CyQuAD: Question-Answer Dataset for Validating Retrieval-Augmented Generation in Cybersecurity

Adithya Mukundan*, Moqsadur Rahman*†, Aaron Sanchez‡, Tina M. Ellis*, Monika Akbar†, Joseph T. Aguayo§, Renuka Chintalapati*, M. Shahriar Hossain†, Siddhartha Shankar Das*, Mohammad F. Babar‡, Monowar Hasan‡, Samantha Silva*, Joe Higbee*, Zhisheng Qi¶, Yu Wang¶, Luis Fernando de la Torre‡, Mahantesh Halappanavar*

*Pacific Northwest National Laboratory, USA
{adithya.mukundan, tina.ellis, renuka.chintalapati, siddhartha.das, samantha.silva, joseph.higbee, hala}@pnnl.gov

†The University of Texas at El Paso, USA
{mrahman20, makbar, mhossain}@utep.edu

§Microsoft, USA
josephaguayo@microsoft.com

‡Washington State University, USA
{aaron.m.sanchez, m.babar, monowar.hasan, luis.delatorre}@wsu.edu

¶University of Georgia, USA
{zhisheng.qi, yu.wang6}@uga.edu

Abstract*—Cybersecurity question answering requires integrating evidence from vulnerability records, weakness taxonomies, attack-pattern catalogs, adversary-behavior frameworks, and public threat reports. Large language models can simplify access to this information, but answers generated from model parameters alone may contain unsupported identifiers, omit important relationships, or rely on outdated information. This paper introduces the Cybersecurity Question-Answer Dataset (CyQuAD), a benchmark of 600 open-ended question–answer pairs, and Graph-based Question Answering over Cybersecurity Data (GraphQuAD), an agentic method for evidence-grounded answer generation. CyQuAD contains 60 manually constructed questions and 540 questions produced through structured prompting and reviewed against associated sources, spanning factual retrieval, multi-hop and cross-standard reasoning, mitigation, hypothetical scenarios, and threat-intelligence analysis. GraphQuAD translates each question into Cypher, executes it against a Neo4j cybersecurity knowledge graph, and revises unsuccessful queries using execution errors, schema warnings, and empty-result feedback. When graph retrieval returns no usable evidence, GraphQuAD applies FAISS-based vector retrieval followed, when necessary, by web search. Using accuracy, completeness, and relevance, we compare GraphQuAD with a method that receives no graph evidence. We achieve around 3× improvement by GraphQuAD in mean accuracy from 0.2565 to 0.7538 and completeness from 0.2313 to 0.7616, while relevance decreases from 0.9526 to 0.9092. Applying fallback-retrieval only to the 60 no-evidence cases further increases full-benchmark accuracy to 0.8067 and completeness to 0.8040, with relevance of 0.8886. These results strongly demonstrate that structured graph evidence improves factual correctness and answer coverage, while targeted fallback retrieval can recover many graph-retrieval failures.

***Index Terms*—Cybersecurity benchmarking, large language models, knowledge graphs, retrieval-augmented generation, text-to-Cypher, agentic AI.**

I. Introduction

Cybersecurity analysis depends on information distributed across vulnerability records, weakness taxonomies, attack-pattern catalogs, adversary-behavior frameworks, and public threat reports. Sources such as Common Vulnerabilities and Exposures (CVE) [1], Common Weakness Enumeration (CWE) [2], Common Attack Pattern Enumeration and Classification (CAPEC) [3], and the MITRE ATT&CK framework [4] provide structured descriptions of these concepts, but answering an analyst's question may require linking several sources, interpreting scenario context, and identifying applicable mitigations. Large language models (LLMs) can make this information easier to query, yet answers generated from model parameters alone may contain unsupported identifiers, omit relationships, or rely on outdated information. In cybersecurity, a response is insufficient when its factual content or supporting evidence is incorrect or incomplete.

Recent cybersecurity benchmarks evaluate LLM knowledge, reasoning, and threat-intelligence capabilities through multiple-choice, classification, and structured-prediction tasks [5]–[8]. However, they provide less insight into open-ended answers that must integrate evidence across cybersecurity standards and practical threat scenarios.

Addressing this gap requires methods that can retrieve and use external evidence. Retrieval-augmented generation can ground an LLM in external evidence [9], while graph-based retrieval can preserve explicit relationships among cybersecurity entities. An end-to-end graph-based method, however, must translate a natural-language question into a valid graph query, execute it against the intended schema, determine whether the returned records are usable, and generate an answer supported by those records. Failures can arise from invalid Cypher [10], schema mismatches, or empty results, even when relevant information may be present in the graph. These failure modes make it necessary to evaluate final answers generated with and without Cypher-retrieved knowledge-graph evidence and to analyze how fallback retrieval can recover from unsuccessful graph retrieval.

We present CyQuAD, a cybersecurity question-answering benchmark in which each reference answer is linked to its supporting sources. We use CyQuAD to evaluate GraphQuAD, which translates questions into Cypher queries and uses the

retrieved graph records to generate answers. Together, they allow us to examine how graph evidence affects the accuracy, completeness, and relevance of cybersecurity answers. Our main contributions are:

- We introduce CyQuAD, a 600-question, source-traceable benchmark for open-ended cybersecurity question answering across factual, relational, mitigation, scenario-based, and threat-intelligence tasks.
- We develop GraphQuAD, an agentic question-to-Cypher and evidence-conditioned answer-generation method with feedback-guided query revision. When the graph returns no usable evidence, GraphQuAD uses FAISS-based vector retrieval [11] followed by web search when needed.
- We evaluate how graph evidence and fallback retrieval affect answer quality by question and subcategory. GraphQuAD roughly triples No-KG mean accuracy (0.2565 to 0.7538) and completeness (0.2313 to 0.7616). Fallback on the 60 no-evidence cases raises full-benchmark accuracy to 0.8067 and completeness to 0.8040, while relevance falls to 0.8886.

II. Related Work

Cybersecurity benchmarks evaluate different aspects of LLM knowledge, reasoning, and operational capability. SecQA uses multiple-choice computer-security questions [5], while SECURE evaluates knowledge and reasoning in industrial-control-system security [6]. CTIBench covers cyber threat intelligence tasks, including CVE-to-CWE mapping, vulnerability-severity assessment, and threat-actor attribution [7]. AthenaBench extends this direction through a dynamically updated evaluation pipeline and tasks covering mitigation and attack techniques [8]. Together, these benchmarks primarily cover cybersecurity knowledge, reasoning, classification, and structured prediction. CyQuAD instead evaluates open-ended answers across six subcategories covering direct retrieval, multi-hop reasoning, cross-standard linking, mitigation, hypothetical scenarios, and threat-intelligence questions. Its questions are grounded in sources spanning CVE, CWE, CAPEC, MITRE ATT&CK, and public threat reports. This organization supports evaluation across questions that require both factual knowledge and relationships among cybersecurity entities.

Retrieval-augmented generation conditions a language model on external evidence rather than relying only on its internal knowledge [9]. Graph-based extensions use entities and relations to support structured retrieval [12], [13]. GraphRAG constructs entity graphs and community summaries [14], while G-Retriever retrieves question-relevant subgraphs for language-model reasoning [15]. Text2Cypher and CypherBench are more directly related to GraphQuAD because they study the translation of natural-language questions into Cypher queries over property graphs [16], [17]. However, these works mainly evaluate query generation or graph retrieval. GraphQuAD evaluates the complete process from question-to-Cypher generation and Neo4j execution to evidence-conditioned answer generation. The study by Hamzic et al. compares vector, graph, query-repair, and hybrid retrieval for cyber threat intelligence [18]. Our framework differs by evaluating standards-linked cybersecurity questions, analyzing cases in which Cypher retrieval returns no usable evidence, and introducing FAISS and web-search fallback for those cases.

In executable environments, CVE-Bench evaluates agents' ability to exploit web application vulnerabilities [19], CyberGym tests whether agents can reproduce vulnerabilities [20], and ExploitGym tests whether they can turn vulnerability triggers into working exploits [21]. CyQuAD complements these benchmarks by evaluating the accuracy, completeness, and relevance of open-ended answers to cybersecurity questions.

III. Methodology

Our methodology has two components. First, we construct CyQuAD, a 600-question benchmark for open-ended cybersecurity question answering. Second, we develop GraphQuAD, an agentic method that converts questions into Cypher, retrieves evidence from a Neo4j cybersecurity knowledge graph, and generates answers from the retrieved evidence. GraphQuAD uses execution feedback to revise unsuccessful queries and invokes vector and web retrieval when the graph path returns no usable evidence.

CyQuAD is not restricted to questions whose answers are fully represented in the knowledge graph. The benchmark draws from cybersecurity standards, public threat reports, and government advisories, whereas the graph contains a structured subset of this information. This distinction allows the benchmark to evaluate both the benefits and limitations of knowledge-graph retrieval.

A. CyQuAD Benchmark Construction

1) Cybersecurity Sources: CyQuAD incorporates information from four structured cybersecurity resources: CVE, CWE, CAPEC, and MITRE ATT&CK. These resources describe vulnerabilities, weakness classes, attack patterns, adversary behaviors, threat groups, campaigns, malware, and defensive actions through standardized identifiers and relationships. The benchmark also uses public threat-intelligence reports and government advisories concerning exploited vulnerabilities, threat actors, campaigns, and real-world incidents. These sources are particularly important for questions whose answers cannot be constructed solely from structured framework records. Each benchmark item records the source references supporting its expected answer.

2) Question Categories and Subcategories: CyQuAD has two categories and six subcategories, with 100 questions per subcategory (Table I). *Knowledge-Based* questions test the ability to retrieve facts and trace relationships across cybersecurity standards. This category includes Direct Fact Retrieval, Multi-Hop Reasoning, and Cross-Standard Linking. *Practical Action* questions test the ability to select mitigations, reason about scenarios, and interpret threat intelligence. This category includes Mitigation, Hypothetical, and Intel to Intersection.

3) Manual Question–Answer Pair Construction: We manually constructed 10 question–answer pairs per subcategory, yielding 60 examples. For Knowledge-Based questions, we

TABLE I
CONSTRUCTION AND DISTRIBUTION OF THE CYQUAD SUBCATEGORIES.

Subcategory	Target capability	Manual	Generated	Total
Direct Fact Retrieval	Direct lookup of a vulnerability, weakness, attack pattern, identifier, or associated property.	10	90	100
Multi-Hop Reasoning	Traversal through multiple related entities, such as CVE → CWE → CAPEC.	10	90	100
Cross-Standard Linking	Connection of concepts represented in different standards, particularly CAPEC and MITRE ATT&CK.	10	90	100
Mitigation	Identification of preventive, corrective, or defensive actions for a weakness, vulnerability, or attack pattern.	10	90	100
Hypothetical	Interpretation of a cybersecurity scenario and identification of the relevant weaknesses, attacks, or mitigations.	10	90	100
Intel to Intersection	Connection of threat actors, campaigns, techniques, vulnerabilities, and attack patterns using public threat intelligence.	10	90	100
Total		60	540	600

used individual framework records and followed documented links across entities and standards. For Practical Action questions, we drew on recommended mitigations, formulated scenarios, and linked activity described in threat reports and advisories to cybersecurity entities. These examples guided the wording, reasoning requirements, and answer format of the generated pairs.

4) Structured Dataset Expansion: The 60 manually constructed examples were expanded into 540 synthetic QA pairs across six categories using a subcategory-specific, structured Chain-of-Thought (CoT) prompting strategy. For each subcategory, the generation prompt specified: (1) the eligible source types, (2) the intended reasoning or linkage pattern, (3) the required output fields, and (4) manually prepared examples demonstrating the expected format. For relational questions, the prompt also specified the sequence of entities that each generated item should connect.

5) Dataset Representation: Each CyQuAD item is stored as a structured JSON record containing a unique question identifier, category, subcategory, natural-language question, optional context, expected answer, and one or more source references. Fig. 1 shows a sample CyQuAD record. This format supports question-level evaluation, comparisons across categories and subcategories, and analysis of retrieval and answer-generation errors.

CyQuAD is constructed directly from the cited sources. Neither manual question construction nor dataset expansion involved querying the GraphQuAD knowledge graph or sampling its traversal paths. One co-author manually verified all 600 question–answer pairs for correctness. The CyQuAD dataset is available on Zenodo at https://zenodo.org/records/24167735.

```
{
   "Q #": 14,
   "Category": "Knowledge Based",
   "Sub Category": "Multi Hop Reasoning",
   "Question": "What related attack patterns (CAPEC
       IDS) can be exploited through CVE-2023-3425",
   "Context": "Out-of-bounds read issue in M-Files
       Server versions below 23.8.12892.6 and LTS
       Service Release Versions before 23.2 LTS SR3
       allows unauthenticated user to read
       restricted amount of bytes from memory.",
   "Answer": "CAPEC-540: Overread Buffers",
   "Source": "https://www.cve.org/CVERecord?id=...",
   "Source 2": "https://cwe.mitre.org/data/..."
}
```

Fig. 1. Example CyQuAD question–answer record.

B. Cybersecurity Knowledge Graph

GraphQuAD operates over a cybersecurity knowledge graph stored in Neo4j that integrates entities from CVE, CWE, CAPEC, and MITRE ATT&CK [22]. The construction pipeline ingests NVD CVE 2.0 JSON feeds, CWE and CAPEC XML releases, and ATT&CK STIX bundles. The graph contains over 350,000 nodes and 300,000 relationships, constructed from source cross-references, hierarchies, and STIX relationship objects. The graph also incorporates predicted links from CVEs to CWE weaknesses using V2W-BERT [23] and from CWE weaknesses to CAPEC attack patterns using VWC-BERT [24]. CVE, CWE, and CAPEC are refreshed daily, and ATT&CK weekly. After initial ingestion of CVE archives from 2002–2025, incremental updates incorporate new and modified vulnerabilities without reprocessing historical records.

The graph connects these cybersecurity entities through the path $\texttt{cve} \xrightarrow{\texttt{related_cwe}} \texttt{cwe} \xrightarrow{\texttt{related_capec}} \texttt{capec} \xrightarrow{\texttt{related_attack}} \texttt{attack}$. This path connects vulnerabilities to weaknesses, CAPEC attack patterns, and corresponding ATT&CK techniques. Although these relationships are stored with defined directions, GraphQuAD can traverse them in reverse when a question begins with a CWE, CAPEC, or ATT&CK entity. These traversal patterns support both multi-hop and cross-standard questions.

The ATT&CK portion of the graph additionally represents intrusion sets, campaigns, malware, tools, and courses of action. Intrusion sets are connected to the attack techniques, malware, tools, and campaigns associated with their reported activity. Campaigns and courses of action are also connected

to relevant ATT&CK techniques. These relationships support questions concerning threat actors, adversary behavior, campaigns, defensive actions, and their intersections with structured vulnerability information.

The graph schema constrains the question-to-Cypher model to the supported node labels, properties, relationships, and identifier formats. CVE nodes store complete identifiers, such as `CVE-2022-26069`, whereas CWE and CAPEC nodes store only their numeric components, such as `89` and `66`. ATT&CK entities store standard identifiers, such as `T1055` and `G0016`. The prompt also specifies preferred query paths and directs the model to use the shortest valid traversal consistent with the question intent. These constraints reduce invalid-label, relationship, property, and identifier-matching errors during Cypher generation.

C. GraphQuAD Question Answering

Fig. 2 summarizes GraphQuAD's agentic question-to-Cypher and evidence-conditioned answer-generation workflow. The process consists of schema-guided Cypher generation, execution and feedback-guided revision, evidence serialization, and final answer generation.

1) Schema-Guided Cypher Generation: For each benchmark item, GraphQuAD receives a question q and, when designated by the benchmark, an optional context c_q. The question-to-Cypher prompt also contains the supported graph schema and query-intent rules. These rules specify preferred traversal patterns for each question type. For example, a question requesting CAPEC patterns associated with a CVE should follow the CVE–CWE–CAPEC path rather than continuing unnecessarily to an ATT&CK node.

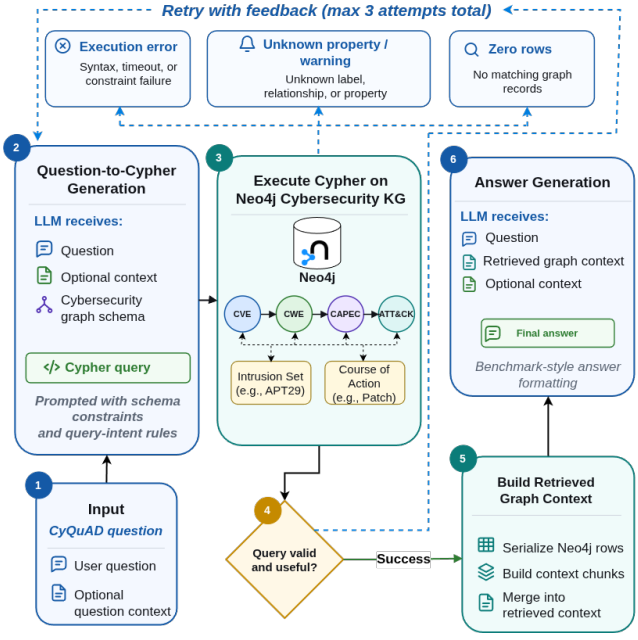


Fig. 2. GraphQuAD's agentic question-answering pipeline. An LLM generates a Cypher query, which GraphQuAD executes against the Neo4j cybersecurity knowledge graph. Execution errors, schema warnings, and empty results provide feedback for query revision. Retrieved records are serialized and used to generate the final answer.

Let $\mathcal{S}$ denote the graph schema and $f_q^{(t-1)}$ the feedback from the previous attempt. At attempt t, the query-generation model produces

$$z_q^{(t)} = \phi\left(q, c_q, \mathcal{S}, f_q^{(t-1)}\right), \qquad 1 \le t \le T, \quad T = 3, \tag{1}$$

where $f_q^{(0)} = \varnothing$. The expected answer is not provided during Cypher generation or answer generation; it is used only during evaluation.

2) Execution and Feedback-Guided Revision: The generated query is executed against the knowledge graph $\mathcal{G}$:

$$\left(R_q^{(t)}, W_q^{(t)}\right) = \text{Execute}\left(z_q^{(t)}, \mathcal{G}\right), \tag{2}$$

where $R_q^{(t)}$ is the set of returned records and $W_q^{(t)}$ contains Neo4j warnings. A query is considered unsuccessful when it raises an execution error, refers to an unknown label, relationship, or property, or returns no records.

For an unsuccessful attempt, GraphQuAD converts the execution outcome into textual feedback. Execution errors provide the database error message, while schema warnings identify unsupported graph elements. Empty-result feedback instructs the model to reconsider identifier formatting and traversal choice. The original question, graph schema, and feedback are then used to generate a revised query. The process is limited to three attempts in total.

A query is accepted when it executes without a relevant schema error and returns one or more records. Failure to obtain usable records within the attempt limit produces a *no-evidence case*. Such a case does not necessarily indicate that the required information is absent from the graph; the query may instead have failed to reach the relevant nodes.

3) Evidence Serialization: Neo4j records may contain graph-specific objects that cannot be inserted directly into an LLM prompt. GraphQuAD therefore converts every returned value into a serializable representation and formats each record as key–value text. At most 20 returned records are retained as individual context chunks and joined to form the retrieved evidence:

$$e_q = \text{Serialize}\left(R_q^{(t^*)}\right), \tag{3}$$

where t^* is the accepted query attempt. This representation preserves the returned identifiers, entity names, properties, and relationships while avoiding Neo4j-specific formatting.

4) Evidence-Conditioned Answer Generation: The answer-generation prompt contains the question q, optional context c_q, and retrieved graph evidence e_q. The prompt instructs the model to generate an answer utilizing the retrieved evidence.

Let $\mathcal{L}$ denote the answer-generation model. The GraphQuAD answer is

$$a_q^{\text{G}} = \mathcal{L}(\text{Prompt}(q, c_q, e_q)). \tag{4}$$

GraphQuAD records generated queries, retrieved evidence, and final answers for question-level analysis.

D. Recovery from Unavailable Graph Evidence

Fig. 3 summarizes GraphQuAD's recovery path when the question-to-Cypher process does not return usable graph evidence. The knowledge graph remains GraphQuAD's primary evidence source, and the recovery path is invoked only when usable records are not obtained within the attempt limit. GraphQuAD first searches a FAISS vector index and retrieves the top-ranked passage. The retrieved passage is used when its cosine similarity to the question meets the routing threshold θ_F; otherwise, Tavily web search is used to obtain relevant page content or result snippets.

Let E_q^G be the graph evidence, $u_q \in \{0, 1\}$ indicate whether that evidence is usable, d_q^* be the highest-ranked FAISS passage, σ_q its cosine similarity, and E_q^W the web evidence. The context selected for answer generation is

$$C_q = \begin{cases} E_q^G, & u_q = 1, \\ d_q^*, & u_q = 0 \wedge \sigma_q \geq \theta_F, \\ E_q^W, & u_q = 0 \wedge \sigma_q < \theta_F. \end{cases} \tag{5}$$

The recovered answer is then generated as

$$a_q^{\mathrm{R}} = \mathcal{L}(\mathrm{Prompt}(q, c_q, C_q)) \,. \tag{6}$$

This routing order preserves the knowledge graph as the preferred source, uses the semantic search as the first recovery mechanism, and accesses the web only when the vector result does not meet the similarity threshold.

IV. Experimental Setup and Evaluation

We compare GraphQuAD with a No-KG method to isolate the effect of question-specific graph evidence. We use GPT-5.1 as the underlying language model for GraphQuAD's Cypher generation and revision, evidence-conditioned answer generation, and No-KG answer generation. Both conditions use the same benchmark questions, optional contexts, answer-generation model, and output formatting rules; they differ only in whether question-specific graph evidence is retrieved and provided for answer generation. The No-KG method does not receive a generated Cypher query or retrieved graph evidence:

$$a_q^{\text{No-KG}} = \mathcal{L}(\mathrm{Prompt}(q, c_q)) \,. \tag{7}$$

For FAISS-based semantic retrieval, we index the textual descriptions of CVE, CWE, CAPEC, and MITRE ATT&CK nodes and use `gemini-embedding-001` to encode these descriptions and incoming questions into dense vector representations. We apply FAISS/web-recovery only to questions when GraphQuAD returns no usable graph evidence.

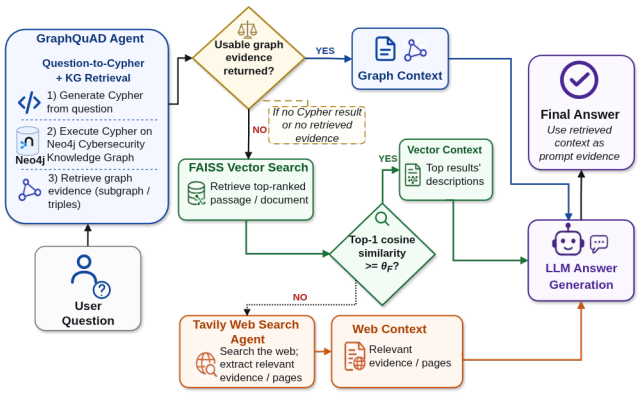


Fig. 3. GraphQuAD recovery path. Usable graph evidence is preferred. When the graph path returns no usable evidence, the method retrieves the top FAISS result and accepts it when cosine similarity is at least θ_F; otherwise, it invokes Tavily web search.

A. Rubric-Based Answer Evaluation

Generated answers are evaluated using three custom discrete metrics implemented through the RAGAS interface [25], with GPT-5.1 serving as the rubric-based evaluator:

- *Accuracy* measures factual correctness and penalizes incorrect or unsupported identifiers, entities, and claims.
- *Completeness* measures whether the response covers the critical information and supporting details in the reference answer.
- *Relevance* measures whether the response directly addresses the question, independently of whether the answer is factually correct.

The evaluator assigns an integer score from 0 to 10 using rubric anchors at 0, 2, 4, 6, 8, and 10. Each score is divided by 10 to obtain a value in $[0, 1]$. Accuracy and completeness receive the question, generated answer, and expected answer. Relevance receives only the question and generated answer so that topical responsiveness is evaluated separately from factual agreement with the reference.

Normalized exact matches and equivalent unordered item sets receive a score of 1.0. For other responses, deterministic identifier checks supplement the rubric-based evaluation. When recognized identifiers appear in both the expected and generated answers, accuracy is capped by identifier precision and completeness by identifier recall. A completeness cap of 0.7 is applied when the expected answer contains substantial explanatory detail but the generated response provides only a brief list. This prevents sparse answers from receiving high completeness scores despite omitting required explanation.

B. Aggregate and Question-Level Comparison

We compute subcategory-level means for accuracy, completeness, and relevance. Let $s_{q,m}^{(d)} \in [0, 1]$ denote the score of method m on question q. Means summarize aggregate performance but do not show how consistently a method approaches the better result across individual questions. We therefore use question-level performance profiles, adapted from Dolan and Moré [26]. For $\mathcal{M} = \{\mathrm{G}, \mathrm{No\text{-}KG}\}$, the performance ratio is

$$r_{q,m}^{(d)} = \begin{cases} 1, & \max\limits_{m' \in \mathcal{M}} s_{q,m'}^{(d)} = 0, \\ \dfrac{\max_{m' \in \mathcal{M}} s_{q,m'}^{(d)}}{s_{q,m}^{(d)}}, & s_{q,m}^{(d)} > 0, \\ \infty, & s_{q,m}^{(d)} = 0 \wedge \max\limits_{m' \in \mathcal{M}} s_{q,m'}^{(d)} > 0. \end{cases} \tag{8}$$

A ratio of 1 indicates that a method attains the highest score on a question, including ties. When all methods receive a score of zero, each is assigned a ratio of 1 to avoid the undefined ratio $0/0$. When a method receives zero but another method receives a positive score, its ratio is set to ∞.

For subcategory c, the performance profile is

$$\rho_{m,c}^{(d)}(\tau) = \frac{1}{|\mathcal{Q}_c^{(d)}|} \sum_{q \in \mathcal{Q}_c^{(d)}} \mathbf{1}\left[r_{q,m}^{(d)} \leq \tau\right], \qquad \tau \geq 1, \tag{9}$$

where $\mathcal{Q}_c^{(d)}$ contains the questions with paired scores and $\mathbf{1}[\cdot]$ is the indicator function. The quantity $\rho_{m,c}^{(d)}(\tau)$ is the fraction of questions for which method m achieves at least $1/\tau$ of the better score. In particular, $\rho_{m,c}^{(d)}(1)$ is the fraction of questions on which the method is best or tied for best. A profile positioned higher and farther to the left indicates stronger and more consistent performance.

C. Cypher Retrieval and Recovery Analysis

Answer quality alone does not identify whether an error originates from retrieval or generation. We therefore partition GraphQuAD runs according to whether the final Cypher query retrieves usable records. For each partition, we report the number of questions and the mean accuracy, completeness, and relevance. This analysis distinguishes cases in which the answer-generation model receives graph evidence from cases in which the Cypher process fails to supply it.

The recovery evaluation then re-answers only the no-evidence cases using the FAISS or web path. Results are reported on this subset and the complete benchmark. This measures whether fallback evidence can recover retrieval failures without replacing successful knowledge-graph retrieval.

V. Results

We first evaluate the effect of knowledge-graph evidence at the aggregate, subcategory, and question levels. We then analyze retrieval outcomes and determine whether FAISS and web retrieval can recover no-evidence cases.

A. Effect of Knowledge-Graph Evidence

We compare GraphQuAD with the No-KG method to isolate the effect of question-specific knowledge-graph evidence. The two conditions use the same questions, optional benchmark context, answer-generation model, and output instructions; they differ only in whether the answer-generation prompt receives Cypher-retrieved graph evidence. The GraphQuAD question-to-Cypher process is shown in Fig. 2 and described in Section III-C.

TABLE II
Mean GraphQuAD (GQ) and No-KG (NK) scores by subcategory.

	Acc.		Comp.		Rel.	
Sub.	GQ	NK	GQ	NK	GQ	NK
DFR	0.7430	0.2930	0.7480	0.2719	0.8390	0.9480
MHR	0.8504	0.0180	0.8664	0.0157	0.9890	0.9630
CSL	0.7892	0.2480	0.7580	0.2393	0.9330	0.9630
MIT	0.7690	0.6170	0.8010	0.5320	0.7616	0.9290
HYP	0.8767	0.1281	0.8722	0.1183	0.9650	0.9380
ITI	0.4943	0.2348	0.5242	0.2105	0.9660	0.9747
All	0.7538	0.2565	0.7616	0.2313	0.9092	0.9526

DFR: Direct Fact Retrieval; MHR: Multi-Hop Reasoning; CSL: Cross-Standard Linking; MIT: Mitigation; HYP: Hypothetical; ITI: Intel to Intersection.

Across all 600 questions, GraphQuAD increases mean accuracy from 0.2565 to 0.7538 and mean completeness from 0.2313 to 0.7616, corresponding to absolute gains of 0.4973 and 0.5303, respectively. As shown in Table II, these gains occur in every subcategory. The largest improvements arise for questions that depend on explicit relations among cybersecurity entities. For Multi-Hop Reasoning, accuracy increases from 0.0180 to 0.8504 and completeness from 0.0157 to 0.8664. For Hypothetical questions, the corresponding changes are from 0.1281 to 0.8767 and from 0.1183 to 0.8722. Cross-Standard Linking also shows substantial improvements, with accuracy increasing from 0.2480 to 0.7892 and completeness from 0.2393 to 0.7580.

The relevance pattern differs from accuracy and completeness. Using all available relevance scores, the No-KG method obtains a mean of 0.9526, compared with 0.9092 for GraphQuAD. GraphQuAD has higher mean relevance on Multi-Hop Reasoning and Hypothetical questions, whereas the No-KG method has higher mean relevance on the other four subcategories. Retrieved graph records can therefore improve factual coverage while also introducing details that extend beyond the narrow scope of a question.

B. Retrieval Baselines and Query-Repair Ablation

Table III compares iterative GraphQuAD, single-attempt GraphQuAD, Tavily-only retrieval, and FAISS-only retrieval across all 600 questions. The GraphQuAD results precede

TABLE III
Retrieval comparisons on CyQuAD. Overall is the arithmetic mean of accuracy, completeness, and relevance.

Method	Acc.	Comp.	Rel.	Overall
GraphQuAD iterative	**0.7538**	**0.7616**	0.9092	**0.8082**
GraphQuAD single-attempt	0.7452	0.7512	**0.9119**	0.8028
Tavily web search only	0.6687	0.5585	0.8507	0.6926
FAISS vector search only	0.4808	0.2856	0.6517	0.4727

fallback recovery. Iterative GraphQuAD achieves the highest accuracy, completeness, and overall score. Compared with FAISS-only retrieval, its accuracy and completeness are higher by 0.2730 and 0.4760, respectively. The corresponding gains over Tavily-only retrieval are 0.0851 and 0.2031. These comparisons demonstrate an advantage for GraphQuAD over the evaluated vector and web retrieval baselines on CyQuAD.

Allowing query revision increases accuracy from 0.7452 to 0.7538 and completeness from 0.7512 to 0.7616, while relevance decreases slightly from 0.9119 to 0.9092. Thus, the repair loop provides modest aggregate gains, with the single-attempt variant already achieving similar performance.

C. Qualitative Case Study

Table IV illustrates the effect of graph evidence on a Multi-Hop Reasoning question asking for CAPEC attack patterns associated with `CVE-2025-0327`. GraphQuAD generates the required CVE–CWE–CAPEC traversal in a single attempt, retrieving three records linking `CVE-2025-0327` to `CWE-269` and subsequently to `CAPEC-58`, `CAPEC-233`, and `CAPEC-122`. Its answer matches all three expected CAPEC ID–name pairs, yielding scores of 1.00 for accuracy, completeness, and relevance.

In contrast, the No-KG method returns ten CAPEC entries, including `CAPEC-233` with an incorrect attack pattern, while omitting `CAPEC-58` and `CAPEC-122`. The remaining entries are not part of the expected answer, resulting in accuracy and completeness scores of 0.10 and 0.20, respectively. Its relevance remains 1.00 because the response addresses the requested information despite being factually inaccurate and incomplete. This example demonstrates how explicit graph traversal improves factual correctness and coverage while highlighting why relevance alone is insufficient for evaluating cybersecurity answers.

D. Question-Level Performance Profiles

Fig. 4 presents the question-level performance profiles defined in Section IV-B. At $\tau = 1$, GraphQuAD is best or tied for best on 91.33% of questions for accuracy and 92.17% for completeness, compared with 27.50% and 26.50%, respectively, for the No-KG method. The mean-score improvements in Table II therefore reflect broad question-level gains rather than a small number of unusually large improvements.

For accuracy, the largest separations occur on Multi-Hop Reasoning, Cross-Standard Linking, and Hypothetical questions. The GraphQuAD/No-KG best-or-tied fractions are $0.98/0.08$, $0.97/0.36$, and $0.98/0.16$, respectively. For completeness, the fractions are $0.99/0.08$, $0.97/0.35$, and $0.97/0.15$. GraphQuAD also leads on Direct Fact Retrieval, Mitigation, and Intel to Intersection for both metrics.

Relevance is more balanced. GraphQuAD and No-KG have mean relevance scores of 0.9092 and 0.9526, respectively. GraphQuAD is best or tied on 85.62% of questions, whereas No-KG is best or tied on 89.97%. GraphQuAD leads on Multi-Hop Reasoning and Hypothetical questions, with best-or-tied fractions of $0.98/0.89$ and $0.94/0.83$. No-KG leads on Direct Fact Retrieval, Cross-Standard Linking, Mitigation, and Intel to Intersection. The largest difference occurs for Mitigation, where the fractions are 0.606 for GraphQuAD and 0.949 for No-KG. Thus, an answer can remain directly responsive to a question even when it is less accurate and complete.

TABLE IV
QUALITATIVE COMPARISON FOR A MULTI-HOP REASONING QUESTION.

Stage	Result
Question	What related attack patterns (CAPEC IDs) can be exploited through `CVE-2025-0327`?
Cypher path	CVE → CWE → CAPEC; one attempt, three records.
Graph evidence	`CVE-2025-0327` → `CWE-269` → {`CAPEC-58`, `CAPEC-233`, `CAPEC-122`}.
GraphQuAD	CAPEC-58: Restful Privilege Elevation; CAPEC-233: Privilege Escalation; CAPEC-122: Privilege Abuse.
No-KG	Ten CAPEC entries; CAPEC-233 is incorrectly named, while CAPEC-58 and CAPEC-122 are omitted.
Expected	CAPEC-122: Privilege Abuse; CAPEC-233: Privilege Escalation; CAPEC-58: Restful Privilege Elevation.
Scores	GraphQuAD: 1.00/1.00/1.00; No-KG: 0.10/0.20/1.00.

Scores: accuracy/completeness/relevance.

E. Score Distributions Across CyQuAD Subcategories

Fig. 5 complements the performance profiles by visualizing the distributions of question-level scores using violin plots [27]. The width of each violin represents the estimated score density within a subcategory, while the internal markers indicate the mean, median, minimum, and maximum.

GraphQuAD accuracy is concentrated at higher values than No-KG accuracy, particularly for Multi-Hop Reasoning, Cross-Standard Linking, and Hypothetical questions. The completeness distributions show the same overall pattern. These distributions support the profile-based finding that graph evidence improves performance across many questions rather than affecting only a small number of cases.

The relevance distributions are closer and, in several subcategories, favor the No-KG method. This pattern is consistent with Fig. 4: graph context can introduce additional entities, relations, or mitigation details, which may make an answer broader than the question requires.

F. Retrieval Outcomes and Fallback Recovery

1) Answer Quality by Retrieval Outcome: GraphQuAD retrieved usable graph evidence for 540 of 600 questions (90%). Retrieval success ranged from 92% to 100% across five subcategories but fell to 57% for Intel to Intersection, which accounted for 43 of the 60 retrieval failures (71.7%). Table V reports answer quality for each retrieval outcome.

Answers with usable graph evidence had higher mean accuracy (0.8132 vs. 0.2192) and completeness (0.8255 vs. 0.1873) than answers without usable evidence, but lower mean relevance (0.9030 vs. 0.9650). Relevance measures whether an answer addresses the question, so a response can score highly even when it omits required identifiers or relationships.

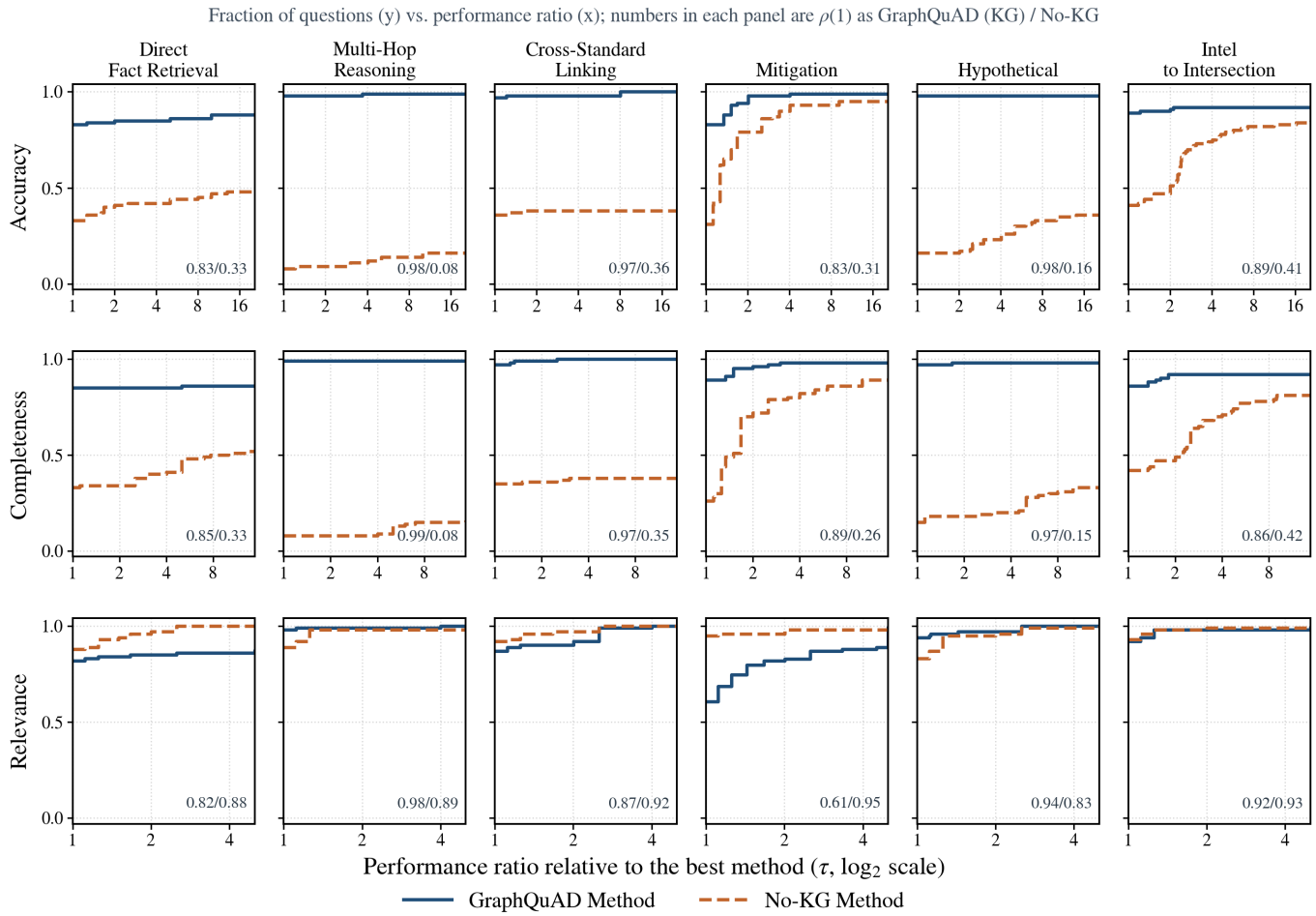


Fig. 4. Subcategory-level performance profiles comparing GraphQuAD with the No-KG method. Rows correspond to accuracy, completeness, and relevance, and columns correspond to the six CyQuAD subcategories. For each question, the performance ratio is the best score obtained by either method divided by the score of the evaluated method. Each curve gives $\rho(\tau)$, the fraction of questions for which a method lies within a factor τ of the better score, with τ shown on a $\log_2$ scale. The numbers in each panel report $\rho(1)$ as GraphQuAD/No-KG. Accuracy and completeness use 600 paired questions; relevance uses 598. GraphQuAD leads on accuracy and completeness in every subcategory, whereas the No-KG method attains the higher $\rho(1)$ for relevance in four of the six subcategories.

2) Recovery on the No-Evidence Subset: We applied fallback retrieval only to the 60 questions for which GraphQuAD retrieved no usable graph evidence (Fig. 3). On these same questions, we tested FAISS cosine-similarity thresholds of 0.75, 0.80, and 0.85. The corresponding scores for accuracy, completeness, and relevance were (0.688, 0.459, 0.648), (0.749, 0.611, 0.760), and (0.740, 0.609, 0.773). We selected 0.80, which yielded the highest accuracy. GraphQuAD used the top FAISS passage when its cosine similarity met this threshold and queried Tavily otherwise.

TABLE V
GRAPHQUAD ANSWER QUALITY BY RETRIEVAL OUTCOME. E AND NE DENOTE QUESTIONS WITH AND WITHOUT USABLE GRAPH EVIDENCE, RESPECTIVELY, AND N DENOTES THE NUMBER OF QUESTIONS.

			Acc.		Comp.		Rel.	
Sub.	N_{E}	N_{NE}	E	NE	E	NE	E	NE
DFR	92	8	0.7859	0.2500	0.7913	0.2500	0.8250	1.0000
MHR	96	4	0.8849	0.0227	0.9008	0.0417	0.9906	0.9500
CSL	98	2	0.7951	0.5000	0.7633	0.5000	0.9327	0.9500
MIT	100	0	0.7690	–	0.8010	–	0.7616	–
HYP	97	3	0.9038	0.0000	0.8992	0.0000	0.9701	0.8000
ITI	57	43	0.6907	0.2340	0.7780	0.1877	0.9614	0.9721
All	**540**	**60**	**0.8132**	**0.2192**	**0.8255**	**0.1873**	**0.9030**	**0.9650**

DFR: Direct Fact Retrieval; MHR: Multi-Hop Reasoning; CSL: Cross-Standard Linking; MIT: Mitigation; HYP: Hypothetical; ITI: Intel to Intersection. One relevance score is unavailable in the usable-evidence group.

On the 60 re-answered questions, mean accuracy rose from 0.2192 to 0.7490 and completeness from 0.1873 to 0.6110, while relevance fell from 0.9650 to 0.7600 (Table VI). Intel to Intersection contributed most to the overall gains. Across its 43 questions, accuracy rose from 0.2340 to 0.7380 and completeness from 0.1877 to 0.5890.

3) Effect on the Complete Benchmark: Table VII reports the effect of replacing only the 60 no-evidence answers while leaving the other 540 GraphQuAD answers unchanged. Mean accuracy increases from 0.7538 to 0.8067, corresponding to

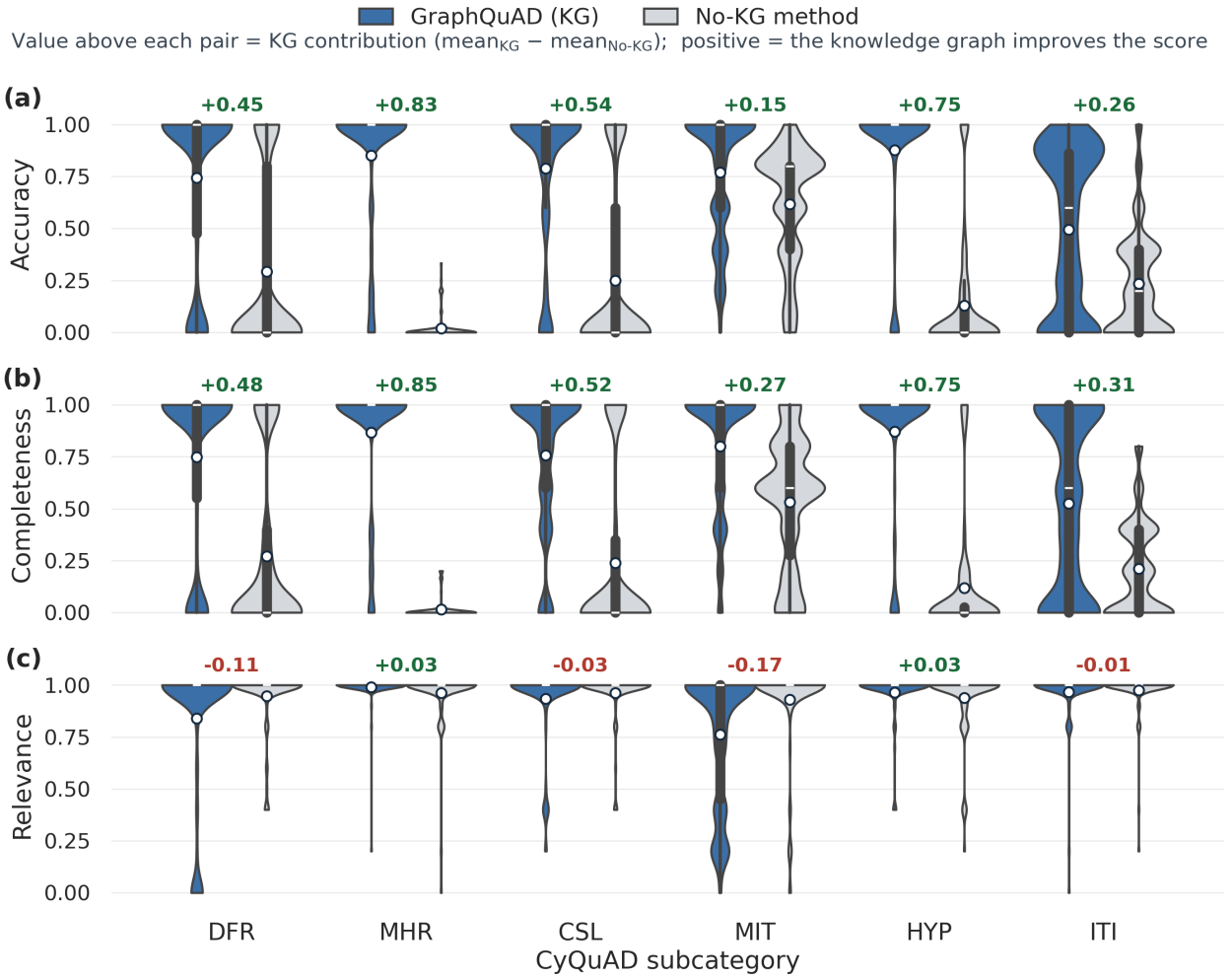


Fig. 5. Question-level score distributions for GraphQuAD and the No-KG method across Direct Fact Retrieval (DFR), Multi-Hop Reasoning (MHR), Cross-Standard Linking (CSL), Mitigation (MIT), Hypothetical (HYP), and Intel to Intersection (ITI). Violin width represents the estimated score density, and the internal markers show the mean, median, minimum, and maximum for (a) accuracy, (b) completeness, and (c) relevance.

TABLE VI
PERFORMANCE BEFORE (B) AND AFTER (A) FAISS/WEB FALLBACK ON THE 60 NO-EVIDENCE QUESTIONS.

		Accuracy		Completeness		Relevance	
Sub.	N	B	A	B	A	B	A
DFR	8	0.2500	1.0000	0.2500	1.0000	1.0000	0.9000
MHR	4	0.0227	0.5500	0.0417	0.3330	0.9500	0.8000
CSL	2	0.5000	1.0000	0.5000	0.5000	0.9500	0.4000
HYP	3	0.0000	0.3330	0.0000	0.3330	0.8000	0.8000
ITI	43	0.2340	0.7380	0.1877	0.5890	0.9721	0.7440
All	**60**	**0.2192**	**0.7490**	**0.1873**	**0.6110**	**0.9650**	**0.7600**

CSL: Cross-Standard Linking; DFR: Direct Fact Retrieval; HYP: Hypothetical; ITI: Intel to Intersection; MHR: Multi-Hop Reasoning. Mitigation has no no-evidence cases.

TABLE VII
EFFECT OF FAISS/WEB FALLBACK ON THE COMPLETE 600-QUESTION CYQUAD BENCHMARK. ONLY THE 60 NO-EVIDENCE QUESTIONS ARE RE-ANSWERED.

Metric	Before	After	Δ	Relative
Accuracy	0.7538	0.8067	+0.0529	+7.0%
Completeness	0.7616	0.8040	+0.0423	+5.6%
Relevance	0.9092	0.8886	−0.0205	−2.3%

an absolute gain of 0.0529 and a relative gain of 7.0%. Mean completeness increases from 0.7616 to 0.8040, an absolute gain of 0.0423 and a relative gain of 5.6%. Mean relevance decreases from 0.9092 to 0.8886, an absolute change of −0.0205 (−2.3%). These results show that targeted FAISS and web retrieval can further improve overall accuracy and completeness by recovering no-evidence cases.

VI. CONCLUSION

This paper introduced CyQuAD, a benchmark of 600 open-ended cybersecurity question–answer pairs, and GraphQuAD,

an agentic method for knowledge-graph-grounded cybersecurity question answering. GraphQuAD generates and executes Cypher queries, revises unsuccessful queries using execution feedback, and invokes FAISS-based vector retrieval and web search when the graph path returns no usable evidence.

Relative to the No-KG method, GraphQuAD increases mean accuracy from 0.2565 to 0.7538 and completeness from 0.2313 to 0.7616. It retrieves usable graph evidence for 540 of the 600 questions, while the lower performance on the remaining cases identifies retrieval reliability as an important source of error. Applying fallback retrieval only to these cases further increases full-benchmark accuracy to 0.8067 and completeness to 0.8040, with a modest decrease in relevance.

These findings show that structured graph evidence can substantially improve the factual correctness and coverage of cybersecurity answers, while targeted fallback retrieval can recover many unsuccessful graph-retrieval cases. Together, CyQuAD and GraphQuAD provide a benchmark and method for developing cybersecurity QA systems that are not only responsive, but also accurate, complete, and supported by retrievable evidence.

Acknowledgment

This work was supported by the U.S. Department of Energy's Office of Science at the Pacific Northwest National Laboratory, which is operated by Battelle for the U.S. Department of Energy under contract DE-AC05-76RL01830. This work was also supported by the DOE Science Undergraduate Laboratory Internship Program (SULI) and the National Science Foundation (NSF) under Grants 2442595, IIS 2524379, NAIRR 250188.

References

[1] MITRE Corporation, "Common Vulnerabilities and Exposures (CVE)," https://www.cve.org/, 2026, accessed: Aug. 2, 2026.

[2] ——, "Common Weakness Enumeration (CWE)," https://cwe.mitre.org/, 2026, accessed: Aug. 2, 2026.

[3] ——, "Common Attack Pattern Enumeration and Classification (CAPEC)," https://capec.mitre.org/, 2026, accessed: Aug. 2, 2026.

[4] ——, "MITRE ATT&CK: Adversarial Tactics, Techniques, and Common Knowledge," https://attack.mitre.org/, 2026, accessed: Aug. 2, 2026.

[5] Z. Liu, "SecQA: A concise question-answering dataset for evaluating large language models in computer security," *arXiv preprint arXiv:2312.15838*, 2023. [Online]. Available: https://arxiv.org/abs/2312.15838

[6] D. Bhusal, M. T. Alam, L. Nguyen, A. Mahara, Z. Lightcap, R. Frazier, R. Fieblinger, G. L. Torales, B. A. Blakely, and N. Rastogi, "SECURE: Benchmarking large language models for cybersecurity," in *2024 Annual Computer Security Applications Conference (ACSAC)*. IEEE, 2024, pp. 15–30.

[7] M. T. Alam, D. Bhusal, L. Nguyen, and N. Rastogi, "CTIBench: A benchmark for evaluating LLMs in cyber threat intelligence," in *Advances in Neural Information Processing Systems*, vol. 37, 2024.

[8] M. T. Alam, D. Bhusal, S. Ahmad, N. Rastogi, and P. Worth, "Athenabench: A dynamic benchmark for evaluating llms in cyber threat intelligence," in *2025 Annual Computer Security Applications Conference Workshops (ACSAC Workshops)*. IEEE, 2025, pp. 443–450.

[9] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," in *Advances in neural information processing systems*, vol. 33, 2020.

[10] N. Francis, A. Green, P. Guagliardo, L. Libkin, T. Lindaaker, V. Marsault, S. Plantikow, M. Rydberg, P. Selmer, and A. Taylor, "Cypher: An evolving query language for property graphs," in *Proceedings of the 2018 international conference on management of data*, 2018, pp. 1433–1445.

[11] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus," *IEEE transactions on big data*, vol. 7, no. 3, pp. 535–547, 2019.

[12] M. Rahman, K. O. Piryani, A. M. Sanchez, S. Munikoti, L. De La Torre, M. S. Levin, M. Akbar, M. Hossain, M. Hasan, and M. Halappanavar, "Retrieval augmented generation for robust cyber defense," Pacific Northwest National Laboratory (PNNL), Richland, WA (United States), Tech. Rep., 2024.

[13] M. Rahman, A. Sanchez, K. Piryani, S. Das, S. Munikoti, L. de la Torre Quintana, M. Hasan, J. Aguayo, M. Akbar, S. Hossain *et al.*, "Generative ai for advanced cyber defense," *AI for Cybersecurity: Research and Practice*, pp. 109–146, 2025.

[14] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, D. Metropolitansky, R. O. Ness, and J. Larson, "From local to global: A graph rag approach to query-focused summarization," *arXiv preprint arXiv:2404.16130*, 2024.

[15] X. He, Y. Tian, Y. Sun, N. V. Chawla, T. Laurent, Y. LeCun, X. Bresson, and B. Hooi, "G-retriever: Retrieval-augmented generation for textual graph understanding and question answering," in *Advances in Neural Information Processing Systems*, vol. 37, 2024.

[16] M. G. Ozsoy, L. Messallem, J. Besga, and G. Minneci, "Text2cypher: Bridging natural language and graph databases," in *Proceedings of the Workshop on Generative AI and Knowledge Graphs (GenAIK)*, 2025, pp. 100–108.

[17] Y. Feng, S. Papicchio, and S. Rahman, "Cypherbench: Towards precise retrieval over full-scale modern knowledge graphs in the llm era," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025.

[18] D. Hamzic, F. Skopik, M. Landauer, M. Wurzenberger, and A. Rauber, "Beyond rag for cyber threat intelligence: A systematic evaluation of graph-based and agentic retrieval," *arXiv preprint arXiv:2604.11419*, 2026.

[19] Y. Zhu, A. Kellermann, D. Bowman, P. Li, A. Gupta, A. Danda, R. Fang, C. Jensen, E. Ihli, J. Benn, J. Geronimo, A. Dhir, S. Rao, K. Yu, T. Stone, and D. Kang, "Cve-bench: a benchmark for ai agents' ability to exploit real-world web application vulnerabilities," in *Proceedings of the 42nd International Conference on Machine Learning*, ser. ICML'25. JMLR.org, 2025.

[20] Z. Wang, T. Shi, J. He, M. Cai, J. Zhang, and D. Song, "Cybergym: Evaluating ai agents' real-world cybersecurity capabilities at scale," in *International Conference on Learning Representations*, 2026.

[21] Z. Wang, N. Schiller, H. Li, S. S. Narayana, M. Nasr, N. Carlini, X. Qi, E. Wallace, E. Bursztein, L. Invernizzi *et al.*, "Exploitgym: Can ai agents turn security vulnerabilities into real attacks?" *arXiv preprint arXiv:2605.11086*, 2026.

[22] M. Rahman, M. Akbar, J. T. Aguayo, M. S. Hossain, T. M. Ellis, S. S. Das, M. F. Babar, M. Hasan, A. Sanchez, L. F. de la Torre *et al.*, "Att&ck to cve: A large-scale automated knowledge graph for threat intelligence," in *2025 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2025, pp. 1044–1054.

[23] S. S. Das, E. Serra, M. Halappanavar, A. Pothen, and E. Al-Shaer, "V2w-bert: A framework for effective hierarchical multiclass classification of software vulnerabilities," in *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2021, pp. 1–12.

[24] S. S. Das, M. Halappanavar, A. Tumeo, E. Serra, A. Pothen, and E. Al-Shaer, "Vwc-bert: Scaling vulnerability–weakness–exploit mapping on modern ai accelerators," in *2022 IEEE International Conference on Big Data (Big Data)*. IEEE, 2022, pp. 1224–1229.

[25] S. Es, J. James, L. E. Anke, and S. Schockaert, "Ragas: Automated evaluation of retrieval augmented generation," in *Proceedings of the 18th conference of the european chapter of the association for computational linguistics: system demonstrations*, 2024, pp. 150–158.

[26] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical programming*, vol. 91, no. 2, pp. 201–213, 2002.

[27] J. L. Hintze and R. D. Nelson, "Violin plots: a box plot-density trace synergism," *The American Statistician*, vol. 52, no. 2, pp. 181–184, 1998.