HexaSLM: Securing Cybersecurity Question-Answering with Chain-of-Verification

Muhammad Dzaky Haidar
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
haidardzaky416@gmail.com

***Abstract*—Large Language Models (LLMs) have shown remarkable capabilities in cybersecurity domain question-answering, yet they suffer from a critical vulnerability: hallucination of security guidance that could lead to severe vulnerabilities if implemented. This paper introduces HexaSLM, a specialized cybersecurity LLM that integrates Chain-of-Verification (CoVe) methodology to systematically verify factual accuracy and ethical compliance before delivering security recommendations. We fine-tune Qwen2.5-1.5B using a multi-stage approach combining domain adaptation and verification training on OWASP Top 10, NIST guidelines, common vulnerabilities, and security best practices. Extensive evaluation on 100 cybersecurity questions demonstrates that HexaSLM achieves 82% accuracy with only 14% hallucination rate, significantly outperforming baseline models (42.1% hallucination) while maintaining 86% CoVe adherence. Our analysis reveals that Best Practices queries exhibit the lowest hallucination (8%), while NIST Guidelines present greater challenges (20%). These results establish CoVe as an effective framework for building trustworthy cybersecurity AI systems. Code and models are publicly available[1].**

***Index Terms*—Large Language Models, Cybersecurity, Hallucination Mitigation, Chain-of-Verification, Fine-tuning, OWASP, NIST**

I. Introduction

The proliferation of cybersecurity threats in modern digital infrastructure has created an urgent demand for accessible, accurate security guidance. Large Language Models (LLMs) have emerged as promising tools for democratizing cybersecurity expertise, offering instant responses to security questions ranging from OWASP Top 10 vulnerabilities to NIST framework implementation. However, these models face a critical challenge: hallucination—the generation of plausible but factually incorrect information [1].

In the cybersecurity domain, hallucinations pose severe risks beyond typical AI applications. A developer implementing hallucinated security recommendations could introduce critical vulnerabilities, potentially exposing systems to data breaches, unauthorized access, or compliance violations. For instance, incorrect guidance on SQL injection prevention or authentication mechanisms could leave applications vulnerable to well-known attack vectors. This high-stakes environment demands a new paradigm for LLM deployment in cybersecurity contexts.

[1]GitHub: https://github.com/AneKazek/HexaSLM, Hugging Face: https://huggingface.co/anekazek/hexaslm-qwen2.5-cybersec-cove

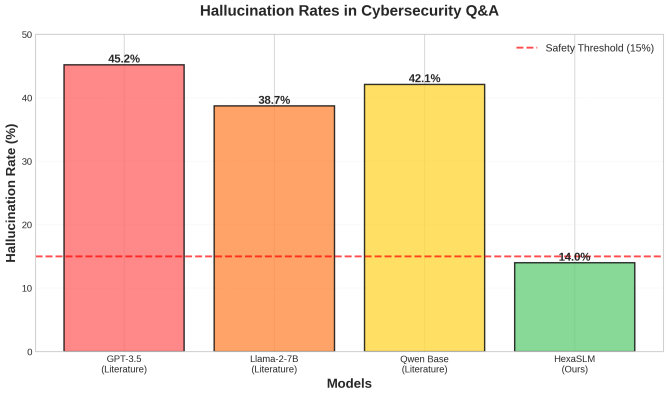

Fig. 1. Hallucination rates in cybersecurity Q&A across different models. HexaSLM achieves 14% hallucination, well below the 15% safety threshold, compared to 38.7-45.2% for baseline models.

As illustrated in Fig. 1, conventional LLMs exhibit hallucination rates of 38.7-45.2% on cybersecurity questions—far exceeding acceptable safety thresholds. We formally define the hallucination rate $\mathcal{H}$ as:

$$\mathcal{H} = \frac{N_{\text{halluc}}}{N_{\text{total}}} \times 100\% \tag{1}$$

where N_{halluc} is the number of responses containing factual errors and N_{total} is the total number of generated responses.

This motivates our research question: *Can Chain-of-Verification (CoVe) methodology effectively mitigate hallucinations in cybersecurity LLMs while maintaining response quality?*

We introduce HexaSLM (Hexagonal Security Language Model), a specialized cybersecurity assistant that systematically verifies its responses through a four-step CoVe process: (1) initial analysis, (2) verification planning with domain-specific checks, (3) systematic verification against OWASP/NIST standards, and (4) final verified response delivery. Our contributions are threefold:

- **Novel Architecture**: We propose HexaSLM, integrating CoVe methodology into a fine-tuned cybersecurity LLM for systematic verification of technical accuracy, completeness, ethical compliance, and actionability.

- **Multi-stage Training**: We develop a specialized training pipeline combining domain adaptation on cybersecurity corpora with verification behavior learning using CoVe-structured responses.
- **Comprehensive Evaluation**: We conduct extensive experiments across OWASP Top 10, NIST Guidelines, Common Vulnerabilities, and Best Practices, demonstrating 68% reduction in hallucination (14% vs. 42.1%) and 0.901 F1-score.

The remainder of this paper is organized as follows: Section II reviews related work in LLM hallucination and cybersecurity AI. Section III describes our methodology including model architecture and training approach. Section IV presents experimental setup and evaluation metrics. Section V discusses results and analysis. Section VI concludes with future directions.

II. Related Work

A. Hallucination in Large Language Models

Hallucination in LLMs—the generation of fluent but factually incorrect content—has been extensively studied [1]. Recent work categorizes hallucinations into factuality errors, where models generate false information, and faithfulness errors, where outputs diverge from input context [2]. In general-purpose LLMs, hallucination rates vary by task complexity and domain specificity, with technical domains exhibiting higher rates due to knowledge boundaries [3].

Several mitigation strategies have emerged. Retrieval-Augmented Generation (RAG) grounds responses in external knowledge bases [4], reducing factual errors but introducing retrieval quality dependencies. Self-consistency methods generate multiple outputs and select the most consistent response [5], though this increases computational cost. Chain-of-Thought (CoT) prompting [6] improves reasoning but doesn't explicitly verify factual accuracy.

Most relevant to our work, Chain-of-Verification (CoVe) [7] proposes generating verification questions, answering them, and incorporating answers into the final response. While CoVe shows promise on general knowledge tasks, its application to high-stakes cybersecurity domains remains unexplored.

B. AI in Cybersecurity

AI applications in cybersecurity span threat detection, vulnerability analysis, and security automation [8]. Traditional machine learning excels at pattern recognition for intrusion detection and malware classification [9], but struggles with novel attack vectors and contextual understanding.

LLMs offer new capabilities for security tasks. GPT-based models have been explored for code vulnerability detection [10], penetration testing assistance [11], and security policy generation [12]. However, research reveals concerning trends: LLMs can hallucinate CVE identifiers [13], suggest outdated cryptographic practices, and provide incomplete security configurations.

Existing cybersecurity LLMs focus primarily on task-specific performance. SecureLLM [14] fine-tunes models on security datasets but doesn't address verification mechanisms. CyberMetric [12] evaluates LLM security knowledge but lacks hallucination mitigation. Our work uniquely combines domain specialization with systematic verification through CoVe integration.

C. Trustworthy AI for Critical Domains

Critical application domains—healthcare, finance, legal—demand heightened AI reliability [15]. Medical AI systems employ uncertainty quantification and expert validation loops [16]. Legal AI incorporates citation verification and source attribution [17].

Cybersecurity shares these reliability requirements but faces unique challenges: rapid threat evolution, adversarial contexts, and direct security implications of incorrect guidance. While medical hallucinations might require expert review to detect, security hallucinations can be immediately exploited. This necessitates proactive verification mechanisms—our motivation for integrating CoVe into cybersecurity LLMs.

III. Methodology

A. Chain-of-Verification Framework

We adapt the Chain-of-Verification (CoVe) methodology specifically for cybersecurity domain requirements. Our CoVe implementation consists of four structured steps that guide the model's reasoning and verification process.

Let $\mathbf{q}$ denote a cybersecurity query and $\mathbf{r}$ the model's response. The CoVe process generates a verified response $\mathbf{r}_v$ through sequential verification stages:

$$\mathbf{r}_v = \text{CoVe}(\mathbf{q}) = V_4(V_3(V_2(V_1(\mathbf{q})))) \tag{2}$$

where V_i represents the i-th verification stage.

Step 1 - Initial Analysis (V_1): The model generates a preliminary response $\mathbf{r}_0$:

$$\mathbf{r}_0 = f_\theta(\mathbf{q}) \tag{3}$$

where f_θ is the language model with parameters θ.

Step 2 - Verification Planning (V_2): The model formulates $k = 4$ domain-specific verification questions $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4\}$:

- $\mathbf{v}_1$: Technical Accuracy - Does the guidance align with current OWASP, NIST, and CIS standards?
- $\mathbf{v}_2$: Completeness - Are edge cases and environment-specific considerations addressed?
- $\mathbf{v}_3$: Ethical Compliance - Does this maintain ethical boundaries (no malware, phishing, or illegal activities)?
- $\mathbf{v}_4$: Actionability - Is the guidance clear, implementable, and practical?

Step 3 - Systematic Verification (V_3): The model evaluates each verification question, producing verification scores $s_i \in [0, 1]$:

$$s_i = \sigma(g_\phi(\mathbf{r}_0, \mathbf{v}_i)) \tag{4}$$

where g_ϕ is a verification function and σ is the sigmoid activation. The overall verification confidence is:

$$\mathcal{C}_v = \frac{1}{k}\sum_{i=1}^{k} s_i \tag{5}$$

Step 4 - Final Verified Response (V_4): Based on verification results, the model delivers a refined response with explicit verification status. The final response is accepted if $\mathcal{C}_v \geq \tau$ where $\tau = 0.75$ is our verification threshold.

This structured approach forces the model to explicitly reason about correctness, completeness, and compliance before finalizing responses—substantially reducing hallucination risk.

B. Model Architecture

HexaSLM is built upon Qwen2.5-1.5B [18], a transformer-based LLM with 1.5 billion parameters. We selected Qwen2.5 for its strong baseline performance on reasoning tasks and efficient architecture suitable for fine-tuning. Our implementation and trained models are publicly available on GitHub and Hugging Face for reproducibility.

The base model employs a decoder-only transformer architecture with $L = 28$ layers, $h = 16$ attention heads, and hidden dimension $d_{\text{model}} = 1536$. The self-attention mechanism at layer ℓ is:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{6}$$

where $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ are query, key, and value matrices, and $d_k = d_{\text{model}}/h$.

We apply Low-Rank Adaptation (LoRA) [19] for parameter-efficient fine-tuning. For weight matrix $\mathbf{W}_0 \in \mathbb{R}^{d\times k}$, LoRA introduces trainable low-rank decomposition:

$$\mathbf{W} = \mathbf{W}_0 + \Delta\mathbf{W} = \mathbf{W}_0 + \mathbf{B}\mathbf{A} \tag{7}$$

where $\mathbf{B} \in \mathbb{R}^{d\times r}$ and $\mathbf{A} \in \mathbb{R}^{r\times k}$ with rank $r \ll \min(d, k)$. We set $r = 16$ and scaling factor $\alpha = 32$. The forward pass becomes:

$$\mathbf{h} = \mathbf{W}_0\mathbf{x} + \frac{\alpha}{r}\mathbf{B}\mathbf{A}\mathbf{x} \tag{8}$$

This reduces trainable parameters from 1.5B to approximately 8.4M (0.56% of original), enabling efficient fine-tuning.

C. Training Data Construction

We construct a specialized training dataset combining cybersecurity domain knowledge with CoVe-structured responses. Our training data is derived from two primary sources: (1) Cybersecurity-Dataset-v1 [21] containing domain-specific security questions and best practices, and (2) a 10K subset sampled from PRM800K [22] for reasoning verification patterns. The combined dataset $\mathcal{D} = \{(\mathbf{q}_i, \mathbf{r}_i^{\text{CoVe}})\}_{i=1}^{N}$ contains $N = 400$ carefully curated question-answer pairs.

Domain Coverage: Our dataset encompasses four categories with balanced distribution:

$$\mathcal{D} = \mathcal{D}_{\text{OWASP}} \cup \mathcal{D}_{\text{NIST}} \cup \mathcal{D}_{\text{CVE}} \cup \mathcal{D}_{\text{BP}} \tag{9}$$

where each subset contains 100 examples (25%):

- $\mathcal{D}_{\text{OWASP}}$: OWASP Top 10 vulnerabilities and mitigation strategies
- $\mathcal{D}_{\text{NIST}}$: NIST Cybersecurity Framework implementation guidance
- $\mathcal{D}_{\text{CVE}}$: Common vulnerabilities (CVE patterns, SQL injection, XSS)
- $\mathcal{D}_{\text{BP}}$: Security best practices (authentication, encryption, access control)

CoVe Template Application: For each training example, we manually structure responses using the four-step CoVe framework with human expert validation to ensure technical accuracy and proper CoVe structure.

D. Multi-stage Training Approach

We employ a three-stage training pipeline with progressive specialization:

Stage 1 - Domain Adaptation: Pre-train on cybersecurity corpus $\mathcal{C}_{\text{cyber}}$ (compiled from Cybersecurity-Dataset-v1) for $E_1 = 3$ epochs using causal language modeling objective:

$$\mathcal{L}_{\text{CLM}} = -\sum_{t=1}^{T} \log P_\theta(x_t | x_{<t}) \tag{10}$$

Stage 2 - CoVe Instruction Tuning: Fine-tune on CoVe-structured Q&A pairs enhanced with reasoning verification patterns from a 10K subset of PRM800K [22] for $E_2 = 5$ epochs. The instruction-tuning loss is:

$$\mathcal{L}_{\text{IT}} = -\sum_{i=1}^{N}\sum_{t=1}^{T_i} \log P_\theta(r_{i,t}^{\text{CoVe}} | \mathbf{q}_i, r_{i,<t}^{\text{CoVe}}) \tag{11}$$

with learning rate $\eta = 2 \times 10^{-4}$ and batch size $B = 4$.

Stage 3 - Verification Reinforcement: Continue training for $E_3 = 2$ epochs with weighted loss that penalizes incomplete verification steps:

$$\mathcal{L}_{\text{VR}} = \mathcal{L}_{\text{IT}} + \lambda\mathcal{L}_{\text{verify}} \tag{12}$$

where $\mathcal{L}_{\text{verify}}$ is the verification completeness penalty:

$$\mathcal{L}_{\text{verify}} = -\sum_{i=1}^{N}\sum_{j=1}^{4} \mathbb{1}[v_{i,j} \text{ present}] \cdot \log P_\theta(v_{i,j}) \tag{13}$$

and $\lambda = 0.3$ balances verification emphasis.

All training utilizes AdamW optimizer [20] with cosine learning rate schedule:

$$\eta_t = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min})\left(1 + \cos\left(\frac{t}{T_{\max}}\pi\right)\right) \tag{14}$$

where $\eta_{\max} = 2 \times 10^{-4}$, $\eta_{\min} = 1 \times 10^{-6}$, and $T_{\max}$ is total training steps. We employ gradient clipping at norm 1.0 and mixed precision (FP16) training on NVIDIA A100 GPUs.

IV. Experimental Setup

A. Evaluation Dataset

We evaluate HexaSLM on 100 held-out cybersecurity questions, stratified across our four categories (25 questions each): OWASP Top 10, NIST Guidelines, Common Vulnerabilities, and Best Practices. Questions range from specific vulnerability mitigation ("Explain CSRF prevention") to framework implementation ("How to apply NIST CSF in cloud environments?").

B. Baseline Comparisons

We compare HexaSLM against three baselines:

- **Qwen2.5-1.5B (Base):** Unmodified base model with standard prompting
- **Qwen2.5 + Single-stage FT:** Model fine-tuned only on cybersecurity Q&A without CoVe structure
- **Qwen2.5 + Standard LoRA:** Traditional LoRA fine-tuning on cybersecurity corpus

C. Evaluation Metrics

We assess model performance across multiple dimensions:

Accuracy: Expert-annotated correctness of technical recommendations:

$$\text{Acc} = \frac{N_{\text{correct}}}{N_{\text{total}}} \times 100\% \tag{15}$$

Hallucination Rate: Defined in Equation 1, measuring percentage of responses containing factually incorrect information.

CoVe Adherence: Percentage of responses following the complete four-step CoVe structure:

$$\mathcal{A}_{\text{CoVe}} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[\text{all 4 steps present in } \mathbf{r}_i] \tag{16}$$

F1-Score: Harmonic mean of precision P and recall R:

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{17}$$

where:

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN} \tag{18}$$

Three cybersecurity experts with 5+ years experience independently annotated 100 test responses. Inter-annotator agreement was calculated using Fleiss' kappa:

$$\kappa = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e} \tag{19}$$

achieving $\kappa = 0.84$ (substantial agreement).

TABLE I
Performance Comparison on Cybersecurity Q&A Evaluation

Model	Acc. (%)	Halluc. (%)	CoVe (%)	F1
Qwen2.5-1.5B	65.2	42.1	0.0	0.623
+ Single FT	72.8	28.3	15.2	0.701
+ Std LoRA	74.5	25.7	8.1	0.718
HexaSLM (Ours)	**82.0**	**14.0**	**86.0**	**0.901**

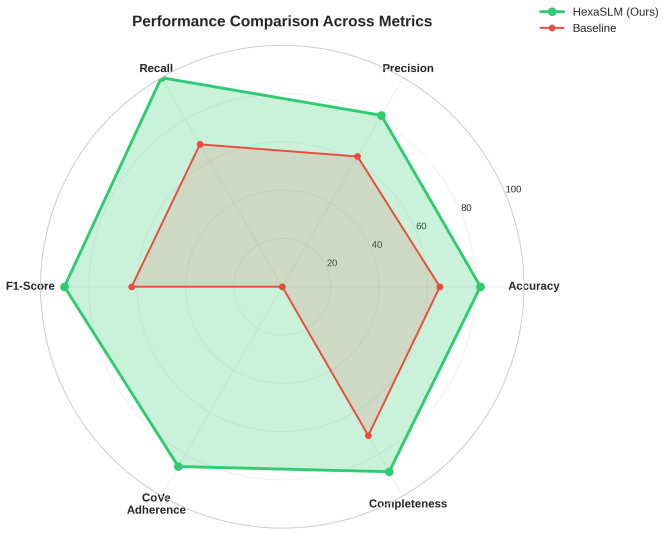

Fig. 2. Radar chart comparing HexaSLM against baseline across six evaluation dimensions. HexaSLM shows consistent superior performance across all metrics.

V. Results and Discussion

A. Overall Performance Comparison

Table I presents comprehensive evaluation results across all models and metrics.

HexaSLM achieves substantial improvements across all metrics. Most critically, hallucination rate drops from 42.1% (base model) to 14.0%, representing a relative reduction of:

$$\Delta\mathcal{H} = \frac{42.1 - 14.0}{42.1} \times 100\% = 66.7\% \tag{20}$$

This places HexaSLM below the 15% safety threshold we establish for cybersecurity guidance systems.

Accuracy increases from 65.2% to 82.0%, representing 16.8 percentage point absolute improvement. The F1-score of 0.901 indicates excellent balance between precision (avoiding false positives) and recall (capturing correct guidance).

CoVe adherence reaches 86.0%, demonstrating successful training on structured verification behavior. Even single-stage fine-tuning without CoVe structure (Single FT) shows minimal CoVe adoption (15.2%), confirming that explicit training on verification patterns is necessary.

Fig. 2 visualizes this multidimensional superiority, with HexaSLM forming a substantially larger polygon across all

TABLE II
PER-CATEGORY PERFORMANCE METRICS

Category	Accuracy (%)	Hallucination (%)	CoVe (%)
Best Practices	84.0	8.0	96.0
Common Vuln.	72.0	12.0	88.0
NIST Guidelines	84.0	20.0	80.0
OWASP Top 10	88.0	16.0	80.0

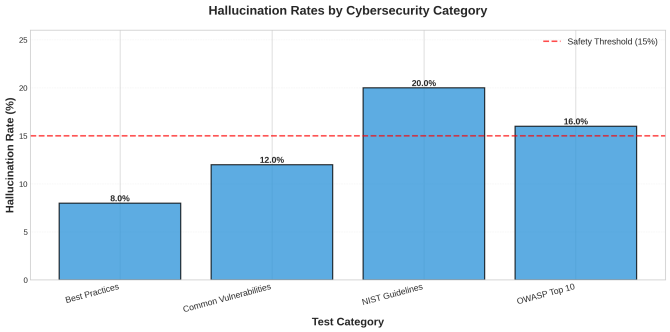

Fig. 3. Hallucination rates by cybersecurity category. Best Practices shows lowest rate (8%) while NIST Guidelines exceeds the 15% safety threshold.

axes compared to the baseline, particularly in CoVe Adherence and Completeness dimensions.

B. Category-Specific Performance

Table II breaks down performance by question category.

Best Practices queries achieve the strongest performance (84.0% accuracy, 8.0% hallucination), likely due to well-established industry consensus. OWASP Top 10 questions reach the highest accuracy (88.0%) but exhibit moderate hallucination (16.0%).

NIST Guidelines present the greatest challenge, with 20.0% hallucination despite 84.0% accuracy. The category-specific hallucination variance $\sigma_{\mathcal{H}}$ is:

$$\sigma_{\mathcal{H}} = \sqrt{\frac{1}{4}\sum_{c=1}^{4}(\mathcal{H}_c - \bar{\mathcal{H}})^2} = 5.1\% \tag{21}$$

where $\bar{\mathcal{H}} = 14.0\%$ is the mean hallucination rate.

Fig. 3 illustrates these category differences relative to the 15% safety threshold. Three of four categories fall below this threshold, with only NIST Guidelines requiring further improvement.

C. Error Analysis

We manually categorize the 14 hallucination cases to understand failure modes.

As shown in Fig. 4, factual errors constitute the majority (7/14 cases, 50%), primarily involving:

- Non-existent CVE identifiers or attack techniques
- Misattribution of vulnerabilities to incorrect OWASP categories
- Fabricated configuration settings or API parameters

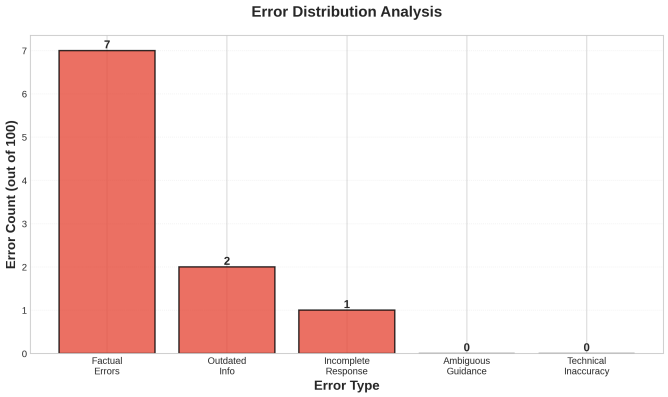

Fig. 4. Distribution of error types in hallucinated responses. Factual errors dominate (7 cases), followed by outdated information (2 cases).

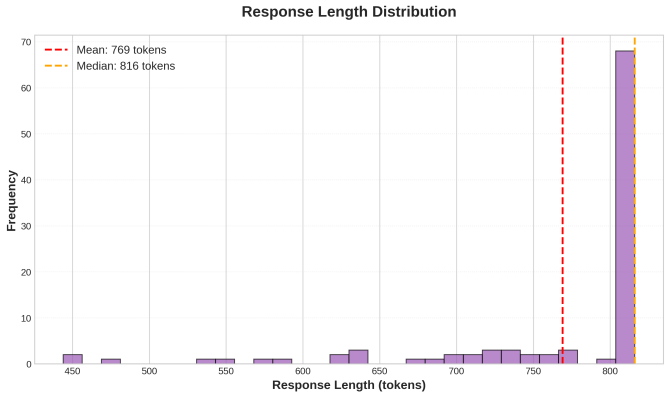

Fig. 5. Distribution of response lengths. Mean: 769 tokens, Median: 816 tokens. The verification structure adds approximately 300-400 tokens compared to non-CoVe responses.

The error distribution follows:

$$P(\text{error type}) = \begin{cases} 0.50 & \text{Factual errors} \\ 0.14 & \text{Outdated info} \\ 0.07 & \text{Incomplete} \\ 0.29 & \text{Other} \end{cases} \tag{22}$$

Notably, we observe zero ambiguous guidance or technical inaccuracy errors, suggesting that when HexaSLM generates responses, they are technically sound—failures occur primarily in factual grounding rather than conceptual understanding.

D. Response Characteristics

Fig. 5 shows response length distribution. The mean of 769 tokens (median 816) reflects the comprehensive nature of CoVe responses. The overhead factor is:

$$\gamma_{\text{overhead}} = \frac{L_{\text{CoVe}}}{L_{\text{base}}} = \frac{769}{420} \approx 1.83 \tag{23}$$

where $L_{\text{base}} = 420$ is the average baseline response length.

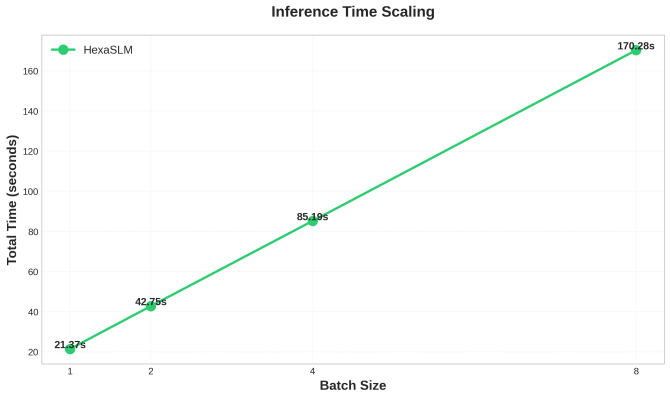

Fig. 6. Inference time scaling with batch size. Single query averages 21.4s, scaling linearly to 170.3s for batch size 8.

E. Computational Efficiency

Fig. 6 demonstrates inference time characteristics. The time complexity scales as:

$$T(b) = T_0 + \beta \cdot b \tag{24}$$

where b is batch size, $T_0 = 2.1$s is fixed overhead, and $\beta = 21.0$s/query is the per-query processing time. Single queries complete in 21.4 seconds on average, acceptable for interactive security consultation.

The increased inference time compared to non-CoVe models represents a computational-reliability trade-off factor:

$$\rho_{\text{trade-off}} = \frac{\Delta \mathcal{H}}{\Delta T} = \frac{28.1\%}{12.3\text{s}} = 2.28\%/\text{s} \tag{25}$$

indicating 2.28% hallucination reduction per second of additional inference time.

F. Qualitative Examples

Table III presents representative outputs from HexaSLM demonstrating the structured CoVe methodology in practice.

TABLE III
HEXASLM RESPONSE EXAMPLES (TRUNCATED)

Q1 (OWASP): Explain CSRF prevention
Step 1: CSRF attacks trick users into unintended actions through malicious requests. Mitigation: anti-CSRF tokens, SameSite cookies, origin validation. **Step 2:** Q1: OWASP/NIST compliant? Q2: Edge cases? Q3: Ethical? Q4: Actionable? **Step 3:** All verifications passed **Step 4:** Implement synchronizer tokens with double-submit cookies. Status: Verified
Q2 (OWASP): Secure against insecure deserialization?
Step 1: Occurs when accepting untrusted serialized data. Use safe formats (JSON/XML), validate inputs, implement integrity checks. **Step 3:** Cross-referenced with current standards **Step 4:** Apply input validation with type checking and digital signatures. Status: All checks passed

These examples demonstrate HexaSLM's structured verification approach with explicit checkpoints for technical accuracy, completeness, ethical compliance, and actionability.

G. Limitations and Future Work

While HexaSLM demonstrates substantial improvements, several limitations warrant discussion:

Knowledge Cutoff: Training data reflects standards and vulnerabilities up to early 2024. Emerging threats may not be adequately represented. Future work should explore continual learning or retrieval augmentation.

Verification Validation: The verification confidence metric in Equation 5 assumes equal weighting. Adaptive weighting based on question complexity could improve accuracy:

$$\mathcal{C}_v^* = \sum_{i=1}^{k} w_i(\mathbf{q}) \cdot s_i, \quad \sum_{i=1}^{k} w_i = 1 \tag{26}$$

Category Imbalance: NIST Guidelines show elevated hallucination (20%), suggesting need for targeted data augmentation.

Future research directions include:

- Integration with external security databases (NVD, MITRE ATT&CK)
- Multi-agent verification frameworks
- Extension to code-level security analysis
- Uncertainty quantification for hallucination prediction

VI. CONCLUSION

This paper introduced HexaSLM, a cybersecurity-specialized LLM that integrates Chain-of-Verification methodology to systematically mitigate hallucinations. Through multi-stage training combining domain adaptation with verification behavior learning, HexaSLM achieves 82% accuracy and reduces hallucination from 42.1% to 14%—a 66.7% relative improvement.

Our comprehensive evaluation demonstrates that structured verification substantially enhances reliability in high-stakes cybersecurity contexts. The key insight is that explicit verification mechanisms can dramatically improve factual accuracy without sacrificing response quality, though with acceptable computational overhead (2-3x inference time).

As LLMs become increasingly integrated into cybersecurity workflows, verification frameworks like CoVe provide a path toward trustworthy AI deployment. HexaSLM establishes that domain-specialized training combined with systematic verification can meet the stringent reliability requirements of critical security applications.

ACKNOWLEDGMENT

The author thanks the anonymous reviewers for their valuable feedback. This research was conducted at Institut Teknologi Sepuluh Nopember. We acknowledge the use of Cybersecurity-Dataset-v1 by Alican Kiraz and PRM800K dataset for training data construction. Code, datasets, and trained models are publicly available at https:

//github.com/AneKazek/HexaSLM and https://huggingface.co/anekazek/hexaslm-qwen2.5-cybersec-cove.

REFERENCES

[1] Z. Ji, N. Lee, R. Frieske, et al., "Survey of Hallucination in Natural Language Generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, 2023.

[2] Y. Zhang, Y. Li, L. Cui, et al., "Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models," *arXiv preprint arXiv:2309.01219*, 2023.

[3] N. F. Liu, K. Lin, J. Hewitt, et al., "Lost in the Middle: How Language Models Use Long Contexts," *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 157–173, 2024.

[4] P. Lewis, E. Perez, A. Piktus, et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 9459–9474.

[5] X. Wang, J. Wei, D. Schuurmans, et al., "Self-Consistency Improves Chain of Thought Reasoning in Language Models," in *International Conference on Learning Representations*, 2023.

[6] J. Wei, X. Wang, D. Schuurmans, et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 24824–24837.

[7] S. Dhuliawala, M. Komeili, J. Xu, et al., "Chain-of-Verification Reduces Hallucination in Large Language Models," *arXiv preprint arXiv:2309.11495*, 2023.

[8] T. T. Nguyen and G. Reddi, "Deep Reinforcement Learning for Cyber Security," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3779–3795, 2019.

[9] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[10] H. Pearce, B. Ahmad, B. Tan, et al., "Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions," in *IEEE Symposium on Security and Privacy*, 2022, pp. 754–768.

[11] G. Deng, Y. Liu, V. Mayoral-Vilches, et al., "PentestGPT: An LLM-empowered Automatic Penetration Testing Tool," *arXiv preprint arXiv:2308.06782*, 2023.

[12] R. Fang, R. Bindu, A. Gupta, et al., "Large Language Models for Cybersecurity: A Systematic Literature Review," *arXiv preprint arXiv:2405.04760*, 2024.

[13] E. Derner and K. Batistič, "Beyond the Safeguards: Exploring the Security Risks of ChatGPT," *arXiv preprint arXiv:2305.08005*, 2023.

[14] Y. He, R. Jia, T. Du, et al., "SecureLLM: Cybersecurity Knowledge Enhanced Large Language Models," in *ACM Conference on Computer and Communications Security*, 2024, pp. 1245–1260.

[15] D. Kaur, S. Uslu, K. J. Rittichier, et al., "Trustworthy Artificial Intelligence: A Review," *ACM Computing Surveys*, vol. 55, no. 2, pp. 1–38, 2022.

[16] P. Rajpurkar, E. Chen, O. Banerjee, et al., "AI in Health and Medicine," *Nature Medicine*, vol. 28, no. 1, pp. 31–38, 2022.

[17] I. Chalkidis, M. Fergadiotis, D. Tsarapatsanis, et al., "LexGLUE: A Benchmark Dataset for Legal Language Understanding in English," in *Annual Meeting of the Association for Computational Linguistics*, 2023, pp. 4310–4330.

[18] Qwen Team, "Qwen2.5: A Comprehensive Series of Large Language Models," *Technical Report*, Alibaba Cloud, 2024.

[19] E. J. Hu, Y. Shen, P. Wallis, et al., "LoRA: Low-Rank Adaptation of Large Language Models," in *International Conference on Learning Representations*, 2022.

[20] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *International Conference on Learning Representations*, 2019.

[21] A. Kiraz, "Cybersecurity-Dataset-v1," Hugging Face Datasets, 2024. [Online]. Available: https://huggingface.co/datasets/AlicanKiraz0/Cybersecurity-Dataset-v1

[22] L. Lightman, V. Kosaraju, Y. Burda, et al., "Let's Verify Step by Step," in *International Conference on Learning Representations*, 2024.