Modeling Semantic Ambiguity: A Knowledge-Driven Framework for Security Attack Technique Extraction

Cheng Meng[1,2], Zhengwei Jiang[1,2], Xinyi Li[2], Fangming Dong[1,2], Qiuyun Wang[1,2], Fangli Ren[1⋆], and Baoxu Liu[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{mengcheng, jiangzhengwei, dongfangming, wangqiuyun, renfangli, liubaoxu}@iie.ac.cn
lixinyi22@mails.ucas.ac.cn

Abstract. MITRE ATT&CK Technique extraction faces inherent semantic ambiguity, leading to systematic labeling inconsistencies that limit the generalization of existing methods. Instead of training on such data, we model the ambiguity as structured textual knowledge. We present the Semantic Ambiguity Modeling Framework (SAMF), a knowledge-driven framework for Technique extraction. SAMF encodes when Techniques overlap and how to distinguish them, and leverages LLMs to automatically discover this knowledge. The framework applies this knowledge through a two-stage process, which can also enhance existing methods. Experimental results demonstrate that SAMF achieves robust cross-dataset performance, outperforming the best supervised method by 24.9% on out-of-distribution data.

Keywords: Complex System Modeling · Semantic Ambiguity · Cyber Threat Intelligence · Large Language Models

1 Introduction

Cyber Threat Intelligence (CTI) relies on the systematic analysis of threat information to enhance organizational defense [12]. The MITRE ATT&CK framework [11] provides a globally recognized standard for characterizing adversary attack Techniques. Mapping unstructured threat reports to these structured Techniques is essential for automated threat analysis, yet currently relies on resource-intensive manual expert analysis [9], making its automation a critical research objective.

The semantic boundaries between ATT&CK Techniques are inherently ambiguous [6,1], leading to systematic labeling inconsistencies across datasets. Existing extraction methods trained on such data may overfit to dataset-specific patterns, which limits their generalization to unseen data [10,15]. The black-box

⋆ Corresponding author: renfangli@iie.ac.cn

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_11

nature of these methods further compounds the problem: when discrepancies emerge, the lack of transparency prevents diagnosis and validation [5].

Retrieval-Augmented Generation (RAG) offers an alternative by grounding Large Language Model (LLM) outputs in official ATT&CK Technique definitions [1]. While these methods avoid fitting to labeling biases, they inherit the semantic ambiguity embedded in the definitions [13]. Although general approaches exist for restructuring retrieved content [8] or enriching category representations [17], they cannot identify which Techniques are semantically ambiguous or how to distinguish them without domain-specific disambiguation efforts.

To improve generalization and reliability, we propose Semantic Ambiguity Modeling Framework (SAMF), a knowledge-driven framework that models semantic ambiguity as structured textual knowledge. SAMF encodes when Techniques overlap and how to distinguish them through a dual-layer representation: shared contextual patterns that cause confusion, and distinguishing features that resolve it. The framework leverages LLMs to automatically discover this knowledge and applies it through a two-stage classification process. The reclassification stage can also enhance existing extraction methods.

Our contributions are threefold:

- Knowledge representation: We design a dual-layer representation that models semantic ambiguity as shared contexts and distinguishing features.
- Framework implementation: Based on this representation, we develop SAMF, an LLM-driven framework that synthesizes structured knowledge and applies it through multi-stage classification.
- Empirical validation: Experiments demonstrate strong performance and cross-dataset generalization, with SAMF achieving state-of-the-art results among RAG-based methods and a 24.9% relative Macro-F1 improvement over the best supervised baseline on out-of-distribution data.

To the best of our knowledge, this is the first work to model semantic ambiguity as textual knowledge for Technique extraction. Code, data, and prompts are available at GitHub[1].

2 Related Work

Recent studies have demonstrated that the ATT&CK framework contains inherent semantic ambiguity, which hinders both method generalization and practical deployment. Current research focuses on supervised learning, without attempting to explicitly model this semantic ambiguity as textual knowledge.

2.1 Semantic Ambiguity in ATT&CK Techniques

A recent systematization study [1] identifies semantic ambiguity as a fundamental challenge in ATT&CK Technique extraction. Their analysis reveals that certain Techniques are frequently misclassified due to subtle semantic differences.

[1] `https://github.com/MengC1024/ATTCK_TTP_EX`

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_11

The study concludes that the ATT&CK framework is not always sufficiently unambiguous, representing a potential source of error. Annotation studies confirm this difficulty: on the AnnoCTR dataset [6], two annotators achieve only 54% F1 when comparing Technique sets identified in documents. To illustrate, consider the sentence "The adversary acquired administrative credentials from the server." One expert might label this as `T1003 (OS Credential Dumping)`, interpreting "acquired" as extraction from memory; another might choose `T1552 (Unsecured Credentials)`, viewing it as credentials found in an insecure location.

This semantic ambiguity undermines practical deployment. Virkud et al. [13] analyze ATT&CK labeling across three major EDR vendors detecting identical threats. They report that 51% of shared malicious entities exhibit zero Technique label agreement. This prevents ATT&CK from serving as a consistent taxonomy in operational contexts.

The limited generalization of existing extraction methods further evidences this fundamental inconsistency [1]. For instance, TTPXHunter [10] reports 92.42% F1-score on its own dataset, while a DistilBERT-based method [7] reports 75.17% accuracy. However, when evaluated on an independently constructed dataset [3][2], their accuracy drops to 22.50% and 24.55% respectively. This persistent difficulty across methods suggests that semantic ambiguity is not merely a labeling issue but a structural challenge embedded in the ATT&CK framework itself.

2.2 Technique Extraction

Recent extraction methods can be categorized by whether they require task-specific training.

Training-based approaches typically fine-tune language models on labeled CTI datasets. For example, TTPXHunter [10] expands coverage to 193 Techniques using contextual data augmentation, while other works fine-tune LLMs with cybersecurity corpora [2] or integrate hierarchical attention mechanisms [16]. Recent work also explores using LLMs to synthesize training data for fine-tuning [3]. These methods often achieve strong in-distribution performance but may inherit annotation biases and degrade on out-of-distribution data.

With the emergence of powerful LLMs, non-training approaches have become viable. These methods leverage LLMs through prompting or retrieval-augmented generation (RAG). RAG-based methods ground outputs in external knowledge such as official definitions [14] or procedure examples [4]. However, directly retrieving original ATT&CK content may inherit its semantic ambiguity. When definitions overlap or lack explicit distinctions, these systems may rely on implicit LLM reasoning to resolve borderline cases.

[2] `https://github.com/starrust/Synthetic-TTP-Extraction-Dataset`

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_11

3 Method

As illustrated in Fig. 1, SAMF addresses the inherent semantic ambiguity in ATT&CK Techniques through a knowledge-driven pipeline. The framework comprises three core parts: *Knowledge Representation*, which structures ambiguity into explicit textual forms; *Knowledge Discovery*, an offline process that synthesizes this knowledge; and *Knowledge Application*, an online process that applies the knowledge to classify CTI text.

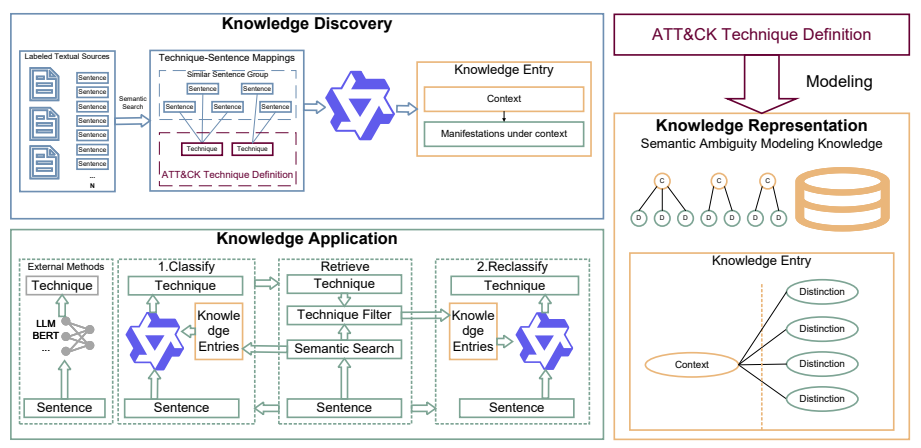

Fig. 1. Overview of the Semantic Ambiguity Modeling Framework (SAMF). The framework consists of three parts: (1) Knowledge Representation structures this knowledge; (2) Knowledge Discovery synthesizes ambiguity knowledge from data; and (3) Knowledge Application employs a retrieve-and-reason process involving Retrieve, Classify, and Reclassify components.

3.1 Knowledge Representation

Standard definitions often fail to delineate the subtle boundaries between overlapping Techniques. To address this, we propose a dual-layer representation defined as a **Knowledge Entry** E:

$$E = (C, \{D_1, D_2, ..., D_n\}) \tag{1}$$

where C denotes the **Context** and $\{D_i\}$ denotes a set of **Distinctions** for candidate Techniques.

This structure is designed to decouple the shared ambiguity (Context) from the specific resolution features (Distinctions). To demonstrate this design in a practical setting, we analyze the scenario presented in Fig. 2. As visualized in the **"Our Knowledge Representation"** panel:

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_11

– **Layer 1: Context (C).** This layer captures the **shared behavioral pattern** where confusion arises. Instead of a vague topic, it provides a specific semantic anchor. In the example, the phrase "malware creates a startup item for persistence" acts as this Context, grouping different Techniques that share this behavior.
– **Layer 2: Distinctions (D).** This layer provides contrastive manifestations to differentiate candidates within that Context.
  • For **T1547**, the distinction is explicitly defined as "modifies system settings for autostart execution via registry keys or startup folders."
  • For **T1059**, the distinction focuses on "uses scripting or command interpreter to execute persistence mechanism."

  By explicitly contrasting these features (modifying settings vs. executing scripts), our Knowledge Representation resolves the ambiguity that vague official definitions leave open.

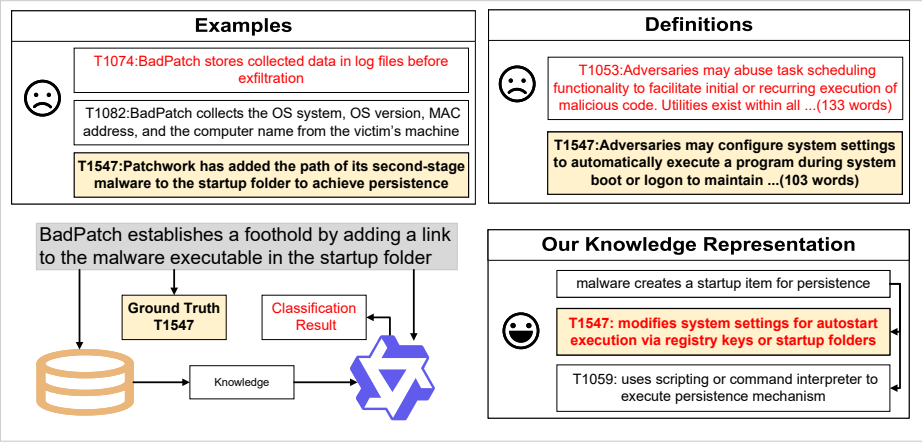

Fig. 2. Comparison of information representations. While Examples and Definitions (top) often contain noise or ambiguity, our **Knowledge Representation** (bottom) **resolves this by decoupling the Shared Context from specific Distinctions** (e.g., differentiating T1547 from T1059), providing clear decision boundaries for classification.

3.2 Knowledge Discovery

The goal of Knowledge Discovery is to automatically construct Knowledge Entries (E) from labeled textual sources and ATT&CK Technique Definitions. As illustrated in Fig. 3, this offline process transforms implicit labeling patterns into structured explicit knowledge through two steps.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_11

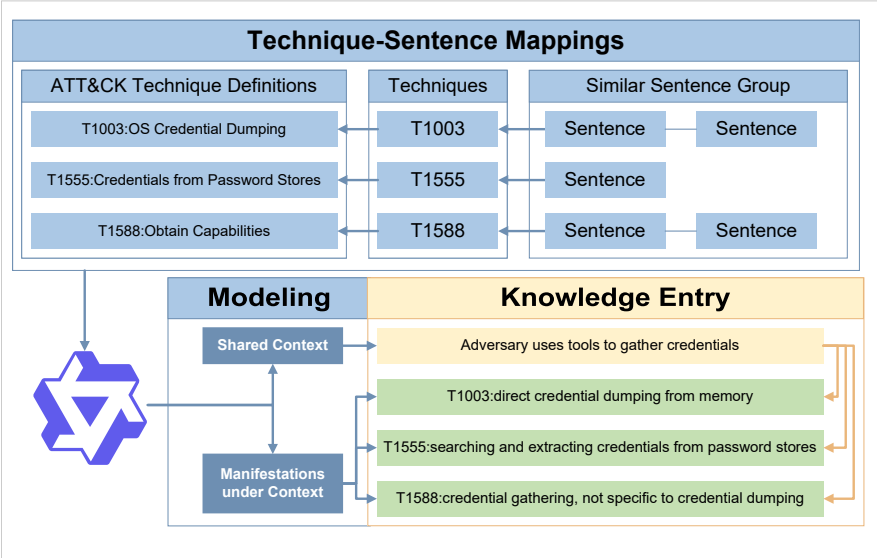

Fig. 3. The knowledge discovery process. By analyzing a group of semantically similar example sentences alongside their official definitions (e.g., T1003 vs. T1555), the model performs ambiguity **Modeling** to extract a Shared Context and specific Manifestations.

First, we construct a **Technique-Sentence Mapping** (M_s) to capture the local ambiguity. For a target sentence s, we employ semantic search to retrieve a similar sentence group G_s of k nearest neighbors from the labeled textual sources. This group serves as concrete example sentences for disambiguation. We then augment this group with official Technique definitions:

$$M_s = \{(s_j, y_j, d_{y_j}) \mid s_j \in G_s\} \tag{2}$$

where s_j is a retrieved example sentence, y_j is its Technique label, and d_{y_j} is the corresponding official definition.

Second, we employ an LLM, denoted as Φ, to perform **Ambiguity Modeling**. This step takes M_s as input to synthesize a structured Knowledge Entry E:

$$E = \Phi(M_s) = (C, \{D_y\}_{y \in Y_s}) \tag{3}$$

where Y_s is the set of unique Techniques in M_s. The output consists of:

- **Shared Context (C):** A description of the shared semantic pattern (e.g., "Adversary uses tools to gather credentials").
- **Distinctions (D_y):** A set of specific manifestations (labeled as "Manifestations under Context" in Fig. 3), where each D_y describes how Technique y uniquely manifests under context C.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_11

By iterating this process over the labeled textual sources, we transform unstructured annotations into a structured Knowledge Base, which enables the retrieval-augmented inference in the next stage.

3.3 Knowledge Application

As depicted in the Knowledge Application panel of Fig. 1, the framework maps an input CTI sentence x to a final ATT&CK Technique y^* through two stages: Classification followed by Reclassification. Both stages can produce independent classification results and utilize the unified Knowledge Entries constructed in Sec. 3.2, but access them via different retrieval pathways. The input to Reclassification is not limited to the output of Classification; it can also accept predictions from external methods, enabling Reclassification to serve as a standalone correction layer.

Stage 1: Classification. The goal of this stage is to generate a preliminary candidate Technique $\hat{y}$.

- **Retrieval:** We perform semantic search between the input sentence x and the Shared Context (C) component of all Knowledge Entries. We retrieve the top-k entries based on cosine similarity.
- **Pre-classification:** For each retrieved Knowledge Entry, an LLM determines which Technique best matches the input sentence x. We retain only the Shared Context (C) and corresponding Distinction (D) of the selected Technique from each entry, filtering out inapplicable distinctions.
- **Classification:** An LLM takes the sentence x and the processed knowledge entries as input to generate the initial classification $\hat{y}$.

Stage 2: Reclassification. This stage functions as a correction mechanism. It validates the initial candidate $\hat{y}$ by focusing on specific behavioral details.

- **Retrieval and Filtering:** We perform semantic search between x and the Distinctions (D) component of all Knowledge Entries. We filter the retrieved results to retain the entries that explicitly contain the initial candidate $\hat{y}$. This isolates the comparison set $\mathcal{D}_{group}$, which is composed of all Distinctions (D) within these entries.
- **Correction:** The LLM takes the sentence x, the initial candidate $\hat{y}$, and the comparison set $\mathcal{D}_{group}$ as input to determine the final, disambiguated label y^*. By contrasting the details in x against the specific nuances in $\mathcal{D}_{group}$, the LLM decides whether to confirm or correct the initial classification.

Note that Reclassification can operate independently: the initial candidate $\hat{y}$ can be provided by external methods (e.g., BERT-based models or other baselines), enabling Reclassification to serve as a standalone correction layer for existing extraction systems.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_11

4 Evaluation

4.1 Datasets and Models

We adopt two datasets from [9][3]. The **Procedures** dataset is derived from official MITRE ATT&CK Procedure Examples, serving as the primary corpus for Knowledge Discovery (using the training split as labeled textual sources) and evaluation (using the test split). The **Expert** dataset contains expert-labeled sentences from CTI reports. We select Expert as the generalization test set because it shares the same ATT&CK version and comparable label diversity with Procedures (121 vs. 152 Techniques), while differing significantly in text length (average 39 vs. 13 words). To align with our research scope and ensure evaluation stability, we mapped all sub-Techniques to their corresponding top-level Techniques.[4]

For fair comparison, we use Qwen3-32B as the unified base LLM across all RAG-based methods and our framework (unless otherwise specified for ablation studies). We use all-mpnet-base-v2 [5] for semantic similarity computation.

4.2 Metrics and Baselines

Given the imbalanced label distribution, we focus on Macro-averaged Precision, Recall, and F1 Score, complemented by overall Accuracy.

We benchmark against the following baselines:

Training-based: TRAM[6], released before datasets, and TTPXHunter [10], trained on the *Procedures* dataset.

LLM without Knowledge: Zero-shot prompting using GPT-5.2, Claude-Sonnet-4.5, DeepSeek-V3.2 (both Chat and Reasoner modes), and Qwen3-32B.

RAG Baselines: Following the categorization in [1], we abstract existing RAG approaches into three knowledge sources: "min-definition" (first sentence of definitions), "definition" (full definitions), and "example" (procedure examples).

Our Method: *SAMF*, our proposed framework with knowledge representation, knowledge discovery, and multi-stage classification.

For all RAG-based methods, we set the retrieval count $K = 10$ in the main results and analyze the impact of K in subsequent experiments.

[3] `https://github.com/tumeteor/mitre-ttp-mapping`

[4] Though our framework supports sub-Technique extraction, we chose to map all sub-Techniques to top-level Techniques (e.g., *T1059.001* → *T1059*) for evaluation stability. Sub-Techniques in the datasets exhibit severe long-tail distributions: in the *Procedures* test set, 34.7% of sub-Techniques have only one example, and 72.7% have five or fewer, making Macro-F1 evaluation unstable and dominated by rare classes. After mapping, the median number of examples per label increases from 2 to 5, yielding more stable evaluation. Additionally, sub-Techniques are strictly hierarchical refinements of their parents and relatively easier to distinguish. In our preliminary study, a simple definition-based baseline achieved 81% Macro-F1 on sub-Technique classification for *Procedures* and 71% for *Expert*, confirming that the main bottleneck lies at the Technique level.

[5] `https://huggingface.co/sentence-transformers/all-mpnet-base-v2`

[6] `https://github.com/center-for-threat-informed-defense/tram`

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_11

4.3 Main Result

Table 1 presents the main results. SAMF achieves the best performance among all non-training methods on both datasets. Notably, SAMF substantially outperforms zero-shot Qwen3-32B (F1: 0.641 vs. 0.158 on *Procedures*, 0.391 vs. 0.111 on *Expert*), demonstrating the significant value of knowledge over raw model capability. On *Procedures*, SAMF (F1: 0.641) outperforms stronger LLMs including Claude-Sonnet-4.5 (0.638) and DeepSeek-Reasoner (0.617), despite these models having significantly more parameters. On *Expert*, SAMF achieves the highest F1 (0.391) across all methods, surpassing GPT-5.2 (0.347) by 12.7%, demonstrating strong generalization capability. For training-based methods, while TTPXHunter achieves the highest in-distribution performance on *Procedures*, it shows limited generalization to *Expert*, where SAMF outperforms it by 24.9% in relative Macro-F1 (0.391 vs. 0.313). We further analyze this gap in Section 5.1.

Table 1. Main results on *Procedures* (in-distribution) and *Expert* (generalization) datasets. Best results in each column are in **bold**.

		Procedures Dataset				Expert Dataset			
	Method	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
Training-based	TRAM	0.655	0.240	0.296	0.255	0.224	0.111	0.165	0.113
	TTPXHunter	**0.952**	**0.901**	**0.886**	**0.883**	0.447	0.341	0.331	0.313
LLM without Knowledge	Qwen3-32B	0.378	0.183	0.186	0.158	0.243	0.130	0.128	0.111
	GPT-5.2	0.766	0.636	0.616	0.592	0.476	0.370	0.381	0.347
	Claude-Sonnet-4.5	0.767	0.674	0.649	0.638	0.453	0.363	0.372	0.339
	DeepSeek-Chat	0.633	0.443	0.436	0.412	0.386	0.277	0.274	0.250
	DeepSeek-Reasoner	0.776	0.638	0.646	0.617	0.463	0.358	0.373	0.334
RAG Baselines	min-definition	0.570	0.490	0.559	0.473	0.392	0.367	0.393	0.338
	definition	0.621	0.538	0.619	0.520	0.388	0.348	0.388	0.330
	example	0.724	0.570	0.554	0.536	0.433	0.335	0.377	0.334
Ours	SAMF	0.772	0.669	0.667	0.641	**0.465**	**0.401**	**0.446**	**0.391**

To isolate the contribution of SAMF, Fig. 4 compares RAG-based methods using the same base model (Qwen3-32B).

SAMF maintains a dominant lead across all retrieval counts (K) on both datasets, **establishing a significant margin even at** $K = 1$. This consistent superiority validates the effectiveness of our knowledge-driven framework in resolving semantic ambiguity. Unlike baselines that are often limited by retrieval noise, SAMF effectively leverages increased retrieval counts (K) to achieve significantly higher performance peaks on both *Procedures* (F1: 0.682) and *Expert* (F1: 0.404).

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_11

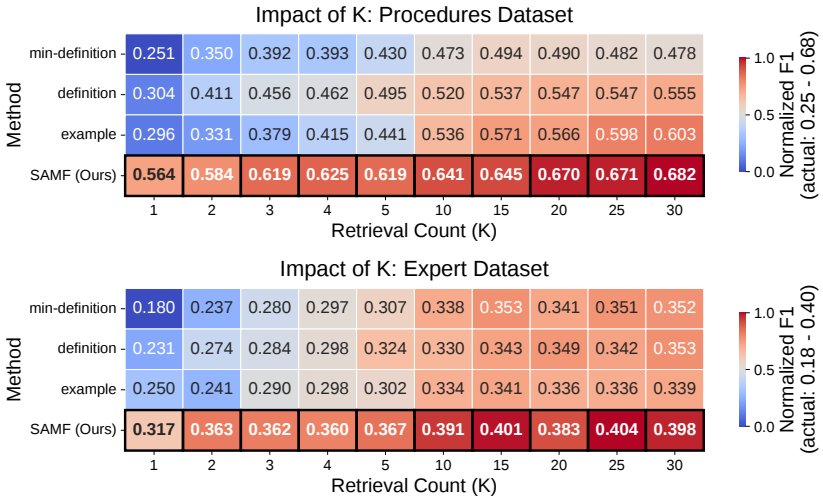

Fig. 4. Impact of retrieval count (K) on RAG-based methods. Bold values and black boxes indicate the highest F1 score for each K.

4.4 Reclassification on External Methods

Fig. 5 and Fig. 6 illustrate the performance impact (ΔF1) of applying our Reclassification stage to various baselines. The results demonstrate that our Reclassification stage functions as a robust and cost-effective enhancement layer: it delivers universal gains and provides efficient correction capabilities even at minimal retrieval depths. We analyze these findings from two key perspectives:

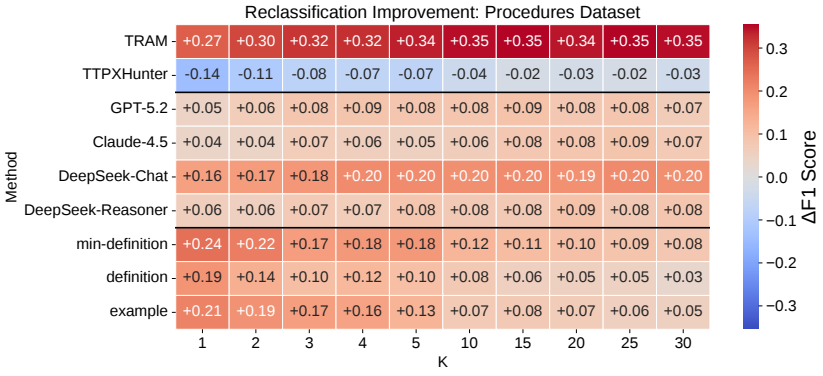

Fig. 5. Reclassification improvement (ΔF1) on *Procedures* dataset.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_11

Fig. 6. Reclassification improvement (ΔF1) on *Expert* dataset.

First, we observe a **consistent improvement trend**, particularly under distribution shift. On the out-of-distribution *Expert* dataset, our method acts as a universal enhancement layer, boosting performance for every single baseline. This confirms that our approach effectively injects generalization capabilities where supervised methods may struggle.

Second, the results highlight the **high efficiency** of our approach. Significant benefits are realized **even at minimal retrieval depths**: on *Procedures*, TRAM improves by +0.27 and DeepSeek-Chat by +0.16 at just $K = 1$. Despite the natural marginal effects observed on stronger baselines, this capacity to rectify errors with minimal external context demonstrates that our structured knowledge is highly discriminative and resource-efficient.

4.5 Ablation Study

Fig. 7 summarizes the contribution of each component.

Impact of Knowledge Discovery. Both *Definitions* and *Example Sentences* prove indispensable. Removing either leads to consistent degradation, with the gap widening at higher K, indicating that high-quality entries are a prerequisite for scaling up retrieval.

Complementary Roles in Knowledge Application. The *Reclassify* stage acts as a precision refiner at low K, while the *Pre-classify* stage functions as a noise filter at high K. Removing both results in the worst consistent performance, validating their complementary roles.

4.6 Model Generalization

We evaluate SAMF's robustness by varying the backbone LLM for Knowledge Discovery and Knowledge Application separately (Fig. 8 and Fig. 9).

For Knowledge Discovery, the lightweight Qwen3-30B-A3B (3B active parameters) generates knowledge as effective as the larger main model, confirming that high-quality knowledge extraction does not require giant models.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_11

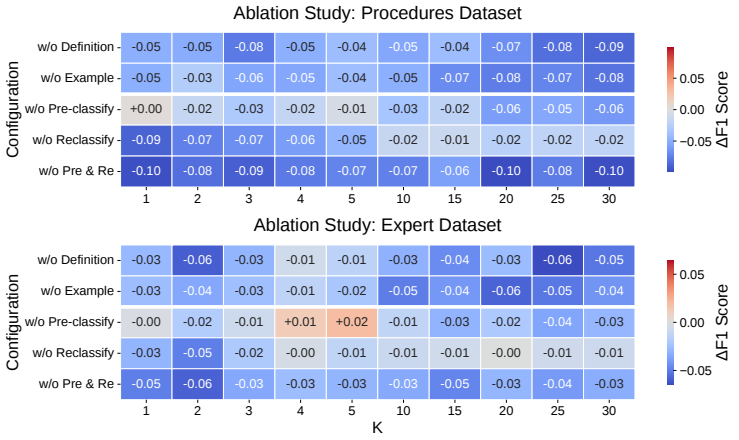

Fig. 7. Ablation study on Knowledge Discovery (top 2 rows) and Knowledge Application (bottom 3 rows). Values indicate ΔF1 relative to the full SAMF model. (Top) Procedures dataset. (Bottom) Expert dataset.

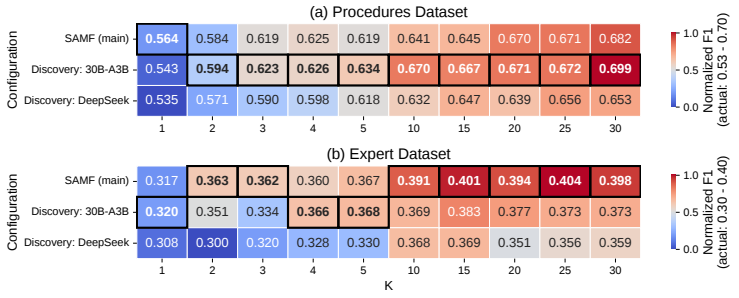

Fig. 8. Model Generalization: Knowledge Discovery. Different LLMs generate knowledge, all use Qwen3-32B for classification. (a) Procedures dataset. (b) Expert dataset.

For Knowledge Application, SAMF with Qwen3-30B-A3B consistently outperforms all RAG baselines across all K values, mirroring the main experiment results. Moreover, it surpasses proprietary models: achieving F1 of 0.645 on *Procedures* (vs. Claude-Sonnet-4.5's 0.638) and 0.374 on *Expert* (vs. GPT-5.2's 0.347), demonstrating that our framework effectively compensates for limited model capacity.

5 Discussion and Limitations

5.1 Generalization Matters: Fitting ≠ Solving

Our results expose a critical distinction between data fitting and task solving. While TTPXHunter dominates the in-distribution Procedures dataset (F1:

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_11

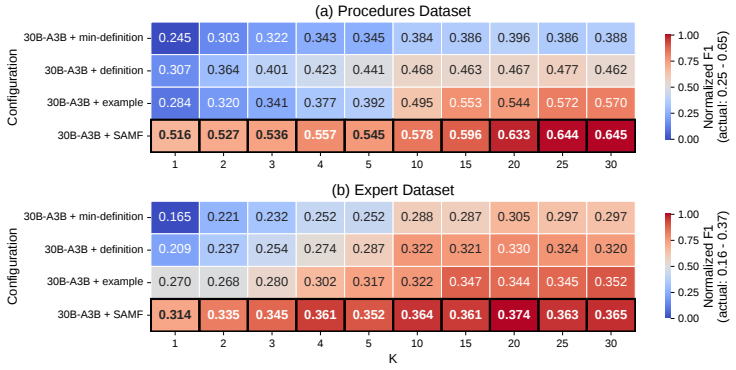

Fig. 9. Model Generalization: Knowledge Application. Qwen3-30B-A3B as classification model with knowledge generated by Qwen3-32B. (a) Procedures dataset. (b) Expert dataset.

0.883), its performance collapses on the out-of-distribution Expert dataset (F1: 0.313), falling behind the zero-shot Claude-Sonnet-4.5 (F1: 0.339). While factors such as annotation inconsistencies, text characteristic differences, and model scale disparity do affect absolute performance across datasets, other methods evaluated under the same conditions do not exhibit comparable degradation—SAMF (0.641 $\rightarrow$ 0.391) and RAG baselines maintain relatively stable performance, while TTPXHunter suffers the most severe drop among all evaluated methods. This suggests that overfitting to source-specific patterns is the dominant factor, highlighting the need for both cross-dataset evaluation and methods with stronger generalization capabilities.

5.2 Modeling vs. Eliminating Ambiguity

The root cause of ambiguity lies within the ATT&CK framework itself: official definitions describe *what* each Technique represents, but often lack explicit guidance on *how* to distinguish similar Techniques. Rather than revising the taxonomy, SAMF complements it by modeling and discovering the missing **decision boundaries**. In other words, our framework does not eliminate the definitional ambiguity itself, but operates within the boundaries of current definitions and annotations.

5.3 Computational Trade-offs

Compared to RAG baselines, SAMF introduces two additional LLM calls per sample (pre-classification and reclassification), resulting in three calls total versus one for standard RAG. However, our structured knowledge representation (Shared Contexts and Distinctions) provides explicit decision boundaries that raw definitions and examples lack, while being more compact (averaging 56

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_11

words per entry vs. 177 words per official definition). The resulting knowledge base contains 7,794 entries covering 174 Techniques, with each entry comparing 3.2 Techniques on average. The knowledge discovery phase is offline, requiring approximately one LLM call per training sample, and is executed only once per taxonomy version, amortizing its cost across all subsequent inferences.

6 Conclusion and Future Work

In this work, we proposed the Semantic Ambiguity Modeling Framework (SAMF) to address the inherent semantic ambiguity in ATT&CK Technique extraction. By restructuring implicit confusion into explicit Shared Contexts and Distinctions, we enable LLMs to effectively navigate complex decision boundaries. Our evaluation demonstrates that SAMF achieves both strong in-distribution performance and robust cross-dataset generalization. The consistent improvements, particularly under distribution shift, highlight the importance of addressing semantic ambiguity in Technique extraction.

Future work includes extending the framework to sub-Technique-level extraction, conducting deeper analysis of how LLM behavioral characteristics and prompt sensitivity influence the framework's performance, and investigating the scalability of knowledge discovery across evolving ATT&CK versions.

Acknowledgments. This work was supported by the Youth Innovation Promotion Association, CAS (No. 2023170) and Beijing Key Laboratory of Network Security and Protection Technology.

References

1. Büchel, M., Paladini, T., Longari, S., Carminati, M., Zanero, S., Binyamini, H., Engelberg, G., Klein, D., Guizzardi, G., Caselli, M., et al.: {SoK}: Automated {TTP} extraction from {CTI} reports–are we there yet? In: 34th USENIX security symposium (USENIX Security 25). pp. 4621–4641 (2025)
2. 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). https://doi.org/https://doi.org/10.1016/j.cose.2024.104213, https://www.sciencedirect.com/science/article/pii/S0167404824005194
3. Dong, F., Jiang, Z., Ma, C., He, Q., Yang, P., Yao, Y., Wang, J.: From threat report to att&ck: Automated extraction and reasoning of ttps using large language models. In: 2025 28th International Conference on Computer Supported Cooperative Work in Design (CSCWD). pp. 860–865. IEEE (2025)
4. Fayyazi, R., Taghdimi, R., Yang, S.J.: Advancing ttp analysis: Harnessing the power of large language models with retrieval augmented generation. In: 2024 Annual Computer Security Applications Conference Workshops (ACSAC Workshops). pp. 255–261 (2024). https://doi.org/10.1109/ACSACW65225.2024.00036
5. Gao, Y., Gu, S., Jiang, J., Hong, S.R., Yu, D., Zhao, L.: Going beyond xai: A systematic survey for explanation-guided learning. ACM Computing Surveys **56**(7), 1–39 (2024)

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_11

6. Lange, L., Müller, M., Torbati, G.H., Milchevski, D., Grau, P., Pujari, S.C., Friedrich, A.: Annoctr: A dataset for detecting and linking entities, tactics, and techniques in cyber threat reports. In: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024). pp. 1147–1160 (2024)
7. Li, L., Huang, C., Chen, J.: Automated discovery and mapping att&ck tactics and techniques for unstructured cyber threat intelligence. Computers & Security **140**, 103815 (2024)
8. Li, Z., Hu, X., Liu, A., Zheng, K., Huang, S., Xiong, H.: *Refiner*: Restructure retrieved content efficiently to advance question-answering capabilities. In: Al-Onaizan, Y., Bansal, M., Chen, Y.N. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2024. pp. 8548–8572. Association for Computational Linguistics, Miami, Florida, USA (Nov 2024). https://doi.org/10.18653/v1/2024.findings-emnlp.500, https://aclanthology.org/2024.findings-emnlp.500/
9. Nguyen, T., Šrndić, N., Neth, A.: Noise contrastive estimation-based matching framework for low-resource security attack pattern recognition. In: Graham, Y., Purver, M. (eds.) Findings of the Association for Computational Linguistics: EACL 2024. pp. 355–373. Association for Computational Linguistics, St. Julian's, Malta (Mar 2024), https://aclanthology.org/2024.findings-eacl.25/
10. Rani, N., Saha, B., Maurya, V., Shukla, S.K.: Ttpxhunter: Actionable threat intelligence extraction as ttps from finished cyber threat reports. Digital Threats: Research and Practice **5**(4), 1–19 (2024)
11. Strom, B.E., Applebaum, A., Miller, D.P., Nickels, K.C., Pennington, A.G., Thomas, C.B.: Mitre att&ck: Design and philosophy. In: Technical report. The MITRE Corporation (2018)
12. Sun, N., Ding, M., Jiang, J., Xu, W., Mo, X., Tai, Y., Zhang, J.: Cyber threat intelligence mining for proactive cybersecurity defense: A survey and new perspectives. IEEE Communications Surveys & Tutorials **25**(3), 1748–1774 (2023)
13. Virkud, A., Inam, M.A., Riddle, A., Liu, J., Wang, G., Bates, A.: How does endpoint detection use the MITRE ATT&CK framework? In: 33rd USENIX Security Symposium (USENIX Security 24). pp. 3891–3908 (2024)
14. Wudali, P.N., Kravchik, M., Malul, E., Gandhi, P.A., Elovici, Y., Shabtai, A.: Rule-att&ck mapper (ram): Mapping siem rules to ttps using llms (2025), https://arxiv.org/abs/2502.02337
15. You, Y., Jiang, J., Jiang, Z., Yang, P., Liu, B., Feng, H., Wang, X., Li, N.: Tim: threat context-enhanced ttp intelligence mining on unstructured threat data. Cybersecurity **5**(1), 3 (2022)
16. Zhang, J., Wen, H., Li, L., Zhu, H.: Unittp: A unified framework for tactics, techniques, and procedures mapping in cyber threats. In: 2024 IEEE 23rd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). pp. 1580–1588 (2024). https://doi.org/10.1109/TrustCom63139.2024.00218
17. Zhang, Y., Yang, R., Xu, X., Li, R., Xiao, J., Shen, J., Han, J.: Teleclass: Taxonomy enrichment and llm-enhanced hierarchical text classification with minimal supervision. In: Proceedings of the ACM on Web Conference 2025. pp. 2032–2042 (2025)

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_11