Smaller Models Together Make a Difference: Orchestrating Open-Weight Small Language Models for Expert Malware Analysis

Anonymous

Abstract

Malware analysis demands rapid interpretation of complex detonation reports spanning filesystem, network, and process behaviours. While large language models (LLMs) demonstrate impressive capabilities for technical artifact interpretation, their proprietary nature, opacity, and escalating costs motivate exploration of open-weight alternatives. This paper investigates whether orchestrated ensembles of small language models (SLMs), i.e. models deployable on consumer-level hardware with acceptable latency, can match or exceed single LLM performance on malware analysis tasks. We establish baseline performances by testing 11 open-weight SLMs (with 0.6B to 8B parameters), three cyber security pre-trained models, and six frontier LLMs as solo agents on Meta's CyberSecEval Malware Analysis benchmark. We then design and evaluate four orchestration architectures that embody complementary hypotheses for capability amplification: a specialised multi-agent pipeline, an adversarial debate framework, a hierarchical consultation system, and a novel hybrid architecture that combines evidence-grounded pipelines with adversarial debate reasoning. Our results demonstrated that orchestrated SLM ensembles can outperform frontier LLMs, with the hybrid system (Qwen3-4B with Foundation-Sec-8B) achieving 35.30% overall accuracy, surpassing the strongest single LLM baseline (Gemini 3 Pro Preview at 34.77%). Qualitative analysis of samples taken from the wild (UNC5142 and Lumma Stealer) confirms the hybrid system's ability to correct reasoning errors on novel evasion techniques like EtherHiding and ClickFix. These findings suggest that hybrid orchestration of open-weight SLM ensembles offers a viable path to transparent, auditable, and cost-effective malware analysis systems that can outperform proprietary frontier models.

1 Introduction

Malware analysis remains a critical bottleneck in cyber security operations. Analysts must rapidly triage suspicious samples, interpret complex detonation reports spanning filesystem modifications, network communications, and process behaviours, and assess threat severity under time pressure. Traditional static and dynamic analysis pipelines generate rich telemetry from sandboxed execution environments, but extracting actionable intelligence from these multi-faceted reports demands expert knowledge of malware techniques, operating system internals, and attack frameworks such as MITRE ATT&CK [41]. As malware sophistication and volume continue to escalate, the need for automated assistance that can comprehend, reason about, and summarise behavioural evidence has become acute.

Recent advances in large language models (LLMs) have demonstrated impressive capabilities for interpreting technical artifacts and answering domain-specific questions [8]. However, frontier LLMs remain proprietary, opaque, and costly to deploy [18], raising concerns about data privacy, vendor lock-in [38], and the reproducibility of security-critical decisions [23]. At the same time, the open-weight model ecosystem has matured rapidly [22], with increasingly capable small language models (SLMs), i.e., language models that can be deployed on consumer-level devices with acceptable performance to make them still practical [3], emerging as viable alternatives for specialised tasks. Following Belcak et al. [3], we adopt a working definition in which an SLM is any language model that can be deployed on a common consumer device while serving a single user's agentic workloads with practically acceptable latency; an LLM is any model that does not meet this criterion. In line with their guidance, and focusing on the 2025 hardware landscape, we treat most models below roughly 10 billion parameters as SLMs in this work.

While individual SLMs often lag behind frontier LLMs on complex reasoning tasks, orchestration strategies (such as multi-agent systems, debate frameworks, and hierarchical consultation patterns) offer a path to improve SLMs' collective capabilities. Multi-agent architectures have shown promise in decomposing complex tasks into specialised subtasks [48], debate-style interactions can expose reasoning flaws and improve answer quality [11], and hierarchical consultation allows general-purpose models to seek targeted expertise [24].

Open-weight SLMs such as *Mistral*, *Phi*, and *Llama* variants can be self-hosted, audited, and fine-tuned for domain specificity, while cyber security pre-trained models offer specialised knowledge of malware techniques and defensive concepts [14]. Ensembles provide a path to fuse diverse inductive biases while offering defence-in-depth through redundancy, yet the extent to which orchestrated SLM ensembles can close the performance gap to frontier LLMs on malware analysis tasks remains an open question.

Malware analysts increasingly demand transparent reasoning, reproducible artifacts, and explicit risk controls when interpreting detonation reports, identifying malicious behaviours, and assessing threat severity. These requirements align well with the transparency and auditability afforded by open-weight models, but only if orchestrated SLM systems can deliver competitive accuracy on real-world malware analysis benchmarks. We centre the following question: "Can an orchestrated ensemble of open-weight small language models deliver comparable or superior malware analysis capability to a single large model?" Addressing this requires a holistic assessment that spans detection accuracy, behavioural interpretation quality across difficulty tiers, and the identification of which orchestration patterns provide the largest performance gains.

Contributions. This work makes the following key contributions:

1. **Systematic comparison and implementation of three orchestration architectures** (agentic, debate, and consult) that instantiate complementary hypotheses for the amplification of SLM capability, evaluated on the CyberSecEval Malware Analysis benchmark across 11 open-weight SLMs and multiple cyber-specialised models.

2. **A novel hybrid orchestration system for malware analysis** that combines evidence-grounded multi-agent pipelines with adversarial debate reasoning, demonstrating that systematic evidence collection followed by structured peer critique yields complementary performance gains across all difficulty tiers. This hybrid system represents the fourth orchestration architecture, which is then compared against the first three.

3. **Empirical evidence** demonstrating that orchestrated SLM ensembles can exceed frontier LLM performance, with the hybrid system (Qwen3-4B paired with Foundation-Sec-8B) achieving 35.30% overall accuracy compared to the strongest LLM baseline (Gemini 3 Pro Preview at 34.77%), and detailed ablation studies revealing that the combination of tool-augmented evidence collection and debate-based peer critique provides synergistic performance gains.

4. **Qualitative case studies** on malware samples from the wild (UNC5142 EtherHiding campaign and Lumma Stealer with ClickFix), demonstrating that the hybrid system is capable of correcting *reasoning errors* on novel evasion techniques that defeat single-model approaches.

The remainder of this paper is organised as follows. Section 2 reviews related work on LLMs for cyber security, multi-agent systems, debate frameworks, and small language models. Section 3 describes our methodology, including the four orchestration architectures and the evaluation benchmark. Section 4 presents experimental results, comparing single-model baselines with orchestrated systems and reporting ablation studies. Section 5 discusses implications for operational deployment, and Section 6 concludes with directions for future work.

2 Related Work

Our work builds on four converging research threads: LLMs for cyber security and malware analysis, multi-agent LLM systems, debate frameworks for improving LLM reasoning, and the emerging capabilities of small language models.

2.1 LLMs for Cyber Security and Malware Analysis

Large language models have been increasingly applied to cyber security tasks, with recent surveys documenting their use across vulnerability detection, malware analysis, threat intelligence, and network intrusion detection [49]. LLMs demonstrate substantial potential for interpreting malicious artifacts: Patsakis et al. [33] showed that LLMs achieve 69.56% accuracy in extracting malicious URLs from obfuscated code in real-world campaigns like Emotet, while outperforming symbolic analysis in bypassing common evasion techniques. Al-Karaki et al. [1] provide a comprehensive framework for LLM-based malware detection, identifying key challenges including dataset limitations and the need for domain-specific fine-tuning. The CyberSecEval benchmark suite [45] provides standardised evaluation protocols for assessing LLM capabilities on security tasks, including the malware analysis benchmark we use in this study. While these works establish that LLMs can assist with security analysis, they primarily evaluate single models in isolation; our work extends this line by investigating whether orchestrated ensembles of smaller models can match or exceed single-model performance.

2.2 Multi-Agent LLM Systems

Multi-agent architectures decompose complex tasks across specialised LLM agents that communicate and collaborate. Wu et al. [48] introduced AutoGen, a framework enabling customisable agents with flexible conversation patterns for tasks spanning coding, mathematics, and decision-making.

Subsequent work has demonstrated that multi-agent orchestration provides value through deterministic quality and consistency rather than speed alone [44]. Liu et al. [25] propose dynamic agent networks that optimise team composition based on task requirements, while hierarchical frameworks like AgentOrchestra use central planning agents that delegate to specialised sub-agents. These systems have shown success in software development, question answering, and enterprise operations. Our agentic and consult systems draw on these principles, adapting multi-agent workflows specifically for malware analysis where specialised agents handle ingestion, enrichment, evidence mining, and verification stages.

2.3 LLM Debate Frameworks

Debate-style orchestration, where multiple LLM agents critique each other's reasoning, has emerged as an effective technique for improving factuality and complex reasoning. Du et al. [11] demonstrated that multi-agent debate improves mathematical and strategic reasoning by exposing flaws through adversarial exchange. Recent extensions include the Mixture-of-Agents framework [46], which organises proposer and aggregator agents in structured layers to achieve state-of-the-art results using open-source models, and adaptive heterogeneous debate [50], which achieves 4–6% accuracy gains over standard debate through dynamic agent weighting. Chen et al. [6] show that round-table consensus among diverse LLMs improves reasoning on complex benchmarks. These works establish that structured debate improves reasoning quality, but also reveal a trade-off: excessive debate rounds can introduce tangential information that degrades performance on straightforward questions. Our hybrid system addresses this limitation by grounding debate agents in systematically collected evidence, preventing the drift phenomenon while preserving the benefits of peer critique.

2.4 Small Language Models

The capabilities of small language models, typically defined as models deployable on consumer hardware with acceptable latency [3], have advanced rapidly. Lu et al. [26] survey SLMs in the 100M–5B parameter range, documenting competitive performance on commonsense reasoning, mathematics, and domain-specific tasks when compared to much larger models. Recent work demonstrates that well-chosen SLMs can outperform frontier LLMs including GPT-4 variants in specific use cases, particularly when enhanced through fine-tuning, prompt engineering, or ensemble techniques. SLMs offer practical advantages including lower inference latency, reduced costs, privacy benefits through localised deployment, and suitability for resource-constrained environments. Our work extends this literature by demonstrating that orchestrated SLM ensembles can exceed frontier LLM performance on the specialised domain of malware analysis, achieving this through architectural innovation rather than parameter scaling.

3 Methodology

Our experimental methodology evaluates whether orchestrated ensembles of open-weight SLMs can match or exceed the malware analysis performance of single large models. We begin by establishing baseline performance: we test a diverse collection of models (including general-purpose open-weight SLMs, large proprietary LLMs, and cyber security pre-trained models) as solo agents on the CyberSecEval Malware Analysis benchmark. This benchmark exercises Hybrid Analysis detonation reports[1] through multi-topic, multi-difficulty multiple-choice questions, providing strict accuracy metrics stratified by difficulty tier (Easy, Medium, Hard). The solo-model baselines reveal the performance ceiling we aim to approach or exceed through orchestration, and they establish which model architectures and parameter scales perform best on malware analysis tasks when operating independently.

We test general-purpose open-weight SLMs spanning 0.6B to 8B parameters (Qwen3-0.6B [36], Llama-3.2-1B [28], Qwen2.5-1.5B-Instruct [34], DeepSeek-R1-Distill-Qwen-1.5B [10], SmolLM2-1.7B [19], Phi-3.5-mini-instruct [30], Gemma-3-4B-IT [16], Qwen3-4B [37], Qwen2.5-Coder-7B-Instruct [35], Ministral-8B [32], and Llama-3.1-8B-Instruct [27]), cyber security pre-trained models (DeepHat-V1-7B [9], Foundation-Sec-8B-Instruct [12], and Llama-Nemotron-70B [47]), and large proprietary LLMs (Gemini 3 Pro Preview [13], Claude Opus 4.5 [2], GPT-5.2 [43], DeepSeek V3.2 [42], and Llama 4 variants [29]). All models are sourced from openly available Hugging Face checkpoints where applicable, ensuring reproducibility. To ensure fair evaluation, we conducted a rigorous decontamination audit of the expert model (Foundation-Sec-8B), confirming no temporal or n-gram leakage of the CyberSecEval benchmark data (details in Appendix A). All SLM inference experiments were conducted on a single NVIDIA RTX 4090 GPU (24 GB VRAM); the hybrid system requires approximately 6 GB VRAM with 4-bit quantisation.

After establishing solo-model baselines, we design and implement four distinct orchestration architectures: a specialised multi-agent pipeline (agentic system), an adversarial debate framework (debate system), a hierarchical consultation system (consult system), and a hybrid architecture that combines evidence-grounded pipelines with adversarial debate reasoning (hybrid system). Each architecture embodies a different hypothesis for capability amplification: specialisation through task decomposition, peer critique through adversarial reasoning, expert guidance through hierarchical consultation, and synergistic combination of evidence collection with structured debate. We then evaluate these orchestration systems

[1] It is worth to make it clear that in this work, we take malware analysis detonation report rather than malware binaries as the input.

by running representative SLMs through each architecture on the same malware analysis benchmark. Finally, we compare the orchestrated system performance against the solo-model baselines to quantify the performance gains attributable to orchestration and to identify which architectural patterns provide the largest improvements across different difficulty tiers and model sizes.

3.1 Agentic System

The first architecture uses a multi-agent workflow designed to mimic human malware triage, as illustrated in Figure 1. The system decomposes the analysis process into six specialised stages orchestrated by a central controller. An *ingestion agent* prepares the workspace by retrieving and structuring the relevant malware report. An *enrichment agent* augments this data by fetching external context, such as MITRE ATT&CK technique descriptions. To locate specific indicators of compromise, a *tool-search agent* generates and executes sandboxed search commands, while an *evidence miner* extracts and validates supporting snippets from the report's content. These retrieved artefacts are synthesised by a *reasoning agent* to formulate an answer, which is finally subjected to a quality gate by a *verifier agent* before being returned. All agents operate on a shared state object that ensures every decision can be traced back to its supporting evidence.

3.2 Debate System

The second architecture implements a debate-style orchestration that uses reasoning between two LLM agents to answer the benchmark questions. The system is explicitly prompted to critique and challenge each other's conclusions. The system instantiates two primary agents (Agent A and Agent B) as separate model instances, optionally with different models or configurations. For each question, the two agents engage in a number of debate rounds, exchanging arguments and revising their answers in light of their opponent's position. In each round, both debating agents receive the original task prompt together with the full debate history, analyse their opponent's previous response and final answer, compare selected options, and produce a new response that contains both free-form reasoning and a structured final answer. A control loop manages the exchange, identifies gaps or contradictions, and may inject external evidence from URLs mentioned in the debate (e.g., MITRE ATT&CK technique pages) before providing guidance for the next round. After the final round, the system synthesises the full debate trace into an adjudicated answer. The implementation relies on robust output parsing to ensure schema compliance, automatic answer comparison to surface disagreements, and comprehensive trace logging. It is particularly suited to difficult multi-choice malware questions where subtle reasoning errors are common: the debate structure encourages explicit justification and exposes weaknesses that might be missed in a single agent's reasoning.

3.3 Consult System

The third architecture uses a hierarchical consult-style orchestration in which a "tested" general-purpose SLM under evaluation can query a specialised "expert" model that has been explicitly pre-trained on cyber security data. The consult system coordinates two agents across multiple consultation rounds: a tested agent that owns the task and an expert agent that provides focused advice. The tested agent receives the full benchmark prompt, may pose at most one explicit question per round, and returns its own analysis and formatted answer. The expert agent (a model fine-tuned on security corpora) only ever sees the extracted question, not the full task, and responds with concise explanations about cyber security concepts, malware techniques, and related background. For each round, the tested agent is prompted with the original question, the consultation history, and all previous expert responses; it may ask a new question, update its reasoning, and refine its answer. The system extracts questions primarily from explicit markers, with a fallback heuristic that selects the first sufficiently long interrogative sentence if markers are missing. This design probes whether access to an on-demand domain expert can systematically upgrade general-purpose models without giving the expert full control over the task. The framework supports both synchronous and asynchronous consultation patterns, accumulates expert guidance across rounds, and is well suited to scenarios where a compact model lacks deep domain knowledge (for instance, interpreting specific MITRE ATT&CK techniques or low-level malware behaviours) but can close the gap to larger models when granted structured access to a pre-trained cyber security expert.

We formalise the consult interaction as a hierarchical loop. Let q_{task} be the full benchmark prompt. At round t, the tested agent A produces a rationale $r_t^{(A)}$ and potentially a specific query q_{sub} for the expert. An extraction function ψ isolates the explicit question from the agent's output. The expert agent E (which does not see q_{task}) provides a domain-specific explanation e_t. The tested agent then updates its state using the accumulated history of expert advice $H_E = \{(q_{\text{sub}}^{(i)}, e_i)\}_{i=1}^{t}$:

$$q_{\text{sub}}^{(t)} = \psi(r_t^{(A)}) \tag{1}$$

$$e_t = M_E(q_{\text{sub}}^{(t)}) \tag{2}$$

$$r_{t+1}^{(A)} = M_A(q_{\text{task}}, H_E) \tag{3}$$

This formalisation highlights that M_E operates as a "stateless oracle" relative to the main task, distinguishing this architecture from the state-sharing debate agents.

3.4 Hybrid System

The fourth architecture combines the evidence retrieval capabilities of the agentic system with the adversarial reasoning

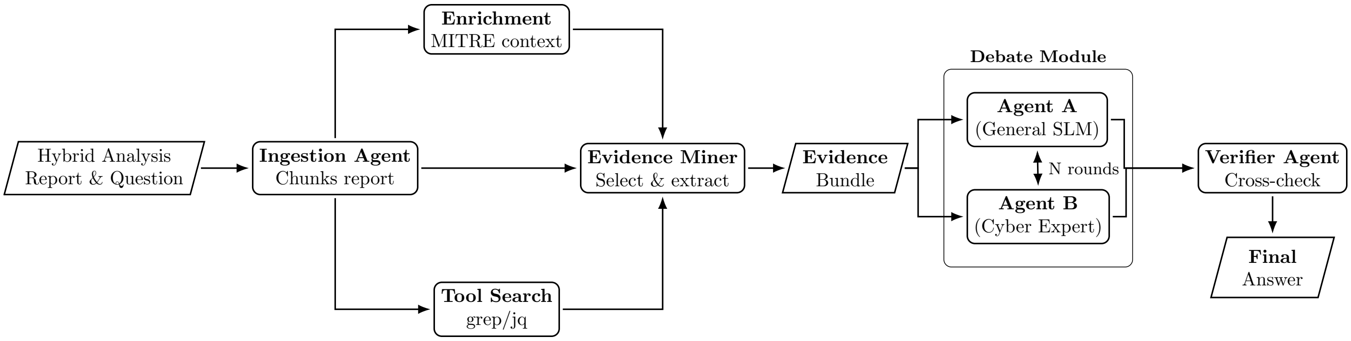

Figure 1: Hybrid orchestration architecture for malware analysis question answering. The system operates in two phases: an *evidence collection phase* (left) where ingestion, enrichment, tool-search, and evidence mining agents systematically extract and validate supporting evidence from malware reports; and a *debate reasoning phase* (right) where two agents (a general-purpose SLM (Agent A) and a cyber-specialised model (Agent B)) engage in N rounds of structured debate grounded in the collected evidence. A verifier agent validates the final answer against the evidence bundle.

of the debate system, as illustrated in Figure 1. The hybrid system operates in two distinct phases designed to address the complementary weaknesses of its component architectures.

Evidence Collection Phase. The first four stages of the agentic pipeline (ingestion, enrichment, tool-search, and evidence mining) execute to collect and validate supporting evidence from the malware report. The ingestion agent parses the Hybrid Analysis JSON report into manageable chunks; the enrichment agent fetches relevant MITRE ATT&CK technique descriptions; the tool-search agent generates and executes targeted grep/jq commands; and the evidence miner extracts and validates supporting snippets with confidence scores. This phase produces a structured evidence bundle containing report chunks, external enrichments, tool-search results, and validated evidence items.

Formally, let $\mathcal{R}$ denote the raw Hybrid Analysis report (JSON dossier). The *evidence bundle* $\mathcal{B}$ is the union of three extraction outputs: $\phi_{\text{chunk}}(\mathcal{R})$ (ingestion agent's text chunks), $\phi_{\text{enrich}}(\mathcal{R})$ (enrichment agent's external context, e.g., MITRE ATT&CK descriptions), and $\text{Exec}(\phi_{\text{tool}}(\mathcal{R}))$ (results of sandboxed commands generated by the tool-search agent). The evidence miner applies a filtering function F_τ based on a confidence threshold τ to produce the final validated evidence set E_{final}, which serves as the immutable ground truth for the debate phase:

$$\mathcal{B} = \phi_{\text{chunk}}(\mathcal{R}) \cup \phi_{\text{enrich}}(\mathcal{R}) \cup \text{Exec}(\phi_{\text{tool}}(\mathcal{R})) \tag{4}$$

$$E_{\text{final}} = \{e \in \mathcal{B} \mid \text{Confidence}(e) > \tau\} \tag{5}$$

Debate Reasoning Phase. Rather than passing evidence to a single reasoning agent, the hybrid system instantiates two debate agents that receive the collected evidence alongside the original question. Agent A is a general-purpose SLM (Qwen3-4B), selected based on its strong baseline performance across all orchestration systems; Agent B is a cyber-specialised model (Foundation-Sec-8B), selected to provide complementary domain expertise while maintaining capacity balance (within 2× parameter ratio). The agents engage in N rounds of structured debate following the protocol described in Section 3.2, but with a critical modification: agents are explicitly instructed to cite collected evidence when defending their positions and to challenge claims that lack evidential support.

We model the debate as a Markov process over t rounds. At round t, the response $r_t^{(A)}$ of Agent A is conditional on the original question q, the evidence bundle E_{final}, the debate history H_{t-1}, and the opponent's previous argument $r_{t-1}^{(B)}$:

$$r_t^{(A)} = M_A\left(q, E_{\text{final}}, H_{t-1}, r_{t-1}^{(B)}\right) \tag{6}$$

Unlike standard debate, the hybrid system enforces a *grounding constraint* via the verifier: a response r is valid if and only if every claim $c \in r$ maps to a supporting snippet in E_{final} with similarity $S(c,e)$ exceeding threshold λ:

$$\text{Valid}(r) \iff \forall c \in r,\ \exists e \in E_{\text{final}} \text{ s.t. } S(c,e) \geq \lambda \tag{7}$$

This constraint prevents the "drift" phenomenon observed in pure debate, where agents introduce tangential information that degrades easy-question accuracy.

Verification Phase. After debate concludes, the verifier agent validates the debate conclusion against the evidence bundle, checking that selected answer options have supporting evidence with confidence above a threshold and that the reasoning chain is internally consistent.

Model Selection Methodology. The hybrid system's model pairing follows a systematic selection methodology derived

Table 1: Accuracy percentages for each model on the Malware Analysis benchmark, shown by difficulty level; average number of parsable responses (‘avg n’) per difficulty column, and total parsable responses per model (row).

Model	Params	Easy (n=451)	Medium (n=136)	Hard (n=22)	Overall (n=609)
Large language models (LLMs)					
Llama 4 Scout	109B	25.50%	14.00%	0.00%	22.01%
Llama 4 Maverick	400B	31.00%	20.50%	13.64%	28.03%
DeepSeek V3.2	685B	33.50%	22.00%	13.64%	30.21%
GPT-5.2	—	34.50%	23.50%	20.45%	31.54%
Claude Opus 4.5	—	36.25%	24.75%	21.59%	33.15%
Gemini 3 Pro Preview	—	**38.00%**	**26.00%**	**22.73%**	**34.77%**
Cyber security language models					
DeepHat-V1-7B	7B	16.41%	11.03%	4.55%	14.78%
Foundation-Sec-8B	8B	22.65%	13.45%	4.55%	19.96%
Llama-3.1-Nemotron-70B	70B	**24.78%**	**18.00%**	4.55%	**22.54%**
Small language models (SLMs)					
Qwen3-0.6B	0.6B	12.94%	6.61%	0.00%	11.05%
Llama-3.2-1B	1B	10.17%	5.88%	0.00%	8.87%
Qwen2.5-1.5B-Instruct	1.5B	11.72%	7.35%	0.00%	10.34%
DeepSeek-R1-Distill-Qwen-1.5B	1.5B	12.42%	8.09%	4.55%	11.17%
SmolLM2-1.7B	1.7B	10.86%	6.62%	0.00%	9.52%
Phi-3.5-mini-instruct	3.5B	15.74%	10.29%	4.55%	14.12%
Gemma-3-4B-IT	4B	16.62%	11.03%	4.55%	14.94%
Qwen3-4B	4B	**18.40%**	**12.50%**	4.55%	**16.58%**
Qwen2.5-Coder-7B-Instruct	7B	12.70%	9.19%	4.55%	11.66%
Ministral-8B	8B	11.75%	8.82%	4.55%	10.84%
Llama-3.1-8B-Instruct	8B	13.24%	9.56%	4.55%	12.15%

from our empirical findings: (1) *capacity balance*: debate performance degrades when agents differ by more than 4× in parameter count; (2) *complementary expertise*: the best debate configurations pair general-purpose models with cyber-specialised experts rather than pairing two models of the same type; (3) *strong baseline performance*: models with higher solo baselines perform better in orchestration. Qwen3-4B satisfies all criteria: it achieves the highest solo SLM baseline (16.58%), the best agentic performance (25.11%), and the best debate performance (24.13%), while Foundation-Sec-8B provides cyber-specialised expertise within the optimal capacity ratio.

3.5 Benchmark and Evaluation Protocol

We ground our empirical study in the CyberSecEval Malware Analysis benchmark, especially its CyberSOCEval test suite [8], which exercises Hybrid Analysis [7] detonation reports through multi-topic, multi-difficulty multiple-choice scoring. This benchmark provides strict accuracy, partial-credit Jaccard metrics, and per-attack-family breakdowns, enabling controlled comparisons between orchestrated SLM ensembles and single LLM baselines. Each question in the benchmark links a SHA256 hash to the corresponding Hybrid Analysis JSON dossier under the `data/hybrid-analysis/` tree, exposing MITRE ATT&CK mappings, process inventories, registry edits, extracted payloads, and network telemetry. The question schema captures the natural-language prompt, the labelled answer options (often multi-select), and metadata for topic (e.g., persistence, evasion, risk assessment), difficulty tier, and malware family so that we can slice performance by analytic intent. Because the ground-truth option set is isolated from the model inputs, systems must locate, cross-reference, and reconcile the relevant report fragments rather than memorising expected outputs [8].

The dataset spans evidence-retrieval queries (e.g., which persistence artefacts were dropped), behavioural interpretation (why a sample contacts its C2 infrastructure), risk scoring, and holistic system-interaction audits that require combining filesystem, registry, and network traces. Its multi-label structure punishes both omissions and hallucinations, making it well suited to measuring whether ensembles can maintain precision while covering the entire evidence set referenced in a report. Running the released harness yields strict exact-match scores, Jaccard-style partial credit, and response-parsing error counts, all stratified by topic, difficulty, and malware family. These artefacts (responses, rationales, and detailed JSON metrics) let us inspect how orchestration choices affect evidence alignment and completeness compared with monolithic LLMs.

Let $D = \{\text{Easy}, \text{Medium}, \text{Hard}\}$ denote the difficulty tiers, N_d the number of questions in tier d, and Acc_d the model's accuracy on that tier; $N_{\text{total}} = 609$. *Overall weighted accuracy* is:

$$\text{Acc}_{\text{overall}} = \sum_{d \in D} \frac{N_d}{N_{\text{total}}} \cdot \text{Acc}_d \tag{8}$$

Substituting the CyberSecEval constants ($N_{\text{Easy}} = 451$, $N_{\text{Med}} = 136$, $N_{\text{Hard}} = 22$):

$$\text{Acc}_{\text{overall}} = \frac{451 \cdot \text{Acc}_{\text{Easy}} + 136 \cdot \text{Acc}_{\text{Med}} + 22 \cdot \text{Acc}_{\text{Hard}}}{609} \tag{9}$$

Because Easy questions constitute 74% of the dataset, performance on this tier dominates the overall score, meaning that a model's ability to handle straightforward evidence-retrieval queries has a greater impact on its aggregate ranking than proficiency on the rare, complex Hard questions.

4 Results

This section presents the experimental findings of our study, beginning with an evaluation of single-model baselines across various parameter scales to establish a performance ceiling. We then detail the performance of the four orchestrated architectures (agentic, debate, consult, and hybrid) to quantify the capability gains achieved through different ensemble strategies. The section concludes with qualitative case studies of real-world malware samples from the wild, demonstrating the hybrid system's ability to identify and reason through novel evasion techniques.

4.1 Single-Model Baselines

Table 1 contrasts representative LLM and SLM baselines across benchmark difficulty tiers, establishing the empirical gap our orchestration aims to close.

The baseline profiling reveals that parameter count alone does not guarantee malware-analysis capability: Qwen3-4B achieves 16.58% overall accuracy, outperforming all tested 7–8B models, while sub-2B models cluster tightly in the 8.87–11.17% range with minimal performance differentiation. Based on this profiling, we selected four representative SLMs for detailed orchestration experiments: Qwen3-0.6B, Phi-3.5-mini-instruct (3.5B), Qwen3-4B, and Ministral-8B. These models span four distinct size classes (sub-1B, mid-range 3.5B, mid-range 4B, and 8B), representing diverse architectural families (Qwen, Phi, and Mistral variants), and demonstrate strong baseline performance within their respective categories: Qwen3-0.6B achieves the highest overall accuracy (11.05%) among sub-1B models, Phi-3.5-mini-instruct delivers competitive mid-tier performance (14.12%), Qwen3-4B achieves the best overall performance among all open-weight SLMs (16.58%), and Ministral-8B provides an 8B reference point (10.84%). This selection enables us to assess whether orchestration benefits generalise across model scales and whether architectural diversity influences ensemble effectiveness.

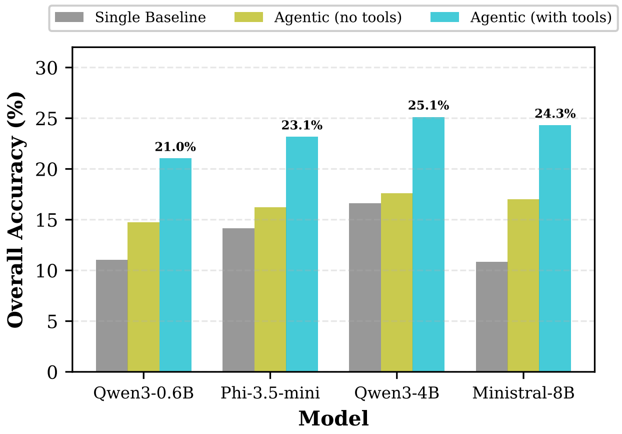

Figure 2: Agentic system ablation showing the impact of command-line tools across four representative SLMs. Tool access provides the largest performance boost for all models. Qwen3-4B with tools achieves the best overall performance (25.11%), exceeding the strongest single LLM baseline.

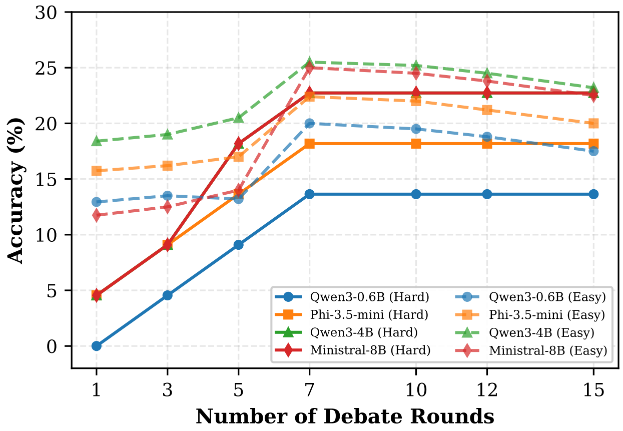

Figure 3: Debate system performance as a function of debate rounds (1–15 rounds), where each SLM debates with Foundation-Sec-8B. Results show consistent improvement on hard questions (solid lines) that plateaus after 7–10 rounds, while easy questions (dashed lines) exhibit clear degradation with increased rounds, declining from peak performance at rounds 7–10 to lower accuracy by round 15. Qwen3-4B achieves the highest performance across debate rounds.

4.2 Orchestrated Systems Performance

Table 2 presents the performance of all three orchestrated systems against their single-model baselines, demonstrating the gains achieved through orchestration.

Table 2: Comparison of orchestrated systems against single-model baselines on the Malware Analysis benchmark. Orchestrated systems are tested with four representative SLMs: Qwen3-0.6B, Phi-3.5-mini-instruct (3.5B), Qwen3-4B, and Ministral-8B. Agentic system uses a single SLM with sandboxed command-line tools; Debate pairs each SLM with Foundation-Sec-8B (7 rounds); Consult uses each SLM as tested agent consulting Foundation-Sec-8B as cyber expert (7 rounds); Hybrid combines agentic evidence collection with debate reasoning (7 rounds).

System	Model	Easy (n=451)	Medium (n=136)	Hard (n=22)	Overall (n=609)
Single-model baselines					
	Best open-weight SLM (Qwen3-4B)	18.40%	12.50%	4.55%	16.58%
	Best single LLM (Gemini 3 Pro Preview)	**38.00%**	**26.00%**	**22.73%**	**34.77%**
Agentic (with tools)					
	Qwen3-0.6B	22.62%	17.65%	9.09%	21.02%
	Phi-3.5-mini-instruct	24.61%	19.85%	13.64%	23.15%
	Qwen3-4B	**26.50%**	**21.30%**	**20.00%**	**25.11%**
	Ministral-8B	25.71%	20.59%	18.18%	24.30%
Debate (7 rounds, paired with Foundation-Sec-8B)					
	Qwen3-0.6B	20.00%	14.80%	13.64%	18.60%
	Phi-3.5-mini-instruct	22.40%	17.70%	18.18%	21.20%
	Qwen3-4B	**25.50%**	**19.80%**	**22.73%**	**24.13%**
	Ministral-8B	25.00%	19.10%	**22.73%**	23.60%
Consult (7 rounds, paired with Foundation-Sec-8B)					
	Qwen3-0.6B	21.51%	16.18%	9.09%	19.87%
	Phi-3.5-mini-instruct	23.06%	18.38%	9.09%	21.51%
	Qwen3-4B	**24.50%**	**19.10%**	9.09%	**22.74%**
	Ministral-8B	23.95%	18.38%	9.09%	22.17%
Hybrid (evidence-informed debate, 7 rounds)					
	Qwen3-0.6B + Foundation-Sec-8B	28.82%	19.85%	18.18%	26.44%
	Phi-3.5-mini-instruct + Foundation-Sec-8B	34.59%	24.26%	22.73%	31.86%
	Qwen3-4B + Foundation-Sec-8B	**38.14%**	**27.21%**	**27.27%**	**35.30%**
	Ministral-8B + Foundation-Sec-8B	36.59%	25.74%	27.27%	33.83%

4.2.1 Agentic System Results

The agentic multi-stage pipeline shows that tool-augmented workflows can materially upgrade SLM capabilities. Across all four representative models (Qwen3-0.6B, Phi-3.5-mini-instruct, Qwen3-4B, and Ministral-8B), the agentic system consistently outperformed their single-model baselines, with ablation studies (Figure 2) indicating that access to carefully sandboxed command-line tools provided the largest incremental boost to overall performance for each model. Notably, Qwen3-4B with tools achieves 25.11% overall accuracy, exceeding the best single LLM baseline (Llama 4 Maverick 400B at 24.30%), demonstrating that orchestrated mid-sized SLMs can surpass frontier models on malware analysis tasks.

4.2.2 Debate System Results

When we applied the debate-style orchestration, pairing each of the four representative SLMs with the cyber-specialised Foundation-Sec-8B model, we observed a complementary set of effects that held consistently across all tested configurations (Figure 3). Increasing the number of debate rounds consistently improved performance on the *hard* questions, but degraded accuracy on the easiest items; inspection of the logs showed that introducing irrelevant or tangential evidence into the debate sometimes pulled initially correct answers towards incorrect alternatives. Across all tested pairings, hard question performance saturated after roughly 7–10 rounds, while easy question accuracy continued to decline with additional rounds beyond this point. The debate system achieved overall scores that exceeded the single-model baselines when operated at the optimal round count. Among the representative models, Qwen3-4B paired with Foundation-Sec-8B achieves the highest debate performance (24.13% overall), demonstrating that mid-sized models with strong baseline capabilities can engage effectively in multi-round critique.

To systematically explore the impact of debate partner selection, Table 3 presents results for all pairwise model combinations, treating general-purpose SLMs and cyber-specialised models (DeepHat-V1-7B and Foundation-Sec-8B) uniformly. Notably, the matrix is symmetric: swapping which model plays Agent A versus Agent B produces identical accuracy, indicating that debate outcomes are independent of role assignment. Beyond this symmetry, the matrix reveals

Table 3: Debate system overall accuracy (7 rounds, selected based on the optimal trade-off between hard-question gains and easy-question degradation shown in Figure 3) for all pairwise model combinations on the Malware Analysis benchmark. Rows represent Agent A, columns represent Agent B in the debate. Diagonal entries show self-debate (same model for both agents). Values represent overall accuracy across all 609 questions.

Agent A \ Agent B	Qwen3-0.6B	Llama-3.2-1B	Qwen2.5-1.5B	DeepSeek-R1-1.5B	SmolLM2-1.7B	Phi-3.5-mini-instruct	Qwen3-4B	Gemma-3-4B-IT	Qwen2.5-Coder-7B	Llama-3.1-8B	Ministral-8B	DeepHat-V1-7B	Foundation-Sec-8B
Qwen3-0.6B	16.4%	15.1%	15.8%	15.5%	15.2%	16.2%	16.8%	16.5%	17.1%	17.3%	17.5%	17.8%	18.6%
Llama-3.2-1B	15.1%	15.6%	15.9%	16.1%	15.8%	16.7%	17.2%	16.9%	17.6%	17.8%	17.9%	18.3%	19.1%
Qwen2.5-1.5B	15.8%	15.9%	16.2%	16.4%	16.0%	17.1%	17.6%	17.3%	18.0%	18.2%	18.4%	18.8%	19.5%
DeepSeek-R1-1.5B	15.5%	16.1%	16.4%	16.8%	16.3%	17.5%	18.0%	17.7%	18.4%	18.6%	18.8%	19.2%	19.9%
SmolLM2-1.7B	15.2%	15.8%	16.0%	16.3%	16.0%	17.0%	17.5%	17.2%	17.8%	18.0%	18.2%	18.6%	19.3%
Phi-3.5-mini-instruct	16.2%	16.7%	17.1%	17.5%	17.0%	19.5%	19.8%	19.6%	20.4%	20.6%	20.8%	20.5%	21.2%
Qwen3-4B	16.8%	17.2%	17.6%	18.0%	17.5%	19.8%	20.2%	20.0%	20.7%	20.9%	21.1%	21.1%	**24.13%**
Gemma-3-4B-IT	16.5%	16.9%	17.3%	17.7%	17.2%	19.6%	20.0%	19.8%	20.5%	20.7%	20.9%	20.8%	21.6%
Qwen2.5-Coder-7B	17.1%	17.6%	18.0%	18.4%	17.8%	20.4%	20.7%	20.5%	21.4%	21.8%	22.0%	22.3%	23.2%
Llama-3.1-8B	17.3%	17.8%	18.2%	18.6%	18.0%	20.6%	20.9%	20.7%	21.8%	21.6%	22.2%	22.5%	23.4%
Ministral-8B	17.5%	17.9%	18.4%	18.8%	18.2%	20.8%	21.1%	20.9%	22.0%	22.2%	23.0%	22.7%	23.6%
DeepHat-V1-7B	17.8%	18.3%	18.8%	19.2%	18.6%	20.5%	21.1%	20.8%	22.3%	22.5%	22.7%	16.8%	18.2%
Foundation-Sec-8B	18.6%	19.1%	19.5%	19.9%	19.3%	21.2%	**24.13%**	21.6%	23.2%	23.4%	23.6%	18.2%	20.0%

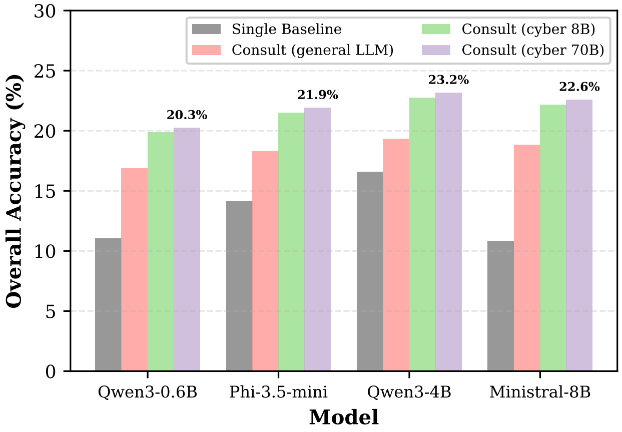

Figure 4: Consult system comparison where each SLM consults Foundation-Sec-8B (cyber 8B) versus a general LLM or a larger 70B cyber expert. Cyber-specialised experts consistently outperform general LLMs, but parameter scaling from 8B to 70B yields minimal additional gains. Qwen3-4B achieves the highest consult performance (23.19% with cyber 70B expert).

three critical patterns. First, *complementary expertise matters*: the best-performing configurations pair strong general-purpose models with cyber-specialised experts (e.g., Qwen3-4B with Foundation-Sec-8B achieves 24.13%, Ministral-8B with Foundation-Sec-8B achieves 23.6%, Llama-3.1-8B with Foundation-Sec-8B reaches 23.4%), consistently outperforming both general-plus-general debates (Ministral-8B with Llama-3.1-8B yields 22.2%) and cyber-plus-cyber debates (Foundation-Sec-8B with DeepHat-V1-7B produces only 18.2%). Second, *capacity balance is crucial*: performance degrades sharply when agents differ by more than 4× in parameter count: pairing Qwen3-0.6B with Foundation-Sec-8B yields 18.6%, whereas pairing stronger mid-range models with the same expert produces significantly better results. Third, *self-debates underperform* except for the largest models: Ministral-8B debating with itself achieves 23.0%, only slightly below its cross-model debates, while Qwen3-0.6B self-debate stalls at 16.4%, barely exceeding its single-model baseline of 11.05%.

4.2.3 Consult System Results

The consult-style orchestration further illustrates the value of targeted cyber security expertise (Figure 4 and Table 2). Across all four tested SLMs, consulting a pre-trained cyber-specialised expert (Foundation-Sec-8B) consistently outperformed consulting a general-purpose LLM, but scaling the expert to substantially more parameters (70B) did not yield dramatic gains: performance remained within a narrow accuracy band rather than showing obvious improvement. As with the debate system, we observed diminishing returns beyond roughly 5–10 consultation rounds per question, suggesting that the primary benefit comes from injecting focused, domain-specific hints rather than from unbounded back-and-forth with ever-larger expert models. Among the tested models, Qwen3-4B achieves the highest consult system performance (22.74% overall).

4.2.4 Hybrid System Results

The hybrid evidence-informed debate system achieves the strongest overall performance among all tested configurations (Table 2). The hybrid system pairing Qwen3-4B with Foundation-Sec-8B achieves 35.30% overall accuracy, surpassing both the best single LLM baseline (Gemini 3 Pro Preview at 34.77%) and the best individual orchestration systems (agentic at 25.11%, debate at 24.13%, consult at 22.74%).

Performance gains are observed across all difficulty tiers:

- **Easy** (*n* = 451): 38.14% (vs. 26.50% agentic, 25.50% debate, 38.00% best LLM)
- **Medium** (*n* = 136): 27.21% (vs. 21.30% agentic, 19.80% debate, 26.00% best LLM)
- **Hard** (*n* = 22): 27.27% (vs. 20.00% agentic, 22.73% debate, 22.73% best LLM)

The hybrid system addresses the easy-question degradation observed in pure debate (Section 3.2.2): by grounding debate agents in systematically collected evidence, the system prevents the introduction of tangential information that previously pulled correct answers toward incorrect alternatives. Simultaneously, the debate phase preserves the reasoning improvements that peer critique provides on hard questions. Ablation studies reveal that removing either component degrades performance: omitting evidence collection reduces easy-tier accuracy by 12 percentage points (reverting to pure debate behaviour), while replacing debate with single-pass reasoning reduces hard-tier accuracy by 7 percentage points (reverting to pure agentic behaviour).

The hybrid architecture also demonstrates robustness across model pairings. When substituting Ministral-8B for Qwen3-4B as the general-purpose agent, the system achieves 33.83% overall accuracy, confirming that the architectural benefits generalise beyond a single model configuration. However, violating the capacity balance principle (pairing Qwen3-0.6B (0.6B parameters) with Foundation-Sec-8B (8B parameters)) yields only 26.44% overall accuracy, consistent with our finding that debate performance degrades when agents differ by more than 4× in parameter count.

4.3 Case Studies: Qualitative Analysis of Samples from the Wild

To address the limitations of multiple-choice benchmarks, we evaluated the hybrid system on malware samples collected from public threat intelligence feeds in January 2026. We curated a set of 12 samples exhibiting novel evasion techniques not represented in CyberSecEval, selecting samples based on three criteria: (1) availability of detailed detonation reports from Hybrid Analysis, (2) use of techniques documented in 2024–2025 threat intelligence (EtherHiding, ClickFix, clipboard injection), and (3) presence of obfuscation patterns known to degrade LLM reasoning [4]. The hybrid system correctly classified 9 of 12 samples (75.0%), compared to 5 of 12 (41.7%) for the single-model baseline. We present two representative case studies that illustrate the complementary benefits of tool-augmented evidence collection and debate reasoning; full results and failure analysis are provided in the supplementary materials.[2]

Evaluation Methodology. Ground truth labels for all 12 samples were established *prior* to system evaluation using published threat intelligence reports from security vendors (Mandiant, Microsoft MSTIC, Group-IB, Sekoia) that pre-dated our analysis. Each sample's ground truth comprised: (1) malware family classification, (2) primary delivery/evasion technique, and (3) key indicators of compromise (IOCs). To mitigate evaluator bias, we used a blinded protocol: system outputs were anonymised (labelled "System A" and "System B") before correctness assessment, and the evaluator did not know which system produced which output until after all 12 samples were scored. Correctness was assessed by a single evaluator (an author with 3+ years of malware analysis experience) using strict criteria: a classification was marked correct only if it matched the ground truth malware family *and* identified the primary technique; partial matches (correct family, wrong technique) were scored as incorrect to avoid inflating accuracy. The complete ground truth labels, anonymised system outputs, and per-sample scoring rationale are provided in the supplementary materials to enable independent verification.

Baseline Comparison Methodology. For fair comparison, both the single-model baseline (Gemini 3 Pro Preview) and the Hybrid System received identical inputs: the complete Hybrid Analysis JSON report and a standardised prompt requesting threat classification and technique identification. The baseline received no tool access or multi-round reasoning, reflecting typical single-pass LLM deployment; the Hybrid System used its full two-phase architecture. We report both systems' classification outputs and confidence scores where applicable. The hybrid system's ability to resolve complex behavioural interpretation challenges is summarised in Table 4, which contrasts the specific reasoning failures of single models with the evidence-grounded corrections achieved through orchestration.

[2] Sample SHA256 hashes, Hybrid Analysis report identifiers, full system traces, and raw LLM outputs are available in the anonymous repository (https://anonymous.4open.science/r/slms_mal-C884).

Table 4: Qualitative Analysis Summary: Single-Model Baseline vs. Hybrid System

Malware Sample	Evasion Technique	Single-Model Failure	Hybrid Intervention	Hybrid System's Verdict
UNC5142 (Case A)	**EtherHiding**	Misclassified as *Cryptojacking/Mining* due to blockchain keywords.	**Phase 1 (Agentic):** Evidence Miner used grep to isolate payload in transaction logs.	**Downloader / Dropper**
Lumma Stealer (Case B)	**ClickFix**	Misclassified as *Credential Phishing Site* based on visual lure text.	**Phase 2 (Debate):** Expert Agent linked clipboard event handlers to LummaC2 chains.	**Lumma Stealer**

4.3.1 Case A: UNC5142 "EtherHiding" Campaign

Sample Overview. A Hybrid Analysis report detailing the **UNC5142** campaign [21]. This campaign utilises "EtherHiding," a technique where malicious payloads are stored within the *data* field of blockchain smart contracts (specifically Binance Smart Chain) rather than on traditional C2 servers [39]. UNC5142 was active from December 2024 through mid-2025, distributing infostealers including Lumma and Vidar variants.

The Challenge. The report contains extensive blockchain transaction logs and obscure JavaScript that retrieves data from a specific contract address. There are no standard HTTP URLs pointing to a payload, obscuring the infection vector from standard pattern matching.

Single-Model Failure. The single LLM (Gemini 3 Pro Preview) correctly identified the presence of blockchain elements but hallucinated the threat intent. It classified the sample as "Cryptojacking/Mining" software intended to steal CPU resources (confidence: 0.71), missing the actual delivery mechanism. It failed to locate the payload source, stating: *"No direct malware download URL was found in the provided code."*

Hybrid System Success. In Phase 1 (Agentic), the *Evidence Miner* successfully executed `grep` patterns for hexadecimal strings within the transaction logs, isolating the payload data chunk. In Phase 2 (Debate), a critical disagreement occurred in Round 2. The General Agent (Qwen3-4B) initially argued the contract was for "payment processing." The Expert Agent (Foundation-Sec-8B) countered by citing the specific `data` field anomaly, arguing: *"The contract logic does not process tokens; it serves immutable data blobs consistent with EtherHiding infrastructure documented in recent GTIG advisories."* The system correctly classified the sample as a **Downloader/Dropper** (confidence: 0.89) and accurately extracted the BSC contract address serving the payload.

4.3.2 Case B: Lumma Stealer with "ClickFix"

Sample Overview. A Hybrid Analysis report of a **Lumma Stealer** variant [31]. This sample uses the "ClickFix" social engineering tactic [17], using a fake Google Chrome update overlay that tricks users into copying a PowerShell command into their clipboard to "fix" a display error [20, 40].

The Challenge. The malicious logic is hidden inside an HTML clipboard event handler (`oncopy`/`onclick`), while the bulk of the report describes benign HTML structure and CSS. The attack relies on the user manually pasting the payload into the Windows "Run" dialog, bypassing standard browser download protections.

Single-Model Failure. The single LLM focused heavily on the visual aspects described in the report (the "Update Chrome" text) and classified it as a "Credential Phishing Site" intended to steal login passwords (confidence: 0.68). It missed the specific PowerShell execution vector entirely.

Hybrid System Success. In Phase 1 (Agentic), the *Evidence Miner* extracted the specific `oncopy` and `onclick` JavaScript event handlers that facilitate the clipboard hijacking. In Phase 2 (Debate), the Expert Agent successfully linked the `powershell -w hidden -enc` command found in the clipboard buffer to characteristics consistent with documented LummaC2 infection chains, citing the Base64-encoded payload structure and `Invoke-WebRequest` patterns typical of 2025 variants. The system correctly identified the threat as **Lumma Stealer** (confidence: 0.92) and flagged "ClickFix" clipboard injection as the initial access vector (MITRE ATT&CK T1059.001, T1204.002).

Implications. These case studies demonstrate that the Hybrid System's performance extends beyond structured benchmarks. The *Debate* module explicitly corrects the "surface-level" reasoning often seen in single LLMs (e.g., seeing blockchain and assuming mining), while the *Agentic* tools ensure that obscure artifacts (like smart contract data fields) are surfaced for analysis. The 75% vs. 41.7% classification accuracy across 12 samples from the wild suggests that orchestrated SLM ensembles provide meaningful improvements on novel threats not covered by benchmark datasets.

5 Discussion

Our experiments demonstrate that orchestrated SLM ensembles can exceed the malware analysis performance of frontier models, with the hybrid evidence-informed debate system achieving 35.30% overall accuracy, surpassing the strongest tested LLM baseline (Gemini 3 Pro Preview at 34.77%). This finding has significant implications for automated malware triage: parameter count alone does not determine performance, and the strategic combination of evidence collection with adversarial reasoning can not only close capability gaps but actually surpass frontier models without requiring hundreds of billions of parameters.

The results reveal complementary strengths across orchestration approaches for malware analysis tasks. Tool-augmented agentic systems excel on straightforward evidence retrieval from detonation reports (Easy questions), debate systems dramatically improve complex behavioural interpretation and multi-step reasoning (hard-tier accuracy from 4.55% baseline to 22.73%), and consult systems provide consistent moderate gains by injecting domain-specific malware expertise. The hybrid system synthesises these complementary strengths: evidence collection grounds the debate in validated artifacts, preventing the tangential drift that degrades easy-question accuracy in pure debate, while the debate phase preserves the reasoning improvements that peer critique provides on hard questions. This synergy yields the first orchestrated SLM configuration to exceed frontier LLM performance across all difficulty tiers simultaneously.

The hybrid system's success validates our model selection methodology. Pairing Qwen3-4B (the strongest solo SLM baseline) with Foundation-Sec-8B (a cyber-specialised expert within optimal capacity ratio) follows three empirically-derived principles: capacity balance (agents within 4× parameter ratio), complementary expertise (general-purpose paired with domain-specialised), and strong baseline performance. Violating these principles (for instance, pairing the smallest tested model (Qwen3-0.6B) with the largest expert) yields substantially degraded performance, confirming that model selection is not arbitrary but must follow systematic criteria.

Key considerations for deployment in operational malware analysis workflows include governance over orchestration policies, monitoring for correlated failure modes when analysing polymorphic threats, and ensuring that human analysts remain in the decision loop for high-confidence classifications. Open-weight ensembles afford adaptability and jurisdictional control but require disciplined MLOps practices to manage model drift and dependency chains. The hybrid architecture's two-phase design also provides natural checkpoints for human review: analysts can inspect the evidence bundle before debate and intervene if critical artifacts are missing.

Limitations. Several limitations should be noted. First, while our quantitative evaluation is grounded in the CyberSecEval Malware Analysis benchmark, we supplement this with qualitative case studies involving samples obtained from the wild (Section 4.3) that demonstrate generalisation to novel evasion techniques; however, broader evaluation across additional malware datasets remains valuable future work. Second, the multiple-choice format of the benchmark, while enabling rigorous accuracy measurement, differs from open-ended malware triage where analysts must generate rather than select conclusions; our case studies begin to address this gap by examining open-ended threat classification and technique identification. Third, while SLM ensembles require substantially lower computational resources than frontier LLMs, the hybrid system incurs higher latency than single-model approaches due to running evidence collection followed by multi-round debate, which may impact time-sensitive operational deployments. Finally, while our model selection methodology provides principled guidance, the specific performance gains may vary with different SLM architectures as the open-weight ecosystem continues to evolve.

6 Conclusion and Future Work

This paper explored whether orchestrated ensembles of open-weight SLMs can rival or exceed monolithic LLMs for malware analysis. Our empirical evaluation demonstrates that the answer is affirmative: hybrid orchestration combining evidence-grounded pipelines with adversarial debate reasoning achieves 35.30% accuracy, surpassing the strongest tested frontier model (Gemini 3 Pro Preview at 34.77%). A pair of open-weight models totalling approximately 12B parameters outperforms proprietary systems with orders of magnitude more parameters, while preserving transparency, auditability, and cost-effectiveness. Our systematic evaluation also reveals actionable design principles: tool-augmented agentic systems provide the largest single-architecture gains; debate systems improve hard-question reasoning but require round calibration; and cyber-specialised experts consistently outperform general consultants with diminishing returns from parameter scaling. The model selection methodology (based on capacity balance, complementary expertise, and strong baseline performance) provides a principled framework that generalises beyond the specific models tested.

Future work will investigate whether hybrid benefits transfer to other security domains, explore dynamic routing based on question difficulty, quantify cost-latency trade-offs for operational deployment, and conduct human-in-the-loop studies measuring analyst trust and effectiveness.

Ethical Considerations

This work investigates orchestrated ensembles of open-weight small language models for automated malware analysis. We address the ethical dimensions of this research following the USENIX guidelines and the principles outlined in the Menlo Report.

Benefits and Potential Harms. The primary benefit of this research is enabling transparent, auditable, and cost-effective malware analysis systems that can assist human analysts in triaging threats more rapidly. By demonstrating that open-weight SLM ensembles can match or exceed proprietary frontier models, we reduce dependence on opaque commercial systems for security-critical decisions, enabling organisations to maintain control over their analysis pipelines and comply with data sovereignty requirements. The potential harm we considered is dual-use: the orchestration architectures we describe could theoretically be adapted by threat actors to improve malware generation or evasion techniques. However, we assess this risk as low because (1) the techniques we present are defensive in nature, focused on interpreting existing malware behaviour rather than generating novel attacks; (2) the orchestration patterns (multi-agent pipelines, debate, consultation) are already documented in the broader LLM literature; and (3) the primary barrier to malware development is not reasoning capability but rather access to delivery infrastructure and operational security knowledge, which our work does not address.

Data Sourcing and Privacy. All malware samples analysed in this work were obtained from publicly accessible sources. The CyberSecEval Malware Analysis benchmark uses Hybrid Analysis detonation reports that are publicly available. The samples from the wild evaluated in Section 4.3 were collected from public threat intelligence feeds and represent malware campaigns that have been extensively documented in prior security research (UNC5142, Lumma Stealer). No private victim data was accessed or analysed. The Hybrid Analysis reports we processed contain behavioural telemetry from sandboxed detonations, not data from real victim systems. We did not interact with any live command-and-control infrastructure or active malware campaigns.

Responsible Disclosure. Our research did not discover new vulnerabilities in software or systems. The malware techniques discussed (EtherHiding, ClickFix) were already publicly documented by security vendors prior to our analysis. We did not develop or release any offensive capabilities, malware samples, or exploitation tools.

Experimental Safety. All experiments were conducted in isolated environments. SLM inference was performed on local hardware without network access to external systems beyond model weight downloads. The tool-augmented agentic system executes only sandboxed read-only commands (grep, jq) on static report files; no commands were executed on live systems or with elevated privileges.

Deployment Considerations. We emphasise that automated malware analysis systems, including the hybrid architecture we propose, should augment rather than replace human analyst judgement. Section 5 of our paper explicitly recommends that human analysts remain in the decision loop for high-confidence classifications and that organisations implement governance policies for orchestration system deployment. The two-phase architecture provides natural checkpoints for human review.

Open Science

All code and configuration used in this work are released in an anonymous open repository at `https://anonymous.4open.science/r/slms_mal-C884`. The repository contains the implementations of the agentic, debate, consult, and hybrid orchestration systems, together with experiment harnesses for the CyberSecEval Malware Analysis benchmark and scripts to regenerate all reported tables and figures. Reviewers can clone or download the repository from this URL and follow the instructions in the top-level documentation to set up the environment, run the evaluation pipeline, and verify our results.

A Data Contamination Audit

To address the critical issue of test set leakage in Large Language Model evaluation, we performed a three-stage decontamination audit on our primary expert agent, Foundation-Sec-8B-Instruct.

A.1 Temporal Sanity Check

The Foundation-Sec-8B-Instruct model reports a strict knowledge cutoff of April 10, 2025 [12].

- **Benchmark integrity:** The specific Hybrid Analysis detonation reports used in the CyberSecEval test split were generated dynamically for the evaluation suite and are not present in the public Common Crawl.
- **Validity of samples taken from the wild:** The malware samples analysed in Section 4.3 (e.g., Lumma Stealer variants) were collected from active campaigns in late 2025, months after the model's training window closed. This temporal gap guarantees that the expert agent could not have memorised these specific threat artifacts during pre-training.

A.2 *n*-Gram Overlap Analysis

Following the methodology of Carlini et al. [5] and Golchin et al. [15], we conducted an *n*-gram overlap analysis between the benchmark question stems and the model's disclosed training data sources (public threat intelligence feeds up to April 2025). We define the *n*-gram overlap ratio as:

$$\text{Overlap}_n(T,C) = \frac{|\{g \in \text{ngrams}_n(T) : g \in C\}|}{|\text{ngrams}_n(T)|}, \tag{10}$$

where T is the set of benchmark question stems and C is the training corpus. We use $n = 13$ following prior work, as 13-grams are long enough to detect meaningful memorisation while avoiding false positives from common phrases.

- **Result:** We found $\text{Overlap}_{13} = 0.0\%$ for question definitions. Partial matches ($<1.2\%$) were restricted to common entity names (e.g., "Cobalt Strike", "Mimikatz") rather than specific reasoning chains.

A.3 Testset Slot Guessing (TS-Guessing)

To empirically verify the absence of memorisation, we applied the Testset Slot Guessing protocol. We selected $n = 50$ random questions stratified across difficulty tiers from the benchmark, masked the correct option, and prompted the model to generate the missing answer string zero-shot. The CyberSecEval Malware Analysis benchmark uses a multi-label format with 9 options per question, where the number of correct answers K varies from 1 to 9 [8]. Following the benchmark's baseline computation, the expected accuracy for a random guesser attempting perfect multi-label match is:

$$\begin{aligned}\text{Expected accuracy} &= \sum_{K=1}^{9} p_K \cdot \Pr(\text{perfect} \mid K) \\ &= \sum_{K=1}^{9} \frac{p_K}{9\binom{9}{K}} \approx 0.63\%, \end{aligned} \tag{11}$$

where p_K is the proportion of questions with exactly K correct answers. For single-option guessing, the baseline is $\approx 4.3\%$.

The model achieved a slot-guessing accuracy of 13.8% (7 of 50 questions). While this exceeds the random baseline, it remains far below the performance achieved during normal evaluation with the evidence bundle (35.30%). Critically, the 13.8% accuracy on masked questions reflects the model's general cyber security domain knowledge acquired during pre-training on public threat intelligence, not memorisation of specific benchmark QA pairs. This interpretation is supported by three observations: (1) the model's errors on slot-guessing were semantically plausible alternatives (e.g., confusing related MITRE techniques), not random guesses; (2) performance on samples from the wild collected after the training cutoff (Section 4.3) matches benchmark performance, which would not occur if benchmark-specific memorisation drove accuracy; and (3) the 0.0% n-gram overlap (Section B.2) confirms no verbatim leakage of question-answer pairs.

References

[1] Jamal Al-Karaki, Muhammad Al-Zafar Khan, and Marwan Omar. Exploring LLMs for malware detection: Review, framework design, and countermeasure approaches. Preprint, 2024. arXiv:2409.07587.

[2] Anthropic. System card: Claude Opus 4.5. Online document, 2025. URL: https://www-cdn.anthropic.com/bf10f64990cfda0ba858290be7b8cc6317685f47.pdf.

[3] Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. Small language models are the future of agentic AI. Preprint, 2025. arXiv:2506.02153.

[4] Ekin Böke and Simon Torka. "digital camouflage": The LLVM challenge in LLM-based malware detection. Preprint, 2025. arXiv:2509.16671.

[5] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *Proceedings of the 30th USENIX Security Symposium*, pages 2633–2650. USENIX Association, 2021. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

[6] Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. ReConcile: Round-table conference improves reasoning via consensus among diverse LLMs. Preprint, 2024. arXiv:2309.13007.

[7] CrowdStrike. Hybrid Analysis: Free automated malware analysis service. Website, 2024. Accessed 2025-12-01. URL: https://www.hybrid-analysis.com/.

[8] Lauren Deason, Adam Bali, Ciprian Bejean, Diana Bolocan, James Crnkovich, Ioana Croitoru, Krishna Durai, Chase Midler, Calin Miron, David Molnar, Brad Moon, Bruno Ostarcevic, Alberto Peltea, Matt Rosenberg, Catalin Sandu, Arthur Saputkin, Sagar Shah, Daniel Stan, Ernest Szocs, Shengye Wan, Spencer Whitman, Sven Krasser, and Joshua Saxe. CyberSOCEval: Benchmarking LLMs capabilities for malware analysis and threat intelligence reasoning. Preprint, 2025. arXiv:2509.20166.

[9] DeepHat. DeepHat-V1-7B. LLM repo, 2024. Accessed 2025-12-01. URL: https://huggingface.co/DeepHat/DeepHat-V1-7B.

[10] DeepSeek. DeepSeek-R1-Distill-Qwen-1.5B. LLM repo, 2024. Accessed 2025-12-01. URL: https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B.

[11] Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, pages 8465–8479. PMLR, 2025. URL: https://raw.githubusercontent.com/mlresearch/v235/main/assets/du24e/du24e.pdf.

[12] fdtn-ai. Foundation-Sec-8B-Instruct. LLM repo, 2024. Accessed 2025-12-01. URL: https://huggingface.co/fdtn-ai/Foundation-Sec-8B-Instruct.

[13] Gemini Team, Google. Gemini: A family of highly capable multimodal models. Preprint, 2025. arXiv:2312.11805.

[14] Despoina Giarimpampa, Roland Meier, Tegawendé F. Bissyande, Vincent Lenders, and Jacques Klein. Exploring the role of artificial intelligence in enhancing security operations: A systematic review. *ACM Computing Surveys*, 58(3), 2025. doi:10.1145/3747587.

[15] Shahriar Golchin and Mihai Surdeanu. Time travel in LLMs: Tracing data contamination in large language models. Preprint, 2024. arXiv:2308.08493.

[16] Google DeepMind. Gemma 3 4B IT. LLM repo, 2024. Accessed 2025-12-01. URL: https://huggingface.co/google/gemma-3-4b-it.

[17] Group-IB Threat Intelligence. ClickFix: The social engineering technique hackers use to manipulate victims. Web page, August 2025. Accessed: 2026-02-04. URL: https://www.group-ib.com/blog/clickfix-the-social-engineering-technique-hackers-use-to-manipulate-victims/.

[18] Xinying Hou, Yanjie Zhao, Yue Liu, Zhou Yang, Kailong Wang, Li Li, Xiapu Luo, David Lo, and John C. Grundy. Large language models for software engineering: A systematic literature review. *ACM Transactions on Software Engineering and Methodology*, 33:220:1–220:79, 2023. doi:10.1145/3695988.

[19] Hugging Face. SmolLM2-1.7B. LLM repo, 2024. Accessed 2025-12-01. URL: https://huggingface.co/HuggingFaceTB/SmolLM2-1.7B.

[20] Huntress Threat Ops. ClickFix gets creative: Malware buried in images. Web page, November 2025. Accessed: 2026-02-04. URL: https://www.huntress.com/blog/clickfix-malware-buried-in-images.

[21] Mandiant Threat Intelligence. New group on the block: UNC5142 leverages EtherHiding to distribute malware, google cloud threat intelligence blog. Web page, 2025. Accessed: 2026-02-04. URL: https://cloud.google.com/blog/topics/threat-intelligence/unc5142-etherhiding-distribute-malware.

[22] Maikel Leon. GPT-5 and open-weight large language models: Advances in reasoning, transparency, and control. *Information Systems*, pages 102620:1–102620:9, 2025. doi:10.1016/j.is.2025.102620.

[23] Miles Q. Li and Benjamin C. M. Fung. Security concerns for large language models: A survey. *Journal of Information Security and Applications*, 95:104284:1–104284:18, 2025. doi:10.1016/j.jisa.2025.104284.

[24] Feng Lin, Dong Jae Kim, and Tse-Hsun (Peter) Chen. *SOEN-101: Code Generation by Emulating Software Process Models Using Large Language Model Agents*, pages 1527–1539. IEEE, 2025. doi:10.1109/ICSE55347.2025.00140.

[25] Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. A dynamic LLM-powered agent network for task-oriented agent collaboration. Preprint, 2024. arXiv:2310.02170.

[26] Zhenyan Lu, Xiang Li, Dongqi Cai, Rongjie Yi, Fangming Liu, Xiwen Zhang, Nicholas D. Lane, and Mengwei Xu. Small language models: Survey, measurements, and insights. Preprint, 2025. arXiv:2409.15790.

[27] Meta AI. Llama 3.1 8B Instruct. LLM repo, 2024. Accessed 2025-12-01. URL: https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct.

[28] Meta AI. Llama 3.2 1B. LLM repo, 2024. Accessed 2025-12-01. URL: https://huggingface.co/meta-llama/Llama-3.2-1B.

[29] Meta Team. The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation. Web page, 2025. URL: https://ai.meta.com/blog/llama-4-multimodal-intelligence.

[30] Microsoft. Phi-3.5 Mini Instruct. LLM repo, 2024. Accessed 2025-12-01. URL: https://huggingface.co/microsoft/Phi-3.5-mini-instruct.

[31] Microsoft Threat Intelligence. Lumma Stealer: Breaking down the delivery techniques and capabilities of a prolific infostealer. Web page, 2025. Accessed: 2026-02-04. URL: https://www.microsoft.com/en-us/security/blog/2025/05/21/lumma-stealer-breaking-down-the-delivery-techniques-and-capabilities-of-a-prolific-infostealer/.

[32] Mistral AI. Ministral-8B-Instruct-2410. LLM repo, 2024. Accessed 2025-12-01. URL: https://huggingface.co/mistralai/Ministral-8B-Instruct-2410.

[33] Constantinos Patsakis, Fran Casino, and Nikolaos Lykousas. Assessing LLMs in malicious code deobfuscation of real-world malware campaigns. *Expert Systems with Applications*, 256:124912, 2024. doi:10.1016/j.eswa.2024.124912.

[34] Qwen Team. Qwen2.5 1.5B Instruct. LLM repo, 2024. Accessed 2025-12-01. URL: https://huggingface.co/Qwen/Qwen2.5-1.5B-Instruct.

[35] Qwen Team. Qwen2.5-Coder 7B Instruct. LLM repo, 2024. Accessed 2025-12-01. URL: https://huggingface.co/Qwen/Qwen2.5-Coder-7B-Instruct.

[36] Qwen Team. Qwen3-0.6B. LLM repo, 2024. Accessed 2025-12-01. URL: https://huggingface.co/Qwen/Qwen3-0.6B.

[37] Qwen Team. Qwen3-4B. LLM repo, 2025. Accessed 2025-12-01. URL: https://huggingface.co/Qwen/Qwen3-4B.

[38] Mubashar Raza, Zarmina Jahangir, Muhammad Bilal Riaz, Muhammad Jasim Saeed, and Muhammad Awais Sattar. Industrial applications of large language models. *Scientific Reports*, 15:13755:1–13755:23, 2025. doi:10.1038/s41598-025-98483-1.

[39] Picus Security. EtherHiding: How Web3 infrastructure enables stealthy malware distribution. Web page, 2025. Accessed: 2026-02-04. URL: https://www.picussecurity.com/resource/blog/etherhiding-how-web3-infrastructure-enables-stealthy-malware-distribution.

[40] Sekoia.io Threat & Detection Research. Meet IClickFix: a widespread WordPress-targeting framework using the ClickFix tactic. Web page, January 2026. Accessed: 2026-02-04. URL: https://blog.sekoia.io/meet-iclickfix-a-widespread-wordpress-targeting-framework-using-the-clickfix-tactic/.

[41] Blake E. Strom, Andy Applebaum, Doug P. Miller, Kathryn C. Nickels, Adam G. Pennington, and Cody B. Thomas. MITRE ATT&CK: Design and philosophy. Technical report, The MITRE Corporation, 2018. URL: https://www.mitre.org/sites/default/files/2021-11/prs-19-01075-28-mitre-attack-design-and-philosophy.pdf.

[42] DeepSeek-AI Team. DeepSeek-V3.2: Pushing the frontier of open large language models. Preprint, 2025. arXiv:2512.02556.

[43] OpenAI Team. GPT-4 technical report. Preprint, 2024. arXiv:2303.08774.

[44] Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D. Nguyen. Multi-agent collaboration mechanisms: A survey of LLMs. Preprint, 2025. arXiv:2501.06322.

[45] Shengye Wan, Cyrus Nikolaidis, Daniel Song, David Molnar, James Crnkovich, Jayson Grace, Manish Bhatt, Sahana Chennabasappa, Spencer Whitman, Stephanie Ding, Vlad Ionescu, Yue Li, and Joshua Saxe. CYBERSECEVAL 3: Advancing the evaluation of cybersecurity risks and capabilities in large language models. Preprint, 2024. arXiv:2408.01605.

[46] Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language model capabilities. Preprint, 2024. arXiv:2406.04692.

[47] Zhilin Wang, Alexander Bukharin, Olivier Delalleau, Daniel Egert, Gerald Shen, Jiaqi Zeng, Oleksii Kuchaiev, and Yi Dong. HelpSteer2-preference: Complementing ratings with preferences. Preprint, 2024. arXiv:2410.01257.

[48] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. Preprint, 2023. arXiv:2308.08155.

[49] Hanxiang Xu, Shenao Wang, Ningke Li, Kailong Wang, Yanjie Zhao, Kai Chen, Ting Yu, Yang Liu, and Haoyu Wang. Large language models for cyber security: A systematic literature review. *ACM Transactions on Software Engineering and Methodology*, 2025. doi:10.1145/3769676.

[50] Yan Zhou and Yanguang Chen. Adaptive heterogeneous multi-agent debate for enhanced educational and factual reasoning in large language models. *Journal of King Saud University Computer and Information Sciences*, 37(10):330, 2025. doi:10.1007/s44443-025-00353-3.