Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

CATBridge: Enhancing CVE-ATT&CK Alignment Capability in Low-Resource Environments Using Professional Knowledge from LLMs

Maomiao Xiao
State Key Laboratory of Mathematical Engineering and Advanced Computing
Hao Hu
State Key Laboratory of Mathematical Engineering and Advanced Computing
Yingchang Jiang
State Key Laboratory of Mathematical Engineering and Advanced Computing
Yichen Li
State Key Laboratory of Mathematical Engineering and Advanced Computing
Jianxiao Yu
State Key Laboratory of Mathematical Engineering and Advanced Computing
Feiyang Li
State Key Laboratory of Mathematical Engineering and Advanced Computing
Yuling Liu
Institute of Information Engineering, Chinese Academy of Sciences
Yuchen Zhang
zycxzhappy@126.com

State Key Laboratory of Mathematical Engineering and Advanced Computing

Research Article

Keywords: Common Vulnerabilities and Exposures (CVE), Adversarial Tactics, Techniques, and Common Knowledge (ATT, &CK), Large Language Model(LLM), Knowledge-Driven, Knowledge Transfer

Posted Date: June 24th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9994775/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

CATBridge: Enhancing CVE-ATT&CK Alignment Capability in Low-Resource Environments Using Professional Knowledge from LLMs

Abstract

Aligning Common Vulnerability Exposures(CVE) with ATT&CK is a crucial step in achieving active defense. Existing machine learning and deep learning-based methods have received attention due to their high detection accuracy and strong generalization capabilities, but they struggle to effectively capture the deep semantic associations between CVE and ATT&CK. Large language models(LLMs), with their ability to understand complex contexts, are likely to become the mainstream approach for CVE-ATT&CK alignment. However, low-resource environments offer limited computation and scarce labeled samples. Prompt-based methods let large-scale LLMs (such as GPT) perform well, but their high computational cost and low inference efficiency limit their practical use in CVE-ATT&CK alignment. In contrast, under the same constrained conditions, small-scale LLMs (such as Qwen-2.5-7B), although they can adapt to specific tasks through fine-tuning and offer better practicality, are limited by insufficient understanding of professional knowledge, making it difficult to comprehensively grasp complex attack behavior patterns. To address this, this paper proposes a multi-stage knowledge enhancement framework called CATBridge, whose core lies in constructing a "teacher-student" large-small model collaboration mechanism, through three progressive stages of targeted data enhancement, domain knowledge enhancement, and reasoning knowledge enhancement, to gradually guide lightweight LLMs to complete CVE-ATT&CK alignment tasks. Experimental results show that the framework performs well in low-resource environments, with the deployed lightweight LLM (7B) achieving an accuracy of 83.50%, an absolute improvement of 33.29 percentage points over SOTA, and an average improvement of 46.24 percentage points over zero-shot large-scale general-purpose LLMs such as GPT-5, DeepSeek-R1 (671B), and Qwen3 (235B).

Keywords: Common Vulnerabilities and Exposures (CVE), Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK), Large Language Model(LLM), Knowledge-Driven, Knowledge Transfer

1 Introduction

With the continuous evolution of cyber attack technologies, vulnerabilities have become core security threats in the digital era [1–3]. Common Vulnerabilities and Exposures (CVE) [4] provides unique identifiers, basic descriptions, and references for publicly disclosed vulnerabilities, but does not directly offer defense response strategies. MITRE's ATT&CK framework [5] systematically models observed attack behaviors, categorizing them into a two-level structure of Tactics and Techniques, and provides mitigation measures for each technique, filling the gap in CVE for vulnerability response. Aligning CVE with ATT&CK is of great significance for the comprehensiveness of vulnerability governance and the specificity of defense strategies. However, the large number of newly discovered vulnerabilities daily makes manual alignment impractical. According to published CVE records [6], 40,077 vulnerabilities were recorded in 2024, an increase of 11,116 compared to 2023 (28,961), highlighting the urgency and necessity of developing automated alignment methods.

Currently, a substantial body of research is dedicated to achieving automated alignment from CVE to ATT&CK. Learning-based methods have received much attention due to their high detection accuracy and strong generalization capabilities. Depending on feature extraction and representation approaches, learning-based automatic alignment methods can be broadly categorized into machine learning-based methods and deep learning-based methods. Machine learning-based methods [7–9] rely on manual feature engineering and static feature extraction, often employing TF-IDF or bag-of-words models to transform CVE samples into structured vectors for alignment. In contrast, deep learning-based methods [10–14] leverage neural networks to automatically learn feature information from CVE samples, utilizing word embeddings or pre-trained language models to generate dense vectors for alignment. However, both categories of methods struggle to effectively capture deep-level semantic associations.

Large language models (LLMs) possess the ability to understand complex contexts and are expected to become the mainstream solution for CVE-ATT&CK alignment [15–18]. Current technical approaches are mainly divided into two categories: one is prompt engineering-driven large-scale LLMs (such as GPT) [19–21], and the other is fine-tuning small-scale LLMs (such as Qwen-2.5-7B)[22–24]. However, in low-resource environments with limited computational resources and scarce high-quality labeled samples, although prompt-based methods [19] can enable large-scale LLMs to exhibit excellent performance, their high computational costs and low inference efficiency severely limit their practical application potential in CVE-ATT&CK alignment tasks. These large-scale LLMs are typically accompanied by high API call costs and face numerous restrictions regarding data privacy and secure access. Even when considering local deployment, the massive hardware resources required significantly reduce their applicability [25, 26].

In contrast, in low-resource environments, small-scale LLMs can adapt to specific tasks through fine-tuning, with more controllable resource costs [22]. However, fine-tuning small-scale LLMs [27, 28] also faces bottlenecks in CVE-ATT&CK alignment tasks. Due to their limited model size, they contain relatively scarce cybersecurity domain knowledge, making it difficult to comprehensively grasp complex attack behavior patterns. Although knowledge can be supplemented through manual construction, this approach is costly and inefficient, significantly limiting the model's practicality.

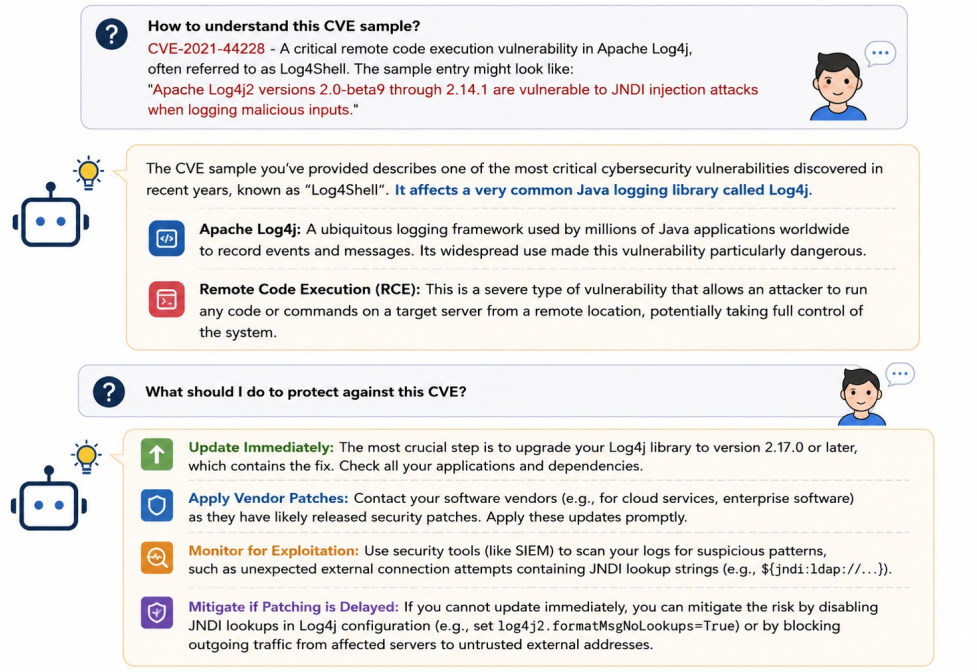


Fig. 1 An example of ChatGPT helping to understand CVE vulnerability descriptions, which can provide detailed background information, term explanations, and solutions.

Fortunately, large-scale LLMs, with their vast training data and parameter scale, can comprehensively understand complex attack behaviors [15, 19]. As shown in Figure 1, large-scale LLMs like ChatGPT [29] can help better understand CVE samples. Therefore, large-scale LLMs like GPT-4 can compensate for the knowledge deficiencies of small-scale LLMs, enabling the latter to analyze CVE samples more efficiently and professionally.

To more effectively utilize the knowledge of large-scale LLMs to enhance CVE-ATT&CK alignment performance [30], we propose extracting the knowledge of large-scale LLMs with strong semantic understanding capabilities to guide the training of small-scale LLMs, enabling them to simulate experts for alignment. However, in low-resource environments, when utilizing the professional knowledge of large-scale LLMs to enhance alignment tasks, there are three main challenges:

C1: Long-tail distribution.CVE data is typically passively collected based on disclosure time, lacking active tracking mechanisms for ATT&CK technology evolution, leading to the long-tail distribution characteristics of CVE samples [4–6]. Sampling statistics show that a few ATT&CK techniques occupy the majority of samples, while actual use cases of many techniques are extremely rare. When training on long-tail distribution data, knowledge extracted from large-scale LLMs will

inevitably bias towards head-class samples. Because tail data receives insufficient exposure, small-scale LLM models struggle to learn the rare attack patterns contained within [31, 32].

C2: Semantic gap. CVE descriptions typically focus on specific technical details of software vulnerabilities (such as "buffer overflow," "SQL injection") [4, 32], while the ATT&CK framework models macro-level attack intentions and behaviors (such as "privilege escalation," "data theft") from the attacker's perspective [5, 33]. To complete alignment from CVE to ATT&CK, a complex reasoning process from "vulnerability exploitation paths" to "attacker tactical objectives" is required. With their vast internal knowledge bases, large-scale LLMs can "intuitively" make such leaps, but their reasoning paths are implicit and opaque [34, 35]. If only using the final alignment results as guidance, small-scale LLMs may only learn superficial text correlations and fail to grasp the underlying logic.

C3: Technical similarity. There is widespread behavioral overlap between ATT&CK techniques, which stems from attackers needing to utilize the limited common functions provided by operating systems or applications to achieve their goals [5, 33]. For example, "T1055: Process Injection" and "T1053.005: Scheduled Task" are common techniques for executing malicious code. Both involve running unauthorized code. However, their core tactical intentions and implementation mechanisms are fundamentally different. "T1055: Process Injection" aims to achieve "Defense Evasion" and "Privilege Escalation" by injecting code into the address space of another legitimate process to run, thereby hiding its own activities and potentially inheriting the target process's privileges. The core intention of "T1053.005: Scheduled Task" is "Persistence" and "Execution", executing code through the operating system's task scheduler at specific times or when triggered by events. When generating knowledge, large-scale LLMs may overemphasize the shared "code execution" surface functional semantics between the two, failing to precisely distinguish the fundamental differences in their tactical intentions. When such knowledge that fails to clearly distinguish the uniqueness of the techniques is used to guide small-scale LLMs, the latter will struggle to learn clear decision boundaries, leading to decreased alignment accuracy.

To overcome the above challenges, this paper proposes a framework based on multi-stage knowledge enhancement, whose core lies in constructing a "teacher-student" large-small model collaboration mechanism. Through three progressive stages of targeted data enhancement, domain knowledge enhancement, and reasoning knowledge enhancement, the professional knowledge from the teacher model (large-scale LLM) is extracted to guide the student model (small-scale LLM) to complete the alignment task. Specifically, **to address the C1 problem,** refined prompt engineering guides the teacher model to targetedly synthesize high-quality CVE samples for scarce categories, to alleviate the sample long-tail distribution problem. **To address the C2 problem,** the teacher model extracts structured domain knowledge such as CWE and CAPEC, builds a vulnerability knowledge graph, extracts knowledge chains from CVE to ATT&CK from it, and through fine-tuning guides the student model to construct a semantic bridge from CVE vulnerability descriptions to ATT&CK techniques. **To address the C3 problem,** the Chain-of-Thought (COT) technique is used to guide the teacher model to generate discrimination summaries for similar techniques,

and through fine-tuning enables the student model to learn the discrimination logic in the summaries, improving the student model's discrimination ability for similar techniques.

The main contributions of this paper are as follows:

- **LLM-driven professional knowledge acquisition:** This work constructs a "teacher-student" large-small model collaboration mechanism, extracts professional knowledge from large-scale LLMs, and uses it to fine-tune small-scale LLMs. It addresses the problems of high overhead of large-scale LLMs and lack of professional knowledge in small-scale LLMs in low-resource environments. To our knowledge, this is among the first works to build a multi-stage teacher-student knowledge enhancement pipeline for CVE-ATT&CK alignment, going beyond prior fine-tuning approaches such as VTT-LLM that rely on chained-template knowledge integration.
- **Multi-stage knowledge enhancement:** This work proposes a multi-stage knowledge enhancement framework, which includes three complementary stages: targeted data enhancement, domain knowledge enhancement, and reasoning knowledge enhancement, significantly improving alignment accuracy.
- **Significant performance in low-resource environments:** In low-resource environments with limited computational resources and scarce high-quality labeled samples, the lightweight student model (7B) deployed by the CATBridge framework achieved 83.50% accuracy, an absolute improvement of 33.29 percentage points over the current best baseline, and an average improvement of 46.24 percentage points over zero-shot general large models such as DeepSeek-R1 (671B), GPT-5, and Qwen3 (235B).

The structure of this paper is as follows: Section 2 reviews related work, Section 3 introduces the basic design and detailed information of each module of CATBridge, Section 4 evaluates model performance from multiple dimensions, and Section 5 summarizes the research findings and contributions, and discusses the limitations and directions for improvement.

2 Related Work

The alignment between existing CVE and ATT&CK frameworks is mainly based on three technical systems: Machine learning [36–38], deep learning [39–41], and large language models (LLMs) [19, 27, 42]. These methods generally follow the following process: first, collect and preprocess CVE samples and related datasets; second, extract and select features, and generate representation vectors of CVE vulnerability descriptions through various combination and processing methods; finally, use these representation vectors to train models to achieve alignment from CVE to ATT&CK frameworks. This section will analyze alignment methods based on machine learning, deep learning, and large language models in sequence, exploring their advantages and disadvantages to provide reference for subsequent research.

Table 1 Comparison of representative CVE-ATT&CK alignment methods.

Category	Representative work	Strengths	Limitations
Machine learning	VULDAT [9], GuardLink [7], El Jaouhari et al. [8]	Simple to implement; low compute cost	Stays at surface text; misses deep semantics
Deep learning	Ampel et al. [10], Grigorescu et al. [11], Othman et al. [14]	Stronger semantic features than ML	Needs large labeled data and compute; weak on rare and cross-domain classes
Prompt-based LLM	Liu et al. [19]	Rich internal knowledge; no task-specific training	High API cost; low inference efficiency; privacy and hardware limits
Fine-tuned LLM	VTT-LLM [27]	Deployable; controllable resource cost	Scarce domain knowledge; weak on complex attack patterns
CATBridge (ours)	This work	Low deployment cost plus transferred expert knowledge	Single-label; single test set (see Section 5)

2.1 Machine learning-based Methods

Machine learning-based alignment methods [36–38] mainly encode CVE vulnerability descriptions into machine-readable vectors through feature engineering, which are used to train classifiers to predict their associated ATT&CK techniques. Different researchers select features from multiple perspectives to construct sequences. The VULDAT framework [9] uses Term Frequency-Inverse Document Frequency (TF-IDF) technology to construct word frequency vector sequences and calculates cosine similarity to identify potential associations. However, this method only stays at the word frequency level, making it difficult to capture deep-level semantic associations, leading to limited alignment accuracy. EL JAOUHARI et al. [8] use the LabelPowerset model to extract keyword sequences from CVE descriptions, optimize feature representation through data augmentation, and then map these feature vectors to corresponding ATT&CK techniques through a Multi-Layer Perceptron (MLP) classifier. However, this method relies on pre-defined MITRE technique IDs. GuardLink [7] introduces the Classifier Chain method, uses Count Vectorizer to encode CVE descriptions, and combines them with metadata features of ATT&CK techniques to achieve alignment through Logistic regression. Although excellent in feature engineering, it still fails to break through the limitations of surface text, lacking deep semantic analysis of contextual interactions.

Machine learning-based methods are simple to implement, but their performance is essentially limited by the completeness of manual feature engineering, making it difficult to break through the limitations of surface text to capture deep semantics and contextual interactions. For example, when mapping a SQL injection vulnerability described as “bypassing authentication by constructing malicious SQL statements”, a TF-IDF-based method might incorrectly associate it with “T1110: Brute Force” only because of keywords like “login” and “bypass”. The root cause is that this method cannot understand the causal relationship between unfiltered parameters and SQL

injection, nor can it accurately capture the overall attack semantics of SQL injection, leading to imprecise alignment results.

2.2 Deep Learning-based Methods

Deep learning-based alignment methods [39–41] mainly focus on semantic modeling of CVE vulnerability descriptions, extracting deep features of vulnerability descriptions through pre-trained models, and combining multiple classification strategies to achieve alignment to ATT&CK techniques. Mendsaikhan et al. [13] use Universal Sentence Encoder (USE) to vectorize CVE vulnerability descriptions and achieve alignment through LabelPowerse, but this method relies on large amounts of labeled data and has high computational overhead. Ampel et al. [10] introduced the RoBERTa model and combined it with self-knowledge distillation to optimize fine-tuning, using task-aware text encoding to enhance semantic associations, but it performs poorly on cross-domain descriptions and rare tactical categories. Grigorescu et al. [11] adopted BERT-based models such as SciBERT and SecBERT combined with adversarial data augmentation, alleviated category bias through multi-output layer architecture, but still faced the problem of poor interpretability. Othman et al. [14] improved the VUL-DAT framework by using the MPNET sentence Transformer model to convert attack descriptions and CVE samples into embedding vectors, calculating cosine similarity for matching to achieve end-to-end mapping, but it is sensitive to input noise and has limited semantic capture capability in cross-knowledge base generalization. Branescu et al. [12] explored multiple Transformer architectures such as SecRoBERTa, TARS, and GPT-4, utilized domain-adaptive pre-training to extract contextual features, and combined with prompt engineering to optimize zero-shot generalization, but with high computational costs and poor performance on semantically complex categories.

Although the semantic feature extraction performance of existing deep learning methods is better than machine learning, their semantic understanding level is still insufficient to reliably complete the CVE-ATT&CK alignment task. At the same time, they usually rely on large-scale labeled data and sufficient computational resources for fine-tuning, which limits their practicality in low-resource environments with scarce labels and limited hardware.

2.3 Large language model-based Methods

The breakthrough of Large Language Models (LLMs) has provided new approaches for aligning CVE with ATT&CK [15–18, 33], mainly evolving along two technical paths: prompt engineering optimization and domain adaptive fine-tuning[20–24]. Among these, prompt-based methods aim to directly leverage the professional knowledge embedded in large-scale LLMs, enabling them to understand and complete relevant tasks [43–45]. Liu et al. [19] were the first to explore the application of prompt engineering in alignment tasks, evaluating ChatGPT's performance in alignment tasks. They designed two paradigms: weak prompts with direct queries, and strong prompts guided by chain-of-thought. The results confirmed that large-scale LLMs like ChatGPT can, to some extent, understand the semantic associations between CVE samples and ATT&CK techniques, demonstrating their potential in

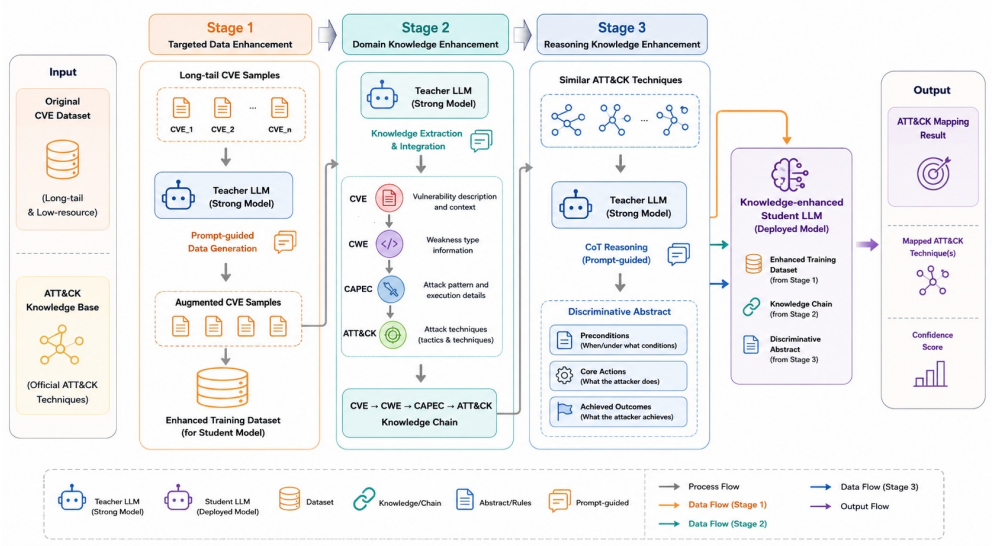


Fig. 2 The full-process architecture of three-stage knowledge enhancement for the CATBridge framework: CVE-ATT&CK alignment based on the collaboration between large-scale LLM (teacher model) and small-scale LLM (student model) involving three core modules of targeted data enhancement, domain knowledge enhancement and reasoning knowledge enhancement

alignment tasks. However, GPT-3 with 175 billion parameters requires 326GB of GPU memory for deployment [26], and its high cost and low computational efficiency limit the widespread use of large-scale LLMs in real-world scenarios. Methods based on fine-tuning small-scale LLMs aim to adapt models to specific domain tasks with controllable resource costs[22]. VTT-LLM [27] is a representative framework in this path, which builds fine-tuning instructions by integrating external knowledge and adopts a two-stage method of "generative model + semantic embedding matching" to achieve CVE to ATT&CK alignment. However, small-scale LLMs contain relatively scarce cybersecurity knowledge, making it difficult for them to understand complex attack patterns.

We believe that with sufficient professional knowledge, small-scale LLMs can make better predictions in CVE-ATT&CK alignment tasks. In this paper, we use large-scale LLMs as domain experts and explore how to effectively acquire and utilize the professional knowledge of large-scale LLMs to enhance CVE sample understanding.

Table 1 summarizes the three method families, their representative works, and their main trade-offs.

3 Design of CATBridge

This section elaborates on how the CATBridge framework achieves CVE-ATT&CK alignment through multi-stage knowledge enhancement. The basic design of the CATBridge framework is shown in Figure 2 and includes the following three modules:

1) Targeted data enhancement module (refer to 3.1): This module treats ATT&CK technique categories with scarce samples as "targets", and through refined prompt engineering, guides large-scale LLMs (teacher models) to targetedly synthesize logically coherent and semantically standardized CVE samples for scarce technique categories.

2) Domain knowledge enhancement module (refer to 3.2): This module utilizes the teacher model to extract structured domain knowledge such as CWE and CAPEC, and builds a vulnerability knowledge graph. On this basis, it extracts knowledge chains from CVE to ATT&CK. Then, through fine-tuning, it guides small-scale LLMs (student models) to construct a semantic bridge from CVE vulnerability descriptions to ATT&CK techniques.

3) Reasoning knowledge enhancement module (refer to 3.3): For highly similar ATT&CK technique pairs, this module utilizes the Chain-of-Thought (COT) technique to guide the teacher model to generate discrimination summaries for similar techniques. Then, through fine-tuning, it enables the student model to learn the discrimination logic in the summaries, improving the student model's discrimination ability for similar techniques.

3.1 Targeted Data Enhancement

To alleviate the long-tail distribution of CVE-ATT&CK alignment samples, existing methods [8, 11, 46] mostly use EDA methods [47] at the vocabulary level for data augmentation. Although it can expand the data scale to a certain extent, its essence is random replacement or disturbance at the surface vocabulary level, which easily distorts the key behavior semantics related to specific ATT&CK techniques, and cannot accurately supplement the knowledge of scarce techniques[48, 49]. Given the powerful semantic understanding and generation capabilities of large-scale LLMs, this paper introduces large-scale LLMs as domain experts to synthesize high-quality CVE samples.

It should be noted that the final alignment performance of this method highly depends on the quality of samples generated by LLMs. Inspired by the "targeted therapy" concept in the medical field (i.e., precise intervention for specific pathogenic genes) [50], this study treats ATT&CK technique categories with scarce samples as "targets", and through refined prompt engineering [43], guides LLMs to targetedly synthesize logically coherent and semantically standardized CVE samples for scarce technique categories, achieving effective supplementation and enhancement of long-tail samples.

Targeted Sample Synthesis Template. In summary, as shown in Appendix A, we have designed a targeted sample synthesis template P_{da}. This template can be formally represented as a prompt function $P_{\text{da}} = \{P_{\text{sys}}, P_{\text{task}}, P_{\text{input}}, P_{\text{output}}\}$, where each component corresponds to system settings, task description, input instructions, and output specifications, collectively forming a guidance mechanism for the teacher LLM. Specifically:

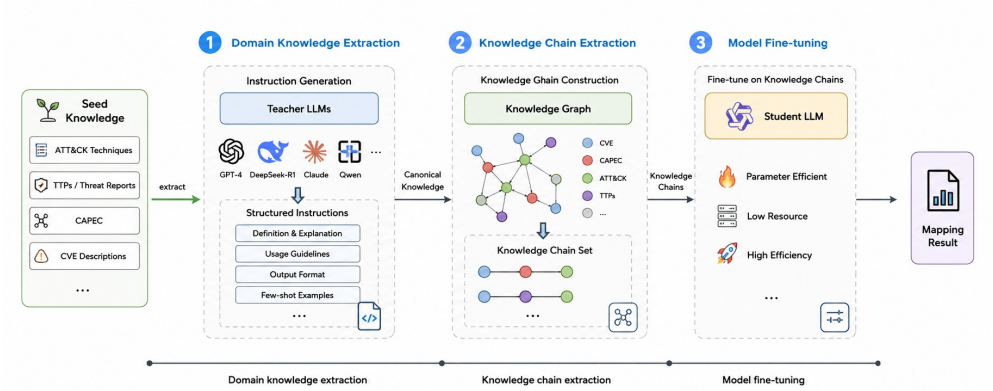


Fig. 3 The execution process of the CATBridge domain knowledge enhancement module: the teacher large-scale LLM extracts CWE and CAPEC structured domain knowledge, constructs a vulnerability knowledge graph, extracts CVE-CWE-CAPEC-ATT&CK knowledge chains and fine-tunes the student small-scale LLM to realize semantic bridging for CVE-ATT&CK alignment

- System Setting(P_{sys}): By role-playing, position the LLM as a "Cybersecurity Vulnerability Modeling Expert," clarifying its core task as generating highly matched and credible CVE descriptions for the specified "target" ATT&CK technique.
- Task Description(P_{task}): Deeply integrate the contextual logic of the vulnerability attack chain, guiding the LLM to focus on the typical attack scenarios, exploitation conditions, and harm manifestations of the target ATT&CK technique, ensuring that the generated CVE descriptions are precisely focused in technical semantics.
- Input (P_{input}): Accept structured targeted input, including a clear "target technique identifier" (i.e., the target ATT&CK technique ID) and "seed CVE samples" extracted from CTI reports, providing the LLM with precise semantic anchors and style references.
- Output Specification (P_{output}): Explicitly define the LLM output as a normative JSON object, including CVE_ID, ATT&CK technique-related fields, and CVE description text, requiring the description content to accurately reflect the target technique features technically, maintain consistency with seed data in style, and be concise and rigorous in writing, to ensure generation quality and effectively suppress semantic drift.

By constructing templates in the aforementioned manner, we can systematically guide the LLM to generate high-quality and highly relevant CVE vulnerability descriptions for the specified ATT&CK "target" technique, thereby achieving precise expansion of long-tail samples.

3.2 Domain Knowledge Enhancement

To bridge the semantic gap between CVE and the ATT&CK framework, some studies [27, 51] have attempted to introduce indirect data sources such as Common Weakness Enumeration (CWE) [52] and Common Attack Pattern Enumeration and Classification (CAPEC) [53] to establish associations between them. However, these studies rely

on simple information cascading or independent processing methods, limiting them to merely representing pairwise associations between CVE, ATT&CK, and these indirect data sources, failing to effectively construct a coherent logical chain across data sources. As illustrated in the figure 3, given the unique advantages of graph structures in representing complex relationships [54–56], this paper represents the knowledge extracted from indirect data sources in the form of a knowledge graph. Extract a complete knowledge chain from CVE to ATT&CK from the knowledge graph and inject it into the student model, achieving precise mapping and efficient utilization of deep logical relationships among multi-source data.

3.2.1 Construction of Vulnerability Knowledge Graph

The vulnerability knowledge graph represents the relationships between nodes in the form of an entity-relationship-entity triplet (h, r, t), where h, r, and t respectively represent the head entity, relationship, and tail entity[57]. This graph is constructed based on the CVE samples enhanced in Section 3.1, as well as the knowledge extracted from indirect data sources such as CWE and CAPEC by the teacher model. Various types of knowledge are abstracted into entities, specifically including the following four types:

E1: **Vulnerability** = {CVE-ID, CVE-Description, Disclosure Time, Vulnerability Type, Exploitation Conditions, CVSS Score, Impact Scope, Harmful Effects}

E2: **Weakness** = {CWE-ID, CWE-Name, CWE-Description, Abstraction Level, Associated Relationships}

E3: **Attack Pattern** = {CAPEC-ID, CAPEC-Description, Exploited Weakness Information, Execution Process Information, Consequence Information}

E4: **Tactic/Technique** = {Tactic-ID, Tactic-Description, Technique-ID, Technique-Description, Mitigation Measures}

The following three semantic relationships are defined between entities to depict the structural associations among different types of vulnerability entities:

R1: **hasWeakness**. This relationship establishes a causal link between the Vulnerability entity and the Weakness entity, indicating that a specific security vulnerability is directly caused by or closely related to the corresponding software weakness.

R2: **exploitedBy**. This relationship characterizes the exploitation relationship between the Weakness entity and the Attack Pattern entity that can exploit that Weakness, revealing how attackers utilize standardized attack techniques to exploit known software flaws.

R3: **mapsToTechnique**. This relationship maps a specific Attack Pattern entity to the Tactic/Technique entity, i.e., the tactics and techniques in the standardized ATT&CK offensive and defensive framework, achieving semantic alignment from "specific attack techniques" to "general attack framework".

Based on the four types of entities and their relationships we defined, as shown in the appendix, we designed key templates for domain knowledge extraction, guiding the teacher model to extract relevant knowledge from the CVE samples enhanced in Section 3.1. Then, the extracted knowledge is organized into structured STIX 2.0 data[58], and a parsing script is written to extract entity objects and relationship

Algorithm 1: Knowledge Chain Extraction Algorithm

Input: Knowledge graph $G = (E, R, \mathcal{T}, \mathcal{L})$, Starting vulnerability entity $v_{\text{start}} \in E_1$

Output: Knowledge chain set $\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$ or $\emptyset$

1 $W \leftarrow \emptyset$;
2 **forall** *triple* $(h, r, t) \in G$ **do**
3 **if** $h = v_{start}$ ***and*** $r = hasWeakness$ ***and*** $\mathcal{T}(t) = E_2$ **then**
4 $W \leftarrow W \cup \{t\}$;
5 $A \leftarrow \emptyset$;
6 **forall** *weakness entity* $w \in W$ **do**
7 **forall** *triple* $(h, r, t) \in G$ **do**
8 **if** $h = w$ ***and*** $r = exploitedBy$ ***and*** $\mathcal{T}(t) = E_3$ **then**
9 $A \leftarrow A \cup \{t\}$;
10 $T \leftarrow \emptyset$;
11 **forall** *attack pattern entity* $a \in A$ **do**
12 **forall** *triple* $(h, r, t) \in G$ **do**
13 **if** $h = a$ ***and*** $r = mapsToTechnique$ ***and*** $\mathcal{T}(t) = E_4$ **then**
14 $T \leftarrow T \cup \{t\}$;
15 $\mathcal{C} \leftarrow \emptyset$;
16 **forall** *weakness entity* $w \in W$ **do**
17 **forall** *attack pattern entity* $a \in A$ **do**
18 **forall** *technique entity* $t \in T$ **do**
19 **if** $(v_{start}, hasWeakness, w) \in G$ ***and*** $(w, exploitedBy, a) \in G$ ***and*** $(a, mapsToTechnique, t) \in G$ **then**
20 $c \leftarrow (v_{\text{start}}, w, a, t)$;
21 $\mathcal{C} \leftarrow \mathcal{C} \cup \{c\}$;
22 **return** $\mathcal{C}$;

objects from it, which are then loaded into the Neo4j graph database [59], ultimately constructing the vulnerability knowledge graph.

3.2.2 Knowledge Chain Extraction

The core task of this stage is to extract a complete knowledge chain from CVE to CWE, then to CAPEC, and finally to ATT&CK from the vulnerability knowledge graph. Algorithm 1 presents the complete process of traversing the vulnerability knowledge graph to extract structured knowledge chains from it:

The algorithm takes the knowledge graph G and the starting vulnerability entity V_{start} as input, producing a complete set of knowledge chains C composed of Vulnerability, Weakness, Attack Pattern, and Tactic/Technique. The knowledge chain extraction can be divided into four stages: Weakness entity extraction (lines 2-4),

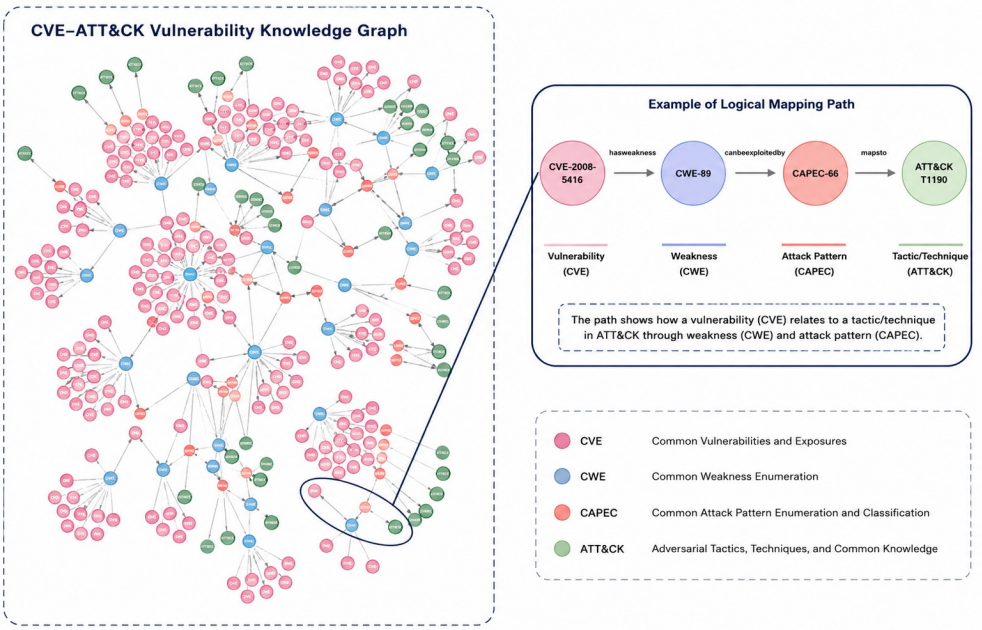


Fig. 4 An Example of CVE-ATT&CK Knowledge Chain Extraction: The Logical Mapping Path from CVE-2008-5416 to CWE-89, CAPEC-66 and T1190 in Vulnerability Knowledge Graph

Attack Pattern extraction (lines 5-9), Tactic/Technique extraction (lines 10-14), and knowledge chain construction (lines 15-21). In the knowledge chain construction stage, we perform a Cartesian product combination of the three extracted entity types, verifying the actual existence of each candidate path in the knowledge graph, and retain only those valid knowledge chains that pass the ternary integrity check. As shown in Figure 4, we used the PHP-Nuke SQL injection vulnerability (CVE-2008-5416) as input, extracting a knowledge chain from CVE-2008-5416 to CWE-89, then to CAPEC-66, and finally to T1190.

3.2.3 Fine-tuning

The core task of this stage is to use the dataset enhanced in 3.1 and the domain knowledge extracted and integrated in 3.2.1 and 3.2.2 to perform supervised fine-tuning (SFT) on the student model (i.e., the alignment model), enabling it to adapt to the task of mapping CVE vulnerability descriptions to ATT&CK techniques.

To balance model performance and training efficiency, we use the Unsloth [60] framework, employing 4-bit quantization and LoRA (Low-Rank Adaptation) [61] technology to efficiently fine-tune the student model. The optimization objective of this stage is the standard supervised fine-tuning (SFT) cross-entropy loss between the student's predicted distribution and the ground-truth ATT&CK labels, as defined in Eq. (1). Crucially, the teacher's domain knowledge is transferred *at the data level* rather than through logits or hidden states: the CVE→CWE→CAPEC→ATT&CK

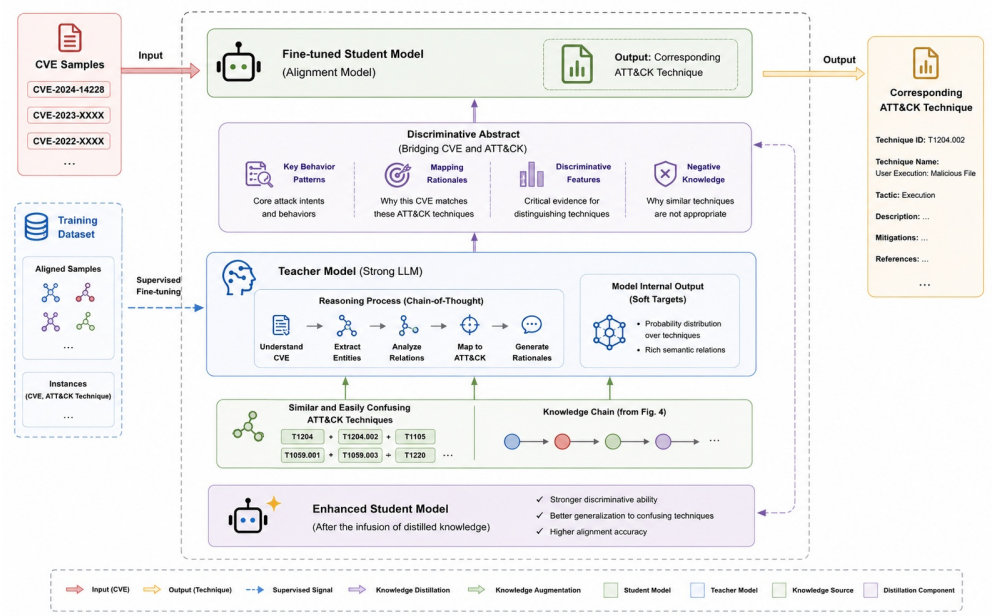


Fig. 5 The full process of the CATBridge reasoning knowledge enhancement module: the teacher large-scale LLM generates discriminative abstracts for similar ATT&CK techniques via the Chain-of-Thought technique, constructs a training dataset by combining the generated abstracts with CVE vulnerability samples and conducts secondary fine-tuning on the student small-scale LLM to realize reasoning logic transfer and effective discrimination of similar ATT&CK techniques

knowledge chains extracted in Sections 3.2.1 and 3.2.2 are serialized into explicit textual context and incorporated into each training instance, so that the student learns the cross-source logical relations directly from the knowledge-chain-augmented inputs. Because the teacher model (DeepSeek-R1, 671B) is accessed only via inference and exposes neither class-level probability distributions nor intermediate-layer representations, no soft-label or feature-matching term is introduced. The objective is computed as:

$$L_{ce} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M}\mathbf{1}\{y_i = j\}\cdot\log\left(\text{Softmax}(p_{i,j})\right) \tag{1}$$

where N is the total number of CVE description samples; M is the number of ATT&CK technique categories; $\mathbf{1}\{\cdot\}$ is the indicator function; y_i is the ground-truth label of the i-th sample; p_i is the student model's raw output vector for the i-th sample; and $\text{Softmax}(p_{i,j})$ is the predicted probability of the true category after Softmax normalization of the student's raw output.

3.3 Reasoning Knowledge Enhancement

When generating knowledge, large-scale LLMs may overemphasize the surface-level functions shared by similar techniques, while ignoring the essential differences in their tactical intentions and implementation mechanisms. This imprecisely differentiated

knowledge can mislead small models, making it difficult for them to learn clear decision boundaries. Existing methods [33] rely on shallow association information like technique co-occurrence statistics and group usage patterns for analysis, and cannot provide discriminative reasoning knowledge. While they can characterize whether techniques are "related", they struggle to explain "why they are different". To accurately capture and distinguish the subtle differences between similar techniques, as shown in Figure 5, this study introduces the Chain of Thought (COT) technique [62, 63], guiding the teacher model to generate discriminative summaries through a structured reasoning process.

3.3.1 Discriminative abstract generation

In this phase, we use the technical pairs that are difficult for the fine-tuned student model from 3.2 to distinguish as input, and utilize the chain of thought technique to output a structured summary to characterize the differences between the technical pairs. As shown in Figure 6, drawing inspiration from the classic attack lifecycle model Cyber Kill Chain [64] and the widely adopted "attack sequence" perspective in modern threat detection analysis [65], we guide the model to conduct systematic analysis along three interrelated and logically progressive dimensions to simulate a complete attack step decomposition:

(1) **Precondition.** Corresponding to the "access" and "preparation" stages of an attack. We instruct the teacher model to focus on the objective constraints prior to the initiation of the attack, namely the system state, access permissions, or environmental configurations that must be present for successful execution of the technique.

(2) **Core Actions.** Corresponding to the "execution" and "means" stages of an attack. We instruct the teacher model to reveal the most representative and fundamental atomic operations executed by the attacker to achieve tactical objectives.

(3) **Achieved Outcomes.** Corresponding to the "results" and "impact" stages of an attack. We prompt the teacher model to analyze the tactical-level outcomes directly resulting from the execution of core actions, reflecting the phased objectives achieved by the technique in the attack chain. This dimension is used to assess the direct impact of the attack actions on the target system and serves as the foundation for subsequent tactical advancements.

3.3.2 Fine-tuning

To transfer the discriminative reasoning of the abstracts to the student model, we conduct a second supervised fine-tuning (SFT) stage. Each training instance pairs the CVE samples of a confusable technique pair with the teacher-generated discriminative abstract (the Precondition / Core Actions / Achieved Outcomes structure of Section 3.3.1), so that the student acquires the discrimination logic by being trained on the teacher's *explicit reasoning text*. The optimization objective is the same cross-entropy SFT loss as in Eq. (1), now computed over this discriminative-abstract-augmented training data. As in Section 3.2.3, the teacher is accessed only via inference and exposes no intermediate representations; the reasoning knowledge is therefore transferred at the data level through the generated abstracts rather than by matching intermediate features.

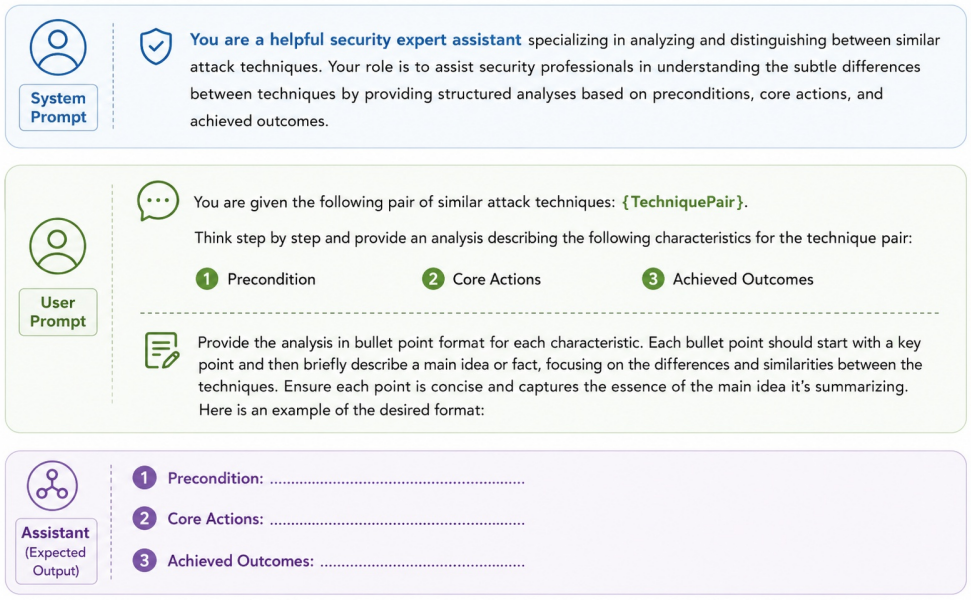


Fig. 6 Three-dimensional Analysis Framework for Generating ATT&CK Technique Discriminative Abstracts: Precondition-Core Actions-Achieved Outcomes Based on Cyber Kill Chain

4 Evaluation

Our evaluation aims to address the following five questions:

RQ1: What is the performance of the CATBridge framework in the CVE-ATT&CK mapping task relative to state-of-the-art (SOTA) methods?

RQ2: How is the quality of the CVE samples synthesized by the CATBridge framework via targeted data enhancement?

RQ3: Is the CATBridge framework applicable to typical low-resource scenarios (few-shot)?

RQ4: Is the CATBridge framework capable of addressing extreme low-resource scenarios (zero-shot)?

RQ5: How do different student models affect the overall performance of the CATBridge framework?

4.1 Evaluation Preparation

Metrics. To accurately evaluate the CATBridge framework, we primarily consider four metrics: accuracy, precision, recall, and F1-score. Accuracy represents the proportion of correctly mapped samples out of the total samples. Precision indicates the proportion of correctly predicted positive mappings out of all predicted positive mappings. Recall represents the proportion of correctly predicted positive mappings out of all actual (ground-truth) positive mappings. F1-score provides a comprehensive evaluation of Precision and Recall through the harmonic mean. The following formulas calculate these metrics:

Table 2 Hyperparameters used for fine-tuning Qwen2.5-7B-instruct

Parameter	Value
Max source length	2048
Max target length	2048
Train batch size	8
Gradient accumulation step	2
Max step	10000
Learning Rate	5e-5
Warmup ratio	0.1
LR scheduler	linear

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{2}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4}$$

$$\text{F1} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{5}$$

Among these, TP (True Positive) and TN (True Negative) represent the number of correctly mapped positive and negative instances, respectively. Conversely, FP (False Positive) and FN (False Negative) represent the number of incorrect mappings. Due to the variability in sample distribution within the test set, this study selects the "weighted average" of these metrics for performance evaluation of the CVE-ATT&CK mapping task.

Dataset. The data used for evaluation are as follows: during the training phase, we constructed the VulnMapAT dataset based on the ENISA dataset [66] and the VULDAT dataset [67], which encompasses 6,451 real vulnerability data entries from 2007 to 2019. For the testing phase, we utilized the ChatGPT-VDMEval [19] as the test set, which contains 7,013 real vulnerability data entries from 2021. To maintain data format consistency, both datasets include four fields: CVE-ID, CVE-Description, ATT&CK Technique ID, and ATT&CK Technique Name. To ensure that the technical classifications used in this study remain consistent with the latest knowledge system, we standardized and aligned some technical categories that existed in the dataset and had been restructured in earlier versions, based on the official update log of MITRE ATT&CK [68].

We formulate CVE-ATT&CK alignment as a single-label classification task over the ATT&CK Enterprise matrix (version 12.1), where each CVE sample is mapped to exactly one ATT&CK technique. The label space contains 30 distinct ATT&CK techniques in total. The training set (VulnMapAT) covers 18 techniques, the test set (ChatGPT-VDMEval) covers 30 techniques, with 18 techniques shared between them.

Table 3 Results of different methods on ChatGPT-VDMEval (%).

Type	Method	Acc. (%)	Prec. (%)	Rec. (%)	F1 (%)
Unsupervised	SMET	24.83	39.46	47.96	33.43
Machine learning	GuardLink	50.21	66.27	45.99	53.90
Fine-tuned LLM	VTT-LLM	47.69	52.93	45.93	43.23
General-purpose LLM (zero-shot)	Qwen3	33.58	37.32	37.86	32.23
	DeepSeek-R1	37.50	47.18	47.50	37.08
	GPT-5	40.70	43.89	45.70	39.71
Knowledge enh.	**CATBridge (ours)**	**83.50**	**90.98**	**93.01**	**85.55**

For the zero-shot study in RQ4, we construct a controlled unseen-technique setting by removing all CVE samples of 8 ATT&CK techniques from the training data and evaluating exclusively on those held-out techniques. The label distribution is strongly long-tailed: 14 techniques have fewer than 75 associated CVE samples in the training set, while the most frequent technique accounts for over 3,600 samples (min / median / max: 1 / 20 / 3,628).

Setup. Our experiments were conducted on a server with the following configuration: 60 GB RAM, AMD EPYC 128-core CPU 9754, and NVIDIA GeForce RTX 4090D (24GB). The server runs the Linux operating system (Ubuntu 22.04). We implemented CATBridge in Python and fine-tuned the student model using the Unsloth library (v2025.3.3).

In the teacher-student model architecture, the teacher model chosen is DeepSeek-R1 [15], with specific configurations: hidden layer size of 7168, 128 attention heads, 61 model layers, and a parameter count of 671B. The student model chosen is Qwen2.5-7B-instruct [69], with specific configurations: hidden layer size of 3584, 32 attention heads, 28 model layers, and a parameter count of 7B. Table 2 provides several key hyperparameters and their corresponding fine-tuning values for Qwen2.5-7B-instruct.

4.2 RQ1: What is the performance of the CATBridge framework in the CVE-ATT&CK mapping task relative to state-of-the-art (SOTA) methods?

To evaluate the performance of the CATBridge framework in the CVE vulnerability description to ATT&CK technique alignment task, we selected multiple representative alignment methods for horizontal comparison, and uniformly used the ChatGPT-VDMEval dataset for evaluation to ensure fairness and reproducibility of the results. The selected methods cover various technical paradigms such as machine learning and fine-tuned LLMs:

GuardLink [7]: Enhances CVE vulnerability descriptions using the EDA method from the TextAttack library, combined with stratified K-fold cross-validation to ensure balanced label distribution in the dataset. Its core is a classification model based on

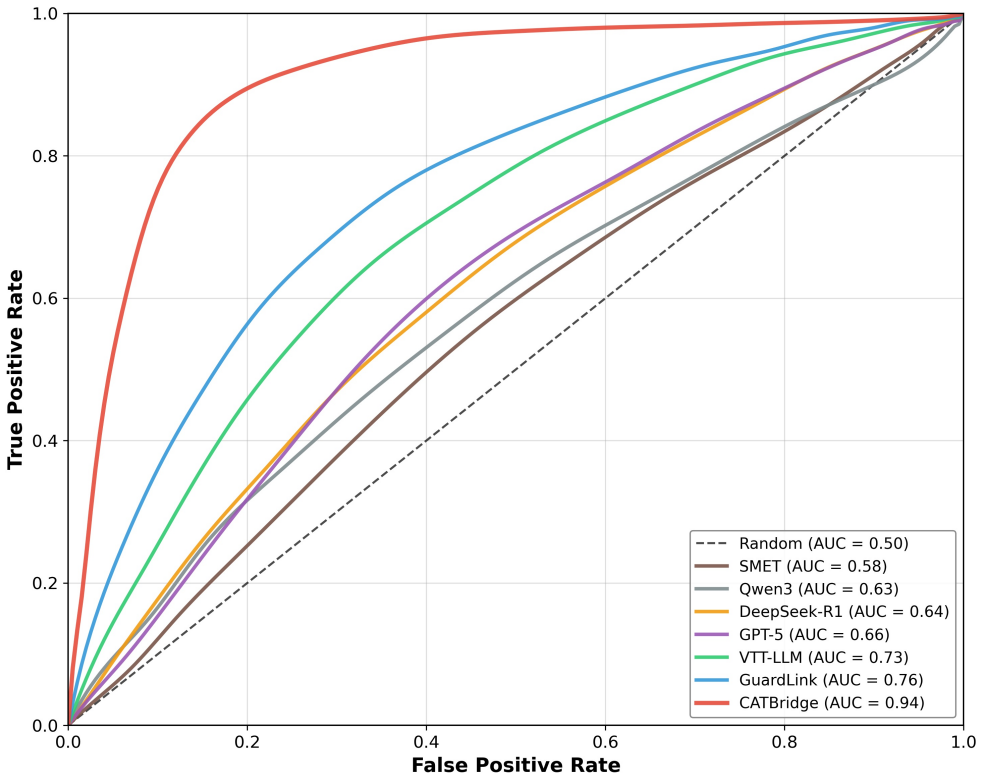


Fig. 7 Micro-averaged ROC Curves of Different CVE-ATT&CK Alignment Models: Comparison of True Positive Rate and False Positive Rate on ChatGPT-VDMEval Test Set

Classifier Chain and logistic regression. In this study, we retrain its core classification model using the VulnMapAT dataset.

SMET [70, 71]: Proposes an unsupervised mapping tool based on semantic similarity, employing a three-component architecture: the attack vector extraction component uses Semantic Role Labeling (SRL) to extract key attack behaviors; the ATT&CK BERT component fine-tunes sentence embedding models based on the SIAMESE network; the logistic regression model estimates matching probabilities and outputs results through ranking aggregation.

VTT-LLM [27]: The first framework to apply fine-tuned LLMs to this task, consisting of a generation module and a mapping module. The generation module integrates CWE and CAPEC knowledge through chained templates and fine-tunes LLMs to generate attack technique descriptions; the mapping module associates the generated descriptions with the ATT&CK technique vector library using text embedding and semantic matching. For a fair comparison, we fine-tune the LLM within the framework using the VulnMapAT dataset according to the original architecture.

Zero-shot prompting-based proprietary large language models: Evaluate the native mapping capabilities of multiple proprietary models, including DeepSeek-R1 [15], Qwen3 [72], and GPT-5 [29]. Guide the models to generate technique IDs through structured prompts without the need for specific training data.

The experimental results are shown in Table 3. The CATBridge framework significantly outperforms baseline methods on all four evaluation metrics, with its Accuracy reaching 83.50%, an absolute improvement of 33.29 percentage points compared to the second-best GuardLink (50.21%). Meanwhile, its Precision (90.98%), Recall (93.01%), and F1-score (85.55%) are also at the highest levels. Importantly, under an identical-model, identical-data setting, the same fine-tuned student without any enhancement reaches only 42.35% accuracy, whereas the full CATBridge yields a +41.15 percentage-point accuracy gain, confirming that the improvement stems from the proposed multi-stage knowledge enhancement pipeline rather than from generic fine-tuning alone.

To isolate the contribution of the CATBridge framework itself from the generic fine-tuning gain, we conduct an ablation study in which the same Qwen2.5-7B-instruct student is trained on the VulnMapAT dataset with successive stages enabled. As shown in Table 4, the gap between the initial SFT baseline (no enhancement) and the full CATBridge row quantifies the marginal gain attributable to the proposed multi-stage knowledge enhancement pipeline.

As shown in Table 4, the three enhancement stages contribute clear and complementary gains over the initial SFT baseline. Starting from the unenhanced student (42.35% accuracy, 50.83% F1-score), Targeted Data Enhancement raises accuracy to 56.18% (+13.83 pp) and F1-score to 62.79% (+11.96 pp), indicating that supplementing scarce long-tail techniques with synthesized CVE samples improves alignment directly rather than simply expanding the feature space. Adding Domain Knowledge Enhancement produces the largest single-stage jump, to 72.46% accuracy (+16.28 pp) and 79.28% F1-score (+16.49 pp), showing that the CVE-CWE-CAPEC-ATT&CK knowledge chains injected as explicit context substantially narrow the semantic gap between low-level vulnerability descriptions and high-level adversary techniques. Finally, Reasoning Knowledge Enhancement lifts performance to the full-framework result of 83.50% accuracy (+11.04 pp) and 85.55% F1-score (+6.27 pp); the simultaneous improvement in precision (90.98%) and recall (93.01%) demonstrates that the discriminative abstracts help the student separate confusable technique pairs without sacrificing coverage. Overall, the pipeline yields a 41.15 pp accuracy improvement over an identical-model, identical-data SFT baseline, confirming that the gains originate from the proposed multi-stage knowledge enhancement rather than from generic fine-tuning.

To more intuitively demonstrate the performance of various methods in the mapping task, we have provided the ROC curves for each method in the test set. As shown in Figure 7, the CATBridge framework achieves a higher true positive rate and a lower false positive rate, reflecting its high accuracy in the mapping task. Additionally, the CATBridge framework has the largest area under the curve (AUC) value, further confirming its strong performance in the context of the task.

Table 4 Ablation study of CATBridge on the full VulnMapAT → ChatGPT-VDMEval setting (Qwen2.5-7B-instruct student)

Configuration	Acc. (%)	Prec. (%)	Rec. (%)	F1 (%)
Initial SFT (no enhancement)	42.35	53.12	48.76	50.83
+ Targeted Data Enhancement	56.18	64.27	61.34	62.79
+ Domain Knowledge Enhancement	72.46	80.13	78.45	79.28
+ Reasoning Knowledge Enhancement (full)	**83.50**	**90.98**	**93.01**	**85.55**

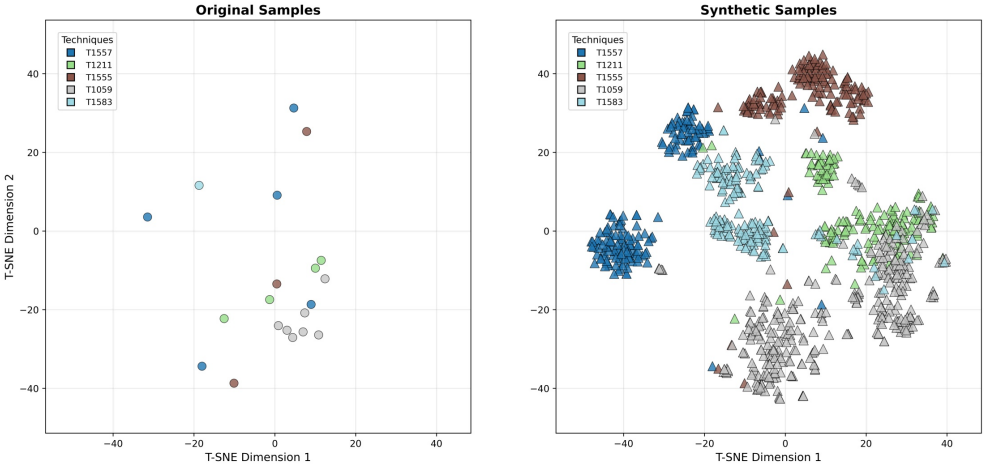


Fig. 8 T-SNE Dimensionality Reduction Visualization of CVE Samples: Feature Distribution Comparison of Original and Targeted Synthesized Samples for Five Typical ATT&CK Technique Categories

4.3 RQ2:How is the quality of the CVE samples synthesized by the CATBridge framework via targeted data enhancement?

To assess the quality of the CVE samples generated by the Targeted Data Enhancement module within the CATBridge framework, we selected five techniques for detailed analysis. On the technical implementation side, we initially converted the CVE vulnerability descriptions into 500-dimensional numerical feature vectors using the TF-IDF method, followed by applying T-SNE dimensionality reduction to project the high-dimensional feature space onto a two-dimensional plane for visualization. As depicted in Figure 8, the CVE samples associated with different ATT&CK techniques form well-defined, separable clusters in the T-SNE projection space, indicating a strong category distinction. Furthermore, the original samples (denoted by circles) and synthetic samples (denoted by triangles) of the same technique category are densely distributed in the projection space, indicating a significant feature relevance. The findings demonstrate that the synthetic samples produced by the CATBridge framework closely align with actual attack techniques in terms of feature representation, while

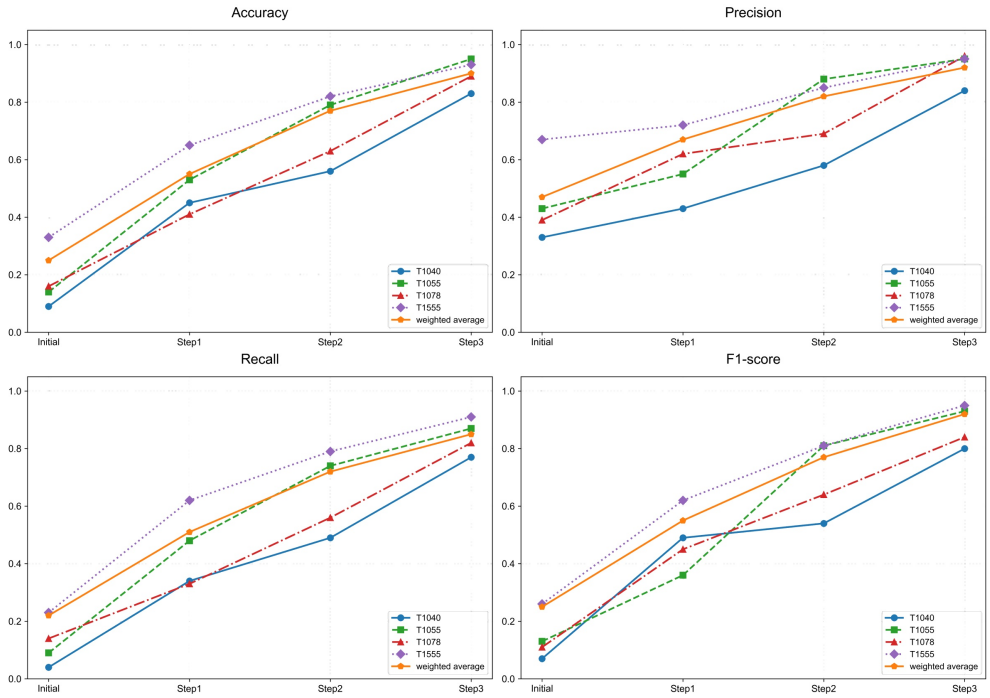


Fig. 9 Micro-level validation of the effectiveness of the CATBridge framework in few-shot scenarios, analyzing three few-shot techniques: T1040 (23 samples), T1055 (46 samples), T1078 (67 samples), one rich-sample technique T1555 (2589 samples), and the overall weighted average.

Table 5 Experimental outcomes in few-shot settings (%).

Stage	**Acc. (%)**	**Prec. (%)**	**Rec. (%)**	**F1 (%)**
Initial fine-tuning	25.33	47.30	22.02	20.03
Targeted data enhancement	55.07	66.62	51.77	54.99
Domain knowledge enhancement	77.48	81.98	71.20	77.35
Reasoning knowledge enhancement	90.94	92.43	85.12	91.55

also possessing distinct category boundaries, effectively retaining the semantic features of the original data, providing qualitative evidence for the validity of the synthesized data. Beyond this feature-space analysis, the downstream contribution of the synthesized samples is quantified in the ablation study (Table 4): adding Targeted Data Enhancement to the initial fine-tuned student raises accuracy from 42.35% to 56.18% (a 13.83 percentage-point gain), providing task-level evidence that the synthesized data improves alignment rather than merely clustering plausibly in feature space. We note that the T-SNE visualization serves as qualitative, supporting evidence; the decisive validation of synthesized-sample utility is the downstream ablation above rather than feature-space proximity alone.

4.4 RQ3:Is the CATBridge framework applicable to typical low-resource scenarios (few-shot)?

To evaluate the efficacy of the method introduced in this paper in few-shot settings, we chose CVE samples that encompass 16 ATT&CK techniques from the VulnMapAT dataset for preliminary fine-tuning. This few-shot setting is a controlled subset constructed specifically for RQ3 and is independent of the full-data split used in RQ1: the 16 techniques comprise the 14 long-tail techniques (fewer than 75 training samples) plus 2 rich-sample techniques as a control group. Within this selection, 14 ATT&CK techniques are associated with CVE samples numbering less than 75, constituting a representative few-shot collection. The remaining two ATT&CK techniques are linked to CVE samples numbering over 2000, which form a rich-sample control group for the purpose of comparative analysis. Furthermore, we drew 100 CVE samples for each of the 16 ATT&CK techniques from the ChatGPT-VDMEval dataset, exclusively for testing.

As shown in Table 5, to quantify the improvement of CATBridge method on model performance, we also tested the model after initial fine-tuning to evaluate its initial performance in few-shot scenarios. With limited initial fine-tuning data, the model struggled to capture sufficiently broad and robust feature patterns, resulting in an Accuracy of only 25.33% and a Recall of only 22.02%. However, The three enhancement stages are targeted data augmentation, domain knowledge enhancement, and reasoning knowledge enhancement. After all three stages, the model improved significantly: Its Accuracy increased from 25.33% to 90.94%, an improvement of 65.61 percentage points; its Precision increased from 47.30% to 92.43%, an improvement of 45.13 percentage points; its Recall increased from 22.02% to 85.12%, an improvement of 63.10 percentage points; its F1-Score increased from 20.03% to 91.55%, an improvement of 71.52 percentage points. The experimental results show that the CATBridge framework successfully solved the common imbalance problem of 'high precision, low recall' in few-shot scenarios, has practical potential with high coverage and low false alarm rates, proving its effectiveness in few-shot scenarios.

To further validate the effectiveness of the CATBridge framework in few-shot scenarios at a micro-level, we selected three few-shot techniques: T1040 (Network Sniffing, 23 samples), T1055 (Process Injection, 46 samples), T1078 (Valid Accounts, 67 samples), and one rich-sample technique T1555 (Credentials from Password Stores, 2589 samples). Figure 9 shows the metric changes for the above techniques at each knowledge enhancement stage. In the initial fine-tuning phase, the F1-scores for the few-shot techniques are generally low, with T1040, T1055, and T1078 having F1-scores of only 7%, 13%, and 11%, respectively, while the rich-sample technique T1555 has an F1-score of 26%. This reflects the model's difficulty in learning effective features from limited data in few-shot scenarios, leading to restricted mapping capabilities. After multiple stages of knowledge enhancement, the recognition performance of all categories has significantly improved, with the initially weakest T1040's F1-Score increasing from 7% to 80%, an increase of 73 percentage points. The F1-Score of the rich-sample technique T1555 has increased from 26% to 95%, an increase of 69 percentage points. Although the final mapping performance for the few-shot categories is still below that of the

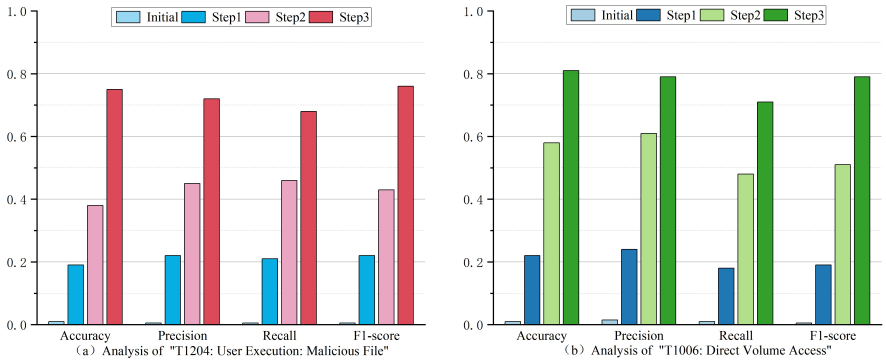


Fig. 10 Micro-level validation of the effectiveness of the CATBridge framework in zero-shot scenarios, selecting two technical categories not in the initial dataset for analysis, namely T1204 and T1006.

Table 6 Experimental outcomes in zero-shot settings (%).

Stage	Acc. (%)	Prec. (%)	Rec. (%)	F1 (%)
Initial fine-tuning	0.96	1.85	0.19	0.37
Targeted data enhancement	22.54	23.08	19.75	21.24
Domain knowledge enhancement	44.02	42.89	46.15	44.48
Reasoning knowledge enhancement	75.73	71.31	74.96	75.59

rich-sample categories, the gap between them has significantly narrowed, fully demonstrating the effectiveness of the CATBridge framework in optimizing the recognition performance of few-shot categories.

4.5 RQ4:Is the CATBridge framework capable of addressing extreme low-resource scenarios (zero-shot)?

To test the effectiveness of the proposed method in zero-shot scenarios, this study removed the CVE samples corresponding to 8 ATT&CK techniques from the VulnMapAT dataset, using the remaining data for initial fine-tuning. This zero-shot setting is a controlled unseen-technique subset constructed specifically for RQ4 and is independent of the full-data split used in RQ1. Additionally, we extracted 200 CVE samples for each of these 8 ATT&CK techniques from the ChatGPT-VDMEval dataset, which were used to test the model's performance.

As shown in Table 6, to quantify the improvement of CATBridge method on model performance, we also tested the model after initial fine-tuning to evaluate its initial performance in zero-shot scenarios. After removing samples corresponding to 8 ATT&CK techniques from the VulnMapAT dataset, the model did not learn the features of these techniques during the initial training phase, and the results met our expectations, with an Accuracy of only 0.96% and an F1-score of only 0.37%, indicating that the model

Table 7 Performance of various LLMs as student models (%).

LLM Name	Acc. (%)	Prec. (%)	Rec. (%)	F1 (%)
Mistral-7B	82.44	90.03	92.44	81.60
Llama3-8B	83.25	**91.35**	92.25	83.58
Gemma2-9B	82.89	89.67	**93.50**	84.68
Qwen2.5-7B	**83.50**	90.98	93.01	**85.55**

could barely recognize these unseen techniques. After multi-stage knowledge enhancement, the model's generalization ability to unseen techniques continuously improved: its Accuracy increased from 0.96% to 75.73%, an improvement of 74.77 percentage points; its Precision increased from 1.85% to 71.31%, an improvement of 69.46 percentage points; its Recall increased from 0.19% to 74.96%, an improvement of 74.77 percentage points; its F1-Score increased from 0.37% to 75.59%, an improvement of 75.22 percentage points. This indicates that in the context of continuously increasing ATT&CK techniques, CATBridge's multi-stage knowledge enhancement method is an effective strategy for improving model generalization ability.

To further validate the effectiveness of the CATBridge framework in zero-shot scenarios from a micro-level, we selected two technical categories not in the initial dataset for analysis, namely "User Execution: Malicious File" and "Direct Volume Access". Figure 10 shows the changes in Accuracy, Precision, Recall, and F1-Score for these two techniques. In the initial fine-tuning stage, without training data, all metrics for these two technical categories were close to 0, and the model did not learn the feature distribution of these techniques. For "User Execution: Malicious File", after Targeted Data Enhancement and Domain Knowledge Enhancement, its F1-Score increased from 0.5% to 43%. Although the metrics improved, because both "User Execution: Malicious File" and "T1202:Indirect Command Execution" involve "execution" behavior, the model could not effectively distinguish between them. Finally, through Reasoning Knowledge Enhancement, the metrics for "User Execution: Malicious File" improved significantly, with F1-Score increasing to 76%. For "Direct Volume Access", its F1-Score increased from the initial 1% to 79%. This demonstrates from a micro-level the effectiveness of the CATBridge framework in zero-shot scenarios.

4.6 RQ5:How do different student models affect the overall performance of the CATBridge framework?

To explore the generalization capabilities of the CATBridge framework with various student models, we trained each model on the VulnMapAT dataset and consistently assessed their performance using the ChatGPT-VDMEval dataset.The experimental outcomes presented in Table 7 reveal that the CATBridge method demonstrates strong performance on the four critical metrics of Accuracy, Precision, Recall, and F1-score, irrespective of the LLM employed. This suggests that the CATBridge framework is not reliant on a particular model structure and exhibits robust generalization capabilities and model compatibility.Further horizontal comparison of different student models

shows that the Qwen2.5-7B-instruct model stands out in several evaluation metrics: it leads in both Accuracy (83.50%) and F1-score (85.55%), and also ranks second in Precision (90.98%) and Recall (93.01%).

5 Conclusion

In this study, we propose CATBridge, a CVE-ATT&CK alignment framework for low-resource environments. It constructs a "teacher-student" large-small model collaboration mechanism. Through three progressive stages of targeted data enhancement, domain knowledge enhancement, and reasoning knowledge enhancement, in low-resource environments with limited computing resources and scarce high-quality labeled samples, it significantly improves the alignment performance of the student model, guiding small-scale LLMs to achieve precise alignment from CVE to ATT&CK. In multiple evaluation experiments under low-resource environment settings, the CATBridge framework demonstrates excellent performance. The deployed small-scale LLM (7B) achieves an alignment accuracy of 83.50%, an absolute improvement of 33.29 percentage points over SOTA, and an average improvement of 46.24 percentage points over zero-shot general large models such as DeepSeek-R1 (671B), GPT-5, and Qwen3 (235B).

To our knowledge, CATBridge is among the first works to introduce a multi-stage teacher-student knowledge enhancement mechanism into the CVE-ATT&CK alignment task, distinct from prior approaches that rely on single-stage fine-tuning or chained-template knowledge integration. Its core value lies in significantly reducing the model's dependence on large-scale labeled data and computational resources, while still enabling reliable threat knowledge correlation analysis in resource-constrained environments. This method effectively supports security personnel in rapid vulnerability localization and governance, and also provides new ideas for the application of lightweight models in cybersecurity knowledge correlation analysis.

Threats to validity. A natural concern is data contamination: the teacher model (DeepSeek-R1) is pre-trained on web-scale corpora. These corpora may already contain CVE records and their documented ATT&CK mappings. Part of the reported accuracy could therefore reflect teacher memorization rather than the framework's generalization ability. We argue that memorization is not the primary driver of the observed gains, for three reasons. First, if the relevant CVE-ATT&CK associations were directly recoverable from teacher memorization, the teacher itself should score highly when queried directly; however, DeepSeek-R1 (671B) attains only 37.50% zero-shot accuracy on the same test set (Table 3), far below the 83.50% of the deployed 7B student, and the other web-scale models (GPT-5, Qwen3) behave similarly. Second, the teacher is never shown the test-set CVE instances during our pipeline; it only synthesizes training-side samples, extracts CWE/CAPEC knowledge, and produces discriminative abstracts, so any benefit must be mediated by the structured knowledge transferred to the student rather than by test-label leakage. Third, the ablation in Table 4 shows that an identical teacher-accessible pipeline without the enhancement stages reaches only 42.35% accuracy, and the +41.15 pp improvement appears only

after the three knowledge-enhancement stages are added, which isolates the contribution of the framework itself. Nevertheless, fully eliminating temporal overlap between the teacher's pre-training data and the evaluation CVEs is not possible with closed-weight teachers; we therefore regard a strict temporal-holdout evaluation, using CVEs disclosed after the teacher's training cutoff, as an important direction for future work.

The interpretability of the current outputs is still limited. This makes it harder to provide reliable decision support in practice. We plan in future work to use technologies such as attention mechanism visualization and reasoning path generation to intuitively present the key features and attack evidence that the model focuses on during the reasoning process, enhance the interpretability of the model's output results, and provide more reliable decision support for security analysts.

Statements and Declarations

Funding. This work was supported by the National Natural Science Foundation of China (Grant No. 61902427) and the National Key Research and Development Program of China (Grant No. 2023YFC3306305).

Competing Interests. The authors have no relevant financial or non-financial interests to disclose.

Ethics declaration. Not applicable.

References

[1] Xu, G., Chen, X., Cai, X., Yu, D.: Exploitability prediction of vulnerabilities based on heterogeneous graphs. Knowledge-Based Systems, 114517 (2025)

[2] Mirtaheri, S.L., Pugliese, A., Pascucci, V.: Automated vulnerability score prediction through lightweight generative ai. Knowledge-Based Systems, 114406 (2025)

[3] Hu, H., Liu, Y., Chen, C., Zhang, H., Liu, Y.: Optimal decision making approach for cyber security defense using evolutionary game. IEEE Transactions on Network and Service Management **17**(3), 1683–1700 (2020)

[4] CVE: CVE. (accessed 1 November 2025) (2025). https://cve.mitre.org/

[5] Mitre: ATT&CK. (accessed 1 November 2025) (2025). https://attack.mitre.org/

[6] CVE: Published CVE Records. (accessed 1 November 2025) (2025). https://www.cve.org/About/Metrics

[7] El Jaouhari, S., Tamani, N., Jacob, R.I.: Guardlink: Dynamic linking of cve to mitre att&ck techniques using machine learning. In: GLOBECOM 2024-2024 IEEE Global Communications Conference, pp. 4811–4817 (2024). IEEE

[8] El Jaouhari, S., Tamani, N., Jacob, R.I.: Improving ml-based solutions for linking of cve to mitre att &ck techniques. In: 2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC), pp. 2442–2447 (2024). IEEE

[9] Othman, R., Russo, B.: Vuldat: Automated vulnerability detection from cyber-attack text. In: International Conference on Embedded Computer Systems, pp. 494–501 (2023). Springer

[10] Ampel, B., Samtani, S., Ullman, S., Chen, H.: Linking common vulnerabilities and exposures to the mitre att&ck framework: A self-distillation approach. arXiv preprint arXiv:2108.01696 (2021)

[11] Grigorescu, O., Nica, A., Dascalu, M., Rughinis, R.: Cve2att&ck: Bert-based mapping of cves to mitre att&ck techniques. Algorithms **15**(9), 314 (2022)

[12] Branescu, I., Grigorescu, O., Dascalu, M.: Automated mapping of common vulnerabilities and exposures to mitre att&ck tactics. Information **15**(4), 214 (2024)

[13] Mendsaikhan, O., Hasegawa, H., Yamaguchi, Y., Shimada, H.: Automatic mapping of vulnerability information to adversary techniques. In: The Fourteenth International Conference on Emerging Security Information, Systems and Technologies SECUREWARE2020 (2020)

[14] Othman, R., Rossi, B., Russo, B.: Cybersecurity defenses: Exploration of cve types through attack descriptions. In: 2024 50th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pp. 415–418 (2024). IEEE

[15] Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al.: Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948 (2025)

[16] Peng, Y., Hu, H., Li, F., Jiang, Y., Tang, J., Liu, Y.: Llm4game: Multi-agent reinforcement learning with knowledge injection for dynamic defense resource allocation in cloud storage. Computer Networks, 111748 (2025)

[17] Zha, X., Dong, Y., Jiang, H., Xu, Z., Wang, C.: Llm-agr: Large language model augmented graph representation learning for recommendation. Knowledge-Based Systems, 114791 (2025)

[18] Lin, X., Zhang, T., Hou, C., Wang, J., Xue, J., Lv, H.: Node importance estimation leveraging llms for semantic augmentation in knowledge graphs. Knowledge-Based Systems, 114521 (2025)

[19] Liu, X., Tan, Y., Xiao, Z., Zhuge, J., Zhou, R.: Not the end of story: An evaluation of chatgpt-driven vulnerability description mappings. In: Findings of the

Association for Computational Linguistics: ACL 2023, pp. 3724–3731 (2023)

[20] Cheng, Y., Chen, J., Huang, Q., Xing, Z., Xu, X., Lu, Q.: Prompt sapper: A llm-empowered production tool for building ai chains. ACM Transactions on Software Engineering and Methodology **33**(5), 1–24 (2024)

[21] Zhang, X., Zhang, C., Li, T., Huang, Y., Jia, X., Hu, M., Zhang, J., Liu, Y., Ma, S., Shen, C.: Jailguard: A universal detection framework for prompt-based attacks on llm systems. ACM Transactions on Software Engineering and Methodology (2025)

[22] Lin, X., Wang, W., Li, Y., Yang, S., Feng, F., Wei, Y., Chua, T.-S.: Data-efficient fine-tuning for llm-based recommendation. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 365–374 (2024)

[23] Zhu, Q., Yang, B., Liu, W., Wu, J.: A simple yet effective difficulty-aware bucketed fine-tuning strategy for llm-based recommendation. Knowledge-Based Systems, 114655 (2025)

[24] Cook, A., Karakuş, O.: Llm-commentator: Novel fine-tuning strategies of large language models for automatic commentary generation using football event data. Knowledge-Based Systems **300**, 112219 (2024)

[25] Ma, L., Yang, W., Li, Y., Fei, B., Zhou, M., Li, S., Jiang, S., Xu, B., Xiao, Y.: Adaptivelog: An adaptive log analysis framework with the collaboration of large and small language model. ACM Transactions on Software Engineering and Methodology (2025)

[26] Ma, L., Yang, W., Jiang, S., Fei, B., Zhou, M., Li, S., Zhao, M., Xu, B., Xiao, Y.: Luk: Empowering log understanding with expert knowledge from large language models. IEEE Transactions on Software Engineering (2025)

[27] Zhang, C., Wang, L., Fan, D., Zhu, J., Zhou, T., Zeng, L., Li, Z.: Vtt-llm: Advancing vulnerability-to-tactic-and-technique mapping through fine-tuning of large language model. Mathematics **12**(9), 1286 (2024)

[28] Chen, M., Zhu, K., Lu, B., Li, D., Yuan, Q., Zhu, Y.: Aecr: Automatic attack technique intelligence extraction based on fine-tuned large language model. Computers & Security **150**, 104213 (2025)

[29] OpenAI: ChatGPT-5. (accessed 1 November 2025) (2025). https://chat.openai.com

[30] Kang, M., Lee, S., Baek, J., Kawaguchi, K., Hwang, S.J.: Knowledge-augmented reasoning distillation for small language models in knowledge-intensive tasks. Advances in Neural Information Processing Systems **36**, 48573–48602 (2023)

[31] Zhang, Y., Kang, B., Hooi, B., Yan, S., Feng, J.: Deep long-tailed learning: A survey. IEEE transactions on pattern analysis and machine intelligence **45**(9), 10795–10816 (2023)

[32] Chen, Z., Chen, J., Feng, Y., Liu, S., Zhang, T., Zhang, K., Xiao, W.: Imbalance fault diagnosis under long-tailed distribution: Challenges, solutions and prospects. Knowledge-Based Systems **258**, 110008 (2022)

[33] Al-Sada, B., Sadighian, A., Oligeri, G.: Mitre att&ck: State of the art and way forward. ACM Computing Surveys **57**(1), 1–37 (2024)

[34] Hsieh, C.-Y., Li, C.-L., Yeh, C.-K., Nakhost, H., Fujii, Y., Ratner, A., Krishna, R., Lee, C.-Y., Pfister, T.: Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In: Findings of the Association for Computational Linguistics: ACL 2023, pp. 8003–8017 (2023)

[35] Shridhar, K., Stolfo, A., Sachan, M.: Distilling reasoning capabilities into smaller language models. Findings of the Association for Computational Linguistics: ACL 2023, 7059–7073 (2023)

[36] Jordan, M.I., Mitchell, T.M.: Machine learning: Trends, perspectives, and prospects. Science **349**(6245), 255–260 (2015)

[37] Jung, A.: Machine Learning: the Basics. Springer, Cham (2022)

[38] Winter, C., Frasincar, F., Peuter, B., Matsiiako, V., Ido, E., Klinkhamer, J.: Automated feature engineering for automated machine learning. Knowledge-Based Systems, 113671 (2025)

[39] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436–444 (2015)

[40] Bishop, C.M., Bishop, H.: Deep Learning: Foundations and Concepts. Springer, Cham (2023)

[41] Awwal, P., Naval, S.: Development of heuristic adapted serial-based deep learning for efficient adversarial malware detection framework in windows. Knowledge-Based Systems, 114032 (2025)

[42] Mizrahi, M., Kaplan, G., Malkin, D., Dror, R., Shahaf, D., Stanovsky, G.: State of what art? a call for multi-prompt llm evaluation. Transactions of the Association for Computational Linguistics **12**, 933–949 (2024)

[43] Zhao, W., Wu, J., Meng, Z.: Apppoet: Large language model based android malware detection via multi-view prompt engineering. Expert Systems with Applications **262**, 125546 (2025)

[44] Nguyen, P., Do, T., Nguyen, L.-M.: Improving hierarchical semantic parsing

with llms: Demonstration selection and chain-of-thought prompting via semantic fragment decoding. Knowledge-Based Systems, 114256 (2025)

[45] Huang, W., Zhou, G., Lapata, M., Vougiouklis, P., Montella, S., Pan, J.Z.: Prompting large language models with knowledge graphs for question answering involving long-tail facts. Knowledge-Based Systems, 113648 (2025)

[46] Kuppa, A., Aouad, L., Le-Khac, N.-A.: Linking cve's to mitre att&ck techniques. In: Proceedings of the 16th International Conference on Availability, Reliability and Security, pp. 1–12 (2021)

[47] Wei, J., Zou, K.: Eda: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196 (2019)

[48] Shorten, C., Khoshgoftaar, T.M., Furht, B.: Text data augmentation for deep learning. Journal of big Data **8**(1), 101 (2021)

[49] Wang, Z., Wang, P., Liu, K., Wang, P., Fu, Y., Lu, C.-T., Aggarwal, C.C., Pei, J., Zhou, Y.: A comprehensive survey on data augmentation. IEEE Transactions on Knowledge and Data Engineering (2025)

[50] Min, H.-Y., Lee, H.-Y.: Molecular targeted therapy for anticancer treatment. Experimental & Molecular Medicine **54**(10), 1670–1694 (2022)

[51] Hemberg, E., Srinivasan, A., Rutar, N., O'Reilly, U.-M.: Sourcing language models and text information for inferring cyber threat, vulnerability and mitigation relationships. AI4Cyber/MLHat: AI-enabled Cybersecurity Analytics and Deployable Defense at KDD (2022)

[52] Mitre: CWE. (accessed 1 November 2025) (2025). https://cwe.mitre.org/

[53] Mitre: CAPEC. (accessed 1 November 2025) (2025). https://capec.mitre.org/

[54] Ghorbani, M., Bahrami, M., Kazi, A., Soleymani Baghshah, M., Rabiee, H.R., Navab, N.: Gkd: Semi-supervised graph knowledge distillation for graph-independent inference. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 709–718 (2021). Springer

[55] Sun, H., Shu, H., Kang, F., Zhao, Y., Huang, Y.: Malware2att&ck: A sophisticated model for mapping malware to att&ck techniques. Computers & Security **140**, 103772 (2024)

[56] Ye, Z., Sun, X., Bo, L., Cao, S., Ren, X., Qi, L., Zhang, J.: Kg4va: Constructing vulnerability knowledge graph for software vulnerability assessment. IEEE Transactions on Services Computing (2025)

[57] Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G.D., Gutierrez, C., Kirrane, S., Gayo, J.E.L., Navigli, R., Neumaier, S., *et al.*: Knowledge graphs.

ACM Computing Surveys (Csur) **54**(4), 1–37 (2021)

[58] OASIS: STIX(TM) Version 2.0. (accessed 1 November 2025) (2017). https://www.oasis-open.org/standard/stix2-0/

[59] Neo4j: Neo4j. (accessed 1 November 2025) (2025). https://neo4j.com/

[60] Daniel Han, M.H.: unsloth. (accessed 1 November 2025) (2025). https://docs.unsloth.ai/

[61] Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., *et al.*: Lora: Low-rank adaptation of large language models. In: International Conference on Learning Representations (ICLR) (2022)

[62] Stechly, K., Valmeekam, K., Kambhampati, S.: Chain of thoughtlessness? an analysis of cot in planning. Advances in Neural Information Processing Systems **37**, 29106–29141 (2024)

[63] Kim, S., Joo, S., Kim, D., Jang, J., Ye, S., Shin, J., Seo, M.: The cot collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 12685–12708 (2023)

[64] Yadav, T., Rao, A.M.: Technical aspects of cyber kill chain. In: International Symposium on Security in Computing and Communication, pp. 438–452 (2015). Springer

[65] Yue, H., Li, T., Wu, D., Zhang, R., Yang, Z.: Detecting apt attacks using an attack intent-driven and sequence-based learning approach. Computers & Security **140**, 103748 (2024)

[66] El Jaouhari, S., Tamani, N., Jacob, R.I.: Linking-CVE-to-Mitre-Techniques. (accessed 1 November 2025) (2024). https://github.com/rocknrj/Linking-CVE-to-Mitre-Techniques

[67] Othman, R., Russo, B.: VULDAT Dataset. (accessed 1 November 2025) (2024). https://doi.org/10.6084/m9.figshare.25828102.v1

[68] Mitre: Updates. (accessed 1 November 2025) (2025). https://attack.mitre.org/resources/updates/

[69] Ahmed, I., Islam, S., Datta, P.P., Kabir, I., Chowdhury, N.U.R., Haque, A.: Qwen 2.5: A comprehensive review of the leading resource-efficient llm with potential to surpass all competitors. Authorea Preprints (2025)

[70] Abdeen, B., Al-Shaer, E., Singhal, A., Khan, L., Hamlen, K.W.: Smet: Semantic mapping of cti reports and cve to att&ck for advanced threat intelligence. Journal of Computer Security, 230218 (2024)

[71] Abdeen, B., Al-Shaer, E., Singhal, A., Khan, L., Hamlen, K.: Smet: Semantic mapping of cve to att&ck and its application to cybersecurity. In: IFIP Annual Conference on Data and Applications Security and Privacy, pp. 243–260 (2023). Springer

[72] Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al.: Qwen3 technical report. arXiv preprint arXiv:2505.09388 (2025)

Appendix A Prompt Template for Targeted Data Enhancement

<System Setup>
'''

You are an expert cybersecurity vulnerability modeling specialist. Your core task is to generate a highly matching, authentic, and credible Common Vulnerabilities and Exposures (CVE) description for a specified target MITRE ATT&CK technique.

'''
<System Setup>
<Task Description>
'''

Your task is to deconstruct the vulnerability by analyzing it in the context of the attack chain for the target ATT&CK technique. Focus on typical attack scenarios, exploitation conditions, and impact manifestations to ensure technical semantic alignment. Specifically, synthesize the vulnerability's core behavioral semantics using the following four feature classes:

1. Internal Attribute: The fundamental flaw, including vulnerability type, root cause, and trigger condition.
2. Attack Operation: The atomic-level actions performed by an attacker, such as initial access, execution, etc., directly reflecting the ATT&CK technique.
3. Scope of Impact: The range of system components, data, or users affected by the attack operation, reflecting the breadth of the vulnerability's influence.
4. Impact Consequence: The consequences of exploitation, including security attribute impacts and business impacts.

Model the logical propagation of the attack by formulating the relationships between these features:

- Exploit ability: Internal Attribute enables Attack Operation.
- Targeting: Attack Operation targets Scope of Impact.
- Determination: Scope of Impact determines Impact Consequence.

Integrate these elements to construct a comprehensive CVE description.
'''
<Task Description>
<Input>

'''
{AT_id},{ AT_name},{ AT_description},{CVE_seed}
'''
<Input>
<Output Description and Requirements>
'''

1. Your output must be a single, valid JSON object containing the following keys: CVE_ID, ATT&CK technique, CVE description.
2. The CVE_description must be a concise, professional summary that strictly adheres to the logical structure derived from the task description. It should technically precisely reflect the characteristics of the target ATT&CK technique, maintain stylistic consistency with the provided CVE seed data, and be succinct and rigorous to ensure quality and prevent semantic drift.

'''
<Output Description and Requirements>

Appendix B Prompt Template for Domain Knowledge Enhancement

<System Setup>
'''
You are a cybersecurity knowledge extraction specialist. Your task is to analyze CVE descriptions and extract structured knowledge entities and relationships conforming to a predefined vulnerability knowledge graph schema.
'''
<System Setup>
<Task Description>
'''
Deconstruct the provided CVE description to identify and extract four core entity types and their relationships:

1. Vulnerability Entities (E1): Extract CVE-specific attributes including vulnerability type, exploitation prerequisites, impact scope, and severity indicators.
2. Weakness Entities (E2): Identify underlying software/hardware flaws (CWE) that cause the vulnerability.
3. Attack Pattern Entities (E3): Extract relevant CAPEC attack patterns that could exploit the identified weakness.
4. Technique Entities (E4): Map to MITRE ATT&CK techniques and associated defensive measures.

Establish precise semantic relationships between these entities:

1. hasWeakness (R1) : Connect Vulnerability (E1) to Weakness (E2).
2. exploitedBy (R2) : Connect Weakness (E2) to Attack Pattern (E3) .
3. mapsToTechnique (R3): Connect Attack Pattern (E3) to Technique (E4).

'''
<Task Description>
<Input>
'''
{AT_id}, {AT_name}, {AT_description}, {CVE_description}
'''
<Input>
<Output Description and Requirements>
'''

1. Output must be a single, valid JSON object with the following structure:{
"entities": {
"vulnerabilities": {"id": "CVE-XXXX", "attributes": {...}},
"weaknesses": {"id": "CWE-XXX", "attributes": {...}},
"attack_patterns": {"id": "CAPEC-XXX", "attributes": {...}},
"techniques": {"id": "TXXXX", "attributes": {...}},
},
"relationships": {
{"type": "hasWeakness", "source": "CVE-XXXX", "target": "CWE-XXX"},
{"type": "exploitedBy", "source": "CWE-XXX", "target": "CAPEC-XXX"},
{"type": "mapsToTechnique", "source": "CAPEC-XXX", "target": "TXXXX"}},
}
2. All extracted knowledge must maintain semantic accuracy with respect to the source CVE description and ATT&CK technique context.
3. Entity attributes should be concise yet comprehensive, covering the key properties defined in your knowledge graph schema.

'''
<Output Description and Requirements>