GRID: GRAPH REPRESENTATION OF INTELLIGENCE DATA FOR SECURITY TEXT KNOWLEDGE GRAPH CONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

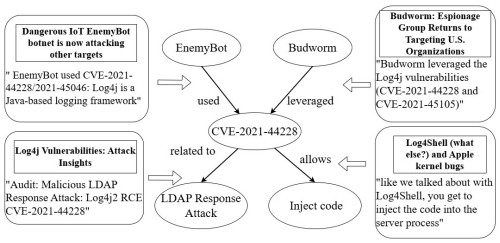
Security knowledge graphs can serve as computable and traceable external memory for security agents, yet existing LLMs lack domain knowledge grounded in real security text, and end-to-end document-to-graph training is hard to supervise with cheap and stable rewards. We present Graph Representation of Intelligence Data (**G.R.I.D.**), an end-to-end framework for security text knowledge graph construction. GRID builds security-domain supervision from CTI articles without manual labels, aligning each article with a traceable graph through graph extraction and knowledge-graph-conditioned article rewriting. It then reformulates document-to-graph learning as a scripted task bank of four-option multi-select questions and triple-level regex targets, which is cheaper and more stable than per-step LLM-judge scoring of full graphs. On a unified benchmark of 249 CTI articles from five sources (GRID, CASIE, CTINexus, MalKG, and SecureNLP), the post-trained Qwen3-4B-Instruct-2507 Task-bank Reward model reaches 84.62% source-averaged precision, 64.91% recall, and 68.53% Avg F1: the best recall and a near-tied top F1, with fewer tokens than the similarly strong CTINexus pipeline. The secondary End2End Reward model, trained with LLM-as-judge rewards, reaches 76.91% precision, 53.85% recall, and 58.06% Avg F1. The offline task bank, reusable across post-training runs, outperforms the online End2End Reward, Choice-only Reward, and End2End SFT without RL, and both article rewriting and article-complexity ordering matter. Finally, this supervision is inexpensive to generate and reusable, in two independent senses: with the open-weight GPT-OSS-120B generating the training data, the Qwen3-4B student still gains 13.8 absolute F1 points over its untrained baseline; and the task bank from our main experiments, reused with no new data generation, trains a different backbone, Llama-3.1-8B-Instruct, lifting it by 24.4 points.

1 INTRODUCTION

Cyber attacks are becoming more frequent, complex, and costly (Gao et al., 2018; Hutchins et al., 2011; Fang et al., 2022; Harris, 2014): in the United States, reported ransomware incidents rose by 22% in 2023 over 2022, and their costs by 74% (Office of the National Cyber Director, 2024). Yet only 35% of security decision-makers believe their organizations fully understand adversaries' tactics, techniques, and procedures (Mandiant, 2023).

Cyber Threat Intelligence (CTI) is therefore important for cyber defense (McMillan, 2013; Wagner et al., 2019). Structured CTI such as Indicators of Compromise (IOCs), Common Vulnerabilities and Exposures (CVEs), and cyber kill chains is useful but often misses attack context (Obrst et al., 2012; Liao et al., 2016; Catakoglu et al., 2016; Recorded Future, 2016; Senki, n.d.; The MITRE Corporation, 2020; Lockheed Martin, 2021; The MITRE Corporation, 2022). Unstructured CTI articles such as technical blogs and threat reports describe attack behaviors, attacker goals, exploited vulnerabilities (Dong et al., 2019), and malware evolution in far more detail (Liao et al., 2016), motivating automatic threat-knowledge extraction (Li et al., 2022; Satvat et al., 2021; Gao et al., 2022; Huang & Xiao, 2024).

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.



Summary:
CVE-2021-44228, commonly known as Log4Shell, is a critical vulnerability in Apache's Log4j logging library that allows attackers to execute code remotely on vulnerable systems. By exploiting this flaw, attackers can bypass authentication protocols, use code injection, and control affected systems. This vulnerability has been used by groups like Budworm and EnemyBot. Budworm targets U.S. organizations and the defense sector, leveraging Log4Shell for attacks via LDAP protocols and DLL side-loading. EnemyBot, on the other hand, uses it to compromise IoT devices and CMS servers, which often lack strong security controls.

Figure 1: Log4Shell (CVE-2021-44228): excerpts from four CTI articles and the knowledge graph extracted from them (left), and an attack summary (right)

Knowledge graphs suit this purpose because they represent security entities and their relations in one structure. Graphs support attack investigation: provenance graphs built from system auditing events enable forensic analysis and attack reconstruction (King & Chen, 2003; Fang et al., 2022; Xu et al., 2022), and graph-based threat knowledge can be matched against them to find known attack behaviors (Milajerdi et al., 2019). Think-on-Graph, G-Retriever, and recent graph-memory systems for agents also report performance gains when LLMs reason over graph-structured knowledge or memory (Sun et al., 2024; He et al., 2024; Chhikara et al., 2025; Rasmussen et al., 2025). Figure 1 shows a Log4Shell example with a knowledge graph and its attack summary from CTI articles.

In this paper, we propose Graph Representation of Intelligence Data (**G.R.I.D.**), a low-cost, complete pipeline for building and assessing LLM-based security knowledge graph extractors, rather than a standalone prompting recipe.

Challenges.

- *Lack of Integration of Security Domain Knowledge*: Most LLMs, especially small ones, are built for general use rather than security knowledge graph extraction, leaving practitioners dependent on expensive commercial APIs.
- *Lack of High-Quality CTI Article-Graph Alignment Data*: Training LLMs for CTI knowledge graph extraction requires supervision that tightly aligns real CTI text with graph outputs, but public annotated CTI datasets are small, narrow in domain, or labeled for other schemas.
- *Expensive Reward Signals for Open-Ended Knowledge Graph Extraction*: End-to-end graph extraction is open-ended, so RL rewards are hard to design, and even an available LLM judge is costly to run on full graphs at every step.
- *Shallow Shortcut Learning in Relation Extraction*: LLMs tend to predict relations from shortcuts such as prompt trigger words and entity-name priors (Tang et al., 2023; Wang et al., 2023) rather than from entity semantics, aliases, hierarchy, and relation constraints in CTI narratives.

Contributions.

- *Automatic Annotation of Article-Graph Alignment Data*: GRID first extracts a traceable knowledge graph that keeps verbatim evidence anchors, then revises the article against the graph to remove uncaptured CTI information while retaining non-CTI context, yielding high-quality article-graph alignments without large-scale manual annotation.
- *Low-Cost Task-Bank Reformulation for RL Training*: GRID reformulates open-ended graph extraction into scripted tasks, four-option multi-select questions and triple-level regex targets, replacing full-graph scoring with cheaper task-level checks. The task bank is generated once offline and reused across post-training runs and backbones; independently, a large open-weight model can also generate such a task bank.
- *Ontology-Guided CTI Knowledge Graph Extraction*: GRID designs a STIX-inspired (OASIS Cyber Threat Intelligence (CTI) TC, 2021) CTI ontology that explicitly models entity types, relation categories, aliases, and hierarchy, so that extraction depends on entity semantics and constraints rather than only shallow textual cues; together with the post-trained model, this lowers deployment cost.
- *Out-of-the-box Benchmark and Trustworthy Automatic Evaluation*: GRID builds a benchmark that combines manually annotated real-world CTI articles with existing security text datasets, paired with a trustworthy automatic evaluator based on text-provable precision and recall.

Evaluations. We evaluate GRID on a unified benchmark of 249 CTI articles from five sources (GRID, CASIE, CTINexus, MalKG, and SecureNLP; 49–50 articles each, averaging 191 to 11,000

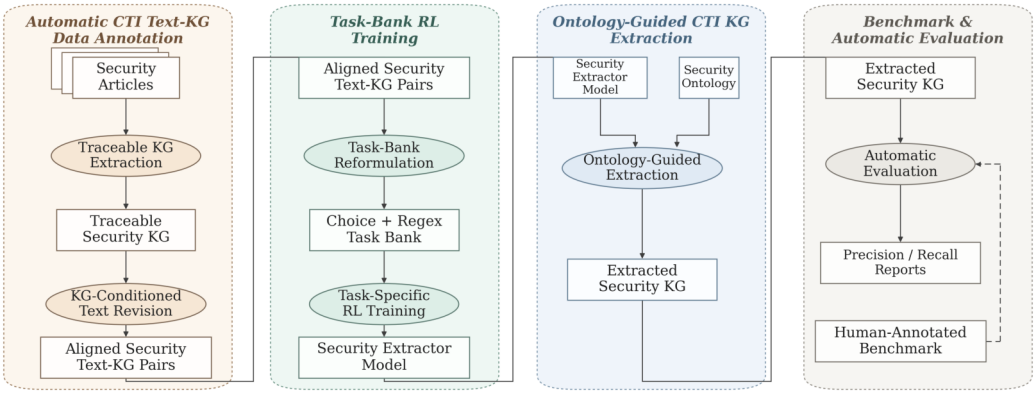


Figure 2: Overview of GRID

tokens and 7.94 to 68.66 ground-truth edges), scored by a calibrated LLM judge that reaches 86.0% agreement with human reviewers. The Task-bank Reward model (a post-trained Qwen3-4B-Instruct-2507) achieves source-averaged precision, recall, and F1 of 84.62%, 64.91%, and 68.53%, the best recall and a near-tied top F1, while using fewer total tokens than CTINexus, with the gap widening on longer articles; the End2End Reward model, trained with an online LLM-as-judge reward, reaches 76.91%, 53.85%, and 58.06%. RQ2 validates Task-bank Reward against online End2End Reward, Choice-only Reward, End2End SFT without RL, and the base model; its task bank costs about $60 once offline and is reused across later SFT and RL runs, versus about $942 of online LLM-as-judge cost for End2End Reward. In RQ3, under the same training budget, the full setting reaches its peak test F1 earlier than the variants without article rewriting or article-complexity ordering, and achieves the highest training reward. RQ4 shows two independent properties of this supervision: with the open-weight GPT-OSS-120B as the training-data generator, the student still gains 13.8 absolute F1 points over its untrained baseline; and the RQ3 task bank, reused unchanged on a different backbone, Llama-3.1-8B-Instruct, raises it by 24.4 points. Code and data are accessible at `https://github.com/anonymousauthorname/ProjectGRID`.

2 Approach

Figure 2 overviews GRID: Sections 2.1–2.3 build the task-bank training data from unlabeled CTI articles, Section 2.4 defines the local scripted rewards used in post-training, and Sections 2.5–2.6 describe ontology-guided two-step extraction and the indexed LLM-judge evaluation; Figure 9 in Appendix B shows the file-level flow.

2.1 Automatic Annotation of Article-Graph Alignment Data

GRID maps each raw CTI article to an aligned pair (a', G') through a two-stage annotate-and-revise loop (Algorithm 1 in Appendix B): it first extracts a traceable knowledge graph under a strict text-provable constraint, then rewrites the article against that graph, removing unsupported security content while preserving graph-grounded evidence and non-security context.

Each KG edge records its subject–relation–object triple together with textual evidence. For example, *"Reveton works with the Citadel malware platform."* yields *(Reveton, works with, Citadel)* together with *raw_sub_name = Reveton*, *raw_obj_name = the Citadel malware platform*, *raw_text_start = Reveton works with*, and *raw_text_end = works with the Citadel malware platform*. These anchors are verbatim phrases, not numeric offsets; the pipeline locates them by exact substring matching, so the LLM never predicts character or token indices.

Revision matters because a CTI fact left in the article without a KG edge creates a reward mismatch: extracting it earns nothing, since the task bank has no target for it (Figure 3).

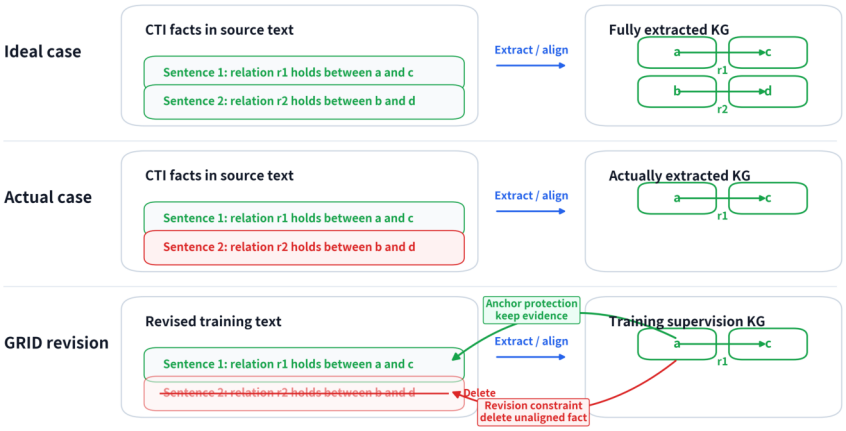


Figure 3: GRID article rewriting keeps KG-anchored evidence and deletes unaligned facts

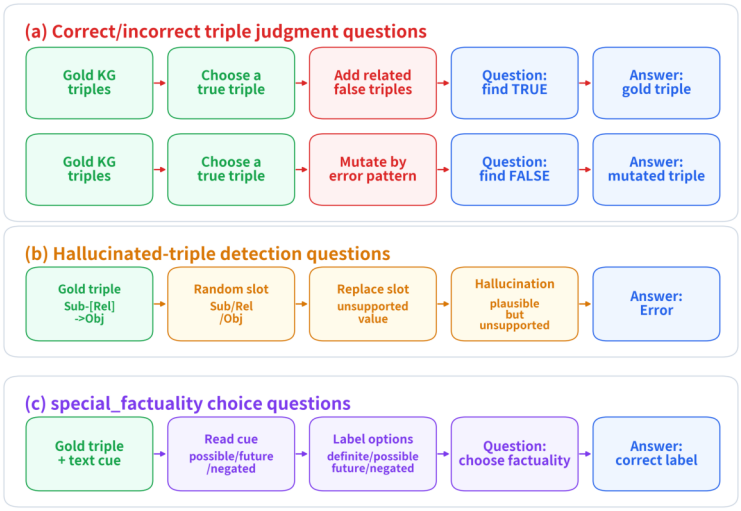


Figure 4: Task-bank question construction from an article-grounded KG

2.2 Task Bank Construction

Rather than rewarding full document-to-graph generation, GRID converts each alignment (a', G') into two easy-to-check RL task types: up to 20 four-option multi-select questions per article, each answered by any subset of 0–4 options (Appendix C), and one triple-level regex target per ground-truth KG edge, which reduces graph supervision to per-edge matching after entity- and relation-level normalization. Figure 4 shows how the three choice-question families are built from an article-grounded KG; Table 6 and Table 7 in Appendix B give the question patterns and regex rules.

2.3 Article Complexity Modeling

GRID assigns each article a scalar complexity score from ascending percentile ranks over the training corpus (Appendix B): C_{base} averages the ranks of article length, entity count, relation count, entity density, and relation density; C_{graph} aggregates the ranks of structural statistics (alias usage, connectivity, clustering, treewidth/core structure, non-bridge ratio, relation span, and crossing density); and $C_{\text{article}}(a) = 0.65\, C_{\text{base}}(a) + 0.35\, C_{\text{graph}}(a)$. The training set is sorted from easier to harder articles, and all tasks derived from one article inherit its score for article-complexity-ordered RL.

2.4 Reward Design for RL Training

The RL stage uses Soft Adaptive Policy Optimization (SAPO), a GRPO-style variant (Gao et al., 2025), with rewards computed locally by task type. A choice task earns 1.0 when the predicted answer set exactly matches the ground truth, 0.5 when the two sets overlap without matching, and 0 otherwise; both task types add a 0.1 format reward when the output contains the `<mythink>` reasoning block and the `<myresult>` answer block required by the prompt (total capped at 1.0). A regex task earns $r_{\text{regex}} = n_{\text{match}}/n_{\text{gt}}$, the fraction of ground-truth regex triplets whose subject, relation, and object are all matched by the same predicted triple. This recall-style reward raises output volume without costing precision: Task-bank Reward has the highest source-averaged precision of all GRID variants (84.62%, Table 2). Nor is selecting every option a shortcut: unless all four options are correct, it earns at most 0.5, while only the exact answer set earns 1.0.

2.5 Ontology-Guided Security Knowledge Graph Extraction

The ontology provides 33 entity types, 44 normalized relation categories, aliases, and hierarchy constraints, which guide the LLM to recover relations not stated through explicit verbs or relative clauses (prompt blocks in Figure 8 and the full inventory, both in Appendix A). It enforces text-provable truth: no completion from outside knowledge, no subject elevation (crediting a component's behavior to its parent entity), no behavior-to-structure conversion (reading "A uses B" as "B is part of A"), and no subject-changing chain deduction ("A uses B" and "B does C" do not yield "A does C"). Aliases, lineage, and categories are recorded through the `alias` and `parent_entity` fields or relation types such as `alias-of` and `categorized-as`, and unsupported isolated entities are removed by a connectivity recheck instead of being linked by guesswork. At inference, Step 1 proposes candidate entities and Step 2 finalizes entities and relations against the original article.

2.6 LLM-Based Automatic Evaluation

GRID scores precision and recall with an LLM evaluator whose two prompts follow the extractor's text-provable principle: a predicted edge counts only if the article supports it, so *(Microsoft, developed, Windows)* is a false positive unless the article states it. The precision prompt audits each predicted edge for direct support in the text or equivalence to a ground-truth edge; the recall prompt audits each ground-truth edge for capture by an equivalent predicted edge or a text-supported, subject-preserving combination of predicted edges. For every audited edge, the judge returns its index, a binary verdict, and the supporting text or matched predicted edge(s); Appendix D, Table 8 summarizes the rules.

3 Evaluation

We address four research questions:

- RQ1: How do the two main GRID systems compare with representative CTI knowledge-graph baselines?
- RQ2: How do the main post-training designs differ in effectiveness and engineering cost?
- RQ3: How much do article rewriting and article-complexity ordering matter within the primary Task-bank Reward setup?
- RQ4: Can a large open-weight model generate the training data, and can a task bank generated once be reused to train other backbones?

3.1 Evaluation Setup

Evaluation Dataset. Our benchmark contains 249 CTI articles (7,617 ground-truth edges) from five sources, keeping only articles whose ground-truth graphs have at least five edges and randomly sampling 50 per source: 49 of 59 manually annotated real-world GRID articles (avg. 1,102 tokens and 15.35 edges) and 50 articles each from CASIE (537 tokens, 7.94 edges), CTINexus (191, 11.80), MalKG (6,632, 48.90), and SecureNLP (11,000, 68.66) (Satyapanich et al., 2020; Cheng et al., 2025; Rastogi et al., 2021; Phandi et al., 2018).

Implementation and Training. We train with VERL (Sheng et al., 2024) and use Qwen3-4B-Instruct-2507 (Qwen Team, 2025a;b) as the base extractor for RQ1–RQ3; RQ4 additionally reuses the RQ3

Table 1: Per-source precision, recall, and F1 (%). Avg columns are arithmetic means over the five sources; bold marks the best value in each column

Method	Source															Avg		
	CASIE			CTINexus			GRID			MalKG			SecureNLP					
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
GRID (Task-bank)	81.80	**77.68**	**77.22**	86.69	81.10	82.58	84.18	**78.11**	**79.62**	84.43	38.55	44.66	**85.97**	49.10	58.58	84.62	**64.91**	68.53
GRID (End2End)	80.88	66.98	70.76	80.30	76.67	76.83	78.71	64.93	67.35	74.94	24.87	31.61	69.71	35.81	43.77	76.91	53.85	58.06
CTINexus	83.44	61.41	67.49	86.75	**91.02**	**87.83**	83.62	71.64	75.91	**88.71**	**39.42**	**48.24**	85.07	53.88	63.80	85.52	63.47	**68.66**
Cognee	48.87	43.75	39.79	63.98	64.58	61.80	57.13	52.01	50.24	63.74	27.73	30.33	68.04	48.74	52.54	60.35	47.36	46.94
LLM-CAKG	85.39	52.76	58.81	79.90	55.63	63.41	80.65	69.61	72.60	78.49	38.27	46.70	80.35	**65.16**	**68.70**	80.96	56.29	62.04
Graphiti	70.50	24.39	32.40	70.87	35.05	44.39	69.97	46.13	51.50	74.12	32.14	39.02	71.78	33.88	43.35	71.45	34.32	42.13
CTIKG	87.41	34.51	44.57	82.67	40.82	50.25	**84.28**	43.06	52.30	79.20	18.91	26.29	84.12	40.61	51.12	83.54	35.58	44.91
GraphRAG	**90.30**	12.43	16.69	**87.09**	33.79	43.81	77.28	35.70	44.07	88.10	21.32	30.25	84.93	20.26	25.62	**85.54**	24.70	32.09
AttacKG+	26.35	19.54	16.67	25.65	36.27	26.55	26.50	25.66	20.06	48.22	13.64	15.04	48.47	44.01	42.95	35.04	27.82	24.25
KnowGL	25.83	0.40	0.53	27.34	1.42	2.35	23.71	1.93	2.20	28.94	2.05	2.85	28.80	5.58	6.89	26.93	2.28	2.96

task bank and recipe to train Llama-3.1-8B-Instruct (Meta, 2024a) (backbone rationale in Section 4). The primary Task-bank Reward model is trained in three stages (SFT and choice-question RL on 800 CTI articles, then 280 steps of task-bank RL on 500; Appendix I), and the secondary End2End Reward model on 1,000 articles, rewarded online by the judged F1 of each full rollout graph. Training-data construction and the End2End training-side reward use GPT-5.3 Codex Medium (OpenAI, 2026a) in RQ1–RQ3; one RQ4 ablation instead uses the open-weight GPT-OSS-120B (OpenAI, 2025) as the generator. At test time, all models receive the full article.

LLM Judge Calibration. Exact string matching would penalize semantically correct but surface-divergent extractions, so all RQs use the indexed edge-wise protocol of Section 2.6 with a frozen GPT-5.4 mini judge (OpenAI, 2026b) (about $0.1 per article; Appendix E). On 378 audit items, each randomly assigned to one of three human reviewers, the judge agrees with humans on 80.6% of precision items (154/191) and 91.4% of recall items (171/187), 86.0% overall (325/378); the lower precision agreement reflects a judge stricter than the humans, who more often override LLM-FP to TP (24 cases) than the reverse (13; Appendix D.1). Precision, recall, and F1 are computed per article and averaged within each source, so a per-source F1 is not the harmonic mean of its precision and recall; Avg columns average the five sources.

Baselines. RQ1 compares the two GRID systems with eight baselines (adaptation details in Appendix F): CTINexus (Cheng et al., 2025), CTIKG (Huang & Xiao, 2024), Cognee (Cognee Contributors, 2025), LLM-CAKG (Wang et al., 2025), Graphiti (Zep, 2025), GraphRAG (Microsoft, 2024a), KnowGL (Rossiello et al., 2023), and AttacKG+ (Zhang et al., 2025). All LLM-based systems run on the same Qwen3-4B-Instruct-2507 (Qwen Team, 2025a;b) with temperature 0.7 and a 32,768-token generation budget, so the comparison asks how far each method takes the same 4B model; only GRID additionally post-trains it (Section 2). We also surveyed aCTIon (Siracusano et al., 2023), LLM-TIKG (Hu et al., 2024), CTI-Thinker (Yang et al., 2026), CodeKGC (Bi et al., 2024), SecTKG (Sun et al., 2023), and HRTC (Yue et al., 2024), but could not include them because of incomplete public implementations, unavailable source data, or access restrictions (aCTIon, for example, has released neither its code nor its data). Among non-LLM methods we keep only KnowGL, since REBEL (Huguet Cabot & Navigli, 2021) and EXTRACTOR (Satvat et al., 2021) scored far lower in preliminary runs.

3.2 RQ1: Comparison with Baselines

Task-bank achieves the best source-averaged recall (64.91%, Table 1). CTINexus's lower recall (63.47%) means it recovers less of the ground-truth security information, although it attains a nearly identical Avg F1 (68.66% vs. 68.53%) and slightly higher Avg precision (85.52% vs. 84.62%), doing best on its own source, the shortest in our suite (191 tokens on average). Task-bank achieves the best recall and F1 on both the GRID source (78.11%, 79.62%) and CASIE (77.68%, 77.22%), and the best precision on SecureNLP (85.97%); among 84 human-confirmed precision errors in a random sample of its judged false positives on 50 articles, 55 are malformed triples or boundary shifts rather than unsupported facts (Appendix H). End2End Reward reaches 58.06% Avg F1, and GraphRAG attains the highest Avg precision (85.54%) but only 32.09% Avg F1 because its recall collapses. GRID runs a fixed two-step interface, whereas CTINexus's multi-stage workflow grows in input tokens with the number of entities and, further, in requests with the number of disconnected subgraphs (Cheng et al., 2025).

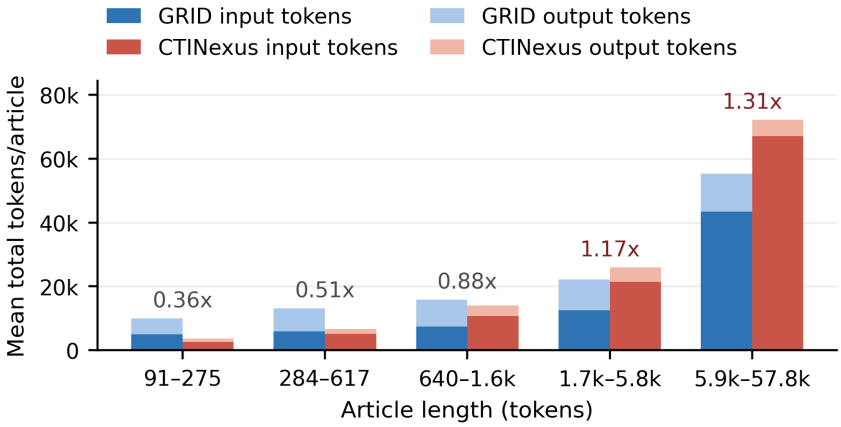


Figure 5: Tokens per article on the 100-article subset by length bin; labels are CTINexus/GRID ratios

Table 2: Post-training variants in GRID (five-source averages; bold marks the best value)

Post-training design	Precision (%)	Recall (%)	F1 (%)
Task-bank Reward	**84.62**	**64.91**	**68.53**
End2End Reward	76.91	53.85	58.06
Choice-only Reward	72.03	33.73	40.56
End2End SFT without RL	69.75	30.48	37.13
No post-training	73.99	27.21	35.44

On a 100-article subset with 20 articles from each source, CTINexus uses 1.44× the input-side tokens of GRID, and its total-token cost grows faster with article length, reaching 1.17× and 1.31× GRID's in the two longest bins (Figure 5).

3.3 RQ2: Post-Training Variants in the GRID Framework

Table 2 adds three weaker variants: Choice-only Reward (the task bank without its regex branch), End2End SFT without RL, and the untrained base model. Task-bank Reward remains the strongest setting at 68.53% Avg F1 and 64.91% recall, ahead of End2End Reward (58.06%), Choice-only Reward (40.56%), End2End SFT without RL (37.13%), and No post-training (35.44%); the regex branch drives recall, which Task-bank raises by 31.18 percentage points over Choice-only.

End2End Reward still trails Task-bank Reward by 11.06 recall and 10.47 F1 percentage points at a much higher cost. Task-bank's one-time offline cost is about $60, and the resulting bank is reused across later SFT and RL runs, whereas End2End Reward judges 512 graphs online at every RL step and reaches about $942 within 13 steps (Appendix I). The two runs are thus compared under this cost constraint, not as a matched ablation: End2End Reward gets more articles (1,000 vs. 800) and larger batches (64 prompts × 8 rollouts vs. 24 × 4 in the final task-bank stage).

3.4 RQ3: Ablation of Article Rewriting and Article-Complexity Ordering

We compare three variants under matched RL settings (the same Qwen3-4B-Instruct-2507 model, 500 training articles, reward function, and hyperparameters, with a shared 225-step budget): the **Full setting**, with revised-article supervision and article-complexity ordering; **w/o article-complexity ordering**, with articles in random order; and **w/o article rewriting**, with supervision regenerated from raw article–KG pairs and the ordering kept. Because each variant is scored at every checkpoint along its RL curve, evaluation uses a fixed 50-article subset (ten per source).

The full setting peaks far earlier: 0.537 at step 100, versus 0.532 at step 225 for w/o article rewriting and 0.382 at step 150 for w/o article-complexity ordering. At step 100 it is already far ahead of both ablations (0.537 vs. 0.256 and 0.265); it also stays above the random-order ablation at every checkpoint and ends with the highest training reward (Table 3, Figure 6).

Table 3: RQ3 ablation on the fixed 50-article subset under a shared 225-step budget (mean article-level F1 on a 0–1 scale, best in bold; train reward is the EMA-smoothed RL training reward at step 225, as plotted in Figure 6(a))

Setting	Best F1	Step of best	F1 at step 100	Train reward
Full setting	**0.537**	100	**0.537**	**0.766**
w/o article rewriting	0.532	225	0.256	0.678
w/o article-complexity ordering	0.382	150	0.265	0.722

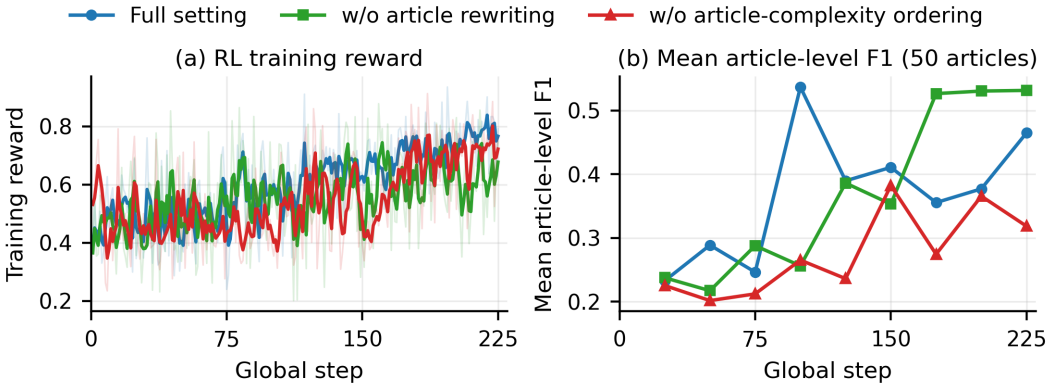


Figure 6: RQ3 ablation curves up to 225 steps. (a) RL training reward: faint lines are raw per-step values and bold lines a first-order EMA with smoothing factor 0.6. (b) Mean article-level F1 on the fixed 50-article subset

3.5 RQ4: Low-Cost Data Generation and Cross-Backbone Reuse

Open-weight training-data generator. Following the RQ3 setting (500 training articles, Qwen3-4B student), we replace the commercial training-data generator with the open-weight GPT-OSS-120B (OpenAI, 2025). On the 72 articles valid for all five checkpoints (Figure 7(a)), F1 rises from 0.438 for the untrained student to 0.537 after SFT and 0.576 after 100 RL steps (+0.138), with recall rising from 0.339 to 0.593. A large open-weight model that fits on a single 80GB GPU (OpenAI, 2025) can therefore generate effective training data without a commercial API.

Reuse on another backbone. Reusing the RQ3 task bank and RL hyperparameters unchanged, with no new data generation, we train Llama-3.1-8B-Instruct (Meta, 2024a), a backbone from a different model family. On the 136 articles valid for its base model and all eight RL checkpoints, F1 rises from 0.189 to 0.433 at step 28 (+0.244), overtaking the untrained Qwen3-4B-Instruct (0.386; Figure 7(b)); the gain holds without the intersection (+0.217 on the 202 articles valid for both the base model and step 28, +0.218 on each checkpoint's own valid articles).

The two ablations are independent: a large open-weight model can generate the supervision, and the RQ3 task bank, generated by GPT-5.3 Codex Medium, is reused across post-training runs (RQ2) and backbones. The generator ablation adds 600 SFT steps before RL, whereas the Llama run starts RL from the base model; both average over common-valid articles (Appendix I), so their scores are not directly comparable with Table 1.

4 Discussion

GRID uses a two-step extractor rather than an agent loop, because agent-loop RL needs a more complex reward interface and current open-source stacks remain immature for stable optimization. We chose Qwen3-4B-Instruct-2507 (Qwen Team, 2025a;b) for faster training and inference than 8B and 14B models under full-parameter post-training on our 4×RTX 6000 Ada setup, and because a recent fine-tuning report (distil labs, 2025) ranks it ahead of Llama-3.1-8B-Instruct (Meta, 2024a) and Llama-3.2-3B-Instruct. The latest publicly released small Llama checkpoint still dates back to September 2024 (Meta, 2024b), and Section 3.5 reports a Llama-3.1-8B-Instruct ablation.

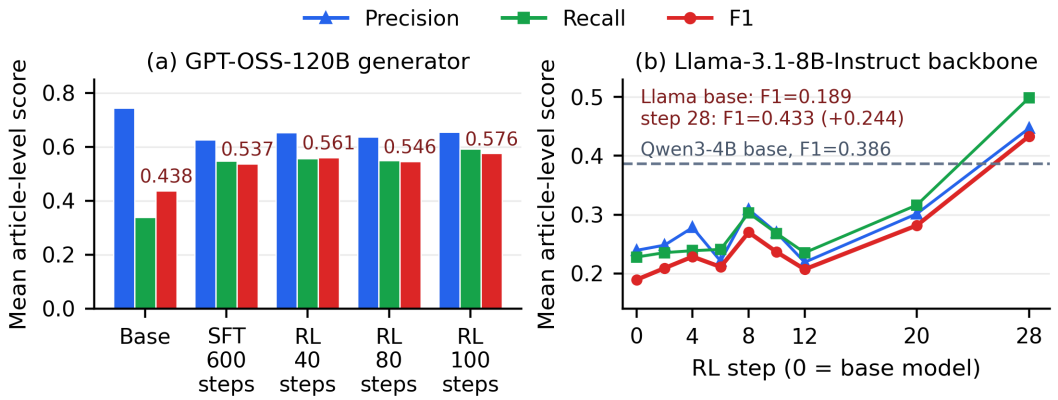


Figure 7: RQ4 ablations. (a) GPT-OSS-120B as the training-data generator (72 articles). (b) Llama-3.1-8B-Instruct trained on the reused RQ3 task bank (136 articles)

5 Related Work

Security knowledge graph extraction. Methods range from supervised pipelines such as EXTRACTOR (Satvat et al., 2021) and LADDER (Alam et al., 2023) to LLM-based systems such as CTIKG (Huang & Xiao, 2024), CTINexus (Cheng et al., 2025), CTI-Thinker (Yang et al., 2026), LLM-TIKG (Hu et al., 2024), LLM-CAKG (Wang et al., 2025), AttacKG+ (Zhang et al., 2025), and aCTIon (Siracusano et al., 2023). Supervised pipelines train task-specific extractors on narrow annotations (e.g., LADDER is trained only on Android-malware reports). Most LLM-based systems improve the inference-time pipeline around a large commercial LLM through prompting and multi-stage extraction (e.g., aCTIon prompts GPT-3.5 to extract STIX bundles); LLM-TIKG and CTI-Thinker LoRA-fine-tune an extractor but release neither code nor weights. RL-based relation extraction such as R1-RE (Dai et al., 2025) classifies relations over given entity pairs rather than extracting article-level graphs. In contrast, GRID post-trains a small open-source extractor on automatically aligned, traceable supervision with a scripted task-bank reward, and releases the pipeline and weights.

CTI datasets. We surveyed the annotated CTI datasets CASIE, SecureNLP, MalKG, CTINexus, LADDER, aCTIon, CTI-HAL (Della Penna et al., 2025), AttacKG (Li et al., 2022), AISecKG (Agrawal et al., 2023), CyberNER (Ech-Chammakhy et al., 2025), CTIBench (Alam et al., 2024), rcATT (Legoy et al., 2020), TRAM (MITRE Center for Threat-Informed Defense, 2023), CS13K (Li et al., 2024), and BVTED (Liu et al., 2024), as well as SciNLP (Duan et al., 2025), a full-text relation benchmark from another field. They are small, narrow in domain, labeled for other schemas, or unreleased (Appendix G gives the reason for each), and none provides traceable article–graph pairs at the scale GRID trains on, so we use CASIE, CTINexus, MalKG, and SecureNLP only as evaluation sources.

6 Conclusion

We presented GRID, which combines article–graph alignment, task-bank post-training, and ontology-guided inference for CTI knowledge graph extraction. On five CTI sources, the Task-bank Reward model achieved the best source-averaged recall and a near-tied top Avg F1 with fewer tokens than the similarly strong CTINexus pipeline; RQ2/RQ3 showed reusable task-bank supervision and gains from KG-conditioned article rewriting plus article-complexity ordering, and RQ4 showed two independent facts: a large open-weight model can generate this supervision, and the existing RQ3 task bank trains another backbone. Since each task is traceable to its source article and scored by a deterministic reward, the task bank also provides a foundation for future agentic training in CTI.

AI use statement

In line with the ICLR 2027 AI Policy for Authors (ICLR, 2026), we disclose our use of generative AI. We used LLMs to generate the training supervision, to filter source articles by topic, to judge precision and recall in evaluation, and to refine and systematize the code. The research idea, the method, and

the initial implementation are the authors' own, and the GRID benchmark articles were annotated manually. The paper was drafted by the authors and polished with AI assistance. We reviewed all AI-assisted work: the judge was calibrated against 378 human-reviewed items (86.0% agreement), and generated alignments and questions were spot-checked manually. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

Reproducibility statement

The data artifacts, code, prompts, and evaluation scripts used in this work are available in the anonymous project repository at https://github.com/anonymousauthorname/ProjectGRID. The corresponding post-trained model weights are available in an anonymous Hugging Face repository, and its link is provided in the GitHub repository. The repository also provides a Dockerfile and a portable Docker entrypoint. The container supports user-specified environment variables, including the input file, OpenAI-compatible LLM endpoint, LLM key, and LLM model. It provides executable entry points for: (1) environment/configuration checking, (2) training-Parquet construction, (3) one-step local RL smoke training and model export, (4) knowledge-graph generation, and (5) evaluation.

Ethics statement

This work uses only publicly available data: the CTI articles in our corpus and benchmark come from public security blogs, threat reports, and the public CASIE, CTINexus, MalKG, SecureNLP, and Primus datasets. We do not collect or process personal data, and the extracted knowledge graphs describe attack behaviors, malware, and infrastructure rather than individuals.

References

Garima Agrawal, Kuntal Pal, Yuli Deng, Huan Liu, and Chitta Baral. AISecKG: Knowledge graph dataset for cybersecurity education. In *Proceedings of the AAAI 2023 Spring Symposium on Challenges Requiring the Combination of Machine Learning and Knowledge Engineering (AAAI-MAKE 2023)*, volume 3433 of *CEUR Workshop Proceedings*, 2023.

Md Tanvirul Alam, Dipkamal Bhusal, Youngja Park, and Nidhi Rastogi. Looking beyond IoCs: Automatically extracting attack patterns from external CTI. In *Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*, pp. 92–108, 2023.

Md Tanvirul Alam, Dipkamal Bhusal, Le Nguyen, and Nidhi Rastogi. CTIBench: A benchmark for evaluating LLMs in cyber threat intelligence. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 37, pp. 50805–50825, 2024.

Zhen Bi, Jing Chen, Yinuo Jiang, Feiyu Xiong, Wei Guo, Huajun Chen, and Ningyu Zhang. CodeKGC: Code language model for generative knowledge graph construction. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 23(3):1–16, 2024.

Onur Catakoglu, Marco Balduzzi, and Davide Balzarotti. Automatic extraction of indicators of compromise for web applications. In *Proceedings of the 25th International Conference on World Wide Web (WWW)*, pp. 333–343, 2016.

Yutong Cheng, Osama Bajaber, Saimon Amanuel Tsegai, Dawn Song, and Peng Gao. CTINexus: Automatic cyber threat intelligence knowledge graph construction using large language models. In *2025 IEEE 10th European Symposium on Security and Privacy (EuroS&P)*, pp. 923–938. IEEE, 2025.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

Cisco. Common types of cyber attacks, 2024. URL https://www.cisco.com/c/en/us/products/security/common-cyberattacks.html.

Cognee Contributors. Cognee, 2025. URL https://github.com/topoteretes/cognee. GitHub repository.

CrowdStrike. CrowdStrike 2024 global threat report, 2024. URL https://www.crowdstrike.com/en-us/resources/reports/crowdstrike-2024-global-threat-report/.

Runpeng Dai, Tong Zheng, Run Yang, and Hongtu Zhu. R1-RE: Cross-domain relationship extraction with RLVR. *arXiv preprint arXiv:2507.04642v1*, 2025.

Sofia Della Penna, Roberto Natella, Vittorio Orbinato, Lorenzo Parracino, and Luciano Pianese. CTI-HAL: A human-annotated dataset for cyber threat intelligence analysis. *arXiv preprint arXiv:2504.05866*, 2025.

distil labs. We benchmarked 12 small language models across 8 tasks to find the best base model for fine-tuning, December 2025. URL https://www.distillabs.ai/blog/we-benchmarked-12-small-language-models-across-8-tasks-to-find-the-best-base-model-for-fine-tuning/. Accessed: 2026-03-29.

Ying Dong, Wenbo Guo, Yueqi Chen, Xinyu Xing, Yuqing Zhang, and Gang Wang. Towards the detection of inconsistencies in public security vulnerability reports. In *28th USENIX Security Symposium (USENIX Security 19)*, pp. 869–885, 2019.

Decheng Duan, Yingyi Zhang, Jitong Peng, and Chengzhi Zhang. SciNLP: A domain-specific benchmark for full-text scientific entity and relation extraction in NLP. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 14473–14486, 2025.

Yasir Ech-Chammakhy, Anas Motii, Anass Rabii, Oussama Azrara, and Jaafar Chbili. CyberNER: A harmonized STIX corpus for cybersecurity named entity recognition. In *2025 IEEE 24th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 2190–2197, 2025.

Pengcheng Fang, Peng Gao, Changlin Liu, Erman Ayday, Kangkook Jee, Ting Wang, Yanfang Ye, Zhuotao Liu, and Xusheng Xiao. Back-propagating system dependency impact for attack investigation. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 2461–2478, 2022.

Chang Gao, Chujie Zheng, Xiong-Hui Chen, Kai Dang, Shixuan Liu, Bowen Yu, An Yang, Shuai Bai, Jingren Zhou, and Junyang Lin. Soft adaptive policy optimization. *arXiv preprint arXiv:2511.20347*, 2025.

Peng Gao, Xusheng Xiao, Zhichun Li, Fengyuan Xu, Sanjeev R. Kulkarni, and Prateek Mittal. AIQL: Enabling efficient attack investigation from system monitoring data. In *USENIX Annual Technical Conference (ATC)*, pp. 113–126, 2018.

Peng Gao, Xiaoyuan Liu, Edward Choi, Sibo Ma, Xinyu Yang, Zhengjie Ji, Zilin Zhang, and Dawn Song. ThreatKG: A threat knowledge graph for automated open-source cyber threat intelligence gathering and management. *arXiv preprint arXiv:2212.10388v1*, 2022.

Elizabeth A. Harris. Data breach hurts profit at Target. The New York Times, 2014. URL https://www.nytimes.com/2014/02/27/business/target-reports-on-fourth-quarter-earnings.html.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-Retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Advances in Neural Information Processing Systems*, 37:132876–132907, 2024.

Yuelin Hu, Futai Zou, Jiajia Han, Xin Sun, and Yilei Wang. LLM-TIKG: Threat intelligence knowledge graph construction utilizing large language model. *Computers & Security*, 145:103999, 2024.

Liangyi Huang and Xusheng Xiao. CTIKG: LLM-powered knowledge graph construction from cyber threat intelligence. In *First Conference on Language Modeling*, 2024.

Pere-Lluís Huguet Cabot and Roberto Navigli. REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 2370–2381, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.findings-emnlp.204.

Eric M Hutchins, Michael J Cloppert, and Rohan M Amin. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. *Leading Issues in Information Warfare & Security Research*, 1:80, 2011.

ICLR. ICLR 2027 AI policy for authors, 2026. URL https://iclr.cc/Conferences/2027/AIPolicyForAuthors. Accessed: 2026-09-19.

Samuel T. King and Peter M. Chen. Backtracking intrusions. In *ACM Symposium on Operating Systems Principles (SOSP)*, pp. 223–236. ACM, 2003.

Valentine Legoy, Marco Caselli, Christin Seifert, and Andreas Peter. Automated retrieval of ATT&CK tactics and techniques for cyber threat reports. *arXiv preprint arXiv:2004.14322*, 2020.

Hongyi Li, Ze Shi, Chengwei Pan, Di Zhao, and Nan Sun. Cybersecurity knowledge graphs construction and quality assessment. *Complex & Intelligent Systems*, 10(1):1201–1217, 2024.

Zhenyuan Li, Jun Zeng, Yan Chen, and Zhenkai Liang. AttacKG: Constructing technique knowledge graph from cyber threat intelligence reports. In *European Symposium on Research in Computer Security*, pp. 589–609. Springer, 2022.

Xiaojing Liao, Kan Yuan, XiaoFeng Wang, Zhou Li, Luyi Xing, and Raheem Beyah. Acing the IOC game: Toward automatic discovery and analysis of open-source cyber threat intelligence. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 755–766, 2016.

Swee Kiat Lim, Aldrian Obaja Muis, Wei Lu, and Chen Hui Ong. MalwareTextDB: A database for annotated malware articles. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1557–1567. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1143. URL https://aclanthology.org/P17-1143/.

Kai Liu, Yi Wang, Zhaoyun Ding, Aiping Li, and Weiming Zhang. BVTED: A specialized bilingual (Chinese–English) dataset for vulnerability triple extraction tasks. *Applied Sciences*, 14(16):7310, 2024.

Lockheed Martin. Cyber Kill Chain, 2021. URL https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html.

Mandiant. Global perspectives on threat intelligence, 2023. URL https://assets.starlinkme.net/gitex-vendor-assets/mandiant/Global%20Perspectives%20on%20Threat%20Intelligence.pdf.

Rob McMillan. Definition: Threat intelligence. Gartner Research, 2013. URL https://www.gartner.com/en/documents/2487216.

Meta. Llama-3.1-8B-Instruct, July 2024a. URL https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct. Released: 2024-07-23; accessed: 2026-03-29.

Meta. Llama-3.2-3B, September 2024b. URL https://huggingface.co/meta-llama/Llama-3.2-3B. Released: 2024-09-25, accessed: 2026-03-29.

Microsoft. GraphRAG, 2024a. URL https://github.com/microsoft/graphrag. GitHub repository.

Microsoft. What is a cyberattack?, 2024b. URL https://www.microsoft.com/en-us/security/business/security-101/what-is-a-cyberattack.

Sadegh M. Milajerdi, Birhanu Eshete, Rigel Gjomemo, and V.N. Venkatakrishnan. POIROT: Aligning attack behavior with kernel audit records for cyber threat hunting. In *ACM Conference on Computer and Communications Security (CCS)*, pp. 1795–1812, 2019.

MITRE Center for Threat-Informed Defense. TRAM: Threat Report ATT&CK Mapper, 2023. URL https://github.com/center-for-threat-informed-defense/tram. GitHub repository, release v1.3.0; accessed: 2026-09-25.

OASIS Cyber Threat Intelligence (CTI) TC. STIX version 2.1. OASIS Standard, 2021. URL https://docs.oasis-open.org/cti/stix/v2.1/stix-v2.1.html.

Leo Obrst, Penny Chase, and Richard Markeloff. Developing an ontology of the cyber security domain. In *Proceedings of the Seventh International Conference on Semantic Technologies for Intelligence, Defense, and Security (STIDS)*, volume 966 of *CEUR Workshop Proceedings*, pp. 49–56, 2012.

Office of the National Cyber Director. 2024 report on the cybersecurity posture of the United States, 2024. URL https://bidenwhitehouse.archives.gov/wp-content/uploads/2024/05/2024-Report-on-the-Cybersecurity-Posture-of-the-United-States.pdf.

OpenAI. gpt-oss-120b, August 2025. URL https://huggingface.co/openai/gpt-oss-120b. Open-weight model; accessed: 2026-09-23.

OpenAI. Introducing GPT-5.3-Codex, February 2026a. URL https://openai.com/index/introducing-gpt-5-3-codex/.

OpenAI. Introducing GPT-5.4 mini and nano, March 2026b. URL https://openai.com/index/introducing-gpt-5-4-mini-and-nano/.

Palo Alto Networks Unit 42. How the EITest campaign's path to Angler EK evolved over time, 2016. URL https://unit42.paloaltonetworks.com/unit42-how-the-eltest-campaigns-path-to-angler-ek-evolved-over-time/.

Peter Phandi, Amila Silva, and Wei Lu. SemEval-2018 task 8: Semantic extraction from CybersecURity REports using natural language processing (SecureNLP). In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pp. 697–706. Association for Computational Linguistics, 2018. doi: 10.18653/v1/S18-1113. URL https://aclanthology.org/S18-1113/.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a. doi: 10.48550/arXiv.2505.09388. URL https://arxiv.org/abs/2505.09388.

Qwen Team. Qwen3-4B-Instruct-2507, 2025b. URL https://huggingface.co/Qwen/Qwen3-4B-Instruct-2507. Accessed: 2026-03-29.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A temporal knowledge graph architecture for agent memory. *arXiv preprint arXiv:2501.13956*, 2025.

Nidhi Rastogi, Sharmishtha Dutta, Ryan Christian, Mohammad Zaki, Alex Gittens, and Charu Aggarwal. Information prediction using knowledge graphs for contextual malware threat intelligence. *arXiv preprint arXiv:2102.05571v1*, 2021. URL https://arxiv.org/abs/2102.05571v1.

Recorded Future. Real-time threat intelligence, 2016. URL https://www.recordedfuture.com/.

Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Nandana Mihindukulasooriya, Owen Cornec, and Alfio Massimiliano Gliozzo. KnowGL: Knowledge generation and linking from text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 16476–16478, 2023.

Kiavash Satvat, Rigel Gjomemo, and VN Venkatakrishnan. EXTRACTOR: Extracting attack behavior from threat reports. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 598–615. IEEE, 2021.

Taneeya Satyapanich, Francis Ferraro, and Tim Finin. CASIE: Extracting cybersecurity event information from text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8749–8757, 2020.

Securelist. Threat categories, 2024. URL https://securelist.com/threat-categories/.

Senki. Open source threat intelligence feeds, n.d. URL https://www.senki.org/operators-security-toolkit/open-source-threat-intelligence-feeds/. Accessed: 2026-09-24.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. *arXiv preprint arXiv:2409.19256*, 2024.

Giuseppe Siracusano, Davide Sanvito, Roberto Gonzalez, Manikantan Srinivasan, Sivakaman Kamatchi, Wataru Takahashi, Masaru Kawakita, Takahiro Kakumaru, and Roberto Bifulco. Time for aCTIon: Automated analysis of cyber threat intelligence in the wild. *arXiv preprint arXiv:2307.10214*, 2023.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. Think-on-Graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Siqi Sun, Cheng Huang, Tiejun Wu, and Yi Shen. SecTKG: A knowledge graph for open-source security tools. *International Journal of Intelligent Systems*, 2023(1):4464974, 2023.

Ruixiang Tang, Dehan Kong, Longtao Huang, and Hui Xue. Large language models can be lazy learners: Analyze shortcuts in in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 4645–4657, 2023.

The MITRE Corporation. Common Vulnerabilities and Exposures (CVE), 2020. URL https://www.cve.org/.

The MITRE Corporation. MITRE ATT&CK, 2022. URL https://attack.mitre.org/.

Trend Micro. Ransomware spotlight: Magniber, 2023. URL https://www.trendmicro.com/vinfo/us/security/news/ransomware-spotlight/ransomware-spotlight-magniber.

VirusTotal. VirusTotal - free online virus, malware and URL scanner, n.d. URL https://www.virustotal.com/en/. accessed 2026-09-24.

Thomas D Wagner, Khaled Mahbub, Esther Palomar, and Ali E Abdallah. Cyber threat intelligence sharing: Survey and research directions. *Computers & Security*, 87:101589, 2019.

Fei Wang, Wenjie Mo, Yiwei Wang, Wenxuan Zhou, and Muhao Chen. A causal view of entity bias in (large) language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 15173–15184, 2023.

Zhihua Wang, Siyuan Fei, Youlin Hu, Dacheng Shan, Shitao Xiao, Lizhao You, and Peijun Chen. Automated attack knowledge graph construction with large language models. In *Proceedings of the 2025 2nd International Conference on Computer and Multimedia Technology*, pp. 700–706, 2025.

Zhiqiang Xu, Pengcheng Fang, Changlin Liu, Xusheng Xiao, Yu Wen, and Dan Meng. DepComm: Graph summarization on system audit logs for attack investigation. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 540–557. IEEE, 2022.

Xiuzhang Yang, Ruijie Zhong, Yuling Chen, Guojun Peng, Di Yao, Chaofan Chen, Chenyang Wang, Dongni Zhang, Yilin Zhou, and Zixuan Yang. CTI-Thinker: an LLM-driven system for CTI knowledge graph construction and attack reasoning. *Cybersecurity*, 9(1):106, 2026.

Yao-Ching Yu, Tsun-Han Chiang, Cheng-Wei Tsai, Chien-Ming Huang, and Wen-Kwang Tsao. Primus: A pioneering collection of open-source datasets for cybersecurity LLM training. *arXiv preprint arXiv:2502.11191*, 2025. URL https://arxiv.org/abs/2502.11191.

HuanZhou Yue, XuRen Wang, Rong Chen, ZhengWei Jiang, YuXia Fu, and Jun Jiang. HRTC: A triplet joint extraction model based on cyber threat intelligence. In *International Conference on Knowledge Science, Engineering and Management*, pp. 214–223. Springer, 2024.

Zep. Graphiti, 2025. URL https://github.com/getzep/graphiti. GitHub repository.

Yongheng Zhang, Tingwen Du, Yunshan Ma, Xiang Wang, Yi Xie, Guozheng Yang, Yuliang Lu, and Ee-Chien Chang. AttacKG+: Boosting attack graph construction with large language models. *Computers & Security*, 150:104220, 2025.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

Prompt Section	
Global Goal	
Text Truth	Text-only facts
	No external facts
	No subject shift
Entity Schema	Canonical entities
	Alias + parent links
	Type-aware normalization
Relation Schema	Behavior / hierarchy / attribution
	Explicit + implicit
	Direction-aware edges
Graph Validity Check	
Output Format	
Input: <Security Article>	

Figure 8: Prompt blocks for the GRID extractor

A Security Ontology Inventory

The ontology used by GRID defines both the structured fields attached to graph elements and the controlled vocabularies used during extraction and validation. Each entity records `name`, `type`, `alias`, and `parent_entity`. Each relation records `sub`, `rel`, `rel_type`, and `obj`: `rel` is the relation phrase as stated in the article (or, when the relation is only implied, its type name or a short paraphrase), and `rel_type` maps it to the closest category in the relation inventory (Table 5; entity types in Table 4), with `other` as the fallback.

Our ontology is STIX-inspired rather than a copy: manual labeling showed that CTI articles repeatedly need the following refinements beyond the STIX inventory.

- Merged attacker/activity labels: `threat-actor-or-intrusion-set`.
- Offensive tools vs. legitimate-but-abused software: `hacker-tool`, `general-software`.
- Component-level types: `detailed-part-of-malware-or-hackertool`, `detailed-part-of-general-software`.
- Analysis- or fallback-oriented types: `security-product`, `malware-analysis-document-or-publication-or-conference`, `abstract-concept`, `generic-noun`, `noise`.

These refinements guide the LLM toward structural relations among entities rather than turning extraction into a closed labeling task; extraction remains open-ended, with the ontology as a lightweight scaffold for reasoning and normalization.

Table 4: Entity types in the GRID ontology

user-account	identity	threat-actor-or-intrusion-set	campaign
malware	hacker-tool	general-software	security-product
detailed-part-of-malware-or-hackertool	detailed-part-of-general-software	attack-pattern	vulnerability
file	process	windows-registry-key	course-of-action
url	domain-name	ipv4-addr	ipv6-addr
network-traffic	infrastructure	email-address	mac-address
indicator	malware-analysis-document-or-publication-or-conference	credential-value	x509-certificate
location	abstract-concept	generic-noun	other
noise			

Table 5: Relation types in the GRID ontology. Relations less specific to cyber threats are merged into one type, e.g., `describes`, `analysis-of`, `characterizes`, and `detects` into `research-describes-analysis-of-characterizes-detects`; `malicious-investigates-track-detects` covers attacker-side reconnaissance, tracking, and anti-analysis detection

exploits	bypasses	malicious-investigates-track-detects	impersonates
targets	compromises	leads-to	drops
downloads	executes	delivers	beacons-to
exfiltrate-to	leaks	communicates-with	resolves-to
hosts	provides	authored-by	owns
controls	attributed-to	affiliated-with	cooperates-with
is-part-of	consists-of	has	depends-on
creates-or-generates	modifies-or-removes-or-replaces	uses	variant-of
derived-from	alias-of	compares-to	categorized-as
located-at	originates-from	indicates	mitigates
based-on	research-describes-analysis-of-characterizes-detects	negation	other

B METHOD DETAILS

This appendix collects the file-level data-flow figure, the algorithm listing, the rule tables referenced in Section 2, and the definition of the article complexity features.

Article complexity features. Every feature is converted to its ascending percentile rank over the full training corpus. C_{base} averages the ranks of article token length, entity count, relation count, entities per 1K tokens, and relations per 1K tokens. C_{graph} averages the ranks of ten statistics of the article's ground-truth graph: the mean alias count per entity, the fraction of entities with aliases, average degree, density, average clustering coefficient, the min-degree treewidth upper bound, the maximum k-core, the fraction of edges that are not local bridges, relation span, and crossing density. For the last two, entities are placed on a line in the order of their first mention (by name or alias) in the article: relation span is the mean rank distance $|\text{rank}(s) - \text{rank}(o)|$ over relations (s, r, o), and

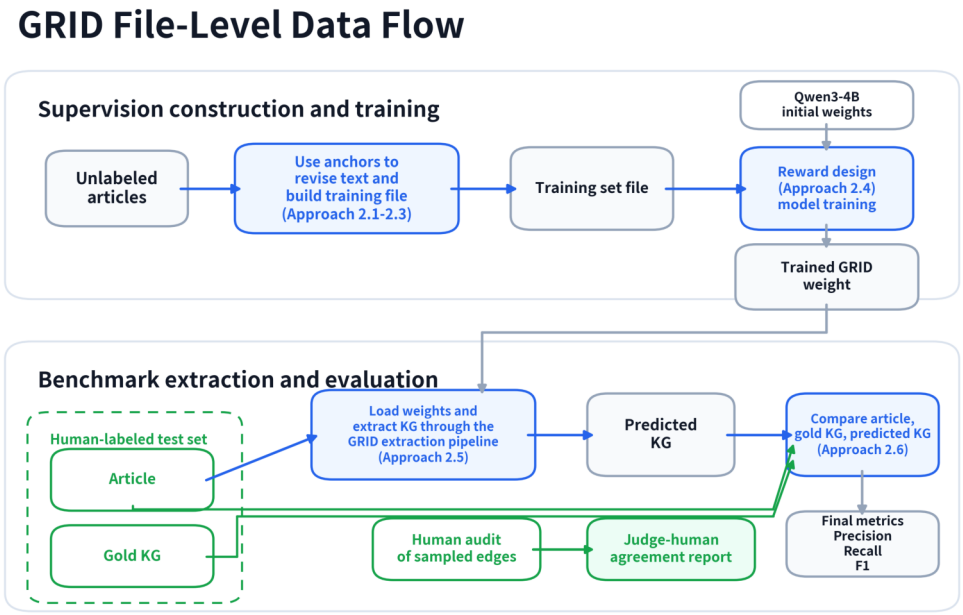


Figure 9: GRID training and evaluation flow at the file level

Algorithm 1: Automatic annotation of article-graph alignment data

Input: Raw CTI article a, traceable extraction prompt P_{trace}, revision prompt P_{rev}
Output: Article-graph alignment (a', G') consisting of revised article a' and text-grounded knowledge graph G'

$G \leftarrow \text{LLMExtract}(a, P_{\text{trace}})$
Parse G into entity list E and relation list R
foreach $r \in R$ **do**
  record the sentence-local subject and object mentions as $(r.\text{sub}, r.\text{obj})$
  record the verbatim evidence anchors as $(r.\text{raw_sub_name}, r.\text{raw_obj_name}, r.\text{raw_text_start}, r.\text{raw_text_end})$
Locate every anchor in a by exact substring matching and mark it as protected
$a' \leftarrow \text{LLMRevise}(a, G, P_{\text{rev}})$, deleting unsupported security mentions while keeping protected anchors and non-security context
$G' \leftarrow (E, R)$
return (a', G')

crossing density is the fraction of pairs of edges with four distinct endpoints whose rank intervals interleave. The article score is $C_{\text{article}} = 0.65\, C_{\text{base}} + 0.35\, C_{\text{graph}}$.

C Task-Bank Question Construction

Task-bank questions are generated by the released prompt-construction routine `prompt_maker_precision_only` (in `src/tools_prompt_nano.py`), which takes the CTI text, its ground-truth KG, and a task specification, and produces four-option multi-select questions in three families.

First, triple judgment questions test whether the model separates text-supported triples from near misses: "find the correct triple(s)" options contain 0–4 supported triples plus textually related but invalid distractors, and "find the incorrect triple(s)" options contain 0–4 incorrect triples built from common extraction errors (subject–object inversion, entity substitution, or a wrong subject, object, or relation) plus supported triples.

Table 6: Choice-question families and patterns in the task bank

Family	Pattern	Definition
Triple judgment	Supported triples	Among the four options, 0–4 are text-supported; the rest are near-miss distractors with a wrong subject, object, or relation.
	Incorrect triples	Among the four options, 0–4 are deliberately incorrect; the rest are supported triples.
Hallucination	Relation illusion	The subject and object are grounded, but their relation is unsupported.
	Object illusion	The subject and relation are grounded, but the object is unsupported.
	Subject illusion	The relation and object are grounded, but the subject is unsupported.
	Total illusion	All three triple elements are unsupported, but the triple remains CTI-plausible.
	Partial illusion	Only one triple element is grounded; the other two are unsupported.
Factuality	`special_factuality` label	A ground-truth edge is selected and the model is asked to predict its `special_factuality` label, such as standard, possible, future, negated, or combinations thereof.

Table 7: Regex rules for triple-level matching

Aspect	Rule
Edge alignment	Maintains one regex triplet per ground-truth edge.
Entity normalization	Normalizes aliases, abbreviations, number, hyphenation, and shortened head nouns for subjects and objects.
Role-preserving generalization	Permits limited generalization only under role-preserving equivalence.
Relation normalization	Normalizes inflection, voice, prepositions, and common CTI paraphrases.

Second, hallucination questions test whether the model rejects CTI-plausible triples that the article does not support, built in the five modes of Table 6.

Third, factuality questions select a ground-truth edge and ask for its `special_factuality` label (standard, possible, future, negated, or a combination), matching the extraction schema, which emits this field only for non-standard factuality.

D Judge Rules for Automatic Evaluation

The automatic evaluator scores precision and recall under an explicit rule set rather than unconstrained semantic similarity. Table 8 summarizes the main judge rules.

D.1 Judge Calibration Against Human Annotations

We compare the LLM judge against human judgments on 378 audit items (191 precision and 187 recall items) spanning the RQ1 and RQ2 methods, each randomly assigned to one of three human reviewers.

Reviewers record agreement or disagreement with each LLM verdict, which we convert into human labels: for precision, agreeing with an LLM-TP or disagreeing with an LLM-FP yields a human TP, and the reverse a human FP; recall is converted the same way into TP and FN.

The resulting agreement (Table 9) is 80.6% for precision (154/191) and 91.4% for recall (171/187), yielding an overall agreement of 86.0% (325/378); Table 10 breaks it down by source and reviewer.

Precision agreement is lower because the judge is stricter than the reviewers: humans concur with 92 of 105 LLM-TP verdicts (87.6%) but with only 62 of 86 LLM-FP verdicts (72.1%), overriding LLM-FP to TP in 24 cases versus 13 in the other direction. The judge rejects edges whose support requires a longer implicit inference. For example, one article states *"We attribute with high confidence*

Table 8: Judge rules in the automatic evaluator

Rule	Meaning
Text-Provable Truth	A relation counts only if it can be supported by the article text itself. The judge must not use external world knowledge, domain defaults, or subject elevation to justify a match.
Subject-Preserving Chain Reasoning	Multi-hop reasoning is allowed only when the full chain is text-supported and the subject remains the same throughout the deduction.
General-Specific Equivalence	Coarser and finer phrasings may match when they refer to the same fact stated in the text rather than to different entities or events.
Action-Technique and Relation Normalization	A surface action, a normalized technique name, and a canonical relation label may match when they denote the same behavior in context.
Relation Hierarchy Tolerance	A predicted relation may be coarser or finer than its ground-truth counterpart only when the article supports the more specific behavior, which also licenses the coarser relation.
Attribute-as-Structure Matching	Structural facts may be represented either as edges or as entity attributes. In particular, `alias` and `parent_entity` fields can satisfy corresponding ground-truth structural relations.
Alias and Hierarchy Equivalence	Alias forms and near-family variants can be treated as equivalent only when the text or entity attributes explicitly support that equivalence.
Indexed Edge-wise Auditing	Precision is judged edge by edge over predicted triples, while recall is judged edge by edge over ground-truth triples. For each audited edge, the judge must return its index, a binary decision, and either supporting text evidence or the matched counterpart predicted edge(s).
Malformed Extraction Rejection	Predictions whose subject or object is a pronoun, a full clause, or another non-entity span are rejected and cannot count as true positives.

Table 9: Judge–human confusion matrices. Rows are LLM labels and columns are human labels; bold marks agreement

Precision calibration

		Human's decision	
		TP	**FP**
LLM	**TP**	**87.6%**	12.4%
	FP	27.9%	**72.1%**

Recall calibration

		Human's decision	
		TP	**FN**
LLM	**TP**	**78.6%**	21.4%
	FN	4.8%	**95.2%**

that Wagemole is a North Korea state-sponsored threat," and the extraction outputs *(Wagemole, is operated by, North Korea)*. The judge marks it FP, whereas the reviewer accepts *"North Korea state-sponsored threat"* as sufficient implicit support for the attribution; the judge would accept only an explicit statement such as *"operated by North Korea."* Precision scores should be read with this conservative pattern in mind.

Table 10: Per-source agreement rates between the LLM judge and three human reviewers

Source	Reviewer 1		Reviewer 2		Reviewer 3	
	Precision agreement	**Recall agreement**	**Precision agreement**	**Recall agreement**	**Precision agreement**	**Recall agreement**
GRID	100.0%	81.8%	77.8%	88.9%	62.5%	100.0%
CASIE	87.5%	80.0%	81.8%	100.0%	100.0%	80.0%
CTINexus	62.5%	100.0%	80.0%	100.0%	100.0%	100.0%
MalKG	70.8%	100.0%	77.3%	88.9%	84.6%	92.9%
SecureNLP	84.2%	93.5%	85.7%	88.0%	77.8%	86.7%

E Evaluation Cost Accounting

Under GPT-5.4 mini's official non-cached pricing ($0.75 per 1M input tokens and $4.50 per 1M output tokens), our logged GRID judge usage of about 22.6K input tokens and 18.5K output tokens per article-level graph evaluation costs about $0.017 for input plus $0.083 for output, or $0.1 per article. For our 249-article benchmark, this is about $25 for evaluating one method. RQ1 and RQ2

evaluate 13 unique methods or variants, so the corresponding benchmark-level judge cost is about $325; RQ3 and RQ4 add further judged runs on top of this figure.

This is acceptable offline because the alternative is exhaustive human auditing: the ground-truth graphs contain 7,617 edges (30.59 per article) and GRID Task-bank predicts 48.71 edges per article, so a human would have to check every predicted edge against the article for precision and every ground-truth edge against the full predicted graph for recall, for each of the 13 methods.

Inside RL, an online judge must score batch size $\times$ rollouts graphs at every step, $64 \times 8 = 512$ in the End2End Reward setting, about two full benchmark evaluations per step. With GPT-5.3 Codex Medium as the training-side judge, the End2End Reward run already costs about $942 within 13 steps (Appendix I), which is prohibitive for longer training. The primary Task-bank Reward model therefore uses the LLM judge only for offline evaluation and the scripted task-bank reward for RL.

F Baseline Adaptation Details

For the seven LLM-based baselines, we first locate and reproduce their public implementations from the corresponding papers, project pages, and code repositories, then apply only the engineering adaptations needed to run them under the same original Qwen3-4B-Instruct-2507 model and unified article-level benchmark protocol without changing each method's core extraction logic. For CTINexus and CTIKG, we retain their released multi-stage extraction workflows and normalize only the final outputs to our common article-level {`entities, relations`} JSON schema. For GraphRAG, we use the official Microsoft graph-extraction prompt and its zero-shot extraction loop. For Graphiti and Cognee, we preserve their released node/edge or cascade-extraction logic while adapting them to full-document chunked execution and article-level merging so that they can be evaluated on long CTI articles under a unified benchmark protocol. For LLM-CAKG, we keep its appendix-style prompts but adapt it to a full-document chunked processing pipeline. For AttacKG+, we retain the original rewrite-then-extract pipeline and adapt it to the same full-document article-level setting. For KnowGL, we evaluate the released Hugging Face model and parse its native decoded outputs into the same triple schema. Across prompt-based baselines, a shared runtime wrapper additionally repairs JSON and normalizes schemas.

G Surveyed CTI Datasets

Table 11 lists the annotated datasets surveyed in Section 5 and why none of them can supervise GRID's end-to-end training, which needs traceable article–graph pairs at scale. CASIE, CTINexus, MalKG, and SecureNLP serve only as evaluation sources (Section 3); SecureNLP refers to the SemEval-2018 Task 8 shared-task dataset, an extension of MalwareTextDB (Lim et al., 2017; Phandi et al., 2018).

H Failure Case Analysis

We inspected precision errors of the GRID Task-bank Reward model on 50 benchmark articles (ten per source, each at most 4,000 tokens): from the 673 edges the judge marked as false positives, we randomly sampled 101 for human review, and 84 were confirmed as errors, i.e., predicted triples not supported by the input article; this audit is separate from the calibration sample of Appendix D.1. These cases cover 23 articles from all five sources. We classified them into five categories: malformed triple, where the output does not form a valid subject–relation–object triple (32/84); boundary shift, where the extracted span shifts away from the correct entity or relation boundary (23/84); unsupported fact, where the triple describes a fact not present in the article (18/84); direction reversal, where the relation and entity pair are text-supported but subject/object are swapped (7/84); and subject elevation, where object and relation are correct but the subject is lifted from a specific span to a broader entity (4/84). These precision failures thus concern well-formed KG slots, local text support, and argument roles rather than missing security vocabulary. Representative examples:

- **Malformed triple (32/84).** Source text: *"read the database password from Magento's configuration file"*. Extracted edge: *(read, the database password from, Magento's configu-*

Table 11: Surveyed annotated datasets and why none can supervise GRID's training

Dataset	What is annotated	Why it does not fit
MalKG (Rastogi et al., 2021)	Entity–relation graphs hand-annotated on 64 malware reports	Small
CTINexus (Cheng et al., 2025)	Entity–relation graphs for 149 short CTI articles, without text anchors	Small; no evidence anchors
AttacKG (Li et al., 2022)	Entities, dependencies, and techniques on 16 manually labeled reports, 9 of them released	Small
LADDER (Alam et al., 2023)	Entities and relations over 140K tokens on 36 Android malware families	Narrow domain
AISecKG (Agrawal et al., 2023)	About 730 triples from six lab manuals of graduate cybersecurity courses	Narrow domain (education); small
SciNLP (Duan et al., 2025)	Full-text entities and relations on 60 ACL papers	Other field (NLP papers)
CASIE (Satyapanich et al., 2020)	Cyber-event triggers and arguments on 1,000 news articles	Other schema (events)
SecureNLP (Phandi et al., 2018)	Token and relation labels on 85 APT reports (SemEval-2018 Task 8)	Small
CyberNER (Ech-Chammakhy et al., 2025)	STIX-aligned entity tags on 610K tokens merged from four NER datasets	Other schema (entities only)
CTI-HAL (Della Penna et al., 2025)	Report passages on seven threat groups mapped to ATT&CK techniques	Other schema (ATT&CK techniques)
rcATT (Legoy et al., 2020)	Report-level ATT&CK tactic and technique labels on 1,490 reports	Other schema (document labels)
TRAM (MITRE Center for Threat-Informed Defense, 2023)	Sentence-level labels for 50 ATT&CK techniques on 150 reports	Other schema (sentence labels)
CTIBench (Alam et al., 2024)	Five question-answering tasks, e.g., multiple choice and threat-actor attribution	Other schema (QA tasks)
CS13K (Li et al., 2024)	13,027 entity–relation triples hand-built from 440+ reports into one merged graph, without article-level pairs	Other schema (merged graph); available on request only
BVTED (Liu et al., 2024)	Bilingual entity–relation triples on 27,311 vulnerability-description sentences	Narrow domain (vulnerability descriptions)
aCTIon (Siracusano et al., 2023)	STIX bundles for 204 reports	Announced as open, but no public release

ration file). Why wrong: *read* is an action rather than an entity, and the relation phrase is incomplete.

- **Boundary shift (23/84).** Source text: *"Administrators need to update the popular Java application framework or put workarounds in place"*. Extracted edge: *(Administrators, or put, workarounds in place)*. Why wrong: the relation boundary is clipped to *or put*.
- **Unsupported fact (18/84).** Source text: *"even if you havent [sic] hired a rental car you might still be concerned that your credit card might have been stung"*. Extracted edge: *(your credit card, might have been stung, by the email)*. Why wrong: the article states a concern, not that the email caused the damage.
- **Direction reversal (7/84).** Source text: *"after he analyzed the router's firmware"*. Extracted edge: *(router's firmware, analyzed, Jan Hörsch)*. Why wrong: the article says Jan Hörsch analyzed the firmware; the prediction swaps subject and object.
- **Subject elevation (4/84).** Source text: *"An unpatched vulnerability in the Magento e-commerce platform could allow hackers to upload and execute malicious code on web servers"*. Extracted edge: *(the Magento e-commerce platform, could allow, hackers)*. Why wrong: the enabling subject is the vulnerability, not the whole platform.

I Additional Training and Testing Details

We collected CTI articles from public online CTI sources such as Mandiant (Mandiant, 2023), CrowdStrike (CrowdStrike, 2024), Trend Micro (Trend Micro, 2023), Securelist (Securelist, 2024), Unit

42 (Palo Alto Networks Unit 42, 2016), Microsoft Security (Microsoft, 2024b), Cisco Talos (Cisco, 2024), and VirusTotal (VirusTotal, n.d.) to construct our CTI corpus, reserved the benchmark articles in this corpus for evaluation, and used the remaining corpus articles together with randomly sampled articles from the Primus dataset (Yu et al., 2025) for training; the 500-article task bank of the primary model and RQ3 is drawn from Primus, while the 1,000-article End2End set comes from our corpus. Each training article contributes up to 20 four-option multi-select questions and one regex target per ground-truth edge. During post-training, the two extraction stages are written into one ontology-guided prompt and optimized as a single generation task; at inference they are issued as two consecutive calls, and the post-trained model transfers to this two-step interface directly (RQ1). Training prompts longer than 4,096 tokens are dropped; at test time, models receive the full article.

During RL training, the maximum response length is 4,096 tokens for Task-bank Reward and 8,192 for End2End Reward. The primary Task-bank Reward model is trained in three stages from Qwen3-4B-Instruct-2507. It first undergoes full-parameter SFT on the 800-article training set (learning rate 10^{-5} with 10% warmup, maximum sequence length 4,096). It is then trained with RL on the multi-select questions of these articles using GSPO (Zheng et al., 2025), with 48 prompts per step and a PPO mini-batch of 24: 180 steps with 4 rollouts at learning rate 3×10^{-6} and no KL term, followed by continued training at learning rate 10^{-5} with 8 rollouts and a low-variance KL loss (coefficient 0.001). Finally, task-bank RL on a 500-article task bank uses SAPO ($\tau_{\text{pos}} = 1.0$, $\tau_{\text{neg}} = 1.05$) with 24 prompts per step, 4 rollouts, a PPO mini-batch of 4, learning rate 3×10^{-6}, and no KL term, and we use the checkpoint after 280 steps. All RL stages of this model sample at temperature 1.0 and top-p 1.0 with 4,096-token prompt and response limits, dropping longer prompts. RQ3 and RQ4 use the same SAPO recipe as this final stage (GRPO-style group advantages, sequence-mean token-mean loss aggregation, no warmup, one PPO epoch), except that rollouts sample at temperature 0.7 and top-p 0.95. All experiments are run on a server with Ubuntu 20.04.6, an AMD Ryzen Threadripper PRO 5955WX CPU, 256 GB of memory, and four NVIDIA RTX 6000 Ada GPUs.

For the End2End Reward setting discussed in RQ2, GPT-5.3 Codex Medium scores each rollout's full graph online with the precision and recall prompts of Section 2.6, and the reward is their F1. We use batch size 64 and rollout count 8 for 13 RL steps; the training-side judge costs about $942 in total, whereas generating the task bank once offline costs about $60 (about $33 for the multi-select questions and $27 for the regex targets).

For the open-weight-generator ablation in Section 3.5, we reuse the 500-article RQ3 pool and the Qwen3-4B-Instruct-2507 student and replace the training-data generator with GPT-OSS-120B; unlike RQ3 and the Llama run, which start RL from the base model, this student first undergoes 600 SFT steps on the generated data. We evaluate the untrained baseline, the SFT checkpoint (600 steps), and RL checkpoints at steps 40, 80, and 100. Their valid-output counts are 247, 138, 139, 161, and 149 of 249 respectively, and their common-valid intersection is 72 articles.

For the cross-backbone reuse ablation, we reuse the RQ3 task-bank Parquet file and RL hyperparameters unchanged and only swap the student to Llama-3.1-8B-Instruct. We evaluate the base model together with RL checkpoints at steps 2, 4, 6, 8, 10, 12, 20, and 28, all under the same GRID two-step inference pipeline, the same frozen GPT-5.4 mini judge, and an 8,192-token generation budget. The nine runs are individually valid on 217 to 233 of the 249 articles, and their common-valid intersection is 136 articles.