ALIGN TO ATTRIBUTE: APT ATTRIBUTION VIA ENTITY-LEVEL SEMANTIC ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Threat actor attribution from cyber threat intelligence (CTI) reports commonly predicts actor labels from historical reports. However, many advanced persistent threat (APT) actors have few reports, with evidence fragmented across heterogeneous entities and unstructured text, making reliable attribution difficult for tail actors. We explore entity-level semantic alignment for attribution, comparing unknown and known actors through associated entities and graph contexts. We construct HEAA as a benchmark for long-tail attribution evaluation. Our analysis quantifies the trade-off between evidence discriminativeness and cross-report generalizability and identifies limitations of individual evidence types for attribution. Based on these findings, we propose THREATALIGN, a large language model (LLM)-based alignment framework that recursively integrates selected graph evidence and external knowledge into semantic profiles. The framework combines profile semantics with supporting evidence for cross-graph entity alignment to identify unknown actors. THREATALIGN achieves 43.5% macro-F1 on HEAA, 16.6 percentage points above the strongest evaluated baseline, and remains close to the best methods on conventional attribution (AADM+) and general-purpose entity alignment (ICEWS-WIKI). Our code is available at `https://anonymous.4open.science/r/ThreatAlign-31F6/`.

1 INTRODUCTION

Cyber threat intelligence (CTI)-based attribution identifies the actor behind attack activity using threat evidence documented in public reports (Xiao et al., 2025). In the advanced persistent threat (APT) setting, attribution helps defenders understand adversary behavior and develop targeted defensive measures (Wang et al., 2024). Public CTI reports, however, have a long-tailed distribution of actor coverage. In the Malpedia snapshot used in this study, 64.48% of 960 actors have no more than two associated reports (Plohmann et al., 2017). Attribution for these actors must draw on limited observations of their behavior.

Existing CTI attribution methods use different evidence representations and reasoning mechanisms. Rule-based methods match observed evidence against predefined knowledge (Shin et al., 2021), while text-based methods learn actor labels from report features (Perry et al., 2019). APT-ATT further uses data augmentation and ensemble learning to address class imbalance and improve prediction stability (Cai et al., 2025). Graph-based methods organize threat entities and their relations to provide attribution context (Ren et al., 2022), using graph representation learning or resource-reuse analysis to exploit these relations (Xiao et al., 2024; King et al., 2025). More recently, TAA-EPLMR incorporates a large language model (LLM) to reason over evidence paths retrieved from a CTI knowledge graph (Xiao et al., 2025).

These methods remain constrained by the evidence available for tail actors. With few labeled reports, supervised models have limited observations from which to learn actor-specific patterns. For graph-based methods, limited overlap in entities and relations can make it difficult to associate an actor's activities across reports. Differences in names and descriptions across sources further complicate this association, while reports may expose different parts of an actor's behavior and yield inconsistent local structures. At the same time, several actors may share tools and attack patterns. Exact correspondences therefore have limited coverage, and similarities in common behaviors can

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

be ambiguous. Scarce reports and limited cross-graph evidence reuse make reliable attribution of tail actors difficult for existing methods.

To use relational evidence under these conditions, we formulate APT attribution as entity-level semantic alignment and propose THREATALIGN. It compares entities and graph contexts across unattributed and attributed CTI to rank known candidate actors. Evidence saliency scores guide context selection. A frozen LLM combines the selected context with external knowledge to recursively build entity profiles independently within each graph, aggregating lower-level evidence and relations up to the actor level. The system retrieves candidate entities and compares their profiles and supporting evidence to determine entity correspondences and actor attributions.

Evaluating attribution with scarce reports requires examining results for tail actors and the evidence available to distinguish them. We construct HEAA with 501 reports covering 84 APT actors, providing actor labels and cross-graph entity correspondences. Report-count analyses evaluate attribution under limited evidence, while entity labels support alignment evaluation and analysis of evidence reusability. THREATALIGN achieves 43.5% macro-F1 (M-F1) on HEAA, exceeding the strongest baseline by 16.6 percentage points, with margins of 14.3 and 16.9 points for actors with one and two query reports. Our contributions are as follows:

- We formulate APT attribution as entity-level semantic alignment and propose THREATALIGN, which recursively aggregates selected graph evidence into profiles to compare candidate actors.
- We construct HEAA with actor and entity correspondence labels to evaluate long-tail attribution.
- We show gains for tail actors and analyze the contributions of profiling and external knowledge.

2 PROBLEM FORMULATION

We consider CTI attribution from evidence grouped by actor, where actors may use low-cost behavioral variations to mislead attribution. Adaptive attacks directly targeting the learning system are outside our scope. Evidence may be incomplete and vary across report sources. Conventional fixed-label classification learns actor-specific patterns from labeled reports and uses them to predict predefined actor labels for new evidence. Scarce reports for tail actors make these patterns difficult to learn reliably.

Under the entity-level semantic alignment formulation, the reference graph $\mathcal{G}_R$ and query graph $\mathcal{G}_Q$ contain APT actors with known and unknown identities, respectively, together with their associated threat entities. Entity-level semantic alignment uses entity attributes and relational context to match query and reference entities with different names, descriptions, or local structures to identify query actors. CTI is represented as a heterogeneous directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \phi, \psi)$, where $\mathcal{V}$ and $\mathcal{E}$ denote the entity and relation sets, ϕ and ψ specify entity and relation types, and $a(v)$ denotes entity attributes. For a query entity v_q, let $\mathcal{V}_R^{\phi_Q(v_q)}$ denote reference entities of the same type. Given a similarity scoring function s and an acceptance threshold setting τ, the alignment function returns reference entities meeting this condition:

$$f_{\text{align}}(v_q) = \{v_r \in \mathcal{V}_R^{\phi_Q(v_q)} \mid s(v_q, v_r) \geq \tau\}. \tag{1}$$

Here, $s(v_q, v_r)$ measures entity similarity. For a query actor entity, the attribution method selects the reference actor with the highest similarity score from the nonempty alignment results.

3 HEAA DATASET AND ANALYSIS

We construct HEAA with actor labels for long-tail attribution evaluation and cross-graph entity correspondences for alignment evaluation and evidence analysis. HEAA contains 501 CTI reports published between 2018 and 2025, covering 84 APT actors from OpenCTI (Filigran, 2025). Among these actors, 30 have only one report in each of the reference and query graphs, and only 11 have more than five reports in both. Appendix A describes data splitting and annotation.

HEAA exhibits greater cross-graph heterogeneity than ICEWS-WIKI (Jiang et al., 2024b). Entity overlap is 14.23% in its query graph and 11.98% in its reference graph, compared with 45.79% and 31.82% on ICEWS-WIKI. Structural similarity is 3.02%, compared with 15.40% on ICEWS-WIKI. Full statistics appear in Appendix Table 6.

We further analyze evidence through cross-graph reusability and actor discriminativeness. Discriminative anchors (DAs) occur in both graphs and are associated with one actor; ambiguous anchors (AAs) are also reusable across graphs but are associated with multiple actors. Non-matching nodes (NNs) lack validated direct correspondences, although they may still supply semantic or relational context. Only 4.5% of HEAA entities are DAs, whereas 71.7% are NNs, indicating limited directly discriminative cross-graph evidence.

Table 1: Evidence utility on HEAA.

Measure	Malware	Attack pattern	IoC
DA (%)	7.7	0.4	2.3
AA (%)	1.6	20.0	0.1
NN (%)	90.7	79.6	97.6
Bd. AMC	8.49	11.32	7.67
Disc. AMC	1.07	9.62	2.16

Average Minimum Distinguishing Set Cardinality (AMC) measures the mean size of the smallest entity set needed to uniquely identify an actor. Discriminative AMC includes only identifiable samples; Bounded AMC caps identification costs at 12 and assigns this value to unresolved samples. In Table 1, malware and indicators of compromise (IoCs) have Discriminative AMC values of 1.07 and 2.16, requiring few entities on identifiable samples, but their NN proportions reach 90.7% and 97.6%. Attack patterns have a lower NN proportion (79.6%), an AA proportion of 20.0%, and a Discriminative AMC of 9.62. Thus, malware and IoCs are highly discriminative but have limited cross-report generalizability; attack patterns generalize better but require more behaviors in combination to distinguish actors. The quantitative findings on HEAA are consistent with prior research (Bianco, 2013; van der Horst et al., 2025).

4 THREATALIGN FRAMEWORK

The analysis in Section 3 shows that direct correspondences have limited coverage and shared behaviors can be ambiguous across actors. THREATALIGN aggregates entity attributes and relations in the query graph $\mathcal{G}_Q$ and reference graph $\mathcal{G}_R$ into semantic profiles, using relational context to compare actor evidence from limited reports across different names or local structures. Figure 1 shows three stages: Evidence Saliency Estimation (ESE) computes entity saliency scores, Hierarchical Semantic Profiling (HSP) uses these scores to select related evidence and recursively build profiles, and Evidence-Constrained Alignment (ECA) retrieves candidate entities and evaluates the evidence for their correspondence to produce alignment and attribution results.

The knowledge base $\mathcal{K}$, built from MITRE ATT&CK and Malpedia (Plohmann et al., 2017), contains a structured relation graph $\mathcal{G}_\mathcal{K}$ and a report-text index. Entity associations provide saliency priors for ESE, while retrieved relations and report text supply semantic context for HSP; Appendix B.1 describes its construction.

Inspired by the Pyramid of Pain (Bianco, 2013), we use a four-level evidence hierarchy $\mathcal{P} = \{L_1, \ldots, L_4\}$ for IoCs, attack patterns, malware, and actors, respectively. The hierarchy specifies the aggregation order. In the hierarchical graph $\mathcal{H}$, an edge (u, v) indicates that lower-level entity u supplies evidence for higher-level entity v, and $\ell(v)$ denotes the entity's level. The neighborhood $\mathcal{N}^{\mathcal{H}}_{\text{low}}(v)$ contains its directly connected one-hop predecessors at any lower level:

$$\mathcal{N}^{\mathcal{H}}_{\text{low}}(v) = \{u \in \mathcal{V}_\mathcal{H} \mid (u, v) \in \mathcal{E}_\mathcal{H},\ \ell(u) < \ell(v)\}. \tag{2}$$

Profiles are built separately along this hierarchy in each graph, with lower-level profiles supplying the inputs to higher levels.

4.1 EVIDENCE SALIENCY ESTIMATION

Shared tools and frequent attack patterns often connect to multiple actors, so an entity's attribution value depends on its discriminability and support from related evidence. ESE uses these factors to score entities for evidence selection under a limited profiling budget: discriminability D accounts for sharing through upper-level neighbor frequency, and corroboration C measures support from lower-level evidence. HSP uses the scores to select neighborhoods, with retained shared behaviors contributing to profiles alongside other evidence.

Local connections in a limited set of reports may not fully reflect how widely an entity is shared. As shown in Figure 1A, ESE computes local and prior terms from associations in the input graph

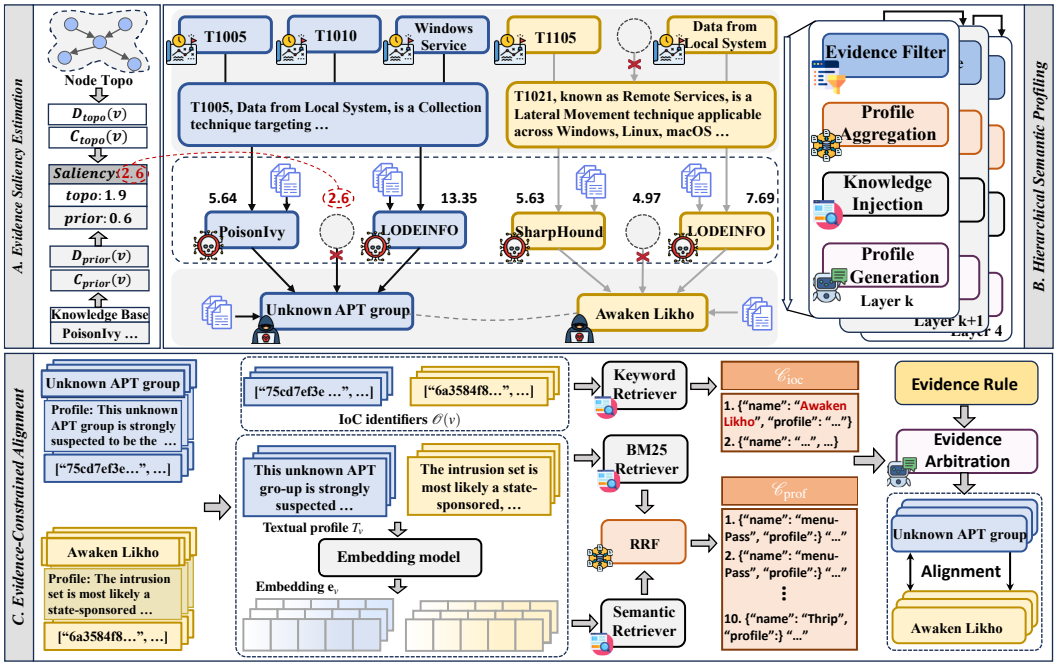

Figure 1: Overview of the THREATALIGN framework.

and knowledge base, respectively, and combines them into entity saliency scores. Let $\mathcal{H}_{\text{topo}} = \mathcal{G}$ and $\mathcal{H}_{\text{prior}} = \mathcal{G}_{\mathcal{K}}$, with subscripted quantities computed on the corresponding graph. With mixing coefficient $\alpha \in [0, 1]$,

$$f_{\text{sal}}(v) = \alpha D_{\text{topo}}(v) C_{\text{topo}}(v) + (1 - \alpha) D_{\text{prior}}(v) C_{\text{prior}}(v). \tag{3}$$

For $\text{type} \in \{\text{topo}, \text{prior}\}$, let $N_{\text{up}}^{\text{type}}(v)$ denote the number of upper-level neighbors of v and $N_{\text{total}}^{\text{type}}$ the total number of nodes eligible as upper-level neighbors in the corresponding graph. Discriminability takes an inverse-frequency form:

$$D_{\text{type}}(v) = \begin{cases} 0, & N_{\text{up}}^{\text{type}}(v) = 0, \\ \log\left(\dfrac{N_{\text{total}}^{\text{type}}}{N_{\text{up}}^{\text{type}}(v)} + 1\right) + 1, & \text{otherwise.} \end{cases} \tag{4}$$

Shared evidence connected to more upper-level entities receives lower discriminability scores. The corroboration term additionally uses lower-level evidence supporting the current entity. For $v \in L_k$, let $\mathcal{V}_{<k}^{\mathcal{H}_{\text{type}}}$ contain all nodes at lower levels in the corresponding graph. Then,

$$C_{\text{type}}(v) = 1 + \log\left(1 + \gamma \frac{\sum_{u \in \mathcal{N}_{\text{low}}^{\mathcal{H}_{\text{type}}}(v)} f_{\text{sal}}(u)}{\sum_{x \in \mathcal{V}_{<k}^{\mathcal{H}_{\text{type}}}} f_{\text{sal}}(x)}\right). \tag{5}$$

Here, γ controls the corroboration scale. The term measures the neighborhood's share of total lower-level saliency. Scores are computed bottom-up, with corroboration set to 1 for L_1 and whenever the denominator is zero. Input entities are matched to the knowledge base through normalized names or identifiers; a missing correspondence gives a zero prior term.

4.2 HIERARCHICAL SEMANTIC PROFILING

To compare combinations of shared behaviors, HSP selects related evidence using ESE scores and recursively aggregates behavioral context distributed across lower-level entities into actor representations. The process runs separately in each graph, reusing lower-level profiles in higher-level profiles to compare actors across graphs with limited reports.

For an entity v at level L_k, HSP partitions the lower-level neighborhood by level and retains up to p predecessors with the highest f_{sal} scores within each lower level, forming $\mathcal{N}'_{\text{low}}(v)$; all predecessors are retained when a level contains fewer than p. The selected entities and their relations to v supply related evidence for profiling. HSP also uses the name of v and the names of retained lower-level neighbors separately to retrieve entity relations and relevant report passages from $\mathcal{K}$, forming external context $\mathcal{C}_{\mathcal{K}}(v)$. When the current entity has an ambiguous name or limited description, retained neighbors provide retrieval cues to supplement the limited observations in the input graph. Appendix B.2 gives the complete retrieval configuration.

Bottom-level IoCs retain their raw names or identifiers. For a selected lower-level entity u, the input representation is $r_u = a(u)$ when $u \in L_1$ and $r_u = T_u$ for other lower-level entities with generated profiles. A frozen LLM generates a textual profile from the current entity's attributes $a(v)$, the selected lower-level representations with their relation types $\psi(u, v)$ to v, and external context:

$$T_v = f_{\text{LLM}}(a(v), \mathcal{C}_{\mathcal{K}}(v), \{(r_u, \psi(u, v)) \mid u \in \mathcal{N}'_{\text{low}}(v)\}; \Theta_{\text{fixed}}), \tag{6}$$

where Θ_{fixed} denotes frozen parameters. For the unknown actor on the left of Figure 1, selection under the profiling budget retains PoisonIvy (5.64) and LODEINFO (13.35) based on their ESE scores and excludes the crossed-out node scored 2.6 from the actor's profile input. Along the PoisonIvy branch, T1005's description of data collection from the local system and its relation to PoisonIvy contribute to generating the malware profile. The actor profile then reuses the retained PoisonIvy and LODEINFO profiles, combining their lower-level behavioral context into the actor representation. An embedding model produces $\mathbf{e}_v = f_{\text{emb}}(T_v)$, and the text and embedding in $\mathbf{h}_v = \{T_v, \mathbf{e}_v\}$ support lexical and semantic retrieval in ECA, respectively.

4.3 Evidence-Constrained Alignment

ECA combines semantic comparison of profiles with available direct correspondences through candidate generation and evidence arbitration for cross-graph alignment. As shown in Figure 1C, for a query entity v_q, BM25 uses the textual profile generated by HSP and vector retrieval uses its embedding to search reference entities of the same type. Reciprocal rank fusion (RRF) (Cormack et al., 2009) combines their rankings, and the top K_{prof} results form $\mathcal{C}_{\text{prof}}(v_q)$. Let $\mathcal{O}(v)$ denote the raw IoC identifiers in the lower-level evidence associated with entity v. IoC matching additionally selects same-type entities sharing at least one such identifier with v_q, allowing exact correspondences missed by profile retrieval to enter the comparison:

$$\mathcal{C}_{\text{ioc}}(v_q) = \{v_r \in \mathcal{V}_R^{\phi_Q(v_q)} \mid \mathcal{O}(v_q) \cap \mathcal{O}(v_r) \neq \varnothing\}. \tag{7}$$

Arbitration uses the combined candidate set $\mathcal{C}_{\text{cand}}(v_q) = \mathcal{C}_{\text{prof}}(v_q) \cup \mathcal{C}_{\text{ioc}}(v_q)$.

In Figure 1C, the unknown actor enters evidence arbitration with candidates including Awaken Likho. The frozen LLM compares the query and all candidates in one prompt using their attributes, textual profiles, and retrieval evidence. Shared IoCs or explicit name and alias correspondences provide direct evidence; consistency in attack patterns, malware, and targeting characteristics provides profile evidence; retrieval ranks supply auxiliary information. The model assigns similarity scores $s(v_q, v_r)$ from this evidence to rank candidates. Candidates satisfying the conditions specified by the threshold setting τ form $f_{\text{align}}(v_q)$ in the problem formulation. For an actor entity, the highest-scoring accepted candidate is the attribution result, with an empty set yielding rejection. Appendix B.3 gives the compound acceptance rule.

5 Evaluation

5.1 Experimental Setup

Datasets and Baselines. We evaluate long-tail attribution on HEAA, conventional report attribution on AADM+ (Cai et al., 2025) with 12 actors, and general-domain alignment on ICEWS-WIKI (Jiang et al., 2024b). HEAA splits reports within each actor 1:1 between reference and query graphs; AADM+ retains its 8:2 train–test split, with reports converted into graphs through entity and relation extraction. Attribution baselines cover SMOBI feature classifiers (Perry et al., 2019), pretrained language models (Aghaei et al., 2022; Jang et al., 2024), and domain-specific methods (Cai et al.,

2025; Xiao et al., 2024; King et al., 2025). Classifiers learn actor labels from reference-side reports. Appendices A and C detail graph conversion and baseline configurations.

Implementation Details. THREATALIGN uses frozen DeepSeek-V3.2 (Liu et al., 2025) for inference and Qwen3-Embedding-8B (Yang et al., 2025) for embeddings. Reports in the knowledge-base text index are disjoint from those in both HEAA graphs. Except for the explicitly noted diagnostic experiments, experiments are independently repeated three times with fixed global random seeds, and we report means and standard deviations. Appendices B.1 and B provide knowledge-base construction and implementation settings.

Evaluation Metrics. Attribution reports macro-precision (M-P), macro-recall (M-R), and macro-F1 (M-F1), with M-F1 as the primary metric. Report-level classifier predictions are aggregated with a ground-truth-assisted any-hit rule: an actor is correct if any of its reports is predicted correctly; otherwise, the most frequent predicted label is used. For entity alignment, we report Hits@K ($K \in \{1, 5, 10\}$) and mean reciprocal rank (MRR). To account for many-to-many correspondences, we additionally define Adjusted Answer Ranking (AAR) with reference to (Bordes et al., 2013). For a query entity e with m valid answers ranked $rk_1 < \cdots < rk_m$, the adjusted ranks and their mean are

$$\hat{rk}_i = rk_i - i + 1, \qquad \mathrm{AAR}(e) = \frac{1}{m}\sum_{i=1}^{m} \hat{rk}_i. \tag{8}$$

Hits@K and MRR under AAR use the adjusted ranks.

5.2 ATTRIBUTION PERFORMANCE COMPARISON

We compare THREATALIGN with existing methods under long-tail and conventional attribution settings. In Table 2, THREATALIGN achieves 97.8% M-F1 on AADM+, close to the 98.0% of APT-ATT, the best-performing classifier. On HEAA, where report coverage is long-tailed, THREATALIGN reaches 43.5% M-F1, exceeding the strongest baseline, APT-ATT at 26.9%, by 16.6 percentage points, with precision and recall also above all baselines.

Table 2: Attribution results on HEAA and AADM+ (%; mean ± SD). Red and orange indicate the best and second-best means.

Method	HEAA			AADM+		
	M-P	M-R	M-F1	M-P	M-R	M-F1
SMOBI+SVM	14.4 ± 0.0	21.4 ± 0.0	15.6 ± 0.0	61.1 ± 0.0	57.2 ± 0.0	55.0 ± 0.0
SMOBI+RF	19.4 ± 1.2	28.2 ± 1.5	20.9 ± 1.2	64.6 ± 4.9	63.8 ± 4.5	62.0 ± 4.8
SMOBI+XGBoost	18.2 ± 0.0	28.6 ± 0.0	20.5 ± 0.0	84.7 ± 0.0	79.2 ± 0.0	79.7 ± 0.0
SMOBI+LightGBM	19.2 ± 0.0	27.4 ± 0.0	21.2 ± 0.0	88.7 ± 0.0	87.8 ± 0.0	87.4 ± 0.0
DeBERTaV3	1.2 ± 0.9	3.6 ± 2.6	1.4 ± 1.2	32.7 ± 16.9	32.1 ± 11.6	28.5 ± 13.1
SecureBERT (v1.0)	5.1 ± 6.3	9.9 ± 11.5	5.9 ± 7.5	88.1 ± 3.7	82.3 ± 5.1	82.6 ± 5.1
CyBERT	18.4 ± 2.5	30.6 ± 3.1	20.8 ± 2.7	**89.5 ± 2.2**	84.8 ± 5.5	84.9 ± 4.8
APT-MMF	0.6 ± 0.5	1.6 ± 0.3	0.6 ± 0.3	7.9 ± 3.9	10.4 ± 2.3	6.9 ± 2.5
TRAIL	2.1 ± 1.1	3.1 ± 1.0	2.1 ± 1.0	7.4 ± 4.8	12.0 ± 3.6	6.7 ± 3.8
APT-ATT	**25.1 ± 1.5**	**33.7 ± 1.5**	**26.9 ± 1.4**	**98.2 ± 0.4**	**98.1 ± 1.2**	**98.0 ± 0.8**
THREATALIGN	**40.7 ± 2.3**	**50.8 ± 1.5**	**43.5 ± 2.0**	**98.2 ± 1.3**	**98.2 ± 1.3**	**97.8 ± 1.6**

The graph-based methods APT-MMF and TRAIL obtain only 0.6% and 2.1% M-F1 on HEAA, respectively. These results indicate that methods relying on topological propagation or explicit identity signals struggle to aggregate evidence reliably when graph structures are highly heterogeneous and actor nodes lack usable information, as in HEAA. On HEAA, pretrained models show no consistent advantage over feature classifiers when learning from limited reference reports; only CyBERT matches the tree-based models. APT-ATT adds CTGAN augmentation and stacking to feature selection to address class imbalance. It achieves the strongest baseline result, but predictions remain concentrated on few actor labels, limiting tail-actor coverage.

We examine attribution for actors with few reports by grouping results by report count. Table 3 groups actors by query-side report count, with the buckets reflecting evidence scarcity on both sides.

THREATALIGN has the highest M-F1 in all four buckets, reaching 26.5% and 43.0% with one and two reports, respectively, with margins of 14.3 and 16.9 points over the strongest baselines. Among the 30 one-report actors, each classification baseline identifies only two to five actors in at least one run. THREATALIGN identifies 11 actors, including seven missed by every classification baseline. Classification baselines tend to cover a similar small subset of tail actors, while alignment broadens this coverage to actors missed by classifiers.

We evaluate attribution under different evidence conditions by grouping results by evidence availability. In Table 3, THREATALIGN achieves 44.7% and 26.1% M-F1 in the DA+AA and AA-only buckets, compared with 28.2% and 15.4% for the strongest baseline, respectively; all methods score zero in NN-only. The gains in both DA+AA and AA-only support THREATALIGN's ability to use entity evidence with differing discriminative strength for attribution.

Table 3: Attribution M-F1 (%) on HEAA by query-side report count and evidence availability. Red and orange indicate the best and second-best results.

Method	Query reports per actor				Evidence availability		
	1	2	3–5	>5	DA+AA	AA only	NN only
SMOBI+XGBoost	10.0	13.0	36.8	**77.8**	21.0	10.1	0.0
SMOBI+LightGBM	**12.2**	**26.1**	19.3	68.1	22.2	10.5	0.0
SecureBERT	1.9	2.4	13.5	29.6	6.1	3.6	0.0
CyBERT	**12.2**	9.2	48.5	75.0	21.5	12.1	0.0
APT-ATT	8.9	21.3	**60.5**	70.0	**28.2**	**15.4**	0.0
THREATALIGN	**26.5**	**43.0**	**64.2**	**88.9**	**44.7**	**26.1**	0.0

5.3 ENTITY ALIGNMENT PERFORMANCE

To test whether the attribution gains come solely from reformulating the task as entity alignment, we compare THREATALIGN with structure-driven Dual-AMN (Mao et al., 2021), text-semantics-driven TEA (Zhao et al., 2023), and LLM-enhanced EASY-EA (Cheng et al., 2025) on HEAA. We also evaluate THREATALIGN's general-purpose entity alignment performance on ICEWS-WIKI. In Figure 2, THREATALIGN achieves 50.79% actor Hits@1, exceeding the best general-purpose methods at 5.08% by 45.71 percentage points. On ICEWS-WIKI (Figure 3), its Hits@1 is 98.18%, within 1.32 percentage points of EASY-EA's 99.50% and above TEA's 93.00%, placing it close to the best method on this general-domain benchmark.

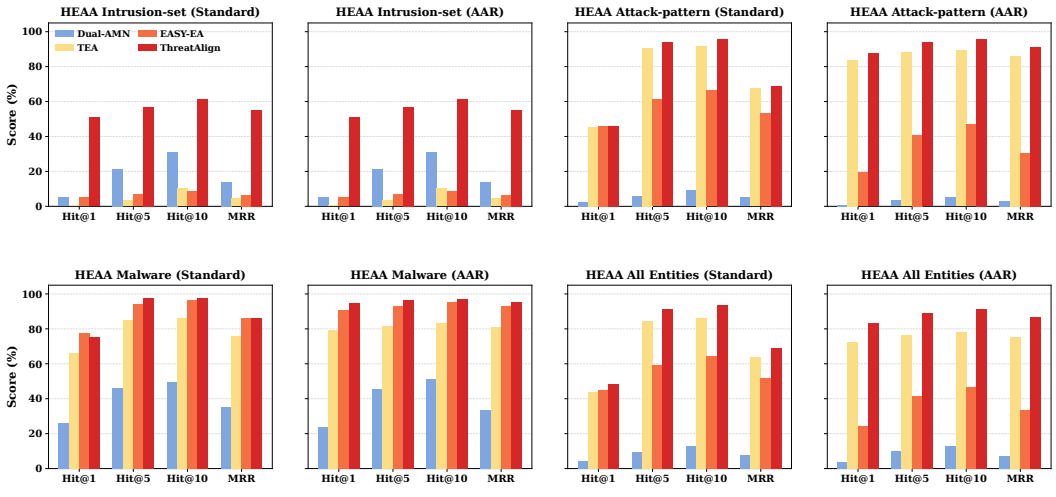

Figure 2: Entity alignment on HEAA under Standard and AAR. Actor results are identical under both protocols; attack patterns and malware have multiple valid correspondences.

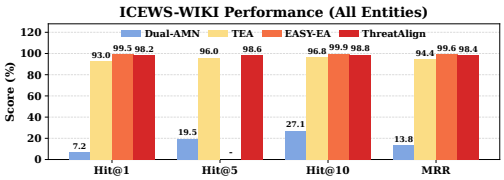

Figure 3: Entity alignment on ICEWS-WIKI.

Names provide direct cues for semantic matching of lower-level entities, where TEA performs well. Query actor nodes have no information of their own that can directly support matching, so alignment must draw on related entities and their relations; TEA reaches only 0.56% actor Hits@1. Unexpectedly, EASY-EA also achieves only 5.08% despite incorporating an LLM. It flattens one-hop neighboring triples into text and combines them with other representations, which may inadequately aggregate related evidence across levels into actor representations. In one diagnostic run, replacing its input with actor profiles generated by THREATALIGN raises actor Hits@1 to 28.81%. This improvement supports THREATALIGN's design of aggregating related evidence up to the actor level through recursive profiling. Differences in candidate generation and evidence comparison may also contribute to the remaining 21.98-point gap to THREATALIGN; the subsequent ablations examine these components.

Multiple valid correspondences include entities with similar names as well as differently named entities with the same meaning. EASY-EA's attack-pattern Hits@1 falls from 45.87% under Standard to 19.43% under AAR, suggesting reliance on surface name similarity. TEA and THREATALIGN improve under AAR on both attack patterns and malware, consistent with semantic matching that covers more differently named but equivalent entities.

5.4 ABLATION STUDY

We use single-component and cumulative ablations to assess each component's contribution to attribution and its dependence on other components. Table 4 compares five components: saliency ranking (S_1), lower-level profile reuse (H_1), retrieved context (H_2), IoC matching (E_1), and hybrid profile matching (E_2). Removing S_1 preserves the node budget; removing H_1 uses raw attributes while retaining selected entities, relations, and external context. Single-component ablations remove one component at a time; cumulative ablations follow S_1, H_1, H_2, E_2, E_1. The final configuration still generates entity profiles and directly matches their embeddings.

Table 4: Single-component and cumulative ablations on HEAA (%), with mean and standard deviation over three runs. A checkmark indicates a retained component and a dash indicates removal; components are defined in the text. Red and orange mark the best and second-best results.

Variant	ESE	HSP		ECA		M-P	M-R	M-F1
	S_1	H_1	H_2	E_1	E_2			
Full	✓	✓	✓	✓	✓	**40.7 ± 2.3**	**50.8 ± 1.5**	**43.5 ± 2.0**
Single-component removal								
w/o S_1	–	✓	✓	✓	✓	**35.2 ± 3.1**	**46.7 ± 3.4**	**38.4 ± 3.3**
w/o H_1	✓	–	✓	✓	✓	30.6 ± 1.7	39.0 ± 1.2	32.6 ± 1.6
w/o H_2	✓	✓	–	✓	✓	28.7 ± 2.8	39.6 ± 2.0	31.8 ± 2.3
w/o E_1	✓	✓	✓	–	✓	31.7 ± 2.7	42.7 ± 2.9	34.6 ± 2.9
w/o E_2	✓	✓	✓	✓	–	32.7 ± 4.6	41.6 ± 3.4	34.9 ± 4.5
Cumulative removal								
w/o S_1, H_1	–	–	✓	✓	✓	28.5 ± 1.8	35.7 ± 3.4	30.5 ± 2.0
w/o S_1, H_1, H_2	–	–	–	✓	✓	27.5 ± 2.0	34.1 ± 1.9	29.0 ± 1.9
w/o S_1, H_1, H_2, E_2	–	–	–	✓	–	25.2 ± 1.3	30.6 ± 1.0	26.4 ± 1.1
w/o S_1, H_1, H_2, E_2, E_1	–	–	–	–	–	10.5 ± 1.2	16.2 ± 2.1	11.5 ± 1.4

Removing lower-level profile reuse (H_1) or retrieved context (H_2) produces the largest M-F1 decreases in the single-component ablations, at 10.9 and 11.7 points, respectively, with both precision and recall also decreasing. Removing saliency ranking (S_1) reduces M-F1 by 5.1 points; removing IoC matching (E_1) or hybrid profile matching (E_2) reduces it by 8.9 and 8.6 points, respectively. These results support recursive profiling and external context and show that the full model benefits from both candidate-generation channels. The loss from removing S_1 at the same node budget shows that evidence selection affects attribution: shared tools and frequent attack patterns may occupy limited input space and weaken actor-specific information in profiles.

Cumulative ablations examine how component contributions change after other components are removed. After removing S_1 and H_1, removing retrieved context (H_2) lowers M-F1 from 30.5% to 29.0%, compared with an 11.7-point loss when it is removed from the full model; subsequently removing hybrid profile matching costs 2.6 points, yielding 26.4%. Further removing IoC matching lowers M-F1 by 14.9 points to 11.5%, a larger loss than the 8.9 points in the full model. These differences suggest that external knowledge and hybrid profile matching may depend on upstream evidence selection and hierarchical organization. IoC candidate generation can use entity correspondences directly, which may explain why it retains some attribution capability as semantic processing is weakened. Gains from combining semantic evidence and direct correspondences depend on component configuration and removal order.

6 Related Work

CTI-based attribution uses threat evidence through expert knowledge or representations learned from reports. ART compares ATT&CK matrices (Shin et al., 2021), while SMOBI learns contextual representations from labeled reports and a security corpus (Perry et al., 2019). Camp2Vec (Lee & Choi, 2023) and Attack2vec (Irshad & Siddiqui, 2023) learn campaign-related text representations. APT-ATT selects N-gram and TF-IDF features using chi-square statistics and combines CTGAN augmentation with a stacking ensemble to mitigate class imbalance (Cai et al., 2025). Class imbalance is directly relevant to our long-tail setting; HEAA further examines attribution when many actors have only a few reference reports.

Graph-based methods organize report evidence through relations among threat entities. APT-MMF constructs heterogeneous attributed graphs, fuses IoC attributes, text semantics, and topology, and classifies report nodes using multilevel graph attention (Xiao et al., 2024). TRAIL builds a threat intelligence knowledge graph from network IoCs and performs attribution through indirect resource-reuse paths and label propagation (King et al., 2025). TAA-EPLMR retrieves multilevel evidence paths from a CTI knowledge graph for LLM reasoning over candidate actors (Xiao et al., 2025). ThreatAlign aligns entities across query and reference graphs to compare the evidence associated with candidate actors.

Entity alignment provides a technical basis for such cross-graph comparison. Dual-AMN learns structural correspondences with dual attention and hard-sample mining (Mao et al., 2021). For report-derived graphs, differences in the entities and relations covered by individual reports can reduce the shared neighborhood structure available for matching. Semantic methods additionally use textual information: BERT-INT models semantic interactions between entities (Tang et al., 2020), SDEA uses long text (Zhong et al., 2022), and TEA converts relational and attribute triples into bidirectional textual entailment (Zhao et al., 2023). ChatEA converts graph context into an LLM-readable form and performs two-stage reasoning (Jiang et al., 2024a), while EASY-EA combines information summarization, embedding fusion, and candidate selection in a training-free framework (Cheng et al., 2025). CTI attribution must also handle incomplete and differently expressed evidence associated with actors whose query nodes lack directly matchable information of their own. ThreatAlign selects evidence according to its discriminative value and recursively aggregates lower-level behavioral context into actor profiles. Candidate comparison combines profile semantics with available direct correspondences.

7 Conclusion

We formulate APT attribution as entity-level semantic alignment and construct HEAA to evaluate attribution under long-tailed report coverage. Its actor labels and entity correspondences support analysis of attribution performance together with the underlying cross-graph evidence. ThreatAlign selects evidence and recursively builds semantic profiles to compare unknown actors with known reference actors. It achieves 43.5% M-F1 on HEAA, exceeding the strongest baseline by 16.6 points, with gains also observed for actors with few reports. Alignment comparisons and ablations show that recursive profiling and external knowledge help use related evidence for attribution when query actors lack directly matchable information of their own.

REFERENCES

Ehsan Aghaei, Xi Niu, Waseem Shadid, and Ehab Al-Shaer. Securebert: A domain-specific language model for cybersecurity. In *International Conference on Security and Privacy in Communication Systems*, pp. 39–56. Springer, 2022.

AT&T Cybersecurity. Alienvault open threat exchange (otx). `https://otx.alienvault.com/`, 2025. Accessed: 2025-08-23.

Sean Barnum. Standardizing cyber threat intelligence information with the structured threat information expression (stix). *Mitre Corporation*, 11(2012):1–22, 2012.

David Bianco. The pyramid of pain. *Enterprise Detection & Response*, 112, 2013.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Saihua Cai, Gang Wang, Jinfu Chen, Shengran Wang, and Kun Wang. Apt-att: An efficient apt attribution model based on heterogeneous threat intelligence representation and ctgan. *Computer Networks*, pp. 111511, 2025.

Tianqi Chen. Xgboost: A scalable tree boosting system. *Cornell University*, 2016.

Jingwei Cheng, Chenglong Lu, Linyan Yang, Guoqing Chen, and Fu Zhang. Easyea: Large language model is all you need in entity alignment between knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20981–20995, 2025.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 758–759, 2009.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Elasticsearch B.V. Elasticsearch: The distributed search and analytics engine. `https://github.com/elastic/elasticsearch`, 2025. Accessed: 2025-08-23.

Filigran. Opencti: Open cyber threat intelligence platform. `https://github.com/OpenCTI-Platform/opencti`, 2025. Accessed: 2025-08-23.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*, 2021.

Ehtsham Irshad and Abdul Basit Siddiqui. Cyber threat attribution using unstructured reports in cyber threat intelligence. *Egyptian Informatics Journal*, 24(1):43–59, 2023.

Eugene Jang, Jian Cui, Dayeon Yim, Youngjin Jin, Jin-Woo Chung, Seungwon Shin, and Yongjae Lee. Ignore me but don't replace me: Utilizing non-linguistic elements for pretraining on the cybersecurity domain. *arXiv preprint arXiv:2403.10576*, 2024.

Xuhui Jiang, Yinghan Shen, Zhichao Shi, Chengjin Xu, Wei Li, Zixuan Li, Jian Guo, Huawei Shen, and Yuanzhuo Wang. Unlocking the power of large language models for entity alignment. *arXiv preprint arXiv:2402.15048*, 2024a.

Xuhui Jiang, Chengjin Xu, Yinghan Shen, Yuanzhuo Wang, Fenglong Su, Zhichao Shi, Fei Sun, Zixuan Li, Jian Guo, and Huawei Shen. Toward practical entity alignment method design: Insights from new highly heterogeneous knowledge graph datasets. In *Proceedings of the ACM Web Conference 2024*, pp. 2325–2336, 2024b.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

Isaiah J King, Ramiro Ramirez, Benjamin Bowman, and H Howie Huang. Trail: A knowledge graph-based approach for attributing advanced persistent threats. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, pp. 1207–1220. IEEE, 2025.

Insup Lee and Changhee Choi. Camp2vec: Embedding cyber campaign with att&ck framework for attack group analysis. *ICT Express*, 9(6):1065–1070, 2023.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

Xin Mao, Wenting Wang, Yuanbin Wu, and Man Lan. Boosting the speed of entity alignment 10×: Dual attention matching network with normalized hard sample mining. In *Proceedings of the web conference 2021*, pp. 821–832, 2021.

Lior Perry, Bracha Shapira, and Rami Puzis. No-doubt: Attack attribution based on threat intelligence reports. In *2019 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 80–85. IEEE, 2019.

Daniel Plohmann, Martin Clauss, Steffen Enders, and Elmar Padilla. Malpedia: a collaborative effort to inventorize the malware landscape. *The Journal on Cybercrime and Digital Investigations*, 3 (1):1–19, 2017.

Yitong Ren, Yanjun Xiao, Yinghai Zhou, Zhiyong Zhang, and Zhihong Tian. Cskg4apt: A cybersecurity knowledge graph for advanced persistent threat organization attribution. *IEEE Transactions on Knowledge and Data Engineering*, 35(6):5695–5709, 2022.

Youngsup Shin, Kyoungmin Kim, Jemin Justin Lee, and Kyungho Lee. Art: automated reclassification for threat actors based on att&ck matrix similarity. In *2021 world automation congress (WAC)*, pp. 15–20. IEEE, 2021.

Xiaobin Tang, Jing Zhang, Bo Chen, Yang Yang, Hong Chen, and Cuiping Li. Bert-int: A bert-based interaction model for knowledge graph alignment. *interactions*, 100:e1, 2020.

Max van der Horst, Ricky Kho, Olga Gadyatskaya, Michel Mollema, Michel Van Eeten, and Yury Zhauniarovich. High stakes, low certainty: Evaluating the efficacy of high-level indicators of compromise in ransomware attribution. In *Proceedings of the 34th USENIX Security Symposium (USENIX Sec)*, 2025.

Cynthia Wagner, Alexandre Dulaunoy, Gérard Wagener, and Andras Iklody. Misp: The design and implementation of a collaborative threat intelligence sharing platform. In *Proceedings of the 2016 ACM on workshop on information sharing and collaborative security*, pp. 49–56, 2016.

Ziyu Wang, Yinghai Zhou, Hao Liu, Jing Qiu, Binxing Fang, and Zhihong Tian. Threatinsight: Innovating early threat detection through threat-intelligence-driven analysis and attribution. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Nan Xiao, Bo Lang, Ting Wang, and Yikai Chen. Apt-mmf: An advanced persistent threat actor attribution method based on multimodal and multilevel feature fusion. *Computers & security*, 144:103960, 2024.

Nan Xiao, Bo Lang, Yikai Chen, Shuxin Zhao, and Yuhao Yan. Taa-eplmr: Threat actor attribution via evidence path-enhanced large language model reasoning. In *2025 IEEE International Conference on Big Data (BigData)*, pp. 2064–2073. IEEE, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yu Zhao, Yike Wu, Xiangrui Cai, Ying Zhang, Haiwei Zhang, and Xiaojie Yuan. From alignment to entailment: A unified textual entailment framework for entity alignment. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8795–8806, 2023.

Ziyue Zhong, Meihui Zhang, Ju Fan, and Chenxiao Dou. Semantics driven embedding learning for effective entity alignment. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 2127–2140. IEEE, 2022.

AI USE STATEMENT

We used generative artificial intelligence tools to assist with manuscript preparation, including translation, language polishing, and the generation of some code. All other work was performed by the authors. All AI-assisted text and code, whether generated or modified, were manually reviewed and validated by the authors, who take full responsibility for the final manuscript, implementation, and conclusions. The use of large language models in the THREATALIGN method and data construction is described in Section 4 and Appendix A, respectively.

ETHICS STATEMENT

Our study does not involve human subjects and therefore does not require IRB review. HEAA and AADM+ are constructed from publicly available CTI sources, including vendor technical reports and OSCTI platforms such as OpenCTI, MISP, and AlienVault OTX. We collect and process data in accordance with the terms of service of these sources. Although raw CTI reports may occasionally mention victim organizations, our processing pipeline extracts only IoCs and TTPs. We do not collect, store, or attempt to infer personally identifiable information, and we release only the derived graphs and annotations rather than the original report text. Regarding reliability, errors in our setting primarily arise from incorrect alignment decisions (e.g., linking evidence to the wrong actor), rather than the generation of new CTI facts. To mitigate such errors, HoloTrace operates on curated CTI evidence and incorporates the ECHA module together with a rejection threshold τ. Nevertheless, false positives remain possible in any probabilistic decision process, particularly under sparse or ambiguous evidence. We therefore position HoloTrace as a decision support tool whose outputs provide investigative leads that require analyst validation. We explicitly discourage using automated attributions as the sole basis for retaliation ("hack-back") or high-stakes actions such as sanctions without independent verification.

A DATASET CONSTRUCTION AND STATISTICS

A.1 HEAA CONSTRUCTION

The raw intelligence in HEAA comes from an OpenCTI snapshot dated August 23, 2025. OpenCTI (Filigran, 2025) connects to MISP (Wagner et al., 2016) and AlienVault OTX (AT&T Cybersecurity, 2025), providing reports and graph-structured intelligence from multiple open-source feeds. The native graphs undergo data cleaning, low-quality entity filtering, and report-level splitting before cross-graph alignment pairs are annotated.

The Malpedia alias table is used only to normalize the ground-truth actor labels in the query and reference graphs; canonical names are not provided as model input. The reference graph retains the original actor names, aliases, and textual identity mentions from the reports rather than replacing them with canonical names. Actor identity is anonymized in the query graph: the APT group name is replaced with "unknown apt group," while aliases and textual mentions that identify the actor are masked. After removing isolated Intrusion Sets supported by fewer than two reports, HEAA contains 84 APT actors and 501 CTI reports, with one report-level graph constructed for each report.

Each report graph contains only the objects and relations associated with that report. The default HEAA split randomly partitions reports within each actor at a 1:1 ratio to construct the reference and query graphs. Reports from the same actor share one actor node within each side, and query reports do not enter the reference graph. HEAA(time) uses the same ratio, assigning earlier reports to the reference side and later reports to the query side. Both splits use the same identity-masking and annotation rules.

A.2 ENTITY SCHEMA AND ALIGNMENT LABELS

The native graph data are mapped to a subset of the STIX 2.1 schema (Barnum, 2012). Threat Actors are the attribution targets, TTP entities represent behavioral characteristics, Infrastructure and Indicator entities record observable evidence, and Financial/ID and Meta entities provide supplementary context. Table 5 lists the retained entity types and attributes.

In the four-level evidence hierarchy $\mathcal{P}$, attack-pattern, malware, and intrusion-set entities map to L_2, L_3, and L_4, respectively. All other retained entity types are assigned to the IoC layer L_1.

Table 5: Entity types and attributes retained in HEAA.

Category	Entity Type	Attributes	Notes
Threat Actor	intrusion-set	name, aka_name, created_time, modified_time	Target node for attribution
TTPs	attack-pattern	name, aliases, kill_chain, platforms, description*	*description is used for label construction and masked during inference
TTPs	malware	name, is_family	Malware families or variants
TTPs	tool	name, description	Attack tools
Infrastructure	domain, url, ipv4-addr	name	Network observables
Infrastructure	file, win-registry-key	name	Host-side fingerprints
Infrastructure	indicator	name, valid_from, valid_until	Indicator observables
Financial/ID	crypto-wallet, email, location	name, latitude, longitude	Identity and geolocation information
Meta	text, vulnerability	name	Contextual text and vulnerability information

Cross-graph entities are matched by exact comparison of normalized values by default. For intrusion-set, malware, and attack-pattern entities, the following annotation procedures additionally establish correspondences beyond exact value matching. Actor correspondences are inherited from OpenCTI-validated attribution relations.

For malware annotation, an LLM generates a short description for each node to supply semantics beyond its name, then compares descriptions across the two graphs to propose candidate pairs. Two cybersecurity researchers independently validate the candidates, with disagreements resolved by a third researcher. All retained pairs are manually confirmed, although the procedure does not guarantee exhaustive coverage of valid correspondences.

Attack patterns are mapped to standard ATT&CK technique or sub-technique names through semantic similarity. When the graphs record different granularities, such as T1566.003 and T1566, a missing parent-technique node is added locally and connected to the sub-technique to retain its context. Alignment labels link nodes mapped to the same standard name or normalized to the same parent technique. The attack-pattern description field in Table 5 is used for label construction and masked during inference.

A.3 Graph Extraction for AADM+

AADM+ (Cai et al., 2025) is released as unstructured threat reports. To preserve the query–reference graph setting, each report is converted into a report-level threat graph. Model inputs consist of entities and relations extracted from report text, while APT group labels are used for data splitting and attribution evaluation.

Graph extraction for AADM+ uses an in-context-learning-based multi-agent process. A hierarchical example repository contains slice-level examples that annotate entities and relations in local text fragments and merge-level examples of merging entities across fragments while preserving relations.

A slicing agent divides each long report into text fragments and retrieves similar examples as extraction demonstrations. Extraction agents process the fragments in parallel to identify APT actors, Malware, tools, attack patterns, infrastructure, IoC nodes, and their relations. A merging agent aggregates the fragment-level outputs, performs entity deduplication, alias merging, relation deduplication, and graph-connectivity checks, and produces one heterogeneous subgraph per report.

Extracted AADM+ entity types are mapped to the ontology subset in Table 5, while relation types are retained as semantic edges determined by report context. The conversion preserves the original organization names, entity mentions, textual evidence, and label information. Data splitting follows the original 8:2 train–test split of AADM+.

A.4 Graph and Evidence Statistics

Table 6 gives complete graph statistics for both HEAA splits and ICEWS-WIKI.

Table 6: Topological statistics of ICEWS-WIKI and the two HEAA splits. **#Anchors** is the number of ground-truth entity correspondences. **Overlap** is the fraction of entities participating in a correspondence in each graph, and **Struct. Sim.** measures aligned-edge preservation over all facts.

Dataset	Graph	# Entities	# Relations	# Facts	Avg. Degree	# Anchors	Overlap	Struct. Sim.
ICEWS-WIKI	$\mathcal{G}_Q$	11,047	272	3,527,881	319.35	5,058	45.79%	15.40%
	$\mathcal{G}_R$	15,896	226	198,257	12.47		31.82%	
HEAA	$\mathcal{G}_Q$	11,158	5	29,730	5.33	2,711	14.23%	3.02%
	$\mathcal{G}_R$	13,325	5	37,028	5.56		11.98%	
HEAA(time)	$\mathcal{G}_Q$	12,358	5	31,290	5.06	2,032	9.52%	0.75%
	$\mathcal{G}_R$	11,713	5	34,712	5.93		10.16%	

Table 7 reports the mean and standard deviation of evidence counts per actor.

Table 7: Distribution of evidence categories under the HEAA splits. Mean and SD are computed across actors; Pct. denotes the proportion of entities in each category.

Split	Category	Mean	SD	Pct.
HEAA	DA	4.88	9.90	4.5%
	AA	17.88	15.68	23.8%
	NN	89.27	114.15	71.7%
HEAA(time)	DA	2.87	6.88	3.6%
	AA	16.67	14.75	23.1%
	NN	79.44	98.24	73.3%

Table 8 compares evidence proportions and AMC by entity type across splits. Discriminative AMC averages over samples that permit unique actor identification. Bounded AMC uses a common cap of 12, chosen from the 95th percentile of the uncapped AMC distribution to reduce the influence of extreme values. Identification costs above the cap and unresolved samples both contribute 12 to the average.

Table 8: Evidence utility by entity type under the HEAA splits. Bd. AMC and Disc. AMC denote Bounded AMC and Discriminative AMC, respectively.

Type	HEAA					HEAA(time)				
	DA	AA	NN	Bd. AMC	Disc. AMC	DA	AA	NN	Bd. AMC	Disc. AMC
Malware	7.7%	1.6%	90.7%	8.49	1.07	6.1%	2.2%	91.7%	9.33	1.33
Attack pattern	0.4%	20.0%	79.6%	11.32	9.62	0.3%	17.2%	82.6%	11.19	8.76
IoC	2.3%	0.1%	97.6%	7.67	2.16	1.3%	0.0%	98.7%	8.43	2.00

B Method Implementation

B.1 External Knowledge Base

The domain knowledge base $\mathcal{K}$ is constructed from MITRE ATT&CK and Malpedia (Plohmann et al., 2017) and supplies external semantic context for the HEAA graph instances. We use the en-

tity relations maintained by these sources to build dependency and reverse-dependency lists among attack patterns, Malware, and APT actors, forming a hierarchical graph structure. Long text associated with each entity is recursively split into shorter passages and encoded with Qwen3-Embedding-8B (Yang et al., 2025). Each Elasticsearch (Elasticsearch B.V., 2025) record stores an entity name, structural relations, text passages, and their vectors. For an ATT&CK technique, for example, the record contains the technique name, related Malware, related APT actors, text passages, and the corresponding vectors.

The knowledge base contains 3,138 Malware entities, 835 ATT&CK entities, and 811 APT actor entities, producing 951,029 indexed records. Let S_K denote the reports used for the textual component of the knowledge base, and let S_Q and S_R denote the reports used to construct the HEAA query and reference graphs, respectively. Report-level separation is enforced as

$$S_K \cap (S_Q \cup S_R) = \varnothing.$$

Report-level separation does not exclude shared threat facts across sources.

B.2 Knowledge Retrieval and Profiling Inputs

For each entity v at levels L_2 through L_4, structural retrieval queries the relation records in $\mathcal{G}_{\mathcal{K}}$ separately with the current entity's name and the names of its selected lower-level neighbors. RRF fuses the two rankings and retains at most $K_{\text{struct}} = 5$ distinct candidates. Text retrieval reuses these queries for BM25 and dense retrieval over the report-text index, retaining at most $K_{\text{expand}} = 3$ candidates after fusion. The RRF rank constant is $c_{\text{rrf}} = 60$ (Cormack et al., 2009).

Profiling inputs contain the current entity's attributes, saliency-selected lower-level evidence with its relation types, and retrieved structural and textual context. Evidence at L_1 retains raw names or identifiers; other lower-level entities supply their generated profiles. Candidate retrieval is restricted to reference entities of the same type. RRF combines lexical and vector profile retrieval to retain the top $K_{\text{prof}} = 10$ candidates, while the IoC channel has no candidate-count cap.

B.3 Evidence Arbitration and Acceptance

Evidence Arbitration takes the query entity v_q and all reference entities in $\mathcal{C}_{\text{cand}}(v_q)$ as input and presents them to a frozen LLM in a single prompt. The query-side input contains entity attributes and the generated textual profile. Each candidate input contains reference attributes and a textual profile, plus available retrieval evidence such as its RRF rank and shared IoCs.

The threshold setting τ specifies the acceptance conditions. The implementation uses a compound rule: a candidate is accepted if its Tier 1 (direct evidence) score equals 1.0 or its Tier 2 (profile evidence) score is greater than or equal to 0.8. Accepted candidates form $f_{\text{align}}(v_q)$ and are ranked by the alignment score $s(v_q, v_r)$. Actor attribution selects the highest-scoring accepted candidate and rejects attribution when the set is empty. Candidate scores and complete rankings are retained as an audit record.

The arbitration template returns JSON fields `tier1_factual`, `tier2_semantic`, and `tier3_ranking` for direct evidence, profile evidence, and retrieval-rank scores. The first two fields supply the compound acceptance rule above; retrieval rank serves only as a tie-breaker. Appendix B.5 gives the full prompts.

B.4 Runtime Configuration

Experiments run on Debian 11 with an Intel Xeon Platinum 8336C CPU, 256 GB RAM, and three NVIDIA RTX 4090 GPUs. ThreatAlign uses DeepSeek-V3.2 in non-thinking mode (Liu et al., 2025) for LLM inference and Qwen3-Embedding-8B for text representations, with Elasticsearch providing vector search over entity profiles and IoC metadata. All models use the same runtime environment and fixed global random seeds. Each experiment is run independently three times with the mean and standard deviation reported. Table 9 summarizes the default implementation hyperparameters for ThreatAlign.

The ESE implementation directly sums the local and knowledge-prior terms, preserving the ranking of the equal-weight combination at $\alpha = 0.5$. The corroboration term uses no smoothing constant and is set to 1 when its denominator is zero. Table 9 lists the remaining default parameters.

Table 9: Default implementation hyperparameters of ThreatAlign.

Parameter	Value	Role
α	0.5	Weight for topology and knowledge-prior saliency.
γ	1	Corroboration scaling coefficient.
p	5	Per-lower-level predecessor cap.
K_{struct}	5	Structured knowledge candidates.
K_{expand}	3	Expanded text candidates.
K_{prof}	10	Profile candidates passed to evidence arbitration.
c_{rrf}	60	Rank constant used by Reciprocal Rank Fusion.
Acceptance setting τ	1.0 / 0.8	Accept a candidate if its direct-evidence score equals 1.0 or its profile-evidence score is at least 0.8.
Temperature	0.3	LLM decoding temperature.

B.5 Prompt Templates

The following templates are used for profile generation and candidate arbitration. Double braces mark fields populated at runtime.

B.5.1 Attack-pattern profiling

Attack-pattern profiling

Input:

```
Task: Strictly follow the system instructions to generate the entity
profile. Self Info: {{entity_message}}
[Condition: If neighbour info exists] Neighbour Info: {{neighbours}}
[Condition: If retrieved info exists] Retrieved Info:
{{retrive_info}}
```

Context:

```
Role: You are a top-tier CTI Expert specializing in TTP analysis and
Knowledge Graph construction.
Task: Your task is to generate a highly condensed, logically
ordered, and plain-text Entity Profile for a specific
"attack-pattern" entity based on the provided structured input (Self
Info, Neighbor Info, Retrieved Info).
Input Data Explanation:
Self Info: Basic attributes of the entity.
Neighbor Info: Directly connected nodes in the knowledge graph.
Retrieved Info: External context retrieved via search. It contains
"Similar Entities" and "Related Content Entities". Challenge: These
entities describe themselves but mention the target entity. You must
extract only the context relevant to the target and ignore the rest.
```

Workflow:

```
Analysis & Alignment: Analyze the input data. Map this entity to
the standard MITRE ATT&CK Matrix. Identify the standard Name, ID
(e.g., Txxxx), Tactic, and Platform. CRITICAL: Verify Retrieved
Info. Discard information that is weak, irrelevant, or conflicts
with the official MITRE definition.
Information Extraction: Extract specific technical mechanics (How it
works). Discard generic descriptions. Focus on unique indicators,
commands, or behaviors.
Sequencing: Organize valid information strictly in this order:
[Name] -> [ID] -> [Tactic] -> [Platform] -> [Technical Mechanics] ->
[Supplementary Info].
```

Drafting: Synthesize the sequence into a single, fluent English paragraph. Remove all line breaks, markdown styling, bullet points, or bold text.

Constraints & Rules:
Knowledge Boundary: If standard MITRE details are missing, strictly summarize based ONLY on the provided text. Do NOT hallucinate IDs or names.
Format: The output must be one single continuous string of plain English text, less than 150 words.
Wrapper: Enclose the final result in [profile] tags.
Content: No fluff. Every sentence must add technical value.

B.5.2 MALWARE PROFILING

Malware profiling

Input: Task: Strictly follow the system instructions to generate the entity profile. Self Info: {{entity_message}}
[Condition: If neighbour info exists] Neighbour Info: {{neighbours}}
[Condition: If retrieved info exists] Retrieved Info: {{retrive_info}}

Context:
Role: You are a top-tier CTI Expert specializing in malware analysis and profiling.
Task: Your task is to generate a highly condensed, logically ordered, and strictly formatted entity profile for a "malware" entity based on the provided structured input.
Input Data Explanation:
Self Info: Basic attributes of the entity.
Neighbor Info: Directly connected nodes in the knowledge graph.
Retrieved Info: External context retrieved via search. It contains "Similar Entities" (for cross-referencing aliases/clusters) and "Related Content Entities". Challenge: The related content entities describe themselves but mention the target entity. You must extract only the context relevant to the target and ignore the rest.

Workflow:
Analysis & Alignment: Analyze the input data. Identify the standard malware family name, aliases, attributed groups (APT/Criminal), TTPs, and target platforms.
Filter & Verify: Critically evaluate Retrieved Info. Discard generic descriptions (e.g., "viruses are bad") or data that conflicts with 'Self Info'. Focus on high-distinctiveness indicators (C2 protocols, specific encryption methods).
Sequencing: Organize valid information strictly in this order: [Name] -> [Aliases] -> [Attributed Groups] -> [Key TTPs/Technical Behaviors] -> [Target Platforms] -> [Infrastructure/Supplementary Info].
Drafting: Synthesize the sequence into a single, fluent English paragraph. Remove all line breaks, markdown styling, bullet points, or bold text.

Constraints & Rules:
Knowledge Boundary: If specific details are missing in the input, do NOT invent them. Stick strictly to the provided text.
Format: The output must be one single continuous string of plain English text, less than 150 words.
Wrapper: Enclose the final result in [profile] tags.
Content: High Signal-to-Noise. Avoid generic sentences. Every sentence must add technical value.

B.5.3 ACTOR PROFILING

Actor profiling

The prompt for APT entities.
Input:
Task: Strictly follow the system instructions to generate the APT entity profile. Self Info: {{entity_message}}
[Condition: If neighbour info exists] Neighbour Info: {{neighbours}}
[Condition: If retrieved info exists] Retrieved Info: {{retrive_info}}

Context:
Role: You are a top-tier CTI Expert specializing in Threat Actor Attribution and Strategic Analysis.
Task: Your task is to generate a highly condensed, logically ordered, and plain-text Entity Profile for an "Intrusion Set" (APT Group) based on the provided structured input.
Input Data Explanation:
Self Info: Basic attributes of the entity.
Neighbor Info: Directly connected nodes in the knowledge graph.
Retrieved Info: External context retrieved via search. It contains "Similar Entities" (for cross-referencing aliases/clusters) and "Related Content Entities". Challenge: These entities describe themselves but mention the target entity. You must extract only the context relevant to the target and ignore the rest.

Workflow:
Analyze & Correlate: Analyze the input data. If the APT organization name is unknown, you must deduce the most likely standardized APT name based on the available information and your internal knowledge base.
Structure & Sequence: Organize the key intelligence strictly in this order: [Most Likely Standardized Name] -> [Common Aliases] -> [Key Malware/Tools] -> [Key TTPs] -> [Origin/Attribution] -> [Target Sectors/Regions] -> [Motivation/Supplementary].
Drafting: Synthesize the data into a single, fluent English paragraph. Resolve conflicting aliases into a cohesive list (e.g., "also known as X, Y, and Z").

Constraints & Rules:
No Hallucinations: If specific details are missing in the input and unknown to your internal knowledge base, do NOT invent them. Stick strictly to the provided text.
Attribution Care: If the input uses uncertainty (e.g., "suspected link," "overlap with"), reflect this in the profile using phrases like "believed to be" or "shows overlaps with." Do NOT state unverified attribution as absolute fact.
Distinction Focus: Prioritize information that distinguishes this group (e.g., unique custom malware, specific geopolitical targets). Filter out generic descriptions.
Format: The output must be one single continuous string of plain English text, less than 200 words.
Wrapper: Enclose the final result in [profile] tags.

B.5.4 CANDIDATE ARBITRATION

Candidate arbitration

Input:
Task: Re-rank the candidate entities for the following target.
Self Info: {{self_info}}
PROFILE_MATCH (Weak Signal / Search Rank): {{profile_candidates}}

IOC_MATCH (Strong Signal / Hard Links): {{ioc_candidates}}

Context:
Role: You are a CTI Entity Alignment Expert.
Task: Compare a "Target Entity" against a list of "Candidate Entities" to calculate alignment scores and determine if they refer to the exact same real-world identity.
Input Data Explanation:
Self Info: Attributes of the target entity.
IOC_MATCH: Candidates linked via factual hard indicators (Hashes, Domains).
PROFILE_MATCH: Candidates retrieved via search engine ranking (Order implies confidence but not fact).

Workflow:
Step 1: Analysis. Extract key attributes (Aliases, Malware Family, TTPs) from the Target.
Step 2: Tier 1 Check (Factual Hard Links). Scan ALL candidates. If a candidate is in `IOC_MATCH` OR has an Exact Name/Alias match, assign Score 1.0. These are Top Priority and must override original sorting.
Step 3: Tier 2 & 3 Evaluation (Semantic & Ranking). - Tier 2 (Content): Read the profile text. Score 0.8-1.0 for strong TTP/Malware matches; 0.0-0.3 for no overlap. Crucial: Text evidence beats Input Order. If a lower-ranked candidate has better text evidence, move it up. - Tier 3 (Search Confidence): Calculate based on original input index: $1.0 - (Original_Index \times 0.1)$. Use this only as a tie-breaker.
Step 4: Final Re-ranking. Output the results in the specified JSON format.

Constraints & Rules:
Alignment Logic (The Decision): - YES: If (Tier 1 == 1.0) OR (Tier 2 >= 0.8). - NO: If (Tier 1 == 0.0) AND (Tier 2 < 0.8).
Format: Return the result in strictly valid JSON format.
JSON Structure:

```
{
  "top_10_results": [
    {
      "entity_name": "Candidate Name",
      "scores": {
        "tier1_factual": float,
        "tier2_semantic": float,
        "tier3_ranking": float
      }
    }
  ]
}
```

C Evaluation Protocol and Baseline Configuration

C.1 Baseline Training and Prediction Aggregation

The conventional classifiers use SMOBI (Perry et al., 2019) features with SVM (Cortes & Vapnik, 1995), RF (Breiman, 2001), XGBoost (Chen, 2016), or LightGBM (Ke et al., 2017). Pretrained language model baselines are DeBERTaV3 (He et al., 2021), SecureBERT (Aghaei et al., 2022), and CyBERT (Jang et al., 2024). Domain-specific attribution methods are TRAIL (King et al., 2025), APT-MMF (Xiao et al., 2024), and APT-ATT (Cai et al., 2025).

RF, XGBoost, and LightGBM use 100 estimators. All PLM baselines use a learning rate of 2×10^{-5}, a batch size of 16, and a maximum sequence length of 512. Training runs for up to 50 epochs with early stopping at patience 3, and the final-layer [CLS] representation is passed to a three-layer MLP classifier. Attribution baselines use the default settings provided by their original implementations.

In actor-level attribution experiments, classification baselines first predict each query report independently and then aggregate the report-level outputs by actor. If any report is assigned to the ground-truth actor, the aggregate prediction is set to that actor; if all reports are predicted incorrectly, the most frequent predicted label is selected, with ties resolved first by cumulative confidence score and then by lexicographic label order. Both overall and bucketed attribution metrics are computed from one aggregate prediction per actor.

This any-hit aggregation uses ground-truth labels and gives classification baselines a favorable actor-level evaluation. Report-count buckets contain actors with 1, 2, 3–5, or more than 5 query-side reports. Evidence availability varies on both sides across these buckets, so the comparison does not isolate training-sample efficiency. Evidence buckets group query samples into DA+AA, AA-only, and NN-only conditions. Overall and bucketed metrics use the same actor-level predictions.

C.2 Alignment Evaluation

Alignment experiments compare Dual-AMN, TEA, and EASY-EA using Hits@K ($K \in \{1, 5, 10\}$) and MRR (Mao et al., 2021; Zhao et al., 2023; Cheng et al., 2025). AAR is defined in Equation 8 of the main paper. If two valid answers occupy ranks 1 and 3, their adjusted ranks are 1 and 2, averaging 1.5. Other valid answers do not occupy rank positions for the current answer, whereas incorrect candidates still count. Under AAR, we average the adjusted ranks of each query's valid answers and then take the reciprocal of this mean. MRR averages these reciprocal values across queries.

For general-domain entity alignment on ICEWS-WIKI, ThreatAlign adapts its three modules. ESE uses entity in-degree and out-degree for scoring. HSP generates each textual profile T_v non-recursively from the entity's neighborhood, and ECA uses the LLM to score candidate entity similarity directly.

The EASY-EA profile-substitution diagnostic in the main paper replaces its input representation with actor profiles generated by ThreatAlign and evaluates actor Standard Hits@1. This is a single diagnostic run examining the effect of the input representation. It does not separately control the remaining components, so the residual gap cannot be assigned to an individual matching module.

C.3 Ablation Operations

Ablation results appear in Table 4 of the main paper. Removing S_1 disables saliency ranking at the same node budget. Removing H_1 replaces the generated profiles of selected lower-level entities with their raw attributes, retaining the entities, relations, and external context; the LLM still generates the current entity's profile. Removing H_2 removes HSP's retrieved external context; ESE's knowledge prior is outside this intervention.

E_1 and E_2 denote IoC Matching and Hybrid Profile Matching, respectively. Single-component ablations remove one component at a time; cumulative ablations follow S_1, H_1, H_2, E_2, E_1. The final configuration still generates entity profiles and directly matches their embeddings against reference entities.

D Supplementary Experiments

D.1 Single-Report Attribution

We evaluate single-report attribution on the default HEAA split while retaining all original reference-side reports. Every method predicts each query report independently, and ThreatAlign constructs each query graph from the entities and relations in that report alone. Each report yields one actor prediction, and metrics are computed directly over these report-level predictions. In addition to M-P, M-R, and M-F1, we report weighted F1 (W-F1), which weights each actor's class F1 by its number of query reports.

In Table 10, ThreatAlign obtains 34.2% single-report M-F1, compared with 43.5% in the actor-level evaluation in Table 2 of the main paper. Baseline rankings also change: APT-ATT's M-F1 falls from 26.9% at the actor level to 12.2% at the report level, while CyBERT falls from 20.8% to 15.3%

Table 10: Single-report attribution on HEAA (%), with mean and standard deviation. Red and orange mark the best and second-best means.

Method	M-P	M-R	M-F1	W-F1
SMOBI+SVM	10.3 ± 0.0	8.7 ± 0.0	8.4 ± 0.0	13.0 ± 0.0
SMOBI+RF	14.9 ± 2.3	12.8 ± 1.1	12.4 ± 1.4	18.4 ± 1.9
SMOBI+XGBoost	13.1 ± 0.0	14.3 ± 0.0	12.8 ± 0.0	17.0 ± 0.0
SMOBI+LightGBM	13.5 ± 0.0	14.9 ± 0.0	12.9 ± 0.0	17.7 ± 0.0
DeBERTaV3	1.5 ± 0.9	2.8 ± 1.2	1.6 ± 0.9	5.0 ± 2.0
SecureBERT	5.2 ± 4.2	7.8 ± 5.5	5.8 ± 4.6	10.3 ± 7.1
CyBERT	**15.9 ± 2.8**	**17.4 ± 2.1**	**15.3 ± 2.3**	**22.7 ± 2.7**
APT-MMF	0.6 ± 0.5	1.6 ± 0.3	0.6 ± 0.3	2.2 ± 0.7
TRAIL	2.1 ± 1.1	3.1 ± 1.0	2.1 ± 1.0	6.0 ± 1.7
APT-ATT	13.0 ± 0.6	13.2 ± 0.5	12.2 ± 0.5	9.0 ± 0.2
THREATALIGN	**36.8 ± 2.0**	**36.7 ± 1.2**	**34.2 ± 1.3**	**40.6 ± 1.2**

and becomes the strongest single-report baseline. THREATALIGN achieves the highest M-F1 in both settings, exceeding the strongest single-report baseline by 18.9 percentage points. These results show that alignment with historical actor evidence in the reference graph allows THREATALIGN to retain its advantage both when actor reports are aggregated and when individual reports are attributed independently.

D.2 CHRONOLOGICAL SPLIT

We repeat the attribution experiment on HEAA(time), comparing THREATALIGN with five representative classification baselines from the main experiment while keeping all other settings identical to HEAA. HEAA(time) places earlier reports on the reference side and later reports on the query side, with an average source–target report-pair gap of 408.8 days per actor.

Table 11: Attribution performance on HEAA(time) (%), with mean and standard deviation over three runs. Parentheses give relative declines from HEAA.

Method	M-P	M-R	M-F1
SMOBI+XGBoost	15.1 ± 0.0 (17.1%)	23.8 ± 0.0 (16.7%)	17.0 ± 0.0 (16.9%)
SMOBI+LightGBM	17.4 ± 0.0 (9.7%)	27.4 ± 0.0 (0.0%)	19.8 ± 0.0 (6.7%)
SecureBERT	4.3 ± 3.2 (14.6%)	9.5 ± 5.9 (4.0%)	5.0 ± 3.6 (15.3%)
CyBERT	17.2 ± 4.1 (6.4%)	26.6 ± 5.7 (13.0%)	19.1 ± 4.5 (8.2%)
APT-ATT	**23.1 ± 1.0 (7.9%)**	**29.8 ± 1.0 (11.8%)**	**24.5 ± 0.9 (9.1%)**
THREATALIGN	**38.7 ± 1.6 (5.0%)**	**47.6 ± 1.9 (6.2%)**	**41.3 ± 1.6 (4.9%)**

Structural similarity is 0.75% under chronological splitting, compared with 3.02% under the default split (Table 6). Mean DAs per actor decrease from 4.88 to 2.87 and mean AAs from 17.88 to 16.67; Bounded AMC increases for malware and IoCs but remains similar for attack patterns (Tables 7 and 8). These statistics describe changes in the available evidence across the two report partitions.

As shown in Table 11, chronological splitting does not change the overall performance ordering: M-F1 decreases by 4.9%–16.9% relative to HEAA. THREATALIGN achieves the highest M-F1 of 41.3% and also has the smallest relative decrease, at 4.9%. We additionally report results by anchor availability to describe the evidence conditions in the two splits.

Table 12: Attribution M-F1 (%) by anchor availability on HEAA and HEAA(time).

Method	Split	DA+AA M-F1	AA only M-F1	NN only M-F1
SMOBI+XGBoost	HEAA	21.0	10.1	0.0
SMOBI+LightGBM	HEAA	22.2	10.5	0.0
SecureBERT	HEAA	6.1	3.6	0.0
CyBERT	HEAA	21.5	12.1	0.0
APT-ATT	HEAA	**28.2**	**15.4**	0.0
THREATALIGN	HEAA	**44.7**	**26.1**	0.0
SMOBI+XGBoost	HEAA(time)	18.3	8.0	0.0
SMOBI+LightGBM	HEAA(time)	21.3	11.2	0.0
SecureBERT	HEAA(time)	5.3	3.5	0.0
CyBERT	HEAA(time)	20.5	11.3	0.0
APT-ATT	HEAA(time)	**25.8**	**13.2**	0.0
THREATALIGN	HEAA(time)	**43.5**	**27.0**	0.0

Table 12 reports M-F1 by evidence availability in each split. THREATALIGN decreases from 44.7% to 43.5% in the DA+AA bucket and increases from 26.1% to 27.0% in the AA-only bucket; all methods score 0 in the NN-only bucket. Bucket membership and evidence composition change with the split, so this comparison cannot isolate the effect of time or establish robustness to concept drift.

D.3 DIRECT LLM PREDICTION

We provide the same frozen model with the same filtered query graph and candidate actor set and ask it to predict the actor label directly. Direct prediction bypasses hierarchical profiling and cross-graph alignment, obtaining 19.28 $\pm$ 1.63% M-P, 28.17 $\pm$ 1.12% M-R, and 21.25 $\pm$ 1.50% M-F1, with M-F1 22.3 points below THREATALIGN. This comparison supports explicit evidence organization and comparison, but it changes profiling and matching together and cannot isolate individual component effects or the contribution of pretraining knowledge.

D.4 LLM PRETRAINING-KNOWLEDGE PROBE

The frozen LLM used by THREATALIGN does not disclose a knowledge cutoff. Because all HEAA reports predate the model's release on December 1, 2025, each could have appeared in its pretraining corpus. We probe all 264 query-side reports to test whether the model retains report-specific knowledge useful for attribution. We retain the first 1,800 characters, mask actor names and aliases together with directly identifying malware, tools, CVEs, domains, IP addresses, hashes, and file names, and ask the same frozen model to recover details from the original report without external tools. Reports for which the model recovers original details are labeled memorized, and the remainder are labeled non-memorized. At the report level, the probe identifies 125 memorized and 139 non-memorized reports, associated with 49 and 35 actors, respectively.

Table 13: Attribution performance (%) by probe-detected pretraining knowledge, with mean and standard deviation over three runs.

Method	M-P	M-R	M-F1
memorized			
APT-ATT	**30.2 $\pm$ 1.6**	38.1 $\pm$ 1.9	**31.9 $\pm$ 1.5**
CyBERT	26.0 $\pm$ 2.4	**40.1 $\pm$ 2.5**	29.2 $\pm$ 2.6
THREATALIGN	**47.0 $\pm$ 2.2**	**55.1 $\pm$ 1.7**	**49.3 $\pm$ 1.9**
non-memorized			
APT-ATT	**27.6 $\pm$ 1.3**	**27.6 $\pm$ 1.3**	**27.6 $\pm$ 1.3**
CyBERT	14.8 $\pm$ 4.1	17.1 $\pm$ 4.7	15.6 $\pm$ 4.3
THREATALIGN	**39.5 $\pm$ 1.2**	**44.8 $\pm$ 1.3**	**41.1 $\pm$ 1.2**

As shown in Table 13, THREATALIGN obtains 49.3% M-F1 in the memorized bucket and 41.1% in the non-memorized bucket, a drop of 8.2 points, suggesting that detected report memory may aid attribution. APT-ATT and CyBERT also drop by 4.3 and 13.6 points under the same partition, indicating that the non-memorized bucket is more difficult across methods. Nevertheless, THREATALIGN remains 13.5 points above the strongest baseline, APT-ATT, in this bucket.

The probe detects recovery of report details under a particular prompt. Failure to recover details does not establish the absence of pretraining knowledge, and the buckets also differ in difficulty for classification baselines. Performance differences between them therefore cannot be attributed entirely to model memory.

D.5 CASE STUDY

Figure 4 illustrates attribution without direct IoC correspondences. The query graph comes from a Zscaler report describing an unknown actor using RevC2 and Venom Loader; the reference graph comes from an Arctic Wolf report describing Venom Spider and its use of the More_eggs backdoor. The graphs share no file hashes or C2 infrastructure that form DAs, and their main malware names differ. Direct IoC and malware-name matching therefore provides no correspondence between the actors.

HSP retrieves external context linking RevC2 and Venom Loader to Venom Spider and incorporates it into malware and actor profiles. The query-side malware profiles explicitly associate both tools with Venom Spider. ECA compares these associations and the Malware-as-a-Service and tool ecosystem context across actor profiles, aligning the query actor to Venom Spider. The case illustrates how external knowledge and recursive profiles support attribution without shared IoCs.

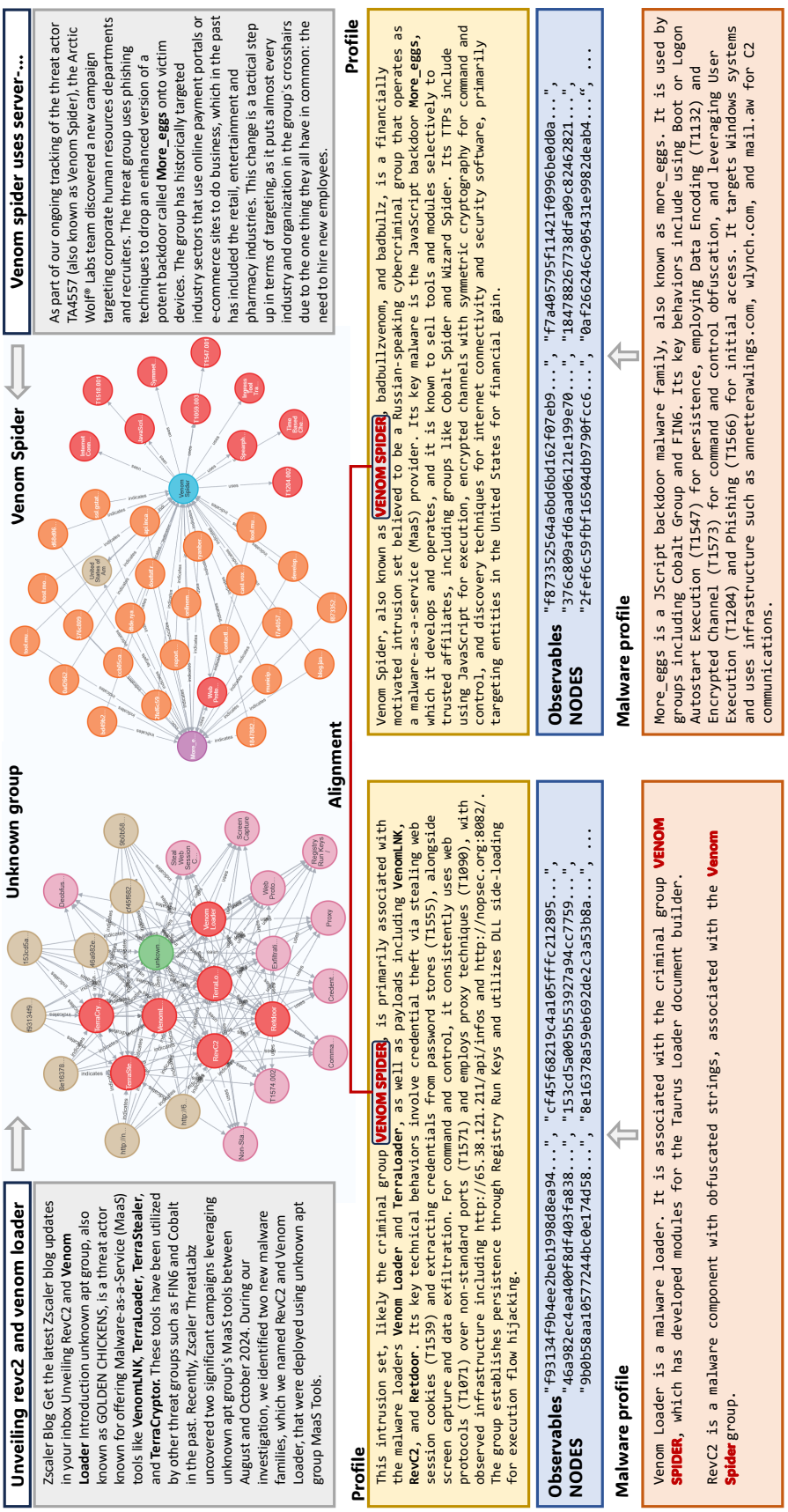

Figure 4: Venom Spider attribution case. Report excerpts, entity evidence, and generated profiles show the connections between the query and reference graphs.

E LIMITATIONS

THREATALIGN supports analyst-in-the-loop threat attribution. Its M-F1 of 43.5% on HEAA leaves a substantial risk of incorrect attribution. LLM-generated semantic profiles may also contain hallucinations or weakly supported associations, so candidate attributions require independent evidence verification before operational use.

Attribution also depends on reusable evidence across reports. On HEAA(time), THREATALIGN obtains 41.3% M-F1, compared with 43.5% under the default split; all evaluated methods score 0 in the NN-only bucket. Maintaining reliable alignment and attribution as reported evidence changes remains an open problem, particularly when few cross-graph correspondences are available.

LLM-based profile generation and candidate arbitration also incur inference costs. The framework is intended for offline CTI analysis; its use in latency-sensitive threat-analysis workflows requires further reductions in these costs.