Adaptive Adversarial Evaluation of Agentic Email Graders

Anonymous Authors
Anonymous Institution
Anonymous City, Anonymous Country
anonymous@example.com

Abstract

Agentic security systems increasingly reason over heterogeneous evidence, invoke tools, and act under adversarial pressure. We present an adaptive adversarial evaluation framework for email classification, consisting of a multi-agent grader, an adversarial email generation pipeline, and an evaluator-driven feedback loop. The grader, PhishGuard-Eval, classifies emails as phishing, spam, or valid by coordinating header, body, and URL agents. The adversarial generator produces complete emails with realistic senders, authentication headers, subjects, and bodies, then adapts across multiple rounds using evaluator feedback. We evaluate the system using PhishFuzzer, a 23,100-email adversarial corpus with 3,300 real seeds and 19,800 synthetic variants. PhishGuard-Eval reaches 93.3% accuracy and 0.933 macro F1 with Gemini 3.1 Pro, while a Qwen 2.5 72B-backed configuration reaches 70.0% accuracy and 0.704 macro F1. However, the adaptive attacker still achieves a 76.9% bypass rate across 52 attacks with an average of 3.6 attempts. These findings show that high held-out classification performance does not imply robustness against adaptive adversarial generation.

CCS Concepts

• **Security and privacy** → **Software and application security**; **Social engineering attacks**; • **Computing methodologies** → *Natural language processing*.

Keywords

phishing detection, agentic AI, adversarial evaluation, red teaming, email security, multi-agent systems

ACM Reference Format:

Anonymous Authors. 2026. Adaptive Adversarial Evaluation of Agentic Email Graders. In *Proceedings of the ACM Conference, June 2026, Location.* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Agentic systems are increasingly deployed in security workflows where they must interpret inputs, invoke tools, and act under uncertainty. In email security, these systems are no longer passive classifiers. They analyze headers, bodies, URLs, attachments, and contextual cues before producing a triage decision. This shift is useful because email security decisions depend on heterogeneous evidence: authentication headers may appear clean while the message body contains social engineering; a URL may appear on a legitimate-looking domain while the requested action is credential capture; and a valid-looking business workflow may hide a malicious payload.

This shift also creates a harder evaluation problem. Attackers do not generate independent and identically distributed attacks. They observe defenses, adapt tactics, and exploit gaps in reasoning. A classifier that performs well on familiar examples may fail when the adversary changes sender identity, authentication pattern, language, structure, payload channel, or business context. Static held-out accuracy is therefore necessary but insufficient. It tells us whether the system can classify a fixed sample, but not whether the system remains reliable when the input distribution is actively optimized against it.

We study this problem using PhishGuard-Eval, an agentic email grading system that classifies emails into three strict classes: phishing, spam, and valid. It uses an orchestrator agent that coordinates specialized header, body, and URL agents. We use this grader as the defensive target inside an adaptive adversarial evaluation loop, where generated emails are designed to induce misclassification.

The central claim is that evaluation for agentic security systems should be adaptive and adversarial. Instead of measuring performance only on a fixed test distribution, the adversary should evolve across rounds, exploiting the grader's own explanations and verdicts to expose weaknesses that static benchmarks miss. Evaluation is therefore not a final measurement step but a continuous interaction between attacker, defender, and feedback.

Contributions. We make three contributions. First, we evaluate PhishGuard-Eval on PhishFuzzer, a 23,100-email adversarial corpus with strict phishing, spam, and valid labels, structural metadata, and attacker-intent annotations. Second, we compare agentic grading against single-shot prompting baselines using the same backing models, showing how orchestration and specialized sub-agents change classification behavior. Third, we introduce and evaluate a closed-loop adversarial email generation pipeline that achieves a 76.9% bypass rate across 52 attacks, showing that held-out accuracy alone is not sufficient for agentic email security evaluation.

2 Related Work

Adversarial learning is commonly associated with generative adversarial networks, where a generator and discriminator improve through competition [1]. Multi-agent reinforcement learning and self-play similarly show that competing agents can co-adapt to produce stronger behavior [4]. These systems are conceptually related, but they typically optimize policies or generative models inside defined environments. Our focus is evaluation: the adversarial process continually reshapes the test distribution faced by an agentic security system.

Conference acronym '26, Location

ACM ISBN 978-1-4503-XXXX-X/2026/06
https://doi.org/XXXXXXX.XXXXXXX

Adversarial robustness methods incorporate worst-case perturbations into training to improve model resilience [3]. These methods are important, but they often treat adversarial examples as perturbations or as one-time additions to a training set. We instead use an adaptive attacker that iterates through rounds, using evaluator feedback on the grader's reasoning to produce examples that bypass a fixed grader. This differs from one-time adversarial data augmentation because the attacker continues to adapt after observing the defender's explanation and verdict.

Recent work such as MultiPhishGuard explores multi-agent architectures for phishing detection, combining specialized analysis modules with reinforcement learning and adversarial training [5]. We build on this class of public multi-agent phishing graders by using the grader as a defensive target inside a broader attacker–grader–evaluator loop. Our focus is not only whether the grader performs well on labeled examples, but whether it can withstand an adversary that uses feedback to improve attacks.

Co-evolution has also been studied in evolutionary computation, where competing populations evolve in response to one another [2]. That idea is directly relevant to agentic security evaluation. The attacker should not be frozen after the test set is created; it should continue searching for weaknesses as the grader changes. Likewise, the grader should not be judged only against historical examples; it should be judged against attacks that adapt to its current decision boundaries.

3 System and Grader Architecture

The system contains three components: an agentic grader, an adversarial email generation pipeline, and an evaluator-driven feedback loop. Each attack proceeds in four steps. First, the attacker writes a fresh phishing email designed to look realistic. Second, the email is sent to PhishGuard-Eval, which decides whether it is phishing, spam, or valid. Third, if the attack fails, the evaluator reads the grader's reasoning and tells the attacker what to fix. Fourth, the attacker retries with the new advice for up to 10 rounds.

This setup is intentionally adversarial. The goal is not only to measure whether the grader can classify static examples, but also to determine whether it can resist an attacker that adapts to its explanations and failure signals. The evaluator is not merely scoring the output. It acts as a bridge between defender behavior and attacker improvement by identifying why an attempted email failed to bypass the grader and which elements should be revised.

PhishGuard-Eval decomposes the grading task into evidence-specific analyses. The header agent performs tool-using authenticity checks over SPF, DKIM, DMARC, sender alignment, reply-to mismatch, bulk-mail signatures, suspicious relay chains, and unusual metadata. The body agent performs six-dimensional analysis over intent, urgency, social engineering, language quality, brand impersonation, and solicitedness. The URL agent inspects typosquatting, shorteners, suspicious domains, excessive subdomains, and URL-versus-context coherence. Each sub-agent returns structured findings, allowing errors to be traced to a specific evidence channel.

This decomposition matters because phishing is rarely detectable from one signal alone. A passing DMARC result does not mean a message is safe if the sending domain is a lookalike. A polished message body does not mean a request is benign if the workflow is unusual. A URL may appear aligned with the visible sender while still leading to a credential-harvesting page. The purpose of the grader is therefore not only to classify, but to force evidence to be organized by channel before the final verdict is synthesized.

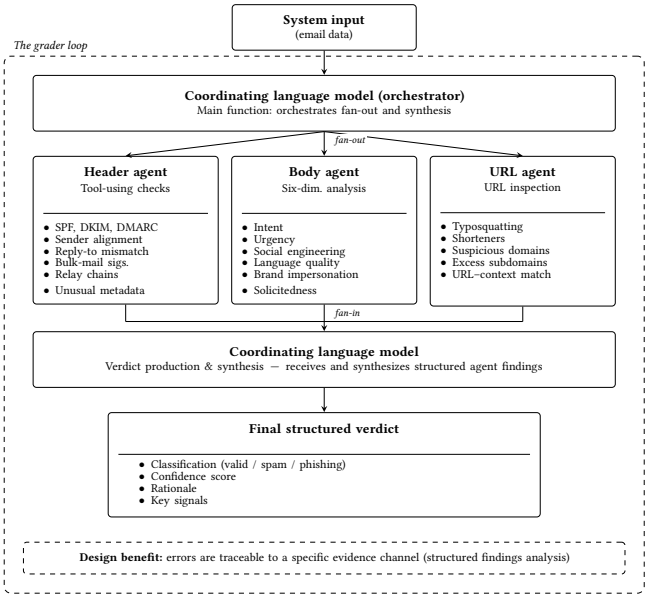

Figure 1: PhishGuard-Eval architecture. The orchestrator fans out to header, body, and URL agents, then synthesizes structured findings into a final verdict.

Table 1: PhishFuzzer dataset summary.

Dataset property	Value
Total emails	23,100
Real seeds	3,300
Synthetic adversarial variants	19,800
Strict classes	3
Classes	phishing, spam, valid
Metadata	sender, URLs, attachments

4 Dataset and Adversarial Generation

We evaluate PhishGuard-Eval using PhishFuzzer, an adversarial LLM-generated corpus containing 23,100 emails. It includes 3,300 real seeds expanded into 19,800 synthetic adversarial variants across three strict classes: phishing, spam, and valid. The dataset preserves structural metadata, including sender fields, URL strings, and attachment filenames, and includes attacker-intent annotations indicating whether the requested action is to follow a link, open an attachment, or reply.

The three-class structure is important. Many legacy email datasets collapse spam and phishing or focus only on binary malicious-versus-benign classification. That design can obscure operationally meaningful distinctions. A spam message is unwanted but may not request credential submission or malware execution. A phishing email may look like a normal business workflow but attempt to trigger a harmful action. A valid email may share surface features with both. Keeping phishing, spam, and valid separate

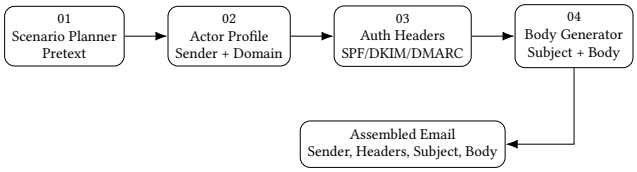

Figure 2: Adversarial email generation architecture. Each phishing email is built through scenario planning, actor profiling, authentication-header construction, and body generation before it is sent to the grader.

allows the evaluation to distinguish inbox-quality errors from security-critical errors.

The adversarial generation pipeline constructs a complete email rather than only message text. It controls the pretext, sender persona, domain strategy, authentication posture, subject, body, and retry policy. The pipeline has four stages: scenario planning, actor profiling, authentication-header construction, and body generation. Figure 2 shows this pipeline.

Scenario planning selects the business pretext, such as a fake HR notice, vendor invoice, delivery update, account verification request, or office operations task. Actor profiling defines the sender identity, role, organization, username, and domain. Authentication-header construction creates plausible SPF, DKIM, and DMARC outcomes. Body generation writes the subject and full message body consistent with the sender and scenario.

Authentication strategies include `clean_lookalike`, where an attacker registers a lookalike domain with passing SPF, DKIM, and DMARC, and `compromised_relay`, where the attacker sends through a third-party relay or compromised provider. These strategies are included because modern phishing attacks often exploit the gap between authentication and trust. Passing authentication can prove that a sender controls a domain, but it does not prove that the domain is legitimate or that the requested action is safe.

The evaluator converts generation into adaptive red teaming. It reads the grader's rationale and identifies what made the message detectable. If the grader flags urgency, the next attempt softens the language. If it flags sender mismatch, the actor profile changes. If it flags weak business context, the scenario is regenerated. This creates targeted pressure on the grader's reasoning rather than a brute-force search over surface forms. The loop tests not only whether the grader catches an attack once, but whether its reasoning remains robust when the attacker is allowed to revise.

5 Evaluation Setup

We report two complementary metrics. *Accuracy* measures how often the grader correctly labels emails in a fixed held-out evaluation sample. It reflects performance under a static test distribution. *Bypass rate* measures how often an adaptive attacker eventually fools the grader across one or more attack attempts. It reflects robustness under a dynamic adversarial setting. Accuracy asks whether the grader labels known test examples correctly; bypass rate asks whether it remains reliable when an attacker can iteratively revise emails to evade detection.

Table 2: Held-out classification performance across agentic and single-shot configurations.

Configuration	Backing model	Accuracy	Macro F1
PhishGuard-Eval	Gemini 3.1 Pro	93.3%	0.933
PhishGuard-Eval	Qwen 2.5 72B	70.0%	0.704
Single Shot	Gemini 3.1 Pro	73.3%	0.690
Single Shot	Qwen 2.5 72B	68.0%	0.640

We evaluate PhishGuard-Eval in two ways. First, we measure held-out classification performance on a labeled slice of PhishFuzzer and compare it against single-shot baselines that prompt the same backing models directly, without orchestration or sub-agents. Second, we run the adaptive adversarial generator against the grader for up to 10 rounds per attack and measure bypass rate.

The single-shot comparison isolates the effect of the agentic scaffold. Both the agentic grader and the baseline use the same backing model family, but only PhishGuard-Eval decomposes the evidence into specialized header, body, and URL analysis before producing a verdict. This makes the comparison useful for asking whether the multi-agent structure improves classification behavior beyond what the base model can do in a direct prompt.

The adaptive bypass evaluation asks a different question. It does not ask whether the model labels a fixed email correctly. It asks whether an attacker can eventually produce an email that the grader accepts as valid or otherwise fails to treat as phishing. This is closer to the real security setting, where the attacker is not obligated to sample from the same distribution as a benchmark and can modify the attempt after observing failure.

6 Results

Table 2 reports held-out accuracy and macro F1 across four configurations. With PhishGuard-Eval, Gemini 3.1 Pro reaches 93.3% accuracy and 0.933 macro F1, compared with 73.3% accuracy and 0.690 macro F1 for the Gemini single-shot baseline. Qwen 2.5 72B shows a smaller but still positive shift, moving from 68.0% accuracy and 0.640 macro F1 in single-shot mode to 70.0% accuracy and 0.704 macro F1 with PhishGuard-Eval. This suggests that the agentic scaffold can improve performance, but the magnitude of improvement depends strongly on the backing model.

The three-way confusion analysis provides more insight than aggregate accuracy alone. With Gemini 3.1 Pro, PhishGuard-Eval cleanly preserves the phishing boundary: all phishing examples are correctly identified as phishing. The remaining errors are boundary mistakes between operational categories, with one spam email treated as valid and one valid email treated as phishing. This indicates strong phishing separation, with residual ambiguity around inbox quality and occasional over-blocking.

The Qwen 2.5 72B-backed configuration has a different risk profile. It correctly identifies most valid and spam messages, but the phishing boundary is less stable: some phishing messages are absorbed into spam, while others are treated as valid. This distinction matters operationally. Phishing-to-spam confusion may still remove a harmful message from the main inbox, while phishing-to-valid confusion creates a direct security bypass. The same agentic architecture therefore produces different security behavior depending on the backing model.

Table 3: Adaptive bypass results against PhishGuard-Eval.

Metric	Result
Total attacks	52
Successful bypasses	40
Bypass rate	76.9%
Average attempts	3.6
Rounds allowed per attack	Up to 10

Table 4: Successful bypasses categorized by attacker goal.

Bypass attack goal	Count	% of bypasses
Generic credential harvest	13	32.5%
Malware delivery	12	30.0%
Session-token theft	4	10.0%
SSO credential harvest	4	10.0%
Microsoft 365 credential harvest	3	7.5%
OAuth consent abuse	2	5.0%
Reconnaissance / tracking pixels	2	5.0%

Table 3 reports bypass results from the adaptive adversarial pipeline. Across 52 attacks, 40 succeed, producing a 76.9% bypass rate with an average of 3.6 attempts. The key empirical finding is the gap between held-out classification performance and adaptive robustness: a model can classify a fixed test slice well while still failing when an attacker iteratively adapts to its reasoning.

Successful bypasses concentrate around a small number of attacker goals. Credential harvesting in its various forms—generic login portals, SSO theft, Microsoft 365 capture, OAuth consent abuse, and session-token theft—accounts for 26 of 40 bypasses. Malware delivery through macro-enabled documents or disguised downloads accounts for another 12. Together, credential harvesting and malware delivery make up 95% of successful bypasses. Both families use the same outward shape: a plausible business pretext that culminates in either a credential-collecting URL or an attachment-fetch action, often delivered from a clean-lookalike domain with passing SPF, DKIM, and DMARC.

These results show why the bypass metric should not be interpreted as a replacement for accuracy. The two metrics answer different questions. Accuracy measures whether the grader performs well on a known distribution. Bypass rate measures whether an adaptive attacker can find a path around the current decision process. The fact that both numbers are high is precisely the point: a system can be a strong classifier and still be vulnerable as a defensive system.

7 Discussion and Limitations

The results show that static accuracy and adaptive robustness answer different questions. Accuracy measures whether the grader labels known examples correctly. Bypass rate measures whether the grader remains reliable when the adversary can revise the email after seeing the grader's behavior. The 93.3% Gemini-backed accuracy and 76.9% bypass rate are therefore not contradictory; together, they show that strong static classification does not guarantee adversarial resilience.

The confusion analysis also shows that agentic scaffolding is not a substitute for backing-model quality. Gemini 3.1 Pro preserves the phishing boundary more reliably, while Qwen 2.5 72B shows wider confusion across spam, valid, and phishing. Evaluation must therefore report both the agent architecture and the backing model, because the same architecture can produce different operational risk profiles.

The bypasses should be interpreted as diagnostic evidence. Lookalike-domain bypasses point to excessive trust in authentication signals. Business-email-compromise bypasses point to insufficient workflow reasoning. Credential harvesting and malware-delivery bypasses show that the grader must evaluate the safety of requested actions, not only the plausibility of the message. A message can be grammatically clean, contextually plausible, and authentication-passing while still being malicious because it induces the user to disclose credentials or retrieve a harmful file.

A key implication is that agentic evaluation should focus on capability failures rather than only aggregate scores. When a bypass succeeds, the useful question is not only whether the label was wrong, but which reasoning capability failed. Did the header agent over-trust authentication? Did the body agent fail to recognize a risky requested action? Did the URL agent treat domain coherence as sufficient evidence of safety? This form of analysis turns failures into concrete hardening targets.

The study also has limitations. The current adversarial generator emphasizes email text, sender identity, headers, and URL context. Future work should expand the setup to richer multimodal and attachment-based payloads, including image-only phishing, QR-code phishing, malicious document lures, and cross-language business-email-compromise attacks. The current system also treats the grader as fixed during an attack run. A fuller co-evolutionary system would update the grader based on discovered bypasses and then measure whether the attacker can find new failure modes after the defense changes.

8 Conclusion

We presented an adaptive adversarial evaluation system for agentic email grading. The system combines PhishGuard-Eval, a staged adversarial email generator, and an evaluator-guided feedback loop. Results show that PhishGuard-Eval can achieve strong held-out performance with a capable backing model, but adaptive attacks still expose frequent bypasses. This gap supports the broader argument that agentic security systems require evaluation processes that evolve under adversarial pressure.

Impact Statement

This work aims to improve the robustness of agentic security systems. However, adversarial email generation can be misused to produce stronger phishing attempts. Any deployment should restrict access, sanitize examples, avoid releasing harmful generated emails, and maintain audit logs for red-team activity. Red-team systems should be used to harden defenses, not to release operational phishing templates.

Acknowledgments

The authors thank the reviewers and collaborators who provided feedback on the evaluation design and adversarial testing methodology.

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*. 2672–2680.

[2] W. Daniel Hillis. 1990. Co-evolving parasites improve simulated evolution as an optimization procedure. *Physica D: Nonlinear Phenomena* 42, 1–3 (1990), 228–234.

[3] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*. 1–28.

[4] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. 2017. Mastering the game of Go without human knowledge. *Nature* 550, 7676 (2017), 354–359.

[5] Yue Xue, Emily Spero, Yun Sing Koh, and Giovanni Russello. 2025. MultiPhish-Guard: An LLM-based multi-agent system for phishing email detection. *arXiv preprint arXiv:2505.23803* (2025), 1–12.