# NETTRACE: EVIDENCE-BOUNDED LLM REASONING FOR NETWORK TRAFFIC INVESTIGATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Tool-using language models can cite actual observations yet draw conclusions those observations do not support. In network investigation, support depends on the evidence set and a claim's entity, time, traffic layer, quantifier, and strength. We formulate evidence-bounded traffic investigation and introduce NETTRACE, combining structured traffic inputs, claim–evidence supervision, and a separately trained scope-aware verifier with conflict checks and revision. On NetTraceBench's 4,800-case test split, NetTrace achieves 98.96 macro-F1 and a 2.84% non-entailing-link rate in the supplementary batch. A same-scale text verifier with matched deterministic scope checks achieves 86.86 and 5.62%, respectively. The F1 difference is 12.10 points; separate controls show advantages under fixed observations and matched answer coverage. Counterfactual pairs, joint-evidence tests, stricter splits, and an independent human audit examine the support mechanism under controlled interventions and distribution shifts.

## 1 INTRODUCTION

Network traffic investigation requires decisions that are useful and auditable. A periodic TLS flow can suggest command-and-control (C2) activity without establishing host compromise. Assessing this hypothesis may require additional flows, protocol metadata, or intelligence. Beyond predicting a label, investigators must acquire observations, connect them across traffic scopes, and limit conclusions to what the evidence supports.

Existing traffic models learn statistical, temporal, or graph features for closed-set prediction (Buczak & Guven, 2016; Wang et al., 2018; Lotfollahi et al., 2020; King & Huang, 2022). Language models offer instruction following and tool-mediated reasoning (Ouyang et al., 2022; Yao et al., 2023), but these capabilities do not inherently preserve traffic structure or ensure evidential support. Flat textual serialization leaves packet order, communication direction, and flow–host relations implicit. Supervision of final answers alone does not explicitly distinguish observed facts from the model's interpretations.

We study *how a language-model investigator can learn from structured traffic and adjust report specificity to the evidence it has acquired*. We distinguish three properties: decision correctness, traceability to observed evidence, and correctness of that evidence. The term *evidence-bounded* denotes a support constraint on what the investigator reports.

Figure 1 presents the end-to-end NetTrace workflow. Stage I adapts a language model to Structure-Preserving Flow Language (SPFL), a compact, task-oriented representation of hierarchical traffic. Training uses report and citation supervision with auxiliary representation losses. Stage II further trains the policy adapter on tool actions, arguments, evidence-linked reports, and revisions. A separate verifier adapter learns whether an observation set supports each atomic claim at its stated scope. During inference, the policy records observation provenance and scope; the verifier triggers further collection, qualification, correction, or abstention when support is incomplete.

Our contributions address three aspects. **Problem:** we formulate traffic-scoped support as a relation between a claim and an evidence set, constrained by entity, time, traffic layer, quantifier, and conclusion strength. **Method:** we learn and apply this relation through structure-aware traffic inputs, evidence memory, citation supervision, joint support verification, relevant-conflict checks, and re-verification after revision. **Evidence:** a matched general-verifier substitution, counterfactual evi-



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.

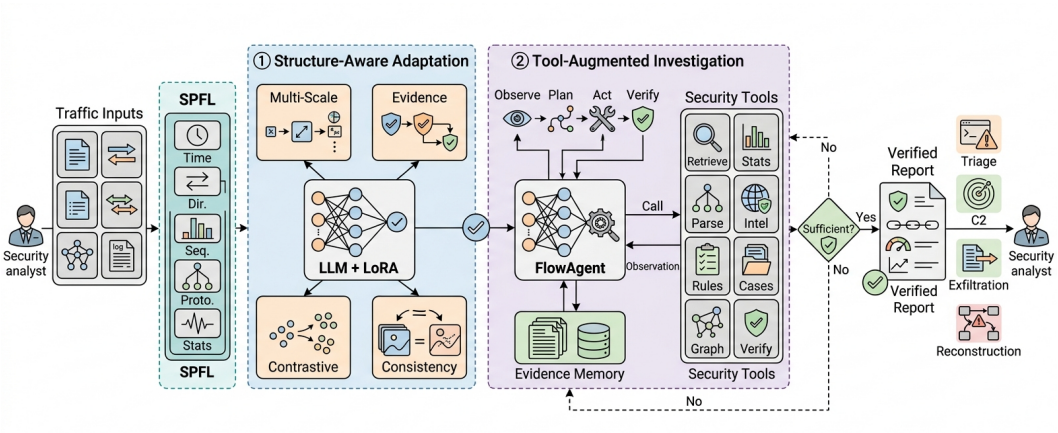



Figure 1: End-to-end overview of NetTrace. SPFL converts traffic into structure-aware inputs; Stage I learns traffic and evidence representations, and Stage II conducts tool-augmented investigation with evidence memory and support verification. A report is released when the observed evidence is sufficient.

dence pairs, fixed-observation and coverage controls, stricter splits, and independent human audits examine the mechanism under controlled interventions and distribution shifts. Four-way labels and a separate verifier are design choices; the contribution lies in the explicit support relation and evidence of its value for traffic investigation.

## 2 Evidence-Bounded Traffic Investigation

Let $X = (P, F, S, H)$ contain packet-, flow-, session-, and host-level traffic, $q$ be an investigation query, and $\mathcal{T}$ the available tools. A policy receives $q$ and a serialization $\phi(X)$, acquires observations through trajectory $\tau$, and outputs

$$o = \left(y, \{(c_j, E_j, v_j)\}_{j=1}^{m}, \tau\right), \qquad m \le 6, \tag{1}$$

where $y \in \{\texttt{target}, \texttt{alternative}, \texttt{insufficient}\}$ is the investigation decision, $c_j$ is an atomic claim, $E_j$ is its cited observation set, and $v_j \in \{\texttt{supported}, \texttt{partial}, \texttt{unsupported}, \texttt{contradicted}\}$ is its support state. We write $c = (p, e, [t_s, t_e), \ell, q_c, s)$ for predicate, entity, half-open time interval, traffic layer, quantifier, and conclusion strength; $\sigma(c)$ comprises the last five fields. The evidence-bounded criterion is

$$\begin{aligned} \mathrm{Release}(c) \Rightarrow \exists\, \varnothing \neq E^{+} \subseteq E_c : \\ \mathrm{ScopeOK}(E^{+}, \sigma(c)) \wedge \mathrm{Sufficient}(E^{+}, c) \\ \wedge \neg \mathrm{UnresolvedConflict}(E_{\mathrm{obs}}, c). \end{aligned} \tag{2}$$

This criterion permits joint support but excludes selecting a favorable subset when observed counterevidence remains unresolved. It applies to substantive report claims, factual titles, and the decision. A qualified claim is a new $c'$ whose narrower scope must itself be supported; adding "possibly" does not establish that relation. Equation 2 specifies the intended reporting behavior. The supplementary configuration implements this criterion through relevant-evidence completion and conflict scanning (Appendix H.2). Section 4 evaluates how reliably this operational approximation satisfies the criterion. We further distinguish support under observed content, $S_{\mathrm{obs}}$, from reliability against external truth, $S_{\mathrm{ext}}$. A corrupted but plausible observation can yield $S_{\mathrm{obs}} = 1$ and $S_{\mathrm{ext}} = 0$.

**What is insufficient?** Let $g_i$ denote the external reference fact and $a_i(\omega, B)$ indicate whether a legal trajectory can establish it in environment $\omega$ within budget $B$. The corresponding investigation target is $y_i^* = g_i$ if $a_i = 1$, and $\texttt{insufficient}$ otherwise. Sufficiency of the evidence actually acquired, $a_i^t$, is a separate property: $a_i^t = 0$ does not imply $a_i = 0$. A truthful early abstention may therefore fail the investigation task. This distinction clarifies the label semantics without treating insufficiency as a third physical state or relabeling cases based on a method's observed trajectory.

# 3 THE NETTRACE FRAMEWORK

## 3.1 STRUCTURE-PRESERVING FLOW LANGUAGE

A flat serialization such as `duration=3.2`, `protocol=tcp`, and `packets=18` leaves hierarchy and order implicit. SPFL exposes these relations through six complementary views:

$$\phi(X) = \langle M(X), T(X), D(X), Q(X), R(X), V(X) \rangle, \tag{3}$$

where $M$ encodes hierarchy and metadata, $T$ temporal statistics, $D$ communication direction, and $Q$ the ordered packet summary. The remaining views encode protocol state ($R$) and payload-derived statistics ($V$). For each retained cue $I_j$ in the invariant set $\mathcal{I}_{\mathrm{SPFL}}$, $I_j(\phi(X)) = g_j(I_j(X))$. Here, $g_j$ is the identity map or a specified bucketing or windowing transform. Thus, order, direction, protocol state, and selected statistics remain recoverable up to $g_j$.

SPFL provides a compact representation by anonymizing identifiers, summarizing raw payload bytes, and windowing long sessions. External tools complement this representation when payload-level content or finer inter-window evidence is required. Appendices D and H.3 specify the representation's scope, units, window rules, and an explicit serialization record.

## 3.2 EVIDENCE-GROUNDED SECURITY REASONING

For a claim $c$, let $\mathcal{G}(c) = \{G_1, \ldots, G_K\}$ denote alternative minimal sufficient support sets, and $E_{\mathrm{pred}}$ its citations. Complete support requires at least one $G_k \subseteq E_{\mathrm{pred}}$, rather than coverage of the union of all alternatives. We define best-path recall and citation-existence error as

$$\mathrm{Coverage}_E = \max_{G \in \mathcal{G}(c)} \frac{|E_{\mathrm{pred}} \cap G|}{|G|}, \qquad \mathrm{ACR} = \frac{|E_{\mathrm{pred}} \setminus E_{\mathrm{obs}}|}{|E_{\mathrm{pred}}|}. \tag{4}$$

Coverage is undefined without a reference support path; ACR is undefined when no citations are provided. Under the supplementary protocol, evidence precision counts citations in the valid support/corroboration set $R^+(c)$; recall uses the best complete path, not the union of alternative paths. Initial-batch precision/recall values are reported separately in Figure 8. Neither discrete measure is back-propagated during training. Equation 8 optimizes citation generation; a separate verifier with parameters $\psi$ learns semantic support:

$$\mathcal{L}_{\mathrm{verify}} = -\frac{1}{N_v} \sum_j \log P_\psi(v_j^\star \mid q, c_j, E_j, \sigma(c_j)). \tag{5}$$

Training instances cover single and joint support, missing key evidence, wrong entities or windows, related-but-insufficient observations, and contradictions. No gradient passes through a generated evidence set, tool execution, verifier decision, or replanning; we use neither REINFORCE nor PPO.

## 3.3 EVIDENCE-BOUNDED INVESTIGATION POLICY

NetTrace uses the eight-tool suite in Table 3. At step $t$, state $s_t = (q, \phi(X), h_{t-1})$ includes the observation history, and $a_t \sim \pi_\theta(a_t \mid s_t)$ specifies a tool and its arguments. The observe–plan–act–verify loop updates evidence memory and checks support before reporting (Appendix D).

The evidence memory stores each observation as $(e_t, \mathrm{tool}_t, \mathrm{args}_t, \mathrm{scope}_t, \mathrm{provenance}_t)$. Observation-producing tools cannot access gold decisions, expert reports, or hidden reference trajectories. The inference budget allows eight observation-tool calls including retries, two verifier–replan rounds, six principal claims, 65,536 cumulative input tokens, and 4,096 generated tokens.

The verifier first applies deterministic checks to reference existence, tool provenance, entity mapping, time scope, and data status. A separate Qwen2.5-7B (Yang et al., 2024) LoRA (Hu et al., 2022) adapter jointly encodes the question, claim, scope, and up to eight observations. A $3584 \to 4$ head predicts the four support labels. Inputs are capped at 3,072 tokens. In the supplementary configuration, evidence selection begins by scanning memory for observations, including counterevidence, relevant to the claim's entity, time, predicate, and version relation. Let $K(c, E_{\mathrm{obs}})$ indicate completion of this relevant-candidate check. Release requires $K = 1$ and no unresolved relevant conflict. Necessary support and counterevidence must fit within eight observations and 3,072 tokens; otherwise, the claim remains partial. Only an explicit same-source version-supersession relation can

retire an older record. A newer favorable observation alone does not override conflicting evidence. With temperature $T = 1.18$, the operational gate releases an unqualified claim only if deterministic checks pass, $P(\texttt{supported}) \geq 0.85$, and $P(\texttt{contradicted}) \leq 0.10$. Partial support triggers collection or qualification; unsupported claims are withdrawn or replaced, and contradictions require correction. Every revised claim $c'$, factual title, and final decision is rechecked; budget exhaustion does not relax the gates. Section 4.4 distinguishes argmax classification from actual release risk.

**A scoped correction.** H0042's 12 MiB TLS transfer initially prompts a sensitive-data exfiltration claim citing volume, protocol, and expired reputation evidence. The gate rejects this claim ($P(S) = .12$). Subsequent authorization and execution records support a window-specific backup claim ($P(S) = .94$), changing the decision to `alternative`. The predicate and evidence path change; hedging would not resolve the claim's lack of support. Appendix H.8 gives the complete record.

### 3.4 Two-Stage Optimization

Token-level report, citation, action, argument, and revision losses use disjoint target regions (Appendix G); contrastive and admissible-perturbation losses are auxiliary. The adapters share a frozen backbone, but the policy adapter $\theta$ and verifier adapter $\psi$ are optimized independently. Thus, $\nabla_\psi \mathcal{L}_{\text{policy}} = 0$ and $\nabla_\theta \mathcal{L}_{\text{verify}} = 0$. The stage objectives are

$$\begin{aligned} \mathcal{L}_{\text{I}} &= \mathcal{L}_{\text{report}} + 1.5\mathcal{L}_{\text{cite}} + 0.2\mathcal{L}_{\text{con}} + 0.1\mathcal{L}_{\text{cons}}, \\ \mathcal{L}_{\text{II,policy}} &= \mathcal{L}_{\text{report}} + 1.5\mathcal{L}_{\text{cite}} + \mathcal{L}_{\text{act}} + \mathcal{L}_{\text{arg}} + 0.5\mathcal{L}_{\text{rev}}, \\ \mathcal{L}_{\text{II,verifier}} &= \mathcal{L}_{\text{verify}}. \end{aligned} \tag{6}$$

Stage I uses 100,800 derived instruction instances for three epochs (4,725 steps). Stage II uses 33,600 policy instances for two epochs (1,050 steps) and 201,600 balanced claim–evidence instances for the verifier (6,300 steps). Stage II policy training starts from the Stage I policy checkpoint. The verifier copies the Stage I adapter, adds the four-way head, and then trains independently. Table 2 in Appendix A specifies the full optimizer and parameter configuration.

## 4 Experiments

We evaluate whether explicit traffic scope and joint support improve investigation beyond a general text verifier, additional reasoning, or more conservative answering. The configuration suffix `-S` identifies results from the supplementary experiment batch. All controlled comparisons use results from the same experiment batch; initial- and supplementary-batch results are reported separately.

### 4.1 Benchmark and Independent Evaluation

NetTraceBench contains 24,000 cases, with 6,000 each for host malicious-network-activity triage (Host), C2 investigation, backup-versus-exfiltration discrimination (Exfil), and multi-stage investigation (Recon). Public sources include CICIDS2017 (Sharafaldin et al., 2018), CSE-CIC-IDS2018 (Communications Security Establishment & Canadian Institute for Cybersecurity, 2018), UNSW-NB15 (Moustafa & Slay, 2015), Bot-IoT (Koroniotis et al., 2019), ToN-IoT (Alsaedi et al., 2020), and CTU-13 (García et al., 2014). Exfil and Recon use controlled executions with Zeek/Suricata observations, authorization records, and event-order references. An IDS label alone cannot establish comprehensive host compromise, a session's C2 function, authorization, or an attack chain. Appendix H.1 provides target, alternative, and insufficient cases for each task.

Construction retains 24,000 of 30,800 candidates after parsing, truth-conflict, duplicate, and scope checks (Appendix B). Each event group combines four query windows and two permitted evidence scopes for one base event, yielding eight correlated cases. Splits contain 2,100/300/600 groups, or 16,800/2,400/4,800 train/validation/test cases; template families are disjoint. Event groups do not represent independent hosts or attack campaigns. Two annotators label decisions and acceptable support paths, with adjudication by a third annotator. The clean test split contains 3,960 determinate-reference and 840 reference-insufficient cases.

| Metric | Object / denominator | Judgment and empty-output handling |
|---|---|---|
| NELR | Claims with nonempty, existing citation sets | Independent joint support; zero denominator undefined. |
| ACR / USR | All citation IDs / all substantive assertions | ID existence / scope overreach; factual titles and decisions enter USR. Empty denominators undefined. |
| Trajectory success | All investigation cases | Correct reference decision, legal calls, complete support path (or justified insufficiency); empty output fails. |
| $S_{\text{obs}}$ / $S_{\text{ext}}$ | Same fixed 6,000-candidate pool | Visible-content / independent-reference binary support accuracy; withdrawal does not change denominator. |
| Accepted error | All released claims | Independently insufficient / released; without releases, error is undefined and acceptance zero. |
| Over-answering | 840 reference-insufficient cases | Determinate outputs / cases; scored alongside selective risk. |

Table 1: Supplementary evaluation protocol. Claim rates use pooled numerator/denominator counts within a run; run summaries use three seeds. Success, risk, and over-answering are case-level quantities. Human audit counts are adjudicated claim units.

**An independent support judge.** Evaluation uses a frozen reference evidence graph and a numerical/relational interpreter. Independent blinded adjudication checks agreement between free text and structured claims. The judge is separate from the policy's verifier and does not receive method identities, verifier probabilities, or hidden reference trajectories as model evidence. It assesses the stronger assertion when prose overstates a structured field. Valid unlisted support paths are independently adjudicated and applied symmetrically to all methods. This independent reference-annotation and blinded-adjudication protocol provides the basis for support evaluation (Appendix H.2).

Let $S_{\text{ind}}(E_c, c)$ be this independent set-support judgment and $\mathcal{C}_+ = \{c : \varnothing \neq E_c \subseteq E_{\text{obs}}\}$. We measure

$$\text{NELR} = \frac{\sum_{c \in \mathcal{C}_+} \mathbf{1}[S_{\text{ind}}(E_c, c) = 0]}{|\mathcal{C}_+|}. \tag{7}$$

Empty citation sets are excluded; the rate is undefined when the denominator is zero. NELR therefore complements unsupported-specificity rate (USR), which covers all substantive assertions, including uncited claims, decisions, and factual titles. Table 1 specifies the remaining denominators.

Run-level summaries use three training seeds, $\{7, 17, 42\}$. Confidence intervals use 10,000 paired group-bootstrap resamples (seed 1701; coverage analysis uses 2701), preserving shared cases and seeds. They quantify group-level uncertainty conditional on these runs. Capture-isolated tests resample parent capture/execution units rather than individual cases. Uncertainty reporting uses the available group-level analyses; aggregate-only results retain their point estimates.

### 4.2 Matched Controllers and Strong Verifier Substitution

The shared-controller comparisons fix the Stage II checkpoint, SPFL, tool access, initial observations, report limits, and observation/revision budgets. ReAct-S uses raw history; Memory-S adds structured memory; VerifierOnly-S checks raw-history evidence; Provenance-S checks citation existence and provenance. SelfCheck-S receives additional self-check/revision computation. TextVerifier-S jointly encodes the complete selected evidence set. It matches NetTrace-S in verifier scale, training instances, hard negatives, validation set, and hyperparameter-search allowance. TextScope-S also applies the same deterministic entity, time, and provenance checks. The principal comparison therefore tests explicit scope relations and set support beyond a generic verifier with matched checks. IndependentReAct-S uses a separately adapted policy; it supplements, rather than changes, the matched-controller analysis.

NetTrace-S achieves 98.96 macro-F1 and 2.84% NELR (Figure 2). Relative to TextScope-S, the aggregate F1 difference is 12.10 points and NELR decreases by 2.78 points (95% CI [-2.90,-2.66]). Trajectory success rises from 80.14% to 86.60%. The updated F1 has no seed SD or paired confidence interval. IndependentReAct-S achieves 84.94 F1, providing a comparison with a separately

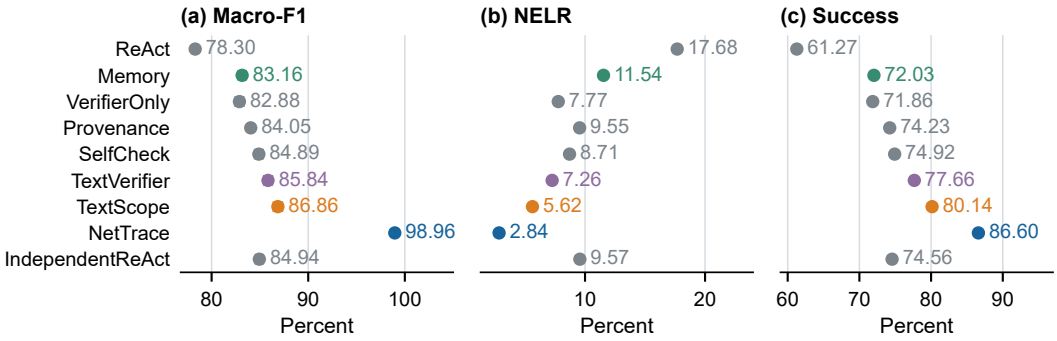



Figure 2: Supplementary batch (−S): 4,800 cases, 600 groups. F1 bars: available three-seed SDs; NetTrace-S: point estimate only. NELR/success: point estimates. IndependentReAct is separately adapted.

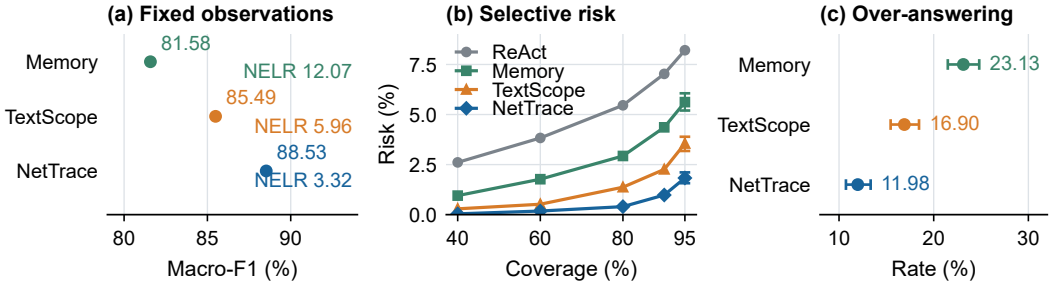



Figure 3: Controls. (a) Fixed-observation F1/NELR. (b) Risk: 3,960 determinate cases, recorded coverage points, paired group-bootstrap 95% CIs at 95% coverage. (c) Over-answering: 840 reference-insufficient cases, 95% CIs.

adapted policy. Resource accounting is separate: Full adds a 40,384,516-parameter verifier alongside a 40,370,176-parameter policy adapter and uses 201,600 verifier training instances.

### 4.3 Information, Coverage, and Mechanism Controls

**Fixed observations and selective risk.** A method-independent six-step plan uses only the query entity and time to acquire flows, statistics, protocol fields, fixed reputation, rule matches, and a visible relation graph. Replay permits no additional observation queries. NetTrace-S retains 88.53 F1/3.32% NELR, versus 85.49/5.96% for TextScope-S and 81.58/12.07% for Memory-S. The paired F1 advantage over TextScope-S has an interval of [2.70,3.37]. Its persistence under fixed observations separates support modeling from additional information acquisition; realized token consumption is reported separately.

The selective-risk analysis ranks determinate proposals by each method's saved confidence score and recomputes the coverage threshold within each bootstrap resample. On the 3,960 determinate-reference cases, risk at 95% coverage is 1.83% for NetTrace-S, 3.55% for TextScope-S, and 5.62% for Memory-S. The NetTrace-S minus TextScope-S interval is [-1.90,-1.49] percentage points. On the separate 840 reference-insufficient cases, over-answering is 11.98%, 16.90%, and 23.13%, respectively. Appendix H.4 gives all coverage points and the score-ranking protocol.

**Scope, joint evidence, and conflicts.** Tests in Figure 4 vary one support-relevant factor at a time while keeping paired inputs otherwise similar. TextScope-S handles entity/time changes well but trails NetTrace-S on layer expansion, quantifiers, missing joint evidence, and relevant counterevidence. Mean pair accuracy is 94.83% versus 82.85% (difference CI [10.69,13.33]). Across 600 cases per joint-support condition, NetTrace-S recognizes sufficient unions in 560 cases, downgrades claims after key-evidence removal in 564, and blocks release after conflicting evidence in 570. TextScope-S achieves 447, 428, and 445, respectively. Renaming and irrelevant-evidence controls test whether sensitivity concerns support rather than arbitrary input changes.

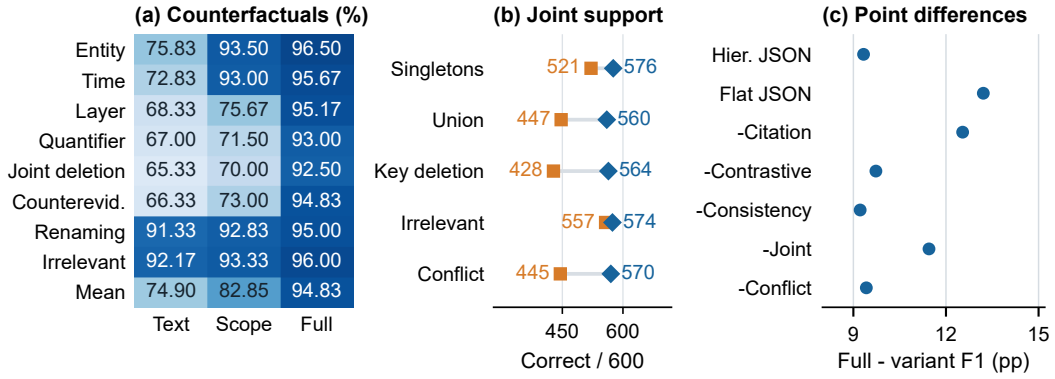



Figure 4: Support controls. (a) Both-member pair accuracy: Text/Scope/Full = TextVerifier-S/TextScope-S/NetTrace-S. (b) Correct/600: TextScope-S (orange squares), NetTrace-S (blue diamonds). (c) Full-minus-variant aggregate F1 differences; updated paired confidence intervals are unavailable.

**What representation and losses contribute.** Hierarchical JSON achieves 89.63 F1/2.98% NELR, compared with SPFL's 98.96/2.84%; information-matched flat JSON achieves 85.75/6.18%. Hierarchical JSON outperforms flat JSON, supporting the value of explicit structure. Without joint reasoning, F1 is 87.51; without conflict scanning, it is 89.54. The configuration without consistency regularization achieves 89.74 F1. These aggregate comparisons describe performance differences, but updated paired intervals are unavailable. They do not establish the statistical significance of individual component contributions or isolate interactions among components (Appendix H.5).

### 4.4 Release Risk and Distribution Shift

**Argmax errors differ from release errors.** The four-class challenge matrix contains 2,400 examples per class. It records 215 insufficient-to-supported predictions: $215/7200 = 2.99\%$ false acceptance among insufficient inputs, but $215/2435 = 8.83\%$ error among supported predictions. In the supplementary gate evaluation, deterministic checks, $P(S) \geq 0.85$, and $P(C) \leq 0.10$ accept 2,160 candidates, of which 48 are insufficient. Accepted error is 2.22% ([1.63,2.85]), candidate acceptance is 22.50%, and supported recall is 88.00%. Acceptance on this balanced challenge differs from investigation answer coverage.

The threshold maximizes acceptance subject to empirical accepted error of at most 3% on 4,800 validation candidates (3,840 supported); temperature 1.18 is retained. In a separate dynamic pool, insufficient claims account for 2,160/12,000 initial candidates, 540/10,800 post-revision candidates, and 260/10,000 released claims (2.60%). Denominators reflect withdrawal as well as revision; these proportions do not define a paired correction rate (Appendix H.6).

**Stricter splits.** Capture/execution isolation yields 84.97 F1 for NetTrace-S versus 80.46 for TextScope-S (difference CI [3.91,5.07]), with 4,800 cases in 24 held-out parent units. Whole-source CTU-13 holdout yields 83.54 versus 78.74 ([3.80,5.96]) on 2,000 Host/C2 cases. Unseen controlled configurations yield 83.96 versus 78.79 ([4.43,5.95]) on 1,200 cases in 150 new execution groups. NetTrace-S retains a consistent advantage across all three tested distribution shifts. The CTU-13 holdout evaluates Host/C2 transfer, and the unseen-configuration evaluation remains within controlled environments.

**Faults and external truth.** Faults are tied to observation IDs, so repeated queries cannot evade them. With 30% missing/corrupted observations, NetTrace-S achieves 82.90/77.15 F1 versus 76.82/70.86 for TextScope-S. In a separate fixed pool of 6,000 claims from 150 groups, missing observations yield equal observed/external support accuracy (92.22%). Corruption separates support under observed content from support against external reference truth. NetTrace-S attains 93.62% observed-content accuracy and 80.50% external-reference accuracy, yielding a 13.12-point gap ([12.23,13.98]). This controlled comparison illustrates why evidence-bounded reporting and external-truth reliability require distinct evaluation targets.

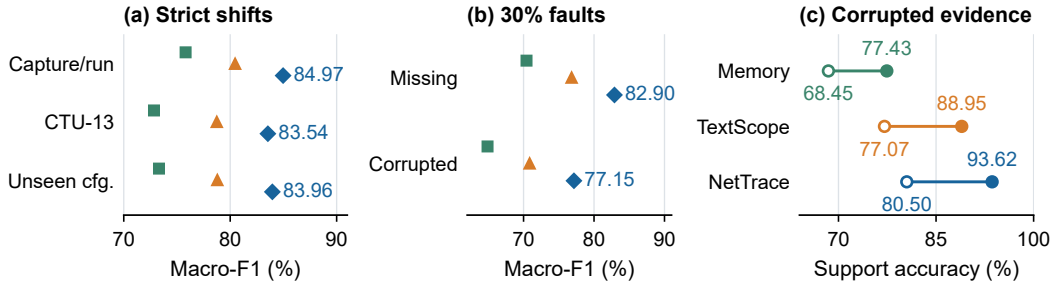



Figure 5: Transfer and faults. (a,b) Memory-S: green squares; TextScope-S: orange triangles; NetTrace-S: blue diamonds. (c) Corrupted-content support: observed (filled) and external-reference (open) accuracy. Connectors join endpoints, not uncertainty intervals.

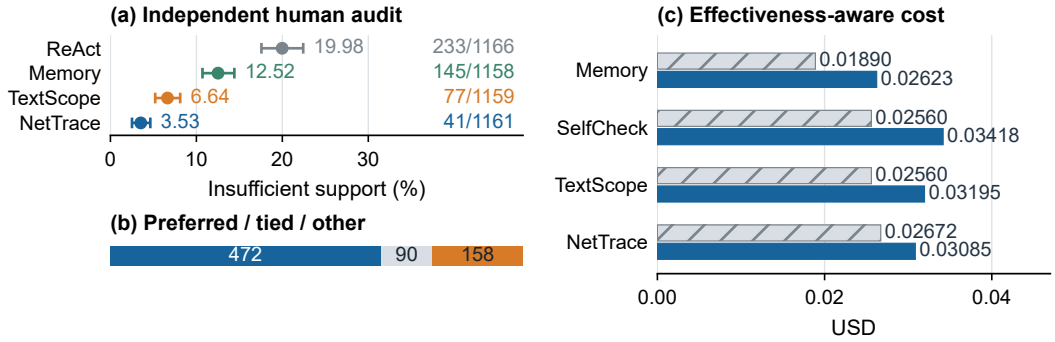



Figure 6: Human audit and cost. (a) Adjudicated support errors, claim counts, case-group 95% CIs. (b) NetTrace-S versus TextScope-S: 472 wins/90 ties/158 losses; 720 judgments, 240 cases. (c) USD per case (hatched) and successful investigation (solid) at the stated rates.

**Reconstruction beyond classification.** On 480 reference-target Recon cases with four stages and three directed relations each, NetTrace-S achieves 96.73 node F1, 89.84 relation F1, and 2.14% order error. TextScope-S achieves 93.64/78.88/5.87%, respectively. Node matching uses stage type, entity, and time. Order error is conditional on connected, comparable pairs and must be interpreted alongside relation F1; accurate stage detection alone does not establish a correct chain.

### 4.5 Human Audit and Cost

Six blinded evaluators assess 240 cases from distinct event groups: 80 per clean, missing, and corrupted condition, and 60 per task. Four methods produce 960 reports; three assignments per report yield 2,880 evaluations, 480 per evaluator. Claim counts vary across reports. Independently adjudicated insufficient citation sets occur in 41/1,161 NetTrace-S claims (3.53%, CI [2.51,4.64]), versus 77/1,159 TextScope-S claims (6.64%, [5.21,8.12]). The paired-difference interval is [-4.31,-2.03] points. NetTrace-S is preferred in 65.56% of paired judgments (case-group interval [61.94,69.17]); preference counts and ties appear in Appendix H.7. The 720 judgments share 240 cases, which serve as resampling units.

At accounting rates of USD 2/GPU-hour and USD 0.2/CPU-service-hour, NetTrace-S costs USD 0.02672 per case and USD 0.03085 per successful investigation, compared with USD 0.02560 and USD 0.03195 for TextScope-S. Relative to Memory-S, NetTrace-S uses additional per-case computation while achieving higher trajectory success and stronger evidence support. Appendix H.7 reports the complete cost comparison.

## 5 Related Work

**Traffic representations.** Encrypted-traffic classification (Rezaei & Liu, 2019), botnet detection (García et al., 2014), and online anomaly detection (Mirsky et al., 2018) motivate specialized

encoders. ET-BERT and YaTC pretrain datagram and hierarchical flow representations (Lin et al., 2022; Zhao et al., 2023). SPFL exposes selected traffic relations for interactive investigation rather than classification alone. Our comparisons focus on tool-using investigation policies and support verification, not classifier rankings.

**Language-model adaptation and investigation.** Instruction tuning (Ouyang et al., 2022), LoRA (Hu et al., 2022), and retrieval augmentation (Lewis et al., 2020) support adaptation. TrafficLLM adapts language models to traffic analysis through traffic-domain tokenization and two-stage fine-tuning for detection and generation (Cui et al., 2025). PentestGPT and LogPrompt address penetration testing and log analysis (Deng et al., 2024; Liu et al., 2024). NetTrace targets defensive investigation with evidence-bounded reports.

**Tool use and evidential support.** ReAct, Toolformer, and Reflexion support tool use and reflection (Yao et al., 2023; Schick et al., 2023; Shinn et al., 2023). ALCE evaluates citation support, while RARR retrieves evidence and revises unsupported content (Gao et al., 2023b;a). Self-RAG learns support critiques; Trust-Align addresses attribution and refusal (Asai et al., 2024; Song et al., 2025). NetTrace examines traffic-scoped joint support, conflicts, and explicit release control, connecting explainable security (Neupane et al., 2022) with hallucination analysis (Ji et al., 2023) while distinguishing support under observed content from external truth.

## 6 Discussion and Limitations

The main evidence comes from independent support evaluation, a matched text-verifier substitution, counterfactual relations, and fixed-observation/coverage controls. Together, these comparisons support the value of explicit traffic-scoped set reasoning in the evaluated configuration. Hierarchical JSON outperforms information-matched flat JSON, supporting explicit structure while leaving serialization-specific contributions open to further study.

SPFL provides a compact traffic representation, complemented by tool observations for payload-level and finer inter-window questions during investigation. Strict splits assess capture-, source-, and configuration-level shifts using one backbone/tool suite and controlled Exfil/Recon environments. Evaluation spans three training seeds, 24 held-out parent units in the capture/execution split, and a 240-case human audit. These settings provide a basis for further evaluation across additional backbones, tool suites, and deployment environments.

Learned support and conflict checks apply the evidence criterion through relevance scanning and conflict detection. The release gate is calibrated on validation data, with risk estimates tied to the tested settings; its reliability rests on these judgments and the available evidence.

## 7 Conclusion

NetTrace learns whether acquired observations jointly support an individual claim at its stated traffic scope. Matched verifier substitution, joint-evidence and counterevidence tests, and independent audits support verification beyond citation existence. Controlled replay and coverage analysis weaken explanations based on information acquisition or abstention. Stricter splits show that NetTrace's advantage persists across the tested capture-, source-, and configuration-level distribution shifts.

## AI USE STATEMENT

OpenAI ChatGPT was used solely for grammar correction and language polishing of author-written text throughout the manuscript.

## REPRODUCIBILITY STATEMENT

The code and associated materials for this work will be made publicly available at `https://anonymous.4open.science/r/NN-63D6/`.

## REFERENCES


Abdullah Alsaedi, Nour Moustafa, Zahir Tari, Abdun Mahmood, and Adnan Anwar. TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems. *IEEE Access*, 8:165130–165150, 2020. doi: 10.1109/ACCESS.2020.3022862.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations (ICLR)*, pp. 9112–9141, 2024. URL `https://proceedings.iclr.cc/paper_files/paper/2024/file/25f7be9694d7b32d5cc670927b8091e1-Paper-Conference.pdf`.

Anna L. Buczak and Erhan Guven. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2):1153–1176, 2016. doi: 10.1109/COMST.2015.2494502.

Communications Security Establishment and Canadian Institute for Cybersecurity. CSE-CIC-IDS2018 on AWS. `https://www.unb.ca/cic/datasets/ids-2018.html`, 2018. URL `https://registry.opendata.aws/cse-cic-ids2018`. Dataset.

Tianyu Cui, Xinjie Lin, Sijia Li, Miao Chen, Qilei Yin, Qi Li, and Ke Xu. TrafficLLM: Enhancing large language models for network traffic analysis with generic traffic representation. *arXiv preprint arXiv:2504.04222*, 2025. doi: 10.48550/arXiv.2504.04222.

Gelei Deng, Yi Liu, Víctor Mayoral-Vilches, Peng Liu, Yuekang Li, Yuan Xu, Tianwei Zhang, Yang Liu, Martin Pinzger, and Stefan Rass. PentestGPT: Evaluating and harnessing large language models for automated penetration testing. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 847–864. USENIX Association, 2024. URL `https://www.usenix.org/conference/usenixsecurity24/presentation/deng`.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16477–16508. Association for Computational Linguistics, 2023a. doi: 10.18653/v1/2023.acl-long.910.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6465–6488. Association for Computational Linguistics, 2023b. doi: 10.18653/v1/2023.emnlp-main.398.

Sebastián García, Martin Grill, Jan Stiborek, and Alejandro Zunino. An empirical comparison of botnet detection methods. *Computers & Security*, 45:100–123, 2014. doi: 10.1016/j.cose.2014.05.011.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. URL `https://openreview.net/forum?id=nZeVKeeFYf9`.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023. doi: 10.1145/3571730.

Isaiah J. King and H. Howie Huang. Euler: Detecting network lateral movement via scalable temporal link prediction. In *Network and Distributed System Security Symposium (NDSS)*, 2022. doi: 10.14722/ndss.2022.24107.

Nickolaos Koroniotis, Nour Moustafa, Elena Sitnikova, and Benjamin Turnbull. Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. *Future Generation Computer Systems*, 100:779–796, 2019. doi: 10.1016/j.future.2019.05.041.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 9459–9474, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Xinjie Lin, Gang Xiong, Gaopeng Gou, Zhen Li, Junzheng Shi, and Jing Yu. ET-BERT: A contextualized datagram representation with pre-training transformers for encrypted traffic classification. In *Proceedings of the ACM Web Conference 2022*, pp. 633–642, 2022. doi: 10.1145/3485447.3512217.

Yilun Liu, Shimin Tao, Weibin Meng, Feiyu Yao, Xiaofeng Zhao, and Hao Yang. LogPrompt: Prompt engineering towards zero-shot and interpretable log analysis. In *Proceedings of the 2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings*, pp. 364–365. ACM, 2024. doi: 10.1145/3639478.3643108.

Mohammad Lotfollahi, Mahdi Jafari Siavoshani, Ramin Shirali Hossein Zade, and Mohammdsadegh Saberian. Deep packet: A novel approach for encrypted traffic classification using deep learning. *Soft Computing*, 24(3):1999–2012, 2020. doi: 10.1007/s00500-019-04030-2.

Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. Kitsune: An ensemble of autoencoders for online network intrusion detection. In *Network and Distributed System Security Symposium (NDSS)*, 2018. doi: 10.14722/ndss.2018.23204.

Nour Moustafa and Jill Slay. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *2015 Military Communications and Information Systems Conference (MilCIS)*, pp. 1–6, 2015. doi: 10.1109/MILCIS.2015.7348942.

Subash Neupane, Jesse Ables, William Anderson, Sudip Mittal, Shahram Rahimi, Ioana Banicescu, and Maria Seale. Explainable intrusion detection systems (X-IDS): A survey of current methods, challenges, and opportunities. *IEEE Access*, 10:112392–112415, 2022. doi: 10.1109/ACCESS.2022.3216617.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 27730–27744, 2022. doi: 10.52202/068431-2011.

Shahbaz Rezaei and Xin Liu. Deep learning for encrypted traffic classification: An overview. *IEEE Communications Magazine*, 57(5):76–81, 2019. doi: 10.1109/MCOM.2019.1800819.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 68539–68551, 2023. doi: 10.52202/075280-2997.

Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*, pp. 108–116, 2018. doi: 10.5220/0006639801080116.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 8634–8652, 2023. doi: 10.52202/075280-0377.

Maojia Song, Shang Hong Sim, Rishabh Bhardwaj, Hai Leong Chieu, Navonil Majumder, and Soujanya Poria. Measuring and enhancing trustworthiness of LLMs in RAG through grounded attributions and learning to refuse. In *International Conference on Learning Representations (ICLR)*, pp. 30783–30823, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/4c88827decab6c046b881a2c3a99c76f-Paper-Conference.pdf.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. doi: 10.48550/arXiv.1807.03748.

Wei Wang, Yiqiang Sheng, Jinlin Wang, Xuewen Zeng, Xiaozhou Ye, Yongzhong Huang, and Ming Zhu. HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. *IEEE Access*, 6:1792–1806, 2018. doi: 10.1109/ACCESS.2017.2780250.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. doi: 10.48550/arXiv.2412.15115.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Ruijie Zhao, Mingwei Zhan, Xianwen Deng, Yanhao Wang, Yijun Wang, Guan Gui, and Zhi Xue. Yet another traffic classifier: A masked autoencoder based traffic transformer with multi-level flow representation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(4):5420–5427, 2023. doi: 10.1609/aaai.v37i4.25674.

**Result-batch convention.** Configuration and benchmark details apply unless a supplementary change is specified. Initial-batch figures and tables retain their values and populations (including the 150-case human study). Results marked −S and the 240-case audit belong to the supplementary batch and are not pooled with initial results. Supplementary protocols and values appear in Appendix H.

## A Training Configuration and Tool Interface

The base model is Qwen2.5-7B-Instruct (Yang et al., 2024). Separate policy and verifier LoRA adapters use rank 16, scale 32, dropout 0.05, and no bias, targeting q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, and down_proj. Each adapter has 40,370,176 trainable parameters. The contrastive projection is a bias-free $3584 \to 256$ layer (917,504 parameters); the verifier head has 14,340 parameters. All base-model weights remain frozen.

All stages use AdamW $(\beta_1, \beta_2) = (0.9, 0.95)$, $\epsilon = 10^{-8}$, weight decay 0.01, cosine decay, 0.03 warmup, gradient clipping at 1.0, and BF16. Contrastive learning uses one positive and four hard negatives per anchor, a normalized 256-dimensional projection, and temperature 0.1. Checkpoints are selected by validation macro-F1, with lower NELR breaking ties. Seeds are $\{7, 17, 42\}$.

| Setting | Stage I policy | Stage II policy | Stage II verifier |
|---|---|---|---|
| Training instances | 100,800 | 33,600 | 201,600 |
| Epochs / steps | 3 / 4,725 | 2 / 1,050 | 2 / 6,300 |
| Sequence length | 4,096 | 4,096 | 3,072 |
| Peak learning rate | $2 \times 10^{-4}$ | $1 \times 10^{-4}$ | $1 \times 10^{-4}$ |
| Global batch size | 64 | 64 | 64 |
| Report / citation | 1.0 / 1.5 | 1.0 / 1.5 | 0 / 0 |
| Contrastive / consistency | 0.2 / 0.1 | 0 / 0 | 0 / 0 |
| Action / argument / revision | 0 / 0 / 0 | 1.0 / 1.0 / 0.5 | 0 / 0 / 0 |
| Verification | 0 | 0 | 1.0 |

Table 2: Two-stage training configuration and active loss terms.

| Tool | Returned evidence | Excluded information |
|---|---|---|
| Flow Retriever | Flows, sessions, authorized event records | Gold decisions, expert reports |
| Traffic Stats | Recomputed periodicity and byte statistics | Attack labels |
| Protocol Parser | Available protocol fields | Handwritten protocol conclusions |
| Threat Intel | Fixed-version reputation snapshot | Test-derived reputation |
| Rule Engine | Fixed-rule matches and details | Answer-derived rules |
| Case Retriever | Training-history incidents only | Test incidents or paraphrases |
| Graph Builder | Graphs from visible entities/connections | Private truth graphs |
| Evidence Verifier | Current question, claim, and observations | Gold labels, hidden observations |

Table 3: Tool-source contract. Seven tools produce observations; the verifier consumes only visible observations.

Single-seed training requires 56 GPU-hours for Stage I, 34 for the Stage II policy, and 24 for the Stage II verifier (114 total). The full study records 342 GPU-hours for three-seed main training, 12 for hard-example generation, and 96 for hyperparameter search. Baselines and ablations require 1,080, with 120 for final confirmation, totaling 1,650 GPU-hours.

## B NetTraceBench Construction and Splits

Figure 7 summarizes the benchmark's composition and construction. Source coverage is task-specific: Exfil and Recon use the controlled Zeek/Suricata track, where authorization and execution records establish reference facts. Their initial-batch evaluation characterizes performance within that track. Exact source and decision counts accompany the figure as CSVs (Appendix I).

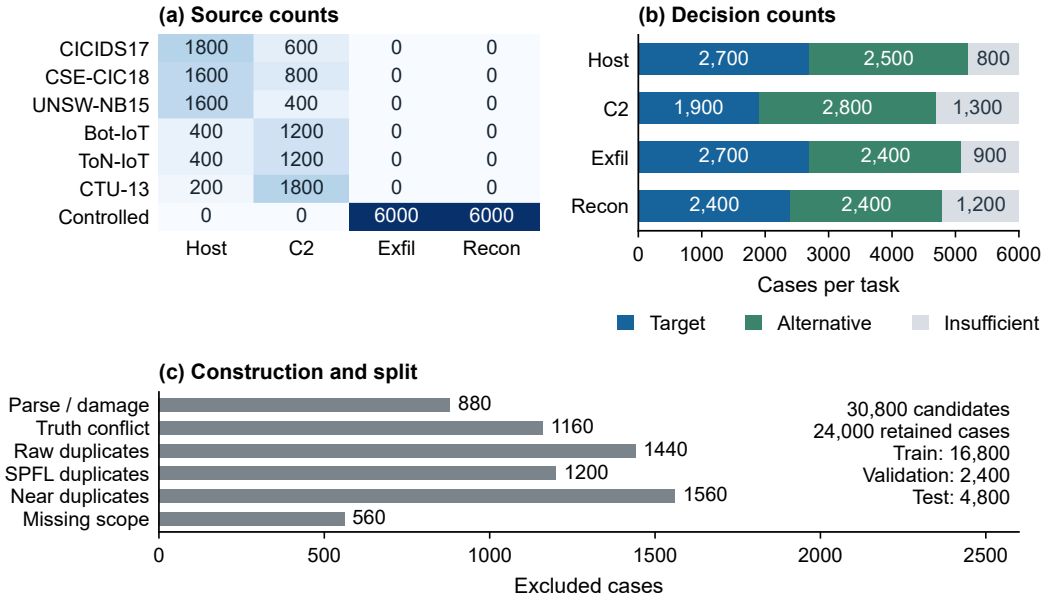



Figure 7: Benchmark composition. (a) Case counts by nominal source and task; zeros indicate no allocated cases. (b) Investigation-decision counts, not attack-family labels. (c) Recorded exclusions and retained split totals. The composition defines coverage; event-group splitting controls leakage. A C2 query does not imply a positive C2 label.

The pipeline excludes 880 parse failures or damaged records, 1,160 unresolved truth conflicts, 1,440 exact raw-record duplicates, and 1,200 exact normalized-SPFL duplicates. It also excludes 1,560 structural/content near duplicates and 560 cases without verifiable entity or time scope from 30,800 candidates. Near-duplicate review uses normalized-SPFL five-token shingles, 128-dimensional MinHash retrieval, and relation-structure review; candidates with Jaccard similarity of at least 0.90 are audited. Cases sharing raw sessions, overlapping event windows, or the same reference execution enter one event group before splitting.

Two annotators independently assess decisions, evidence sets, entity and time scope, and acceptable alternative trajectories; 1,920 cases (8.0%) require third-annotator adjudication. Recorded effort totals 6,400 hours for double annotation and 320 hours for adjudication. The reference trajectory is not uniquely required: any legal trajectory with the correct decision and an accepted support set succeeds.

## C Initial-Batch Verifier and Controlled Attribution

Figure 10 distinguishes classifier errors from the effects of representation, adaptation, tools, and training objectives. Existence-only checking accepts every instance in this challenge set; adding entity/time checks yields 83.3% false acceptance. A same-scale text classifier without explicit scope checking yields 10.7%; omitting hard-negative training from the full architecture yields 8.9%. These comparisons are reported as point estimates. The initial comparator matches classifier scale; the supplementary TextVerifier-S/TextScope-S substitutions also match training and revision-budget controls. The original confusion counts describe argmax classification; thresholded gate behavior and dynamic-release outcomes are evaluated separately in Appendix H.6.

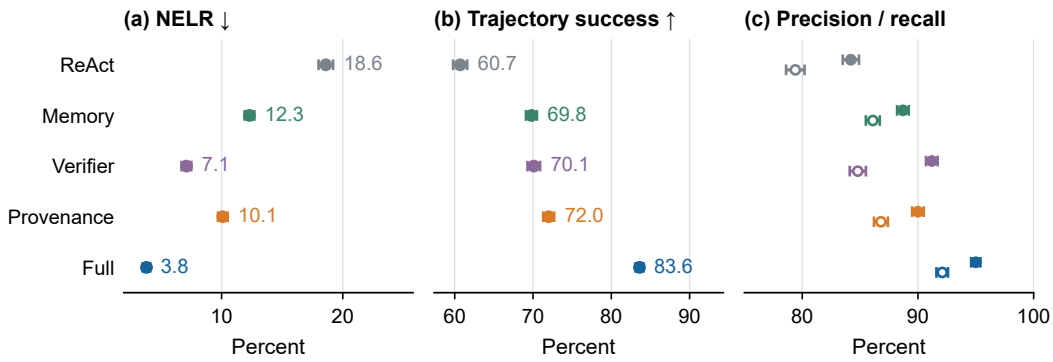



Figure 8: Initial batch: evidence quality under matched clean conditions. (a) NELR; (b) trajectory success; (c) evidence precision (filled) and recall (open). Bars show SD across three seeds on 4,800 test cases. These automatic metrics are distinct from the independent human audit. Exact values appear in the accompanying CSVs.

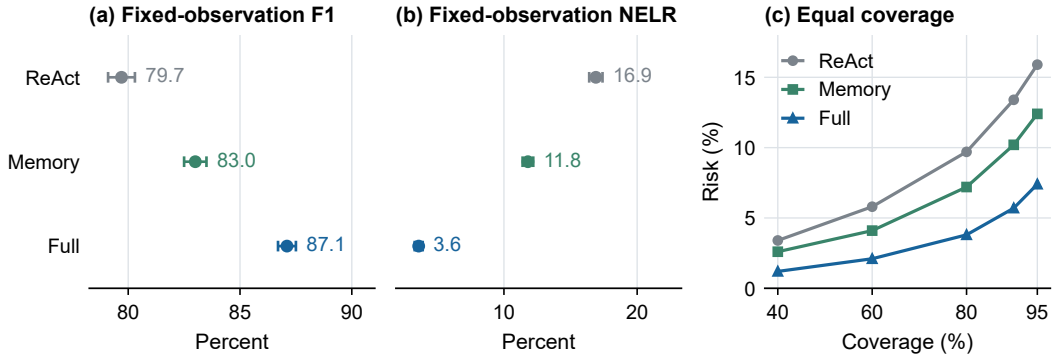



Figure 9: Initial batch: information-acquisition and coverage controls. (a,b) Fixed-observation replay without additional queries; bars show seed SD. (c) Conditional risk on 3,960 determinate-reference cases, excluding 840 reference-insufficient cases. Lines join five recorded coverage points; paired uncertainty is not estimated for this initial-batch curve.

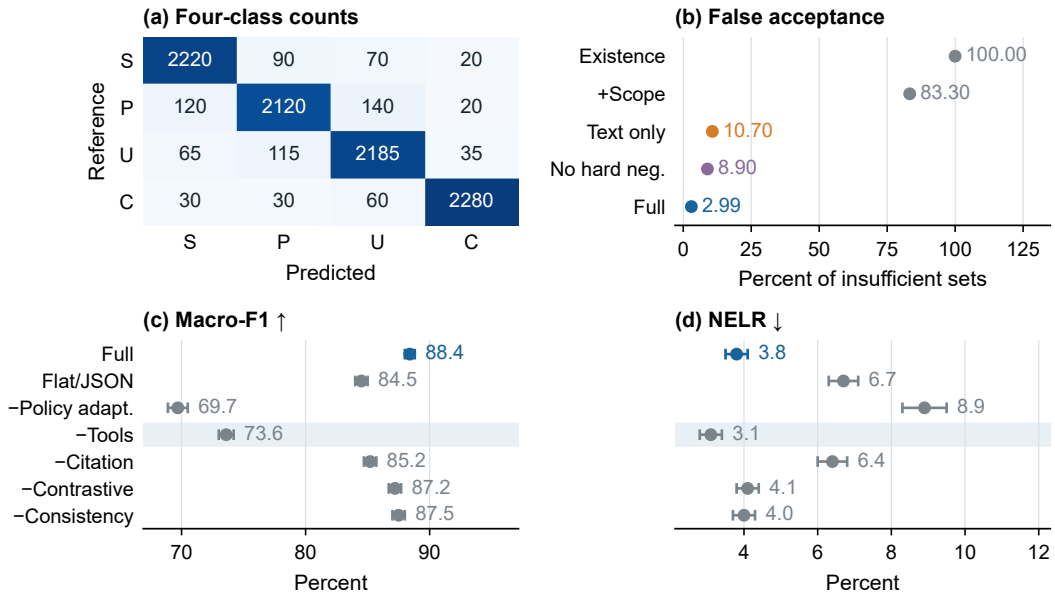



Figure 10: Initial batch: verifier and component controls. (a) Four-class counts, with 2,400 examples per reference class (S/P/U/C). (b) False acceptance on 7,200 insufficient sets. (c,d) End-to-end ablations with seed SD; the shaded no-tools condition reduces both NELR and decision performance. Initial argmax classification and supplementary release-gate evaluation use distinct protocols.

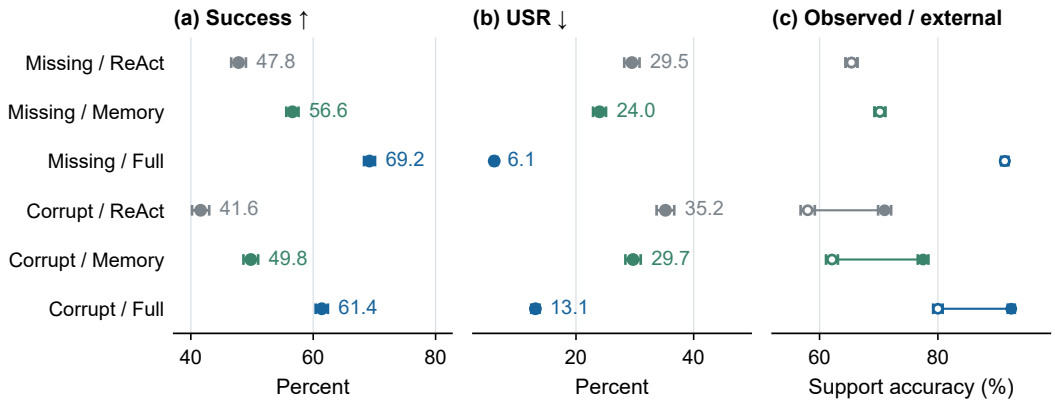



Figure 11: Initial batch with 30% observation faults. (a) Trajectory success; (b) USR; (c) observed-content (filled) and external-truth (open) sufficiency accuracy. Bars show SD across three seeds. Connectors join support endpoints, not uncertainty intervals; equal endpoints under missingness overlap.

## D Representation and Investigation Illustrations

The operational scenario and investigation loop below complement Figure 1. "Verified report" in these illustrations refers to a support check.

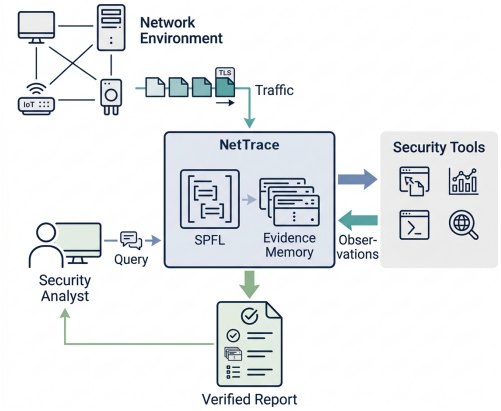



Figure 12: Operational scenario: an analyst query, traffic context, and tool observations inform the report.

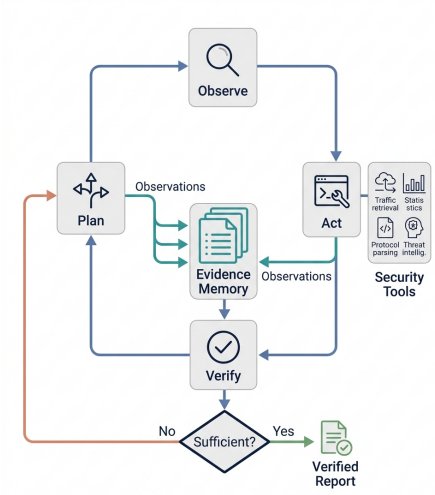



Figure 13: The observe–plan–act–verify loop accumulates observations and checks report support.

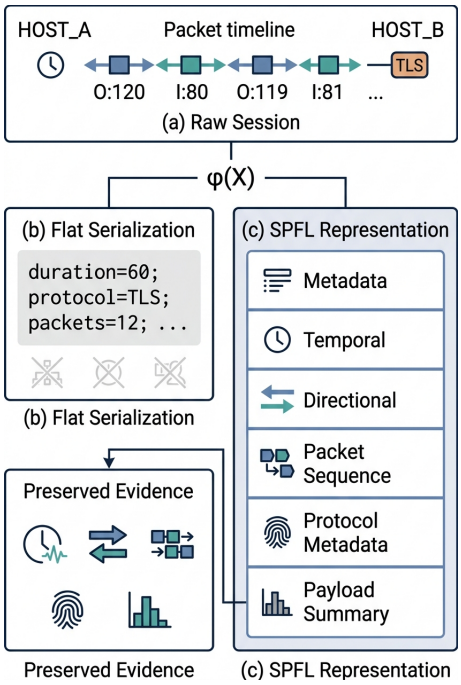



Figure 14: SPFL retains selected hierarchical, temporal, directional, packet-order, protocol, and payload-summary cues. This task-oriented, lossy representation preserves these cues according to its stated summarization and windowing rules.

## E Initial-Batch Human Evaluation and Claim Audit

We sample 150 non-overlapping cases: 50 clean, 50 with 30% missing observations, and 50 with 30% corrupted observations, distributed as 38/38/37/37 across the four tasks. ReAct-Adapted, Memory-only, Provenance-only, and Full produce 600 reports. Six blinded evaluators each rate 300 reports; every report receives three independent ratings, totaling 1,800 report ratings. Evaluators first assess support from the query and visible observations, then assess correctness against independent reference facts.

Figure 15 compares the five anchored rating dimensions. The claim audit and paired preferences appear in Figure 16. Human ratings show the clearest improvements in correctness, evidence sufficiency, and claim control, while conciseness remains comparable across methods.

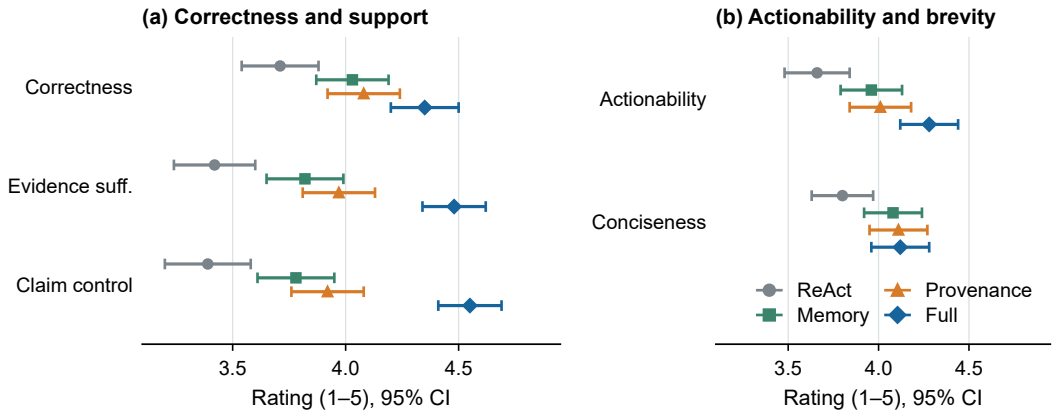



Figure 15: Initial-batch human ratings on 150 cases. (a) Correctness, evidence sufficiency, and claim control; (b) actionability and conciseness. Methods are ReAct (gray circles), Memory (green squares), Provenance (orange triangles), and Full (blue diamonds). Bars show event-level bootstrap 95% CIs from 600 reports, each rated by three evaluators.

The aggregate audit contains 900 principal claim–citation-set units per method. With 150 reports and at most six claims per report, this is the maximum possible count. The aggregate defines the claim denominator but does not specify report-level claim counts or absent/withdrawn-claim handling. Judgments share cases: the 450 preferences are not 450 independent investigations; paired-case interval estimation uses individual judgment records. Existing but insufficient citation sets occur for 167 ReAct claims (18.56%), 108 Memory-only claims (12.00%), 91 Provenance-only claims (10.11%), and 34 Full claims (3.78%). Across 450 paired judgments per comparison, Full wins/ties/loses 342/45/63 against ReAct and 306/54/90 against Memory-only. Krippendorff's $\alpha$ is 0.74 [0.68,0.79] for ordinal ratings and 0.78 for binary support judgments.

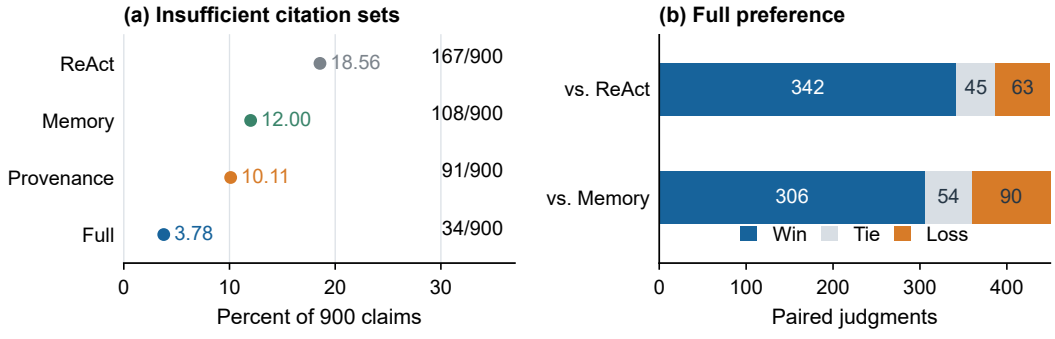



Figure 16: Initial-batch human audit. (a) Existing but insufficient citation sets among 900 principal claims per method, with exact count labels. (b) Full wins/ties/losses among 450 paired judgments per comparator. The displayed counts and proportions have no inferred uncertainty intervals.

# F Initial-Batch Inference and Training Cost

Measurements use one A100-80GB, 32 CPU cores, and 256 GB of memory, with the model and indexes preloaded.

Figure 17 compares monetary cost per case and per successful evidence-qualified investigation.

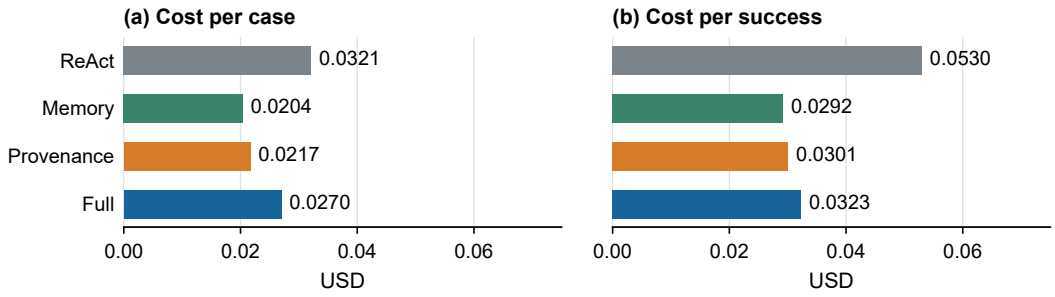



Figure 17: Initial-batch monetary inference cost with preloaded models and local snapshots: (a) USD per case; (b) USD per successful evidence-qualified investigation. Values use the reported accounting rates; sampling intervals are not estimated.

At rates of USD 2.0/GPU-hour and USD 0.2/CPU-service-hour, costs per case / successful evidence-qualified investigation are USD 0.0321/0.0530 for ReAct and 0.0204/0.0292 for Memory-only. Corresponding costs are 0.0217/0.0301 for Provenance-only and 0.0270/0.0323 for Full. These costs quantify Full's quality–cost trade-off alongside improved evidence support.

Figure 18 distinguishes token volume, observation-tool calls, and verifier batches. Full uses 12,900 input and 1,410 output tokens per case, versus 10,100 and 1,200 for Memory-only; it performs 1.4 verifier batches. These resources are reported separately rather than combined into a single call count.

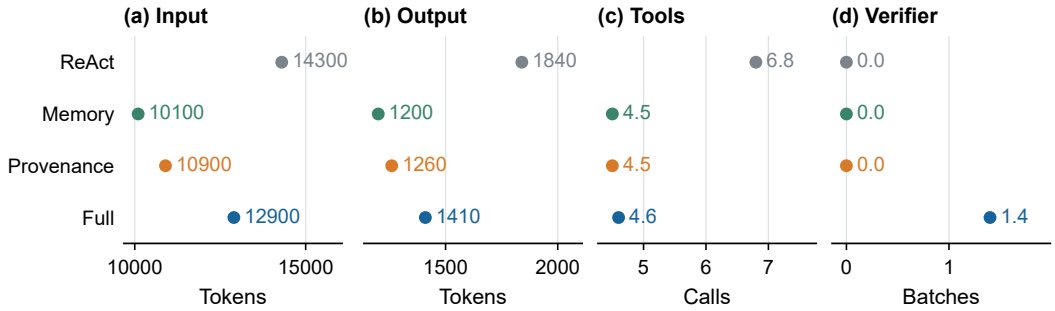



Figure 18: Initial-batch mean resources per case: input/output tokens, observation-tool calls, and verifier batches. Labels retain exact counts. Calls and verifier batches have different units and are reported separately; sampling intervals are not estimated.

# G Training Objectives

## G.1 Multi-Granularity Instruction Tuning

We construct instructions at six granularities: packet, flow, session, host, campaign, and incident. The target sequence is partitioned into disjoint regions $T_{\text{report}}, T_{\text{cite}}, T_{\text{act}}, T_{\text{arg}}, T_{\text{rev}}$ for report fields, evidence identifiers, tool actions, tool arguments, and verifier-triggered revisions. For region $k$,

$$\mathcal{L}_k = -\frac{1}{\max(1, |T_k|)} \sum_{t \in T_k} \log P_\theta(o_t^\star \mid o_{<t}^\star, q, \phi(X), h). \tag{8}$$

Prompt and tool-response tokens contribute no generation loss, and each target token belongs to exactly one region. Citation targets always refer to observations visible at that point in the trajectory.

### G.2 Traffic Semantic Contrastive Alignment

To reduce reliance on superficial fields, we use security-aware contrastive learning (van den Oord et al., 2018). For an anchor $x_i$, a positive $x_i^+$ shares its attack semantics in a different environment. A hard negative $x_i^-$ instead matches superficial cues but differs in its label and evidence relation. One example contrasts a TLS-based C2 beacon with benign cloud synchronization traffic. Pairs with ambiguous security interpretations are excluded. Given a pooled representation $z_i = \mathrm{Pool}(\mathrm{LLM}_\theta(\phi(x_i)))$ and $s(u, w) = \exp(\mathrm{sim}(u, w)/\kappa)$, the contrastive loss is

$$\mathcal{L}_{\mathrm{con}} = -\log \frac{s(z_i, z_i^+)}{s(z_i, z_i^+) + \sum_j s(z_i, z_j^-)}, \tag{9}$$

where $\mathrm{sim}(\cdot, \cdot)$ denotes cosine similarity and $\kappa$ is a temperature.

### G.3 Perturbation Consistency Regularization

We restrict $\tilde{x} = \mathcal{A}(x)$ to entity-consistent renaming, unordered-metadata reordering, and relative-time recoding that preserves absolute-time meaning. These transformations must preserve both the decision and annotated support relation; arbitrary port changes, timing perturbations, or packet dropping are not treated as invariant. We enforce

$$\mathcal{L}_{\mathrm{cons}} = D_{\mathrm{KL}}(P_\theta(y \mid x) \,\|\, P_\theta(y \mid \tilde{x}))\,, \tag{10}$$

which encourages consistency under the specified transformations. This auxiliary objective is assessed in the ablation in Appendix H.5.

## H Supplementary Experiment Protocols and Results

The supplementary batch uses configurations marked `-S` in the experiment records. All numerical values below are transcribed from those records; initial-batch results remain separately labeled. The initial batch retains its 88.4 F1, 150-case audit, and original cost measurements as a distinct set of results. The shared-controller design assesses contributions from inference components; the separately adapted baseline addresses a different type of policy comparison.

### H.1 Task facts, answerability, and support paths

For a fixed external reference $g_i$, environment $\omega$, and budget $B$, the benchmark asks whether a legal trajectory within $B$ can acquire a sufficient support path. If so, $y_i^* = g_i$; otherwise, the reference investigation decision is `insufficient`. Current evidence $E_t$ can be insufficient even when the reference decision is determinate. Stopping at that point may accurately describe the observations but fails to complete the investigation. Under faults, external facts remain unchanged: fixed-reference utility and fault-conditioned answerability are distinct evaluations. Reference labels must not change after a particular method's trajectory has been observed.

An insufficient result is evidence-qualified only if it accurately identifies the relevant evidence gap within the permitted access/budget. Merely omitting an answer receives no reward. Released claims and any decision are checked at their respective scopes; accuracy against the reference decision alone does not establish claim support.

The task records below specify the fact–question–decision–support mapping. A reference execution or authorization record can define the hidden label but constitutes legitimate model evidence only when acquired through the permitted interface. These task-level records make the label mapping explicit; the underlying captures and annotation database remain separate reproduction artifacts.

| Task | Target: facts and required support | Alternative: facts and required support | Insufficient: missing requirement |
|---|---|---|---|
| Host | H01's scoped scan matches malicious execution and flow records, supporting a finding of malicious network activity within that window. | H02's administrative scan matches approval, target list, and execution; the observed activity has an authorized explanation. | H03 has periodic TLS and high outbound volume but no evidence establishing malicious function; comprehensive compromise is not inferred. |
| C2 | C01 links controller commands, client responses, and reference behavior in the same session. | C02's vendor, update job, and execution receipt jointly explain the session. | C03's 60-second periodicity and historical bad reputation do not establish the current session's control function. |
| Exfil | E01 links protected content, the destination, transfer execution, and a complete authorization snapshot establishing lack of permission. | E02's task, destination, interval, and content digest match valid backup authorization. | E03's 12 MiB outbound transfer alone establishes neither sensitive content nor authorization status. |
| Recon | R01 has four execution-linked stages and three verified directed relations. | R02's alerts are explained by independent, authorized jobs. | R03 has temporally close alerts but lacks entity/execution links needed to establish the chain. |

Table 4: Supplementary task records and support requirements. A missing authorization record alone does not establish unauthorized activity; temporal proximity alone does not establish a stage relation.

### H.2 Support semantics and operational checks

A supported claim satisfies all necessary semantic conditions, has compatible scope, and has no unresolved relevant conflict. A partial label indicates support for a necessary premise but not for a required role, range, quantifier, or strength. Entity/time overlap alone does not establish partial support. An unsupported label indicates relevant context without the required semantic relation. Contradicted covers evidence negating a necessary fact or an unresolved relevant conflict.

The supplementary release contract is

$$\begin{aligned}\mathrm{Release}(c) \Rightarrow\ & \mathrm{Exists}(E_c) \wedge \mathrm{ScopeOK}(E_c, c) \\ & \wedge K(c, E_{\mathrm{obs}}) \wedge P_\psi(S \mid c, E_c) \geq .85 \\ & \wedge P_\psi(C \mid c, E_c) \leq .10 \wedge \neg\widehat{\mathrm{Conflict}}(c, E_{\mathrm{obs}}).\end{aligned} \tag{11}$$

$K$ records completion of the relevant-candidate check. The memory scan collects supporting and opposing observations by entity, time, predicate, and explicit source-version relations. If necessary support or relevant counterevidence exceeds eight observations/3,072 tokens, the claim remains partial. Explicit same-source version supersession may retire an older record; favorable recency alone cannot. Revisions constitute new claims; report titles and decisions undergo the same check. The operational contract depends on the accuracy of learned conflict detection and relevance scanning.

For alternative support paths $\mathcal{G}(c) = \{G_1, \ldots, G_K\}$ and valid supporting/corroborating evidence $R^+(c)$,

$$P_E = \frac{|E_{\mathrm{pred}} \cap R^+(c)|}{|E_{\mathrm{pred}}|}, \qquad R_E = \max_k \frac{|E_{\mathrm{pred}} \cap G_k|}{|G_k|}. \tag{12}$$

Complete support requires an entire $G_k \subseteq E_{\mathrm{pred}}$ together with the conflict check. If $G_1 = \{e_1, e_2\}$ and $G_2 = \{e_3\}$, citing $e_3$ alone yields full path recall. Additional valid corroboration is not penalized merely for belonging to another path. These metrics are undefined for empty denominators or claims without an applicable support path. Independently adjudicated new paths are shared across methods.

**Independent evaluator protocol.** A frozen reference evidence graph and numerical/relational interpreter assess structured claims; blinded semantic adjudication checks their agreement with free text. The judge evaluates the stronger assertion when prose overstates a structured field. Valid unlisted support paths are independently adjudicated and applied symmetrically across methods. Evaluation records retain the assessed objects, ranking fields, and outcomes.

### H.3 SPFL SERIALIZATION RECORD

The supplementary serialization uses a UTC epoch with nonnegative integer millisecond offsets, 60,000-ms half-open windows, and packets sorted by timestamp, then capture ordinal. Windows with at most 64 packets retain them all; longer windows retain the first 32 and last 32, recording omitted counts and original ordinals. Direction is relative to an explicitly identified initiator. IP-byte and transport-payload-byte units remain separate. Entity aliases are consistent across views within a capture namespace; absent protocol or payload fields have an explicit missing state rather than an inferred value.

The four-packet record contains offsets 0, 5, 60,000, and 60,005 ms; directions out, in, out, in; IP bytes 120, 80, 119, 81; and payload bytes 80, 40, 79, 41. It yields two windows, a duration of 60,005 ms, 239 outbound IP bytes, 161 inbound IP bytes, and 240 payload bytes. The exact serialized record is included as `supplementary_records/records/spfl_s_record.json`. In compact field notation:

```
time_unit: ms; ip_byte_unit: byte; payload_byte_unit: byte
window[0]: [0,60000); retained ordinals: [0,1]
window[1]: [60000,120000); retained ordinals: [2,3]
duration: 60005; out_ip: 239; in_ip: 161; payload: 240
```

This notation displays values from the record; the JSON file retains the actual schema. The serialization record lists seven passing checks covering the stated representation properties and transformations of retained fields. The serialization summarizes raw payloads, records omitted packet counts, and preserves the original ordinals of retained packets. These representation checks complement the evaluation of investigation-level performance.

### H.4 CONTROLLED REPLAY, COVERAGE, AND STRICT SPLITS

Replay uses the method-independent six-query plan described in the main text, without acquiring further observations. Replanning and verifier budgets have matched ceilings; realized token counts are tracked separately. Each saved main-result score ranks a proposed determinate label; ordering is recomputed after group resampling to attain each coverage level. These scores serve as ranking statistics in the coverage analysis; their calibration is distinct from the release-gate evaluation. Figure 3 reports the full coverage curve and the complementary over-answering results. The 95% risk and over-answering intervals use paired event-group resampling.

Capture/execution isolation divides 120 parent units of 200 cases into 84/12/24 train/validation/test units. The 24 test parents, not their 4,800 derived cases, serve as resampling units. CTU-13 holdout removes all 2,000 nominal-source cases and rebuilds training/validation from the remaining 22,000; it covers Host/C2 tasks. The unseen controlled configuration test varies scheduling, naming, bandwidth, and inter-stage intervals across 150 new execution groups (1,200 cases). This evaluation remains limited to controlled-environment extrapolation.

### H.5 ABLATIONS, JOINT SUPPORT, AND SHORTCUT DIAGNOSTICS

The generic verifier jointly encodes the selected evidence set for cross-observation assessment. TextScope-S adds the same deterministic checks as Full, while Full learns explicit scope/set relations. Both use the same verifier training instances and hard negatives. Counterfactual pair scores require both labels to be correct and preserve dependence on the same underlying base event.

Figure 4(b) reports all five joint-evidence conditions. Negative and positive transitions are tested separately; more evidence need not imply more support.

### H.6 THRESHOLD SELECTION AND ITERATIVE RELEASE

The original four-class argmax counts remain in Figure 10. The supplementary ledger separately records supported/contradicted probabilities and deterministic-check outcomes, distinguishing the operational gate from argmax classification. All thresholds below retain the contradicted-probability gate and deterministic checks. False acceptance among 7,200 insufficient inputs and error among accepted candidates use different denominators.

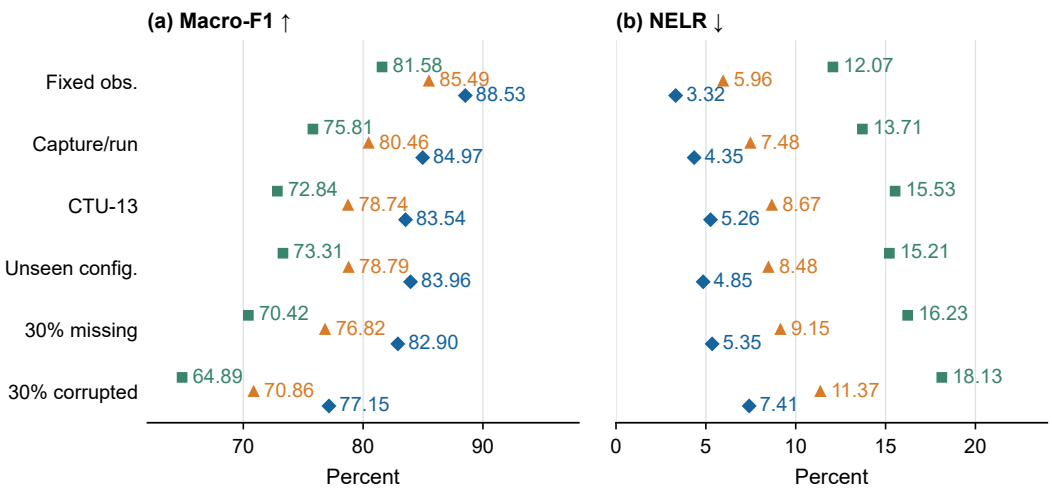



Figure 19: Complete supplementary protocol endpoints. Green squares: Memory-S; orange triangles: TextScope-S; blue diamonds: NetTrace-S. Cases/resampling units are 4,800/600 for replay and faults, 4,800/24 for capture/execution holdout, 2,000/250 for CTU-13, and 1,200/150 for unseen configurations. Parent units are used for capture/execution; event groups are used otherwise.

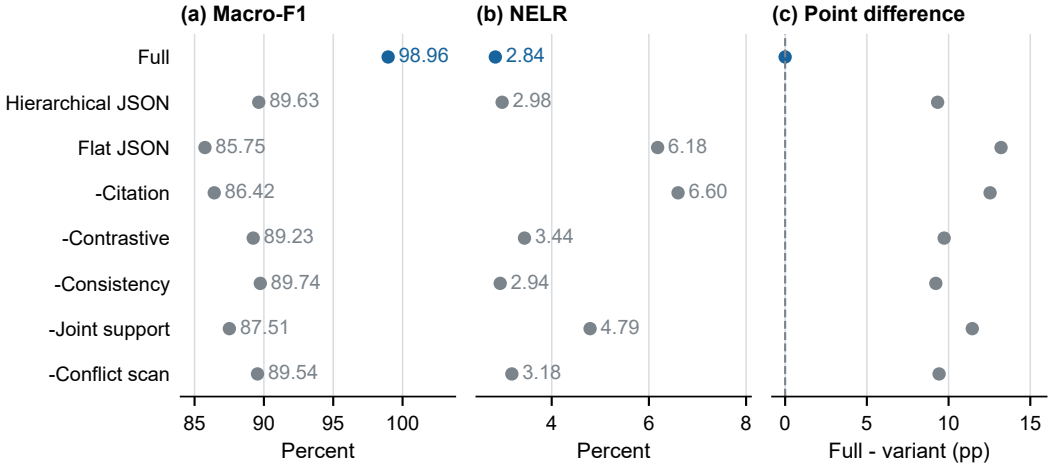



Figure 20: Supplementary ablations: absolute F1 and NELR, and Full-minus-condition aggregate F1 differences. Updated paired confidence intervals are unavailable; point differences alone do not establish statistical significance. All displayed values are retained in the figure-data CSVs.

Post-revision binary-support calibration in this pool yields a ten-bin ECE of 0.01133 and a Brier score of 0.04088. The validation pool, balanced challenge set, and dynamic pool differ in support prevalence and denominators. Threshold selection and the retained temperature characterize empirical release risk for these populations. Distribution-free guarantees would require a calibration framework and assumptions beyond this empirical evaluation.

### H.7 Faults, structural reconstruction, audit, and cost

The observed/external support analysis uses the same 6,000 candidates and 150 event groups for every method, regardless of whether it releases a claim. Missingness leaves the remaining content unchanged. Under corruption, visible content can support a claim that the independent reference facts do not support.

Figure 21(b–d) gives all three structural reconstruction metrics.

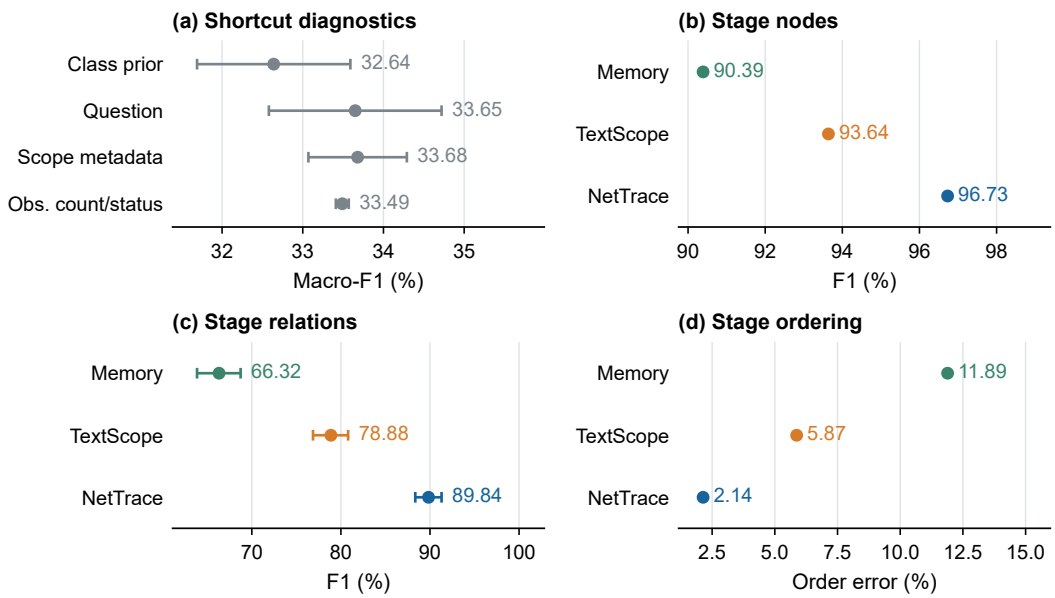



Figure 21: Shortcut diagnostics and structural reconstruction. (a) Restricted-input macro-F1 with seed SD; the four inputs test low-complexity shortcuts, not every possible shortcut. (b–d) Reconstruction on 480 target-reference cases: node F1, relation F1 with 95% CIs, and order error. Order error is conditional on connected, comparable stage pairs and must be interpreted alongside relation recovery.

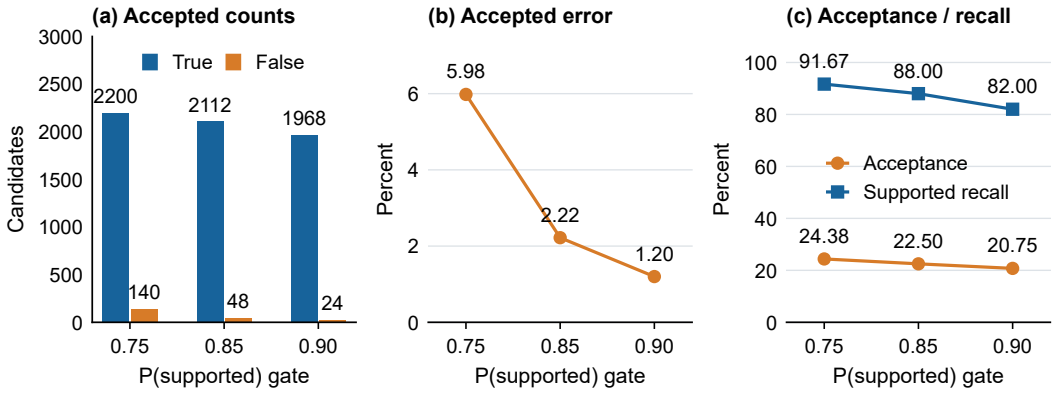



Figure 22: Release gates on the balanced 9,600-candidate challenge set. (a) True/false accepted counts; (b) error among accepted candidates; (c) candidate acceptance and supported recall. Deterministic checks and the contradicted-probability gate apply throughout. At 0.85, 48/2,160 gives 2.22% accepted error; 48/7,200 gives 0.67% false acceptance among insufficient inputs.

The human audit uses the actual number of substantive claim–citation units in each report, without padding reports to six claims. Withdrawn candidates are excluded from released claims. Uncited assertions are evaluated for USR and correctness but do not enter existing-citation NELR. Case groups remain intact across reviewers and methods during resampling. The ledger contains adjudicated claim-error counts and individual rating records. The adjudication rubric and study-governance documentation are maintained in the project records. Figure 6(a) reports all four methods, actual claim denominators, and case-group intervals. These independent human judgments are distinct from automatic NELR.

Paired preference compares NetTrace-S with TextScope-S on the same 240 cases, with three judgments per case. Wins/ties/losses are 472/90/158 (65.56/12.50/21.94%). The win-rate interval is [61.94,69.17]%; the 720 individual judgments are not independent cases. Agreement on nominal

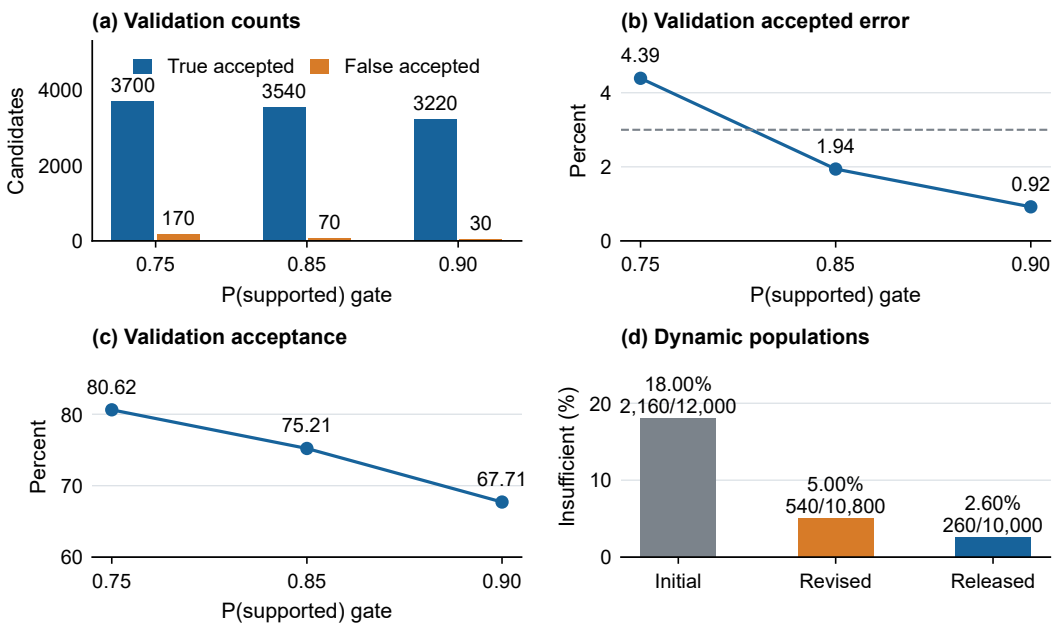



Figure 23: Validation selection and the separate dynamic pool. (a–c) Validation uses 4,800 candidates, including 3,840 supported. Acceptance is maximized subject to empirical accepted error of at most 3% (dashed line); the selected gate is 0.85 and temperature is 1.18. (d) Dynamic insufficient-claim rates, with numerator/denominator labels. Withdrawal and revision change the populations; these stage-wise rates are not paired correction probabilities.

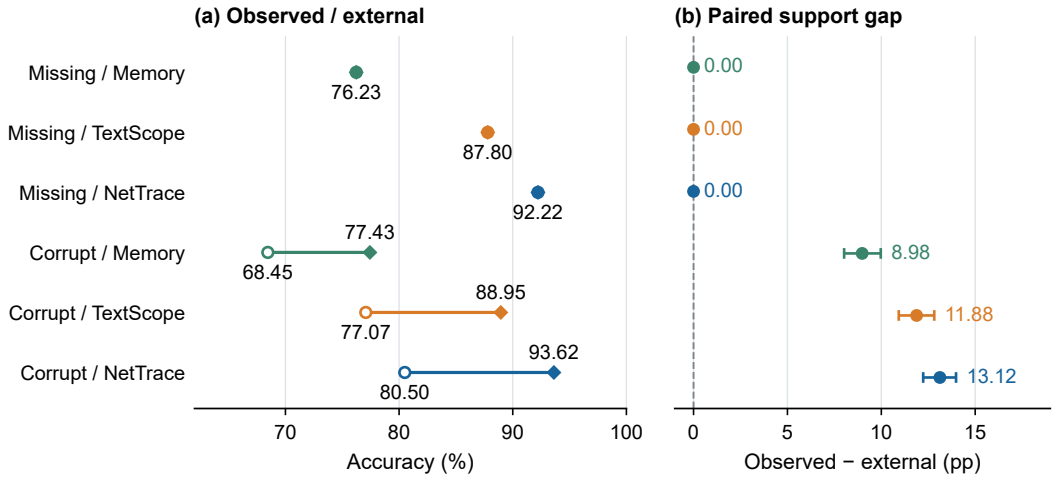



Figure 24: Observed-content and external-reference support on the same 6,000 candidates from 150 event groups. (a) Filled diamonds denote observed content; open circles denote external reference truth. Equal endpoints under missingness overlap. (b) Observed-minus-external gaps and paired event-group 95% CIs. The separation characterizes this corruption protocol.

support labels is $\alpha = 0.803$. The supplementary audit uses six evaluators; 240 denotes the number of audited cases, not raters.

Cost conversion uses the stated accounting assumptions: USD 2/GPU-hour and USD 0.2/CPU-service-hour. Runtime records provide the basis for this comparison. NetTrace-S's per-case cost is 41.42% higher than Memory-S's, alongside gains in trajectory success and evidence support. Initial-batch hardware/resource measurements are retained in Appendix F and apply to that batch. Figure 6(c) gives the complete supplementary cost comparison.

### H.8 A SCOPED INVESTIGATION RECORD

The supplementary case asks whether H0042's 12 MiB transfer to D0017 during 2026-01-12 02:00–02:10 is an authorized backup or unauthorized exfiltration. The initial report asserts "H0042 is exfiltrating sensitive data over TLS" and cites e01/e02/e03. The observations below distinguish citation existence from actual support for the claim.

| ID | Observed content | What the observation establishes |
|---|---|---|
| e01 | Traffic Stats: 12,582,912 bytes in the queried interval | Transfer volume and scoped endpoints; not content sensitivity or authorization. |
| e02 | Protocol Parser: TLS session | Protocol; the cited protocol fact alone does not establish sensitive-content exfiltration. |
| e03 | Historically good reputation, valid through 2025-12-31 | Historical reputation; neither valid for the January query nor evidence of transfer authorization. |
| e04 | Backup BK17 authorization: subject, destination, time, approved manifest M17 | A scoped authorization relation to be matched with the observed transfer. |
| e05 | BK17 execution receipt matching interval, volume, and content digest | Execution link that, jointly with e01/e04, supports authorized backup. |

Table 5: Supplementary investigation record. The final two observations address the evidence gap identified by the initial verification.

The initial verifier probabilities $(S, P, U, C) = (.12, .75, .10, .03)$ fail the release gate. The claim lacks the required sensitivity/authorization support, and the cited reputation is out of scope. Further investigation acquires e04 and e05. The final title is "02:00–02:10 transfer investigation", and the decision is `alternative`. Its substantive claims are: (i) the scoped transfer is 12 MiB (e01); (ii) the transfer matches BK17's authorization and execution, supporting the authorized-backup explanation (e01/e04/e05). This conclusion applies specifically to H0042's queried transfer. The revised backup claim has probabilities $(.94, .04, .01, .01)$ and passes re-verification.

The correction changes the predicate and support path rather than merely hedging "exfiltrating sensitive data". The record illustrates how further observations support a different predicate and provide the evidence path for it.

### H.9 SUPPLEMENTARY MATERIAL AND PROVENANCE

The submission package contains supplementary summaries, compressed case/candidate/rating ledgers, and SPFL records. An accompanying author correction records Full macro-F1 of 98.96; original outcome ledgers remain unchanged. Human-audit ledgers contain report assignments, individual judgments, and adjudicated claim counts; case counts are distinct from evaluator counts.

These materials support inspection of individual outcomes, aggregate tables, and the documented analyses. The recorded 14 ledger checks and seven serialization checks assess artifact consistency within their stated scope.

The project records retain execution and evaluation resources, including training-environment and model/configuration materials, traffic captures, tool snapshots, and prompts. They also retain the evaluator implementation, claim grammar, annotation and adjudication records, method-specific confidence-score mappings, and optimizer-search records. Together, these resources document the reported pipeline and its evaluation. Optimizer-search records support training-resource accounting for the separately adapted policy comparison. Study-governance documentation is maintained with the project records. This manuscript revision describes the experiment records from existing runs.

## I EXACT FIGURE DATA AND STATISTICAL CONVENTIONS

Every numerical result table is represented in the figures and accompanying `figure_revision/data` CSV files. The mapping in `figure_revision/figure_manifest.json` identifies all 25 original result tables. CSVs retain the reported precision, counts, and available intervals; the corrected Full F1 has no accompanying uncertainty estimate. Initial-batch figure data are also retained in `visualization/csv`.

SD denotes variability across three training seeds. Human-rating intervals are event-level bootstrap 95% confidence intervals; supplementary paired intervals use the resampling units specified in the corresponding protocols. Initial-batch risk–coverage and cost results retain their originally reported point estimates. Plot lines connect reported operating points; the visualizations introduce no additional uncertainty estimates.