.g., "Which threat groups are connected to CVE-2026-25760?") requires multi-hop traversal from multiple entity types. For each benchmark question, we manually verified a reference answer together with the supporting entities and relations in the knowledge graph, enabling evaluation of both retrieval quality and generated-answer quality.

4.3 Systems

We evaluate four systems spanning two retrieval structures (vector and graph) and two control strategies (one-shot and agentic):

- **Basic RAG:** one-shot top-k cosine retrieval ($k = 12$) with flat context.
- **GraphRAG:** one-shot hybrid retrieval with exact and semantic seeding, graph traversal, bridge expansion, re-ranking, and reasoning-subgraph construction.

- **Agentic RAG:** Basic RAG augmented with query reformulation, evidence assessment, and claim verification.
- **Agentic GraphRAG:** GraphRAG augmented with the same controller, including query reformulation, traversal-depth adjustment, and verification over nodes, edges, and paths.

The agentic systems share the controller, retrieval budget, stopping policy, and temperature-zero configuration described in Section 3.7. Their first retrieval attempts match their one-shot counterparts. The resulting four-system design characterizes the observed effects of graph-structured retrieval and agentic control. The one-shot systems use the provider's default temperature, whereas the agentic systems use temperature 0; this difference is considered when interpreting comparisons between one-shot and agentic systems. Accordingly, comparisons within the one-shot pair and within the agentic pair hold the generation procedure fixed. Comparisons between a one-shot system and its agentic counterpart measure the complete controller workflow—including adaptive retrieval, evidence assessment, verification, and controller-specific generation—rather than adaptive retrieval alone.

4.4 Evaluation Metrics and Judge Protocol

We evaluate retrieval effectiveness, answer-ID fidelity, and end-to-end answer quality. For retrieved identifiers R and manually verified gold identifiers G, we report Hit@$k = \mathbb{I}[R \cap G \neq \emptyset]$, precision $|R \cap G|/|R|$, recall $|R \cap G|/|G|$, and their harmonic-mean F1. Identifiers are uppercased and stripped of whitespace, with synthetic edge and path identifiers excluded. For agentic systems, we evaluate the first attempt and evidence pooled across all attempts; pooling cannot reduce recall but may lower precision and F1. We apply the same metrics separately to identifiers written in the final answer, with details in Appendix A.2. The independent judge `nvidia/nemotron-3-ultra-550b-a55b` evaluates all 160 saved answers on accuracy, grounding, completeness, relevance, clarity, and hallucination control using a 1–5 scale (Zheng et al., 2023); the overall score is their mean. Under the shared abstention-aware rubric, accuracy and completeness are measured against the reference answer: abstentions receive no credit for omitted facts but may be rewarded for grounding, hallucination control, and clarity, while partial answers receive credit only for correct facts. Results are macro mean $\pm$ population standard deviation over 40 questions and are descriptive because paired significance tests were not conducted.

4.5 Implementation Details

Entities are embedded with `all-MiniLM-L6-v2`. Basic RAG retrieves $k = 12$ nodes by cosine similarity, whereas GraphRAG uses exact-identifier resolution, semantic seeding, weighted traversal, bridge expansion, and re-ranking. Gemini 2.5 Flash generates answers for all systems. The agentic systems use the same retrieval functions as their conventional counterparts, making their first retrieval attempts identical. The shared controller allows at most three retrieval calls and uses Gemini 2.5 Flash at temperature zero for planning, evidence assessment, generation, and verification. Agentic RAG can reformulate its query, while Agentic GraphRAG can additionally request a traversal depth in $\{1, 2, 3\}$. The conventional and agentic systems use different generation prompts and decoding configurations. Comparisons with an agentic counterpart therefore measure the complete controller workflow, including adaptive retrieval, verification, and generation. All saved answers are evaluated by the same judge and abstention-aware rubric described in Section 4.4.

5 Results and Discussion

5.1 End-to-End Answer Quality

We first examine how retrieval structure and agentic control affect the quality of the final answers. Table 1 shows that Agentic GraphRAG achieves the highest overall score, 4.04 ± 0.94, compared with 3.82 ± 0.92 for GraphRAG, 3.05 ± 0.40 for Agentic RAG, and 3.00 ± 0.56 for Basic RAG. Agentic control therefore yields a larger numerical improvement with graph retrieval ($+5.8\%$) than with flat retrieval ($+1.7\%$). Agentic GraphRAG leads in accuracy, completeness, relevance, and clarity, but scores slightly below GraphRAG in grounding and hallucination control. This indicates a coverage–grounding trade-off: its broader answers address more of the requested information

Table 1: End-to-end answer quality across the four systems. Scores are reported on a 1–5 scale as mean ± population standard deviation over 40 questions. Bold indicates the highest mean in each row.

Metric	Basic RAG	Agentic RAG	GraphRAG	Agentic GraphRAG
Accuracy	1.23 ± 0.88	1.20 ± 0.87	3.12 ± 1.81	$\mathbf{3.83 \pm 1.58}$
Grounding	4.55 ± 1.14	$\mathbf{4.70 \pm 0.98}$	4.38 ± 1.07	4.15 ± 1.31
Completeness	1.10 ± 0.62	1.00 ± 0.00	2.48 ± 1.53	$\mathbf{2.92 \pm 1.44}$
Relevance	1.73 ± 0.74	1.75 ± 0.70	3.62 ± 1.70	$\mathbf{4.00 \pm 1.36}$
Clarity	4.78 ± 0.52	4.85 ± 0.36	4.78 ± 0.52	$\mathbf{5.00 \pm 0.00}$
Hallucination Control	4.62 ± 1.13	$\mathbf{4.80 \pm 0.87}$	4.55 ± 1.00	4.35 ± 1.15
Overall Score	3.00 ± 0.56	3.05 ± 0.40	3.82 ± 0.92	$\mathbf{4.04 \pm 0.94}$

Table 2: Retrieval effectiveness based on overlap between retrieved and gold node identifiers. Values are macro mean ± population standard deviation.

Metric	Basic RAG	Agentic RAG (pooled)	GraphRAG	Agentic GraphRAG (pooled)
Hit@k	0.700 ± 0.458	0.725 ± 0.447	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$
Retrieval ID Precision	0.081 ± 0.068	0.045 ± 0.038	$\mathbf{0.571 \pm 0.347}$	0.490 ± 0.321
Retrieval ID Recall	0.200 ± 0.214	0.226 ± 0.243	0.553 ± 0.267	$\mathbf{0.666 \pm 0.217}$
Retrieval ID F1	0.109 ± 0.095	0.072 ± 0.061	0.491 ± 0.253	$\mathbf{0.516 \pm 0.243}$

but introduce claims that are more difficult to support consistently. Conversely, Agentic RAG's frequent abstentions produce high grounding and hallucination-control scores but low accuracy and completeness.

5.2 Retrieval Effectiveness and Adaptive Expansion

Table 2 reports identifier retrieval using pooled evidence for the agentic systems. GraphRAG and Agentic GraphRAG retrieve at least one gold identifier for every question, compared with Hit@k values of 0.700 for Basic RAG and 0.725 for Agentic RAG. Agentic GraphRAG achieves the highest recall (0.666) and F1 (0.516), whereas GraphRAG achieves the highest precision (0.571). Because every benchmark question contains an explicit identifier resolved before graph expansion, the perfect Hit@k of the graph systems partly reflects successful seed resolution; recall and F1 better characterize complete evidence coverage. Adaptive expansion produces markedly different effects across the two retrievers. Agentic RAG adds 13.6 identifiers per question but increases recall by only 0.027 and reduces F1 from 0.109 to 0.072. In contrast, Agentic GraphRAG adds only 3.2 identifiers, improves recall by 0.113 (20.5%), and increases F1 from 0.491 to 0.516. Graph-guided retries therefore expand evidence more selectively than semantic re-querying, although their lower precision indicates that they still introduce non-gold identifiers. Agentic GraphRAG also achieves the highest generated-answer identifier F1 (0.588, versus 0.434 for GraphRAG); complete answer-ID results and extractor limitations appear in Appendix A.2.

5.3 Performance Across Question Types

We further examine whether system performance varies with question difficulty and reasoning type. By benchmark difficulty, Agentic GraphRAG obtains the highest mean score on easy (4.38) and hard (3.81) questions, while GraphRAG obtains the highest score on medium questions (4.25 versus 4.21). Agentic RAG does not consistently improve over Basic RAG and performs worse on easy questions (2.94 versus 3.38). The category analysis reveals that the largest benefit of agentic control occurs on multi-hop reasoning questions, where Agentic GraphRAG improves over GraphRAG from 3.33 to 4.36, a gain of 30.9%. Smaller improvements occur for exact lookup, attack relationships, and mixed metadata. However, Agentic GraphRAG underperforms GraphRAG on cross-domain questions (4.08 versus 4.54), showing that adaptive control does not improve every question type. Because each category contains only four to eight questions, these results should be interpreted

Table 3: End-to-end latency and its retrieval and LLM components. Values are seconds per question, reported as mean ± population standard deviation. Bold indicates the lowest mean in each row.

Metric	Basic RAG	GraphRAG	Agentic RAG	Agentic GraphRAG
Total latency	$\mathbf{1.73 \pm 0.50}$	4.23 ± 4.02	23.15 ± 10.57	30.96 ± 20.59
Retrieval latency	0.09 ± 0.04	$\mathbf{0.07 \pm 0.04}$	0.17 ± 0.03	0.13 ± 0.09
LLM latency	$\mathbf{1.64 \pm 0.51}$	4.16 ± 4.00	22.98 ± 10.57	30.83 ± 20.55

as descriptive trends. Complete difficulty, category, and stress-test results are reported in Appendices A.3.2.

5.4 Latency and Efficiency

We finally examine the computational cost of graph retrieval and agentic control. As shown in Table 3, conventional Basic RAG has the lowest mean end-to-end latency at 1.73 seconds, followed by GraphRAG at 4.23 seconds. The additional latency of conventional GraphRAG is primarily associated with longer LLM generation rather than graph retrieval, which accounts for only 0.07 seconds on average. Agentic control substantially increases execution time. Agentic RAG requires 23.15 seconds per question, an increase of 21.42 seconds (1,238%) over Basic RAG. Agentic GraphRAG requires 30.96 seconds, an increase of 26.73 seconds (632%, or approximately 7.3×) over GraphRAG. Agentic GraphRAG is also 7.81 seconds (33.7%) slower than Agentic RAG. The latency increase is dominated by language-model inference rather than retrieval. LLM processing accounts for 22.98 of the 23.15 seconds required by Agentic RAG and 30.83 of the 30.96 seconds required by Agentic GraphRAG. This overhead reflects the controller's planning, evidence assessment, answer generation, and claim-verification calls. Agentic GraphRAG therefore provides the strongest answer quality and retrieval recall, but at a substantial inference-time cost. These latency comparisons are descriptive. Conventional and agentic runs were collected at different times, using one observation per question and different decoding configurations. API service load and retry behavior may also affect the measurements.

6 Conclusion

We presented an agentic, multi-layer GraphRAG framework for cybersecurity question answering. The framework integrates heterogeneous repositories through cross-layer bridge relationships, retrieves query-specific reasoning subgraphs, and uses a shared bounded controller for evidence assessment, adaptive retrieval, claim verification, revision, and abstention. Among the four evaluated systems, Agentic GraphRAG achieves the highest answer-quality score of 4.04, compared with 3.82 for GraphRAG and 3.00 for Basic RAG. Agentic control increases GraphRAG retrieval recall from 0.553 to 0.666, retrieval F1 from 0.491 to 0.516, and multi-hop performance from 3.33 to 4.36. In contrast, semantic query reformulation provides limited benefit over flat RAG, indicating that adaptive control is most effective when supported by structured, connected evidence. These gains introduce a quality–efficiency trade-off: Agentic GraphRAG increases mean latency from 4.23 to 30.96 seconds because of additional planning and verification calls. The present evaluation is limited to 40 cybersecurity questions, one knowledge graph, and a single primary generator and judge configuration. Future work will evaluate larger benchmarks, additional domains and models, improve cross-layer bridge coverage and graph control, and reduce the cost of iterative retrieval and verification.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*, 2023.

Cybersecurity and Infrastructure Security Agency. Known exploited vulnerabilities catalog, 2024. URL `https://www.cisa.gov/known-exploited-vulnerabilities-catalog`.

Cybersecurity and Infrastructure Security Agency (CISA). Aa21-048a: Ransomware best practices. https://www.cisa.gov/news-events/cybersecurity-advisories/aa21-048a, 2021a. Accessed: 2026-07-28.

Cybersecurity and Infrastructure Security Agency (CISA). Ar21-048c: Mar-10330311-1.v1: Fivehands ransomware. https://www.cisa.gov/news-events/analysis-reports/ar21-048c, 2021b. Accessed: 2026-07-28.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495*, 2023.

Darren Edge et al. From local to global: A graphrag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

ESET. Eset research. https://www.eset.com/uk/research/, 2026. Accessed: 2026-07-28.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y. Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023a. arXiv:2210.08726.

Yunfan Gao et al. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023b.

Google Cloud. Google threat intelligence. https://cloud.google.com/security/products/threat-intelligence, 2026. Accessed: 2026-07-28.

Aidan Hogan et al. Knowledge graphs. *ACM Computing Surveys*, 54(4):1–37, 2021.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2024. arXiv:2403.14403.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023. arXiv:2305.06983.

Kaspersky. Scarcruft. https://apt.securelist.com/apt/scarcruft, 2026. Accessed: 2026-07-28.

Patrick Lewis et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. Query rewriting for retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023. arXiv:2305.14283.

MITRE Corporation. Common weakness enumeration (cwe), 2024. URL https://cwe.mitre.org.

National Institute of Standards and Technology. National vulnerability database, 2024. URL https://nvd.nist.gov.

Red Canary. Atomic Red Team. https://github.com/redcanaryco/atomic-red-team, 2026. Accessed: 2026-07-26.

Kanwar Bharat Singh. TechRAG: Evidence-gated multimodal agentic RAG for technical literature reasoning. *arXiv preprint arXiv:2606.01613*, 2026. URL https://arxiv.org/abs/2606.01613.

Blake E. Strom, Andy Applebaum, Doug Miller, Kathryn C. Nickels, Adam G. Pennington, and Cody B. Thomas. Mitre att&ck: Design and philosophy. Technical Report MP18042018, MITRE, 2018.

Trend Micro. Raspberry robin malware targets telecom, governments. https://www.trendmicro.com/en_us/research/22/l/raspberry-robin-malware-targets-telecom-governments.html, 2022. Accessed: 2026-07-28.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023. arXiv:2212.10509.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Chao Zhang, Yuhao Wang, Derong Xu, Haoxin Zhang, Yuanjie Lyu, Yuhao Chen, Shuochen Liu, Tong Xu, Xiangyu Zhao, Yan Gao, Yao Hu, and Enhong Chen. TeaRAG: A token-efficient agentic retrieval-augmented generation framework. *arXiv preprint arXiv:2511.05385*, 2025. URL https://arxiv.org/abs/2511.05385.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

AI Use Statement

Generative AI tools were used to assist with language editing and software development. The authors designed the methodology, conducted the experiments, verified all generated code and text, analyzed the results, and take full responsibility for the manuscript.

A Additional Architecture Diagrams

Figures 2 and 3 show the multi-layer knowledge graph and hybrid GraphRAG retrieval pipeline, respectively.

A.1 Shared Controller Algorithm

Algorithm 1 Shared bounded agentic controller

Require: Question q, retriever adapter A, retrieval budget $B = 3$

$F \leftarrow \textsc{Plan}(q)$
$E \leftarrow \emptyset$, $H \leftarrow \emptyset$
$\rho \leftarrow q$
for $i = 1$ **to** B **do**
  $H \leftarrow H \cup \{\rho\}$
  $E \leftarrow E \cup \textsc{Novel}(\textsc{Retrieve}_A(\rho), E)$
  $\sigma \leftarrow \textsc{CheckEvidence}(q, F, E)$
  if not $\textsc{AllFactsSupported}(\sigma)$ **and** $i < B$ **then**
    $\rho' \leftarrow \textsc{Reformulate}(q, F, \sigma, H)$
    if $\rho' \notin H$ **then**
      $\rho \leftarrow \rho'$
      continue
    end if
  end if
  $E^\star \leftarrow \textsc{CitedSupportedEvidence}(E, \sigma)$
  if $E^\star = \emptyset$ **then**
    return $\textsc{Abstain}()$
  end if
  $\hat{a} \leftarrow \textsc{Generate}(q, F, E^\star, \sigma)$
  $C \leftarrow \textsc{VerifyClaims}(\hat{a}, E)$
  if $\textsc{AllMaterialClaimsSupported}(C)$ **then**
    return $\hat{a}$
  end if
  if $i < B$ **then**
    $\rho' \leftarrow \textsc{TargetedRequest}(C, H)$
    if $\rho' \notin H$ **then**
      $\rho \leftarrow \rho'$
      continue
    end if
  end if
  return $\textsc{ReviseOrAbstain}(\hat{a}, C)$
end for
return $\textsc{Abstain}()$

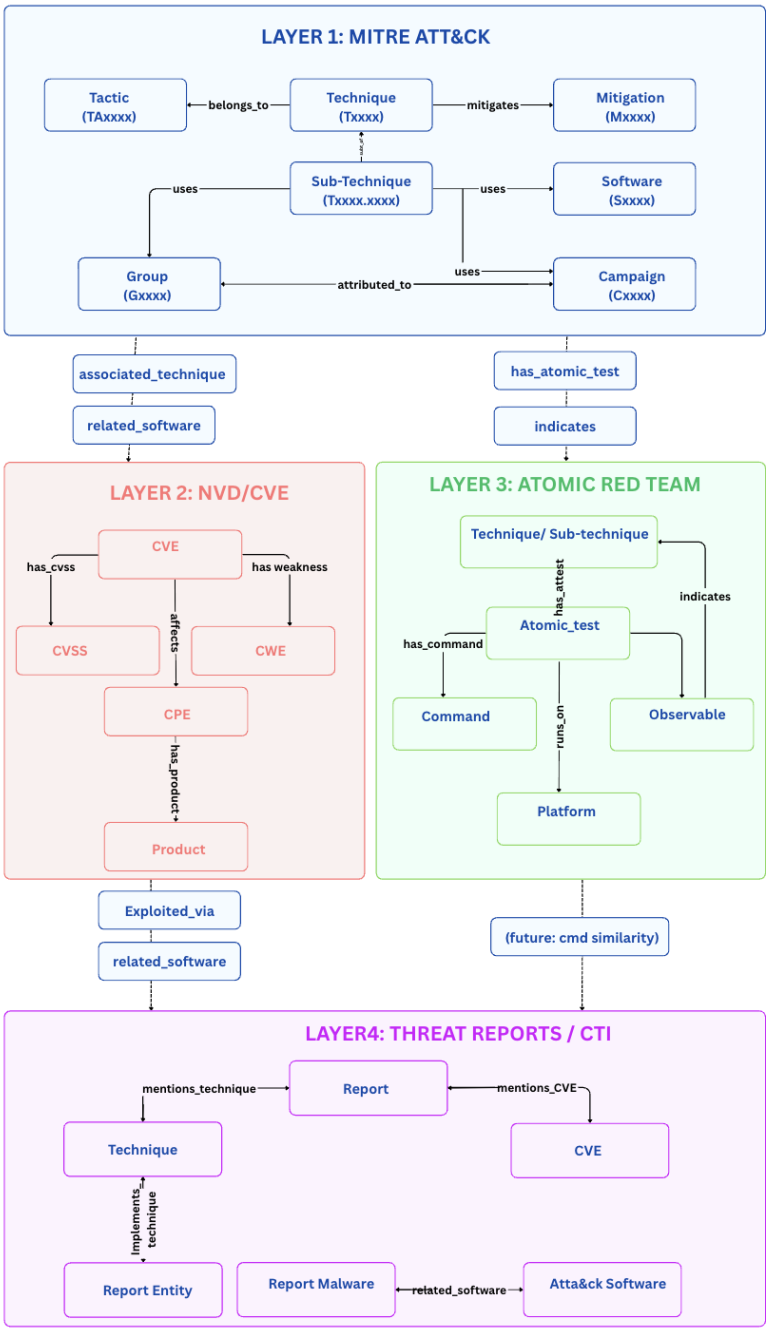

Figure 2: Multi-layer cybersecurity knowledge graph. Repository-specific layers preserve native entities and relations, while bridge edges connect related entities across sources.

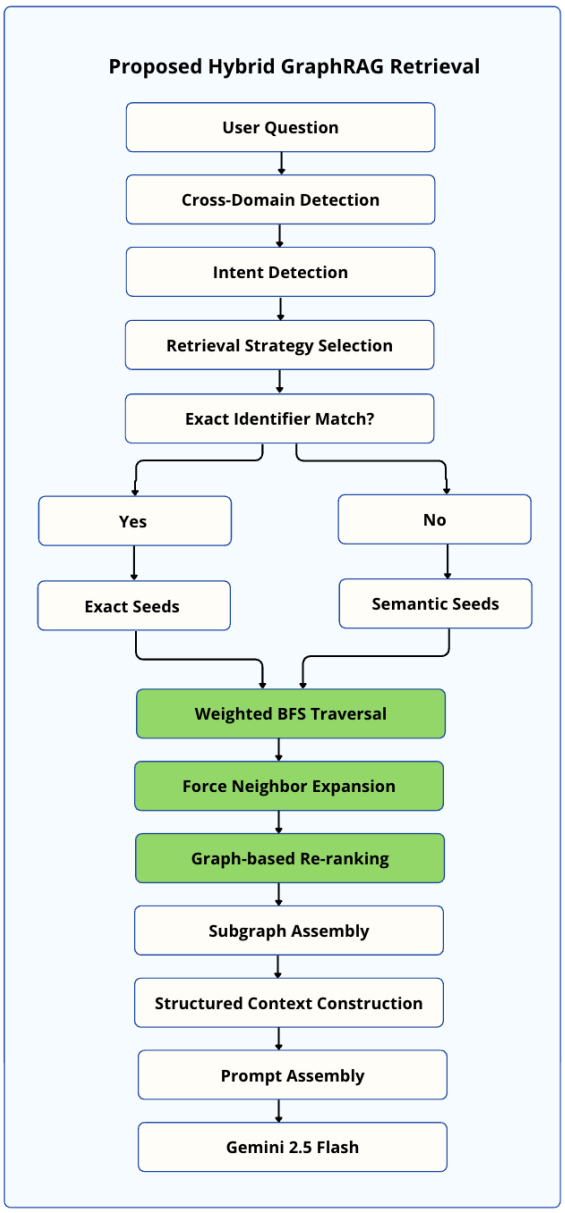

Figure 3: Hybrid GraphRAG retrieval pipeline. Exact or semantic seeds are expanded, re-ranked, and assembled into a query-specific reasoning subgraph.

A.2 Generated-Answer Identifier Fidelity

Answer-ID fidelity compares identifiers explicitly mentioned in the final answer with manually verified gold identifiers. We retain the legacy identifier extractor for comparability with the original evaluation. Because the extractor does not recognize CWE identifiers, answer-ID recall may underestimate coverage for questions containing gold CWE identifiers.

Table 4: Generated-answer identifier fidelity over 40 questions (macro mean ± population standard deviation).

Metric	Basic RAG	Agentic RAG	GraphRAG	Agentic GraphRAG
Answer-ID Precision	0.067 ± 0.238	0.143 ± 0.340	0.547 ± 0.444	$\mathbf{0.763 \pm 0.313}$
Answer-ID Recall	0.029 ± 0.114	0.044 ± 0.108	0.377 ± 0.328	$\mathbf{0.533 \pm 0.273}$
Answer-ID F1	0.038 ± 0.146	0.063 ± 0.155	0.434 ± 0.359	$\mathbf{0.588 \pm 0.264}$

A.3 Detailed Performance Breakdown

This appendix reports answer quality by benchmark difficulty, primary question category, and stress-test type. Because several groups contain few questions, the subgroup results should be interpreted as descriptive.

A.3.1 Results by Benchmark Difficulty

Table 5 reports answer quality across the three benchmark difficulty levels.

Table 5: Overall answer quality by benchmark difficulty. Values are mean ± population standard deviation. Bold indicates the highest mean in each row.

Difficulty	Basic RAG	Agentic RAG	GraphRAG	Agentic GraphRAG
Easy	3.38 ± 0.68	2.94 ± 0.59	4.17 ± 0.82	$\mathbf{4.38 \pm 0.62}$
Medium	2.72 ± 0.61	3.20 ± 0.36	$\mathbf{4.25 \pm 0.80}$	4.21 ± 0.91
Hard	3.02 ± 0.32	3.01 ± 0.29	3.43 ± 0.86	$\mathbf{3.81 \pm 1.01}$

A.3.2 Results by Question Category

Table 6 reports results for the questions assigned to the five primary question categories.

Table 6: Overall answer quality by primary question category. Bold indicates the highest mean in each row.

Category	Basic RAG	Agentic RAG	GraphRAG	Agentic GraphRAG
Exact Lookup	3.23	3.13	4.17	**4.37**
Attack Relationships	2.97	3.06	4.31	**4.56**
Cross-Domain	2.83	3.11	**4.54**	4.08
Mixed Metadata	3.04	3.08	3.29	**3.38**
Multi-Hop Reasoning	2.93	3.02	3.33	**4.36**

A.3.3 Stress-Test Breakdown

Table 7 reports the remaining ten questions by stress-test type. Because several groups contain only one or two questions, these results are diagnostic rather than conclusive.

Table 7: Answer quality by stress-test type. Bold indicates the highest mean in each row.

Stress Type	Basic RAG	Agentic RAG	GraphRAG	Agentic GraphRAG
Depth-3 Traversal	**3.17**	2.50	2.33	2.08
Set Intersection	3.17	3.00	**3.83**	3.33
Multiple CVEs	3.17	3.17	3.00	**4.83**
Multiple Tactics	3.00	3.08	**5.00**	**5.00**
Negative Evidence	3.00	3.08	2.92	**3.42**
Evidence Overlap	**3.00**	**3.00**	2.83	2.17
Evidence Pruning	3.00	3.17	3.17	**4.50**
All Stress Questions	3.07	2.97	3.33	**3.58**