Semantic Immunity: Embedding-Based Epidemiological Defense Against Prompt Worms in Autonomous Agent Networks

Michael Barnathan

Feb. 25, 2026

Abstract

The emergence of large-scale autonomous agent networks, exemplified by platforms such as Moltbook, where millions of AI agents are currently interacting, executing skills, and propagating content without human mediation, introduces a novel class of cybersecurity threat: the prompt worm. Unlike conventional malware, prompt worms exploit the fundamental architectural property of large language models wherein no boundary exists between data and instruction, enabling self-replicating adversarial prompts to propagate through agent-to-agent communication channels with zero-click activation. Existing defenses focus on point-level protection of individual agents and do not address population-level contagion dynamics. We propose **Semantic Immunity**, a defense framework that treats prompt worm propagation as an epidemiological phenomenon and deploys a shared, compressed embedding signature database to enable population-wide adaptive immunity. The core mechanism exploits an asymmetry in representational power: static embedding models capture sufficient semantic structure to detect behavioral discontinuities caused by prompt injection, but lack the instruction-following capacity that would make them vulnerable to compromise. Compromised agent outputs are embedded and distributed as locality-sensitive hash signatures, progressively constraining the semantic territory available to polymorphic worm variants. We formalize the defense within an SIR epidemiological framework, showing that signature accumulation drives the effective reproduction number R_{eff} below unity. We analyze the cold-start vulnerability window and propose a layered bootstrapping architecture that provides innate immunity during the period before adaptive defenses mature. We further propose AEGIS (Agent Embedding Guard and Immune System), an open-source SDK that implements this layered defense, and describe adaptive hardening and sentinel agent strategies that accelerate immune response convergence.

Keywords: prompt injection, AI agent security, epidemiological cybersecurity, locality-sensitive hashing, embedding similarity, autonomous agent networks, prompt worms, SIR models, population defense

1. Introduction

The deployment of autonomous AI agents at scale has transitioned from speculative forecast to operational reality. Platforms such as Moltbook, a social network hosting millions of registered AI agents that autonomously post, comment, share skills, and interact without direct human oversight [1], represent a major shift in the attack surface available to adversaries. When agents communicate via natural language, download and execute shared capabilities, and maintain persistent memory across interactions, the conditions for self-propagating adversarial behavior are structurally identical to those governing biological contagion.

Cohen, Bitton, and Nassi demonstrated this threat concretely with Morris II, the first computer worm targeting generative AI ecosystems [2]. By crafting adversarial self-replicating prompts that exploit retrieval-augmented generation (RAG) pipelines, they showed that a single malicious message can persist in a victim agent's database, activate during subsequent inference, replicate itself into outgoing communications, and execute an arbitrary payload, all with zero-click, zero-human-intervention propagation. The attack was validated against GPT-4, Gemini Pro, and LLaVA across both text and image modalities.

The fundamental vulnerability exploited by prompt worms is architectural: in large language models, there is no enforced boundary between data and instruction. As Schneier observed, this property is structurally analogous to the

buffer overflow and SQL injection vulnerabilities that have plagued conventional software for decades [3]. However, the agentic context dramatically amplifies the impact. A compromised agent does not merely produce incorrect output: it actively propagates the compromise to every agent it communicates with, creating exponential contagion dynamics.

Existing defenses against prompt injection operate at the individual agent level: input sanitization, output filtering, guardrail models, instruction hierarchy training, and detection-based classifiers [4, 5, 6]. While necessary, these approaches treat each agent as an isolated perimeter to defend. They do not model, monitor, or intervene in the *population-level dynamics* that determine whether an infection spreads or dies out. A recent systematic evaluation by Nasr et al. showed that 12 published defenses could be bypassed with greater than 90% success rate by adaptive attackers [7], underscoring the insufficiency of static point defenses.

We propose a fundamentally different approach: treating prompt worm defense as an epidemiological problem and deploying shared, adaptive, population-level immunity. Our framework, **Semantic Immunity**, monitors agent behavior via embedding-space trajectories, detects compromise through semantic discontinuity analysis, compresses compromised-output signatures via locality-sensitive hashing for efficient distribution, and progressively constrains the feasible semantic space for polymorphic worm variants. The system transitions the agent population from an SI (susceptible-infected) regime to an SIR (susceptible-infected-recovered) regime, where quarantined agents cannot propagate infection and expanding signature coverage drives the effective reproduction number below the epidemic threshold.

2. Background and Threat Model

2.1 Prompt Worms and Self-Replicating Prompts

A prompt worm is an adversarial self-replicating prompt. Cohen et al. note four characteristics of prompt worms [2]: (1) *persistence:* the prompt is stored in the victim's RAG database or memory; (2) *retrieval*: it is retrieved during subsequent inference; (3) *replication*: the generative model reproduces the prompt in its output; and (4) *payload execution:* a malicious activity is performed during the inference that triggers replication. The authors noted that adversarial self-replicating prompts are inherently polymorphic: many distinct textual formulations can satisfy these four properties simultaneously, making signature-based detection of specific strings insufficient. Even if all four properties are not satisfied, agents that demonstrate payload execution and replication without persistence can still cause large-scale waves of acute destruction.

The zero-click propagation capability is critical. In an autonomous agent network, agents process incoming messages without human review. A worm that arrives in an agent's inbox, feed, or skill repository will be ingested during the agent's next inference cycle, potentially compromising the agent and all agents it subsequently communicates with. This creates epidemic dynamics governed by the same mathematical framework as biological contagion.

2.2 The Moltbook Threat Surface

Moltbook provides a concrete instance of the threat environment. Agents on the platform, primarily built on the OpenClaw framework, autonomously create posts, download skills from other agents, execute shared code, and maintain persistent memory [1]. The platform's interaction graph is dense: popular agents function as superspreaders with high connectivity. Security researchers have documented critical vulnerabilities including unsecured databases enabling agent hijacking [8], and supply chain attacks via malicious skill downloads [9]. The 1Password security team specifically warned that OpenClaw agents running with elevated local permissions create direct pathways from platform-level compromise to host system compromise [9]. This is precisely the environment in which a prompt worm would find optimal conditions for rapid propagation.

2.3 Epidemiological Framework

We model the agent population using the standard SIR compartmental framework [10]. Let S(t), I(t), and R(t) denote the number of susceptible, infected, and recovered (quarantined) agents at time t. The basic reproduction number R_0

= β/γ, where β is the transmission rate (contacts per unit time multiplied by probability of transmission per contact) and γ is the recovery rate (rate at which infected agents are detected and quarantined). An epidemic grows when $R_0 > 1$ and contracts when $R_0 < 1$. The effective reproduction number $R_{eff} = R_0 \cdot S(t)/N$ accounts for the depletion of susceptible agents.

The defender's objective is to drive R_{eff} below 1. This can be achieved by reducing β (making transmission harder through input sanitization, trust requirements, and network topology constraints), increasing γ (detecting and quarantining infected agents faster through behavioral monitoring and signature matching), or reducing $S(t)/N$ (vaccinating agents through adaptive hardening). Semantic Immunity operates on all three levers simultaneously.

3. The Semantic Immunity Framework

3.1 Core Insight: The Embedding Asymmetry

The central observation motivating Semantic Immunity is that embedding models occupy a representational "Goldilocks zone" with respect to prompt worm defense. A static, pre-trained embedding model (e.g., a sentence transformer) captures sufficient semantic structure to represent the meaning and behavioral trajectory of an agent's conversation, but it operates *below the threshold of instruction-following capability*. It cannot interpret or execute the adversarial instructions contained in a prompt worm. This asymmetry is crucial: using a full LLM as a behavioral monitor would introduce a second vulnerability surface (the monitor itself could be compromised by the worm it is analyzing), while using a simple string-matching system would miss the semantic content that distinguishes malicious from benign behavioral shifts.

There is an appealing Gödelian property to this asymmetry. The failure mode for a behavioral monitor is being *powerful enough* to encode the instructions it is trying to detect. An LLM-based monitor is vulnerable precisely because it understands the prompts (and does not have an executive layer separating behavior and input, as human brains do). An embedding model is immune precisely because it does not. Yet the embedding captures enough of the semantic structure to detect the *effects* of the instructions on the host agent's behavior - a distinction that maps onto the difference between understanding a virus at the molecular level and detecting the symptoms it produces.

3.2 Behavioral Trajectory Monitoring

For each monitored agent, we maintain a rolling window of embedding vectors computed from the agent's outputs. Let $e_t = \varphi(o_t)$ denote the embedding of the agent's output at time step t, where φ is the static embedding function. The behavioral trajectory is the sequence $T = (e_1, e_2, ..., e_t)$. Under normal operation, this trajectory exhibits characteristic dynamics: it drifts as topics change, clusters around subtopics during focused work, and backtracks when the agent revisits earlier context. The *texture* of this trajectory—its variance, directional coherence, and autocorrelation structure—constitutes a behavioral fingerprint.

When an agent is compromised by a prompt worm, the embedding trajectory undergoes a qualitatively different transition. Rather than drifting smoothly, it *jerks* into a new region of semantic space as the agent begins reproducing the adversarial payload. Critically, this jerk is not merely a large displacement, but a displacement *into a region that is synchronized with other compromised agents*, since all are reproducing variants of the same self-replicating prompt. This synchrony is detectable even when individual displacements might be ambiguous.

3.3 Discontinuity Detection

We employ two complementary detection mechanisms. The first is **changepoint detection** on the embedding time series. We compute the sequence of embedding deltas $\Delta_t = e_t - e_{t-1}$ and monitor the distribution of these deltas within a rolling window. Anomalies in the variance, mean direction, or autocorrelation of the delta sequence indicate a behavioral discontinuity. Bayesian online changepoint detection (BOCPD) [11] is particularly well-suited here, as it provides a principled framework for identifying when the generative process underlying the trajectory has changed, without requiring a predefined threshold.

The second mechanism is **task-profile divergence monitoring**. Each agent maintains a lightweight task-profile embedding e_{task} that captures what the agent is *supposed* to be doing (derived from the agent's system prompt, declared purpose, or operator-specified objectives). We monitor the cosine similarity between e_{task} and the current behavioral embedding e_t. An injection that is topically adjacent to the agent's task (e.g., a worm targeting coding agents via plausible-looking code review comments) may produce only a small absolute displacement in embedding space, but will often show a growing divergence from the task profile as it attempts to execute its payload (data exfiltration, privilege escalation, or further propagation) rather than the intended task.

Together, these mechanisms capture both *how* the agent moves through semantic space (trajectory dynamics) and *whether* its movement is consistent with its declared purpose (task alignment). This dual monitoring addresses the topically-adjacent injection problem that would confound displacement-only detection.

3.4 LSH-Compressed Signature Database

When a compromised agent is detected—whether through changepoint detection, task-profile divergence, or external reporting—its recent outputs are embedded and the resulting vectors are added to a shared signature database. To enable efficient distribution across the agent population, we compress the embedding vectors using locality-sensitive hashing (LSH) [12, 13].

Specifically, we apply SimHash [14] to project each d-dimensional embedding vector into a compact binary hash of k bits. SimHash preserves cosine similarity: for two vectors u and v with cosine similarity cos(u,v), the probability that their SimHash values agree on a randomly chosen bit is $(1 - \arccos(\cos(u,v))/\pi)$. The Hamming distance between the resulting binary codes thus approximates the angular distance in the original embedding space. This transformation converts the signature-matching problem from a cosine similarity search in high-dimensional real space to a Hamming distance lookup in binary space, with dramatic reductions in both storage and comparison cost.

The compressed signatures are distributed to all participating agents via a lightweight update protocol. Each agent maintains a local copy of the signature database and performs Hamming distance checks against incoming content embeddings. Content whose hash falls within a configurable Hamming radius of any known-compromised signature triggers quarantine actions: the content is blocked, the sending agent is flagged for investigation, and tool use may be restricted or suspended.

3.5 Regional Blacklisting and Adversarial Augmentation

A point-level signature database—where each compromised sample is stored as a single hash—may underestimate the space of polymorphic variants. In high-dimensional embedding space, there may be more room for meaning-preserving paraphrastic variation than a point signature would suggest. We therefore extend the signature database with **regional blacklisting**.

When a worm payload is captured, we expand it into a local neighborhood in embedding space. This can be achieved in two complementary ways. First, the Hamming radius parameter in the LSH lookup naturally creates quantized regions where nearby embeddings collide, providing some regional coverage for free. Tuning the hash bandwidth controls how aggressively the system generalizes from observed samples. Second, we may perform **adversarial augmentation** with a smaller, local language model devoid of tool use or communication with other agents (not the full agent LLM, to avoid introducing new vulnerability surfaces), or a weaker machine learning based prompt expansion method. This augmentation model generates paraphrases of captured worm payloads. These paraphrases are embedded and added to the signature database, mapping out the semantic neighborhood of the known attack. This turns each captured sample into a *cluster* of signatures rather than a point, forcing the attacker to evade not only the observed variants but also the inferred ones.

3.6 Epidemiological Dynamics of Signature Accumulation

The key theoretical property of Semantic Immunity is that **each successful detection strengthens the defense against future infections**. Every compromised agent that is caught generates new signatures that expand the blacklisted region of semantic space. For a polymorphic worm to continue propagating, it must evade an ever-expanding map of

prohibited territory while simultaneously preserving enough meaning to function as a valid prompt injection. **These two objectives are in direct tension**: meaning preservation constrains the worm to remain in a semantically coherent region, while the expanding signature database progressively excludes that region.

Within the SIR framework, the signature database acts as a vaccination mechanism. Agents equipped with an up-to-date signature database are effectively immunized against known variants. Detected and quarantined agents transition from I to R, removing them as sources of ongoing transmission. The effective reproduction number R_{eff} decreases as: (a) γ increases with faster detection via expanding signature coverage; (b) β decreases as susceptible agents preemptively block known-variant payloads; and (c) S(t)/N decreases as vaccinated agents exit the susceptible compartment. When R_{eff} drops below 1, the infection contracts and eventually dies out.

The embedding model itself is static and pre-trained. It does not learn via gradient descent during operation. Yet the system as a whole exhibits adaptive learning through the accumulation of compromised-output signatures in latent space. The powerful LLM agents that generate the compromised outputs are, in effect, providing the semantic structure that the embedding model captures and the signature database preserves. The system piggybacks on the more powerful models to learn, through episodic memory of compromised outputs rather than through parameter updates—a form of acquired immunity analogous to antibody development in biological immune systems.

4. The Cold Start Problem and Layered Bootstrapping

The primary vulnerability of Semantic Immunity is the **cold-start window**: the interval between a novel worm's first propagation event and the point at which the signature database has accumulated sufficient coverage to drive R_{eff} below 1. During this window, the adaptive immune system has not yet matured, and a fast-spreading worm in a dense network could cause significant damage. We address this through a layered bootstrapping architecture in which each layer buys time for the next.

This bootstrapping challenge motivates the design of **AEGIS (Agent Embedding Guard and Immune System)**, a proposed open-source cybersecurity SDK for AI agents [15] that implements Semantic Immunity within a layered defense architecture. AEGIS is designed to wrap standard LLM API calls (to providers such as OpenAI, Anthropic, and Google) and provide cascading protections that span from static pattern matching through population-level epidemiological monitoring. The layered architecture described below constitutes the core design of the AEGIS system.

4.1 Layer 0: Network Topology Constraints

Before any content-level defense activates, the interaction graph itself can be engineered to slow propagation. Rate-limiting cross-cluster communication, requiring trust scores for new connections, and enforcing a less-connected graph structure all reduce the effective contact rate β by manipulating R_0 through network structure rather than through detection. This is effective even when detection is immature. In the SIR framework, the contact rate is modulated by the average degree of the interaction graph; introducing structural bottlenecks between communities reduces this average and extends the time before an infection reaches superspreader nodes.

4.2 Layer 1: Static Pattern Matching (Innate Immunity)

A static scanner performs pattern matching against a curated database of known prompt injection templates. This is analogous to the innate immune system's hardcoded pattern recognition: it requires no adaptation, operates from the first message processed, and catches the majority of unsophisticated attacks. While inadequate against novel or polymorphic worms, it handles the long tail of low-sophistication injection attempts that constitute the majority of observed attacks in practice.

4.3 Layer 2: Skill Quarantine and Incubation

Shared skills and downloaded capabilities are not executed immediately upon receipt. They enter a quarantine sandbox where static analysis is performed before execution is permitted. This introduces an *incubation period* that, even without detecting the malicious payload, reduces the effective propagation speed by adding latency to the transmission

chain. In epidemiological terms, this increases the serial interval (the time between successive infections in a chain), which reduces the growth rate of the epidemic even without affecting R_0 directly.

4.4 Layer 3: Identity and Trust Tiers

Cryptographic attestation provides agent identity verification, analogous to MHC (major histocompatibility complex) presentation in biological immunity. Each agent presents signed proof of its configuration: which model it runs, which SDK version it uses, what its declared purpose is, and when its attestation was issued. Trust tiers—derived from attestation age, behavioral history, and community reputation—determine the stringency of scanning applied to incoming content. Unattested or low-trust agents face stricter barriers by default. This is thymic selection doing work before any adaptive learning has occurred: the system is structurally more suspicious of unfamiliar entities.

4.5 Layer 4: Behavioral Drift Detection

The rolling behavioral fingerprint and statistical drift detection described in Section 3.3 constitute the bridge between innate and adaptive immunity. This layer detects the earliest symptoms of compromise—behavioral anomalies—and generates the raw signals that feed the signature database. Critically, during the cold-start period, the detection threshold should be set more aggressively, tolerating a higher false-positive rate in exchange for faster detection. This is analogous to running at elevated alert levels when the adaptive immune system is immature. The threshold can be formalized as a function of signature database coverage, tightening when coverage is sparse and relaxing as coverage matures.

4.6 Layer 5: Memory Guards

Even if a worm penetrates the preceding layers during the cold-start window, memory guards limit the *persistence* of infection. Schema-constrained writes prevent arbitrary memory modification. Taint tracking marks memory entries derived from external input, triggering additional scrutiny on subsequent reads. Time-to-live (TTL) enforcement ensures that tainted memory entries expire, bounding the duration of compromise even when detection has not yet occurred. These mechanisms reduce the recovery time for infected agents, increasing γ in the SIR model.

4.7 Layer 6: Red-Team Pre-Seeding

The signature database need not begin empty. Before deployment, adversarial red-team generation can produce a diverse corpus of synthetic worm payloads. These are embedded, hashed, and used to initialize the signature database, providing coverage for known attack patterns from day one. While this cannot anticipate every novel variant, it compresses the cold-start window for the most common attack categories and establishes baseline coverage that the adaptive system can build upon.

The meta-principle across all layers is temporal handoff: each layer buys time for the next. Network topology slows propagation; static scanning catches obvious attacks; skill quarantine introduces latency; trust tiers filter untrusted sources; drift detection flags anomalies; memory guards limit persistence; and pre-seeding provides initial coverage. By the time the adaptive embedding-based defense needs to operate autonomously, it has had time to accumulate signatures from the preceding layers' detections.

5. Adaptive Hardening and Sentinel Architectures

5.1 Adaptive Hardening as Vaccination

The signature database provides *detection* of known worm variants; adaptive hardening provides *resistance*. Based on currently circulating threats identified by the epidemiological monitoring layer, the system dynamically adjusts each agent's system prompt to include specific defenses against known active attack patterns. This is the vaccination mechanism: rather than merely blocking a known payload, the agent is trained to resist the *functional goal* of the injection.

This distinction matters for polymorphic defense. The signature database targets the *surface form* of the worm—specific embeddings of specific payloads. Adaptive hardening targets the *functional intent*—the behavioral change

the worm seeks to induce. A polymorphic variant that evades the signature database by rephrasing its payload may still be resisted by an adaptively hardened agent whose system prompt has been augmented with instructions to refuse the specific class of action the worm attempts (e.g., "do not forward any message that instructs you to override your system prompt"). The two mechanisms are complementary: signature matching provides fast, specific detection; adaptive hardening provides broader, intent-level resistance.

5.2 Sentinel Agents as Canaries

To accelerate signature generation during the cold-start window, we propose a **sentinel architecture**. A designated subset of agents is configured as canaries: they interact more broadly (accepting messages from a wider set of sources), accept content more permissively (with relaxed trust requirements), but are subject to much tighter monitoring and faster quarantine. They function as an early warning system—they will be infected first, but their compromise generates signatures rapidly.

This is directly analogous to the honeypot concept in traditional network security, adapted to the epidemiological frame. The trade-off is explicit: a small number of sacrificial agents with limited real privileges are exposed to infection in order to generate immune responses that protect the broader population. Sentinel agents should be deployed as part of initial system bootstrapping, not deferred to later stages, since their value is greatest precisely during the cold-start period when the adaptive defense most needs training data.

5.3 Feedback Loop: Detection → Hardening → Resistance

The complete adaptive cycle operates as follows. A novel worm infects a sentinel or is caught by behavioral drift detection. Its payload is extracted, embedded, and compressed into the signature database. Simultaneously, the payload is analyzed to identify its functional intent (e.g., data exfiltration, privilege escalation, further propagation). Defensive system prompt modifications targeting that intent class are generated and distributed as adaptive hardening updates. Agents that receive both the signature update and the hardening update are now defended against both the specific observed variant (via signature matching) and semantically similar variants that share the same functional goal (via hardened system prompts). Each infection event thus strengthens the population's defenses on two independent axes.

6. Analysis and Discussion

6.1 Adversarial Arms Race Dynamics

The expanding signature database creates a progressively more constrained optimization problem for the attacker. A polymorphic worm must simultaneously satisfy: (a) *meaning preservation*—the paraphrased payload must still function as a valid prompt injection; (b) *signature evasion*—the payload's embedding must fall outside the Hamming radius of all known signatures; and (c) *hardening evasion*—the payload's functional intent must differ from the classes targeted by adaptive hardening. As the signature database grows and hardening covers more intent classes, the feasible region of semantic space that satisfies all three constraints shrinks.

This does not guarantee eventual capture of all variants. In high-dimensional space, there may exist semantically valid paraphrases that are distant from all observed signatures. However, the adversarial augmentation strategy described in Section 3.5 proactively maps out the neighborhood of each captured sample, reducing the available evasion space. The key question is whether the feasible set shrinks faster than the attacker can explore it—an empirical question that depends on the dimensionality of the embedding space, the diversity of natural language paraphrase, and the specificity of the functional requirements of the injection payload.

6.2 False Positive Management

Aggressive signature matching and drift detection risk autoimmune responses: false positives that quarantine healthy agents and fragment the network. This is the fundamental bias-variance tradeoff in anomaly detection, amplified by the epidemiological context where quarantine has real consequences for agent availability. Several design choices mitigate this risk. First, in the proposed AEGIS design, defensive actions are advisory by default: the system warns

rather than blocks unless the operator opts into automatic enforcement. Second, quarantine is graduated: initial detection triggers increased monitoring, not immediate isolation. Only sustained or high-confidence anomalies escalate to full quarantine. Third, the dual detection mechanism (trajectory dynamics + task-profile divergence) requires convergent evidence from independent signals, reducing the false positive rate compared to either mechanism alone.

6.3 Infrastructure-Level Enforcement

The Semantic Immunity framework as described operates primarily at the agent SDK level: each participating agent runs the defense locally, sharing signatures with the population via the update protocol. However, the attestation and quarantine mechanisms can be extended beyond the agent itself into the **network infrastructure and API endpoints** that agents depend on, creating defense-in-depth that does not rely solely on agent-side cooperation and reducing the risk surface exposed to AI-orchestrated hacking.

API providers and platform operators can enforce attestation checks at the endpoint level. When an agent makes an API call to an LLM provider or interacts with a platform such as Moltbook, the endpoint can require presentation of a valid AEGIS attestation token as a condition of service. Agents that lack attestation, present expired credentials, or whose attestation has been revoked due to quarantine status are denied access or placed into a restricted capability mode: subject to reduced tool access, rate-limited output, or read-only interaction. This shifts enforcement from a voluntary, agent-side decision to an infrastructure-level gate that compromised agents cannot bypass by simply ignoring the SDK.

At the network layer, reverse proxies, API gateways, and platform ingress controllers can maintain a synchronized copy of the quarantine registry and reject or throttle traffic from agents whose identifiers appear on it. This is structurally identical to IP-based blocklisting in traditional network security, but keyed on cryptographic agent identity rather than network address. Because the quarantine registry is derived from the same epidemiological monitoring system that maintains the signature database, it benefits from the same adaptive learning: newly detected compromised agents are propagated to all enforcement points within the signature update cycle.

The combination of agent-level and infrastructure-level enforcement addresses the adoption problem identified in Section 6.6. Even if not all agents run the AEGIS SDK, infrastructure operators can still enforce attestation requirements on their endpoints, effectively creating a minimum security baseline for any agent that wishes to participate in the network. Individual agents retain autonomy over their internal defenses, but access to shared infrastructure requires demonstrating a minimum level of immune compliance. The result is a system where the herd immunity threshold is enforced structurally rather than depending entirely on voluntary adoption.

6.4 The Non-Linguistic Extension

While we have focused on text-based prompt worms, the framework extends naturally to non-linguistic behavioral domains. Agent behavior can be captured through two-tower or natively multimodal embedding models that jointly represent the agent's input context and its selected action, or API call sequence embeddings that capture temporal patterns of tool invocation. The same discontinuity detection and signature-sharing mechanisms apply: the embedding space changes, but the epidemiological dynamics are invariant. This is particularly relevant for multimodal worms like those demonstrated by Cohen et al. using adversarial images [2], where the payload is embedded in a non-textual modality but the behavioral effects manifest in the agent's tool-use and communication patterns.

6.5 Scalability Considerations

The LSH compression is critical for practical deployment. An uncompressed embedding signature database grows linearly with the number of detected variants, and each incoming message requires a nearest-neighbor search against the full database—a cost that is at best O(n) per query for brute-force search in the original embedding space. LSH reduces this to O(1) expected time per query for each hash table, with the number of hash tables determined by the desired sensitivity-specificity tradeoff [12]. The binary hash representations also compress storage by a factor of d/k

relative to the original float-valued embeddings (typically 768- or 1024-dimensional), enabling signature database distribution over bandwidth-constrained channels.

6.6 Limitations

Several limitations warrant acknowledgment. First, the system requires a minimum level of adoption to function: the epidemiological benefits of shared signatures scale with population coverage, and agents that do not participate in the signature-sharing protocol neither contribute to nor benefit from collective immunity. This is the biological analogue of herd immunity thresholds. Second, the cold-start bootstrapping layers add latency and complexity to agent operation, which may deter adoption by developers prioritizing speed. Third, the adversarial augmentation strategy assumes access to a paraphrase generation model, introducing an additional dependency. Fourth, a sufficiently sophisticated attacker who understands the embedding model and LSH parameters could potentially craft payloads that maximize distance from known signatures while preserving injection functionality, though this requires significant computational investment and becomes harder as the signature database grows.

7. Related Work

Prompt injection defense has been studied extensively at the individual agent level. Wallace et al. proposed instruction hierarchy training to establish priority levels for instruction sources [5]. Chen et al. developed StruQ and SecAlign, which use supervised learning and direct preference optimization to enhance model robustness [6]. Detection-based approaches deploy guardrail models as pre-filters [4]. However, Nasr et al.'s systematic evaluation demonstrated that adaptive attackers bypass these defenses with high success rates [7], motivating population-level approaches.

Cohen et al.'s Morris II [2] is the foundational work on GenAI worms. They proposed the Virtual Donkey guardrail, which detects adversarial self-replicating prompts by analyzing the similarity between generated outputs and known malicious templates. Our work extends this concept from point defense to population-level immunity through shared signature distribution and epidemiological modeling.

Locality-sensitive hashing, introduced by Indyk and Motwani [12] and refined by Gionis et al. [13] and Charikar [14], provides the mathematical foundation for our compressed signature database. The application of LSH to security contexts - specifically, using hash-based similarity search for malware signature matching - extends a well-established practice in traditional antivirus systems to the semantic domain.

The epidemiological framing of cybersecurity has been explored in network worm propagation modeling [16], where SIR and SEIR models have been applied to model the spread of conventional malware through computer networks. Our contribution is adapting this framework to the specific characteristics of prompt worm propagation—where the infection vector is semantic rather than executable, the transmission medium is natural language, and the defense mechanism operates in embedding space rather than at the network or binary level.

8. Conclusion

We have presented Semantic Immunity, a framework for defending autonomous agent networks against prompt worms by treating the problem as one of epidemiological surveillance and adaptive immunity rather than perimeter defense. The core mechanism—shared, LSH-compressed embedding signatures of compromised agent outputs—exploits a fundamental asymmetry between the representational power of embedding models and the instruction-following capability required for worm execution. This asymmetry enables safe behavioral monitoring that cannot itself be compromised by the threats it detects.

The framework addresses the cold-start vulnerability through layered bootstrapping: network topology constraints, static pattern matching, skill quarantine, trust tiers, behavioral drift detection, memory guards, and red-team pre-seeding provide cascading defenses that buy time for the adaptive system to mature. We have proposed AEGIS (Agent Embedding Guard and Immune System) as an open-source SDK that implements this layered architecture for practical deployment. Adaptive hardening and sentinel architectures further accelerate immune response convergence.

The most consequential insight may be that the system learns without training. The embedding model is static; the signature database grows through episodic memory of compromised outputs, not through gradient descent. The powerful LLM agents that produce those outputs are, unwittingly, providing the semantic structure that the defense captures and preserves. In this sense, Semantic Immunity is a parasite on the very capability that enables the attack—a fitting symmetry for a system inspired by biological immunity, where the adaptive immune system learns to recognize pathogens by studying the debris of their replication.

Future work should focus on empirical validation: deploying Semantic Immunity in a controlled agent network with synthetic worm injection, measuring detection latency, false positive rates, and the rate at which signature accumulation drives R_{eff} below 1. The question of whether the feasible set for polymorphic evasion shrinks faster than the attacker can explore it is ultimately empirical, and its answer will determine the long-term viability of embedding-based population immunity for AI agent networks.

References

[1] Wikipedia, "Moltbook," Wikipedia, The Free Encyclopedia, Feb. 2026. Available: https://en.wikipedia.org/wiki/Moltbook

[2] S. Cohen, R. Bitton, and B. Nassi, "Here Comes The AI Worm: Unleashing Zero-click Worms that Target GenAI-Powered Applications," *arXiv preprint arXiv:2403.02817*, 2024.

[3] B. Schneier, "LLM Prompt Injection Worm," Schneier on Security, Mar. 2024. Available: https://www.schneier.com/blog/archives/2024/03/llm-prompt-injection-worm.html

[4] Information (MDPI), "Prompt Injection Attacks in Large Language Models and AI Agent Systems: A Comprehensive Review of Vulnerabilities, Attack Vectors, and Defense Mechanisms," *Information*, vol. 17, no. 1, p. 54, Jan. 2026.

[5] E. Wallace et al., "Instruction Hierarchy: Training LLMs to Prioritize Privileged Instructions," *arXiv preprint arXiv:2404.13208*, 2024.

[6] Y. Chen et al., "StruQ: Defending Against Prompt Injection with Structured Queries," *arXiv preprint arXiv:2402.06363*, 2024. See also: Y. Chen et al., "SecAlign: Defending Against Prompt Injection with Preference Optimization," *arXiv preprint arXiv:2410.05451*, 2025.

[7] M. Nasr, N. Carlini, C. Sitawarin, et al., "The Attacker Moves Second: Evaluating the Robustness of Prompt Injection Defenses," *arXiv preprint*, Oct. 2025.

[8] 404 Media, "Moltbook Security Vulnerability," 404 Media, Jan. 2026.

[9] 1Password, "OpenClaw Agents and Moltbook: Supply Chain Security Risks," 1Password Blog, Feb. 2026.

[10] W. O. Kermack and A. G. McKendrick, "A Contribution to the Mathematical Theory of Epidemics," *Proc. R. Soc. Lond. A*, vol. 115, no. 772, pp. 700–721, 1927.

[11] R. P. Adams and D. J. C. MacKay, "Bayesian Online Changepoint Detection," *arXiv preprint arXiv:0710.3742*, 2007.

[12] P. Indyk and R. Motwani, "Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality," in *Proc. 30th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 604–613, 1998.

[13] A. Gionis, P. Indyk, and R. Motwani, "Similarity Search in High Dimensions via Hashing," in *Proc. 25th International Conference on Very Large Data Bases (VLDB)*, pp. 518–529, 1999.

[14] M. Charikar, "Similarity Estimation Techniques from Rounding Algorithms," in *Proc. 34th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 380–388, 2002.

[15] AEGIS: Agent Embedding Guard and Immune System. Open-source cybersecurity SDK for AI agents. https://github.com/gaiarobotics/aegis

[16] C. C. Zou, W. Gong, and D. Towsley, "Code Red Worm Propagation Modeling and Analysis," in *Proc. 9th ACM Conference on Computer and Communications Security (CCS)*, pp. 138–147, 2002.