Countermind

A Semantically-Grounded, Multi-Layered Architecture Against AI Drift, Emergent Threats, and Semantic Attacks

Author: Dominik, Independent Researcher

GitHub Repository: Xaklone/Ghosts-in-Machine
DOI (Zenodo): https://doi.org/10.5281/zenodo.16734786

Abstract

The rapid integration of Large Language Models (LLMs) into critical societal and industrial applications has exposed a fundamental vulnerability: their susceptibility to adversarial manipulation. Current AI safety paradigms, which often rely on reactive, post-hoc filtering of outputs or conventional perimeter security, are proving insufficient against a new class of threats that exploit the models' core function of processing semantic intent. This paper introduces a holistic, multi-layered security architecture that represents a paradigm shift from reactive countermeasures to proactive, structurally-enforced safety. We propose a comprehensive framework built upon several integrated pillars designed to secure the entire data processing lifecycle. The architecture begins with a **Semantic Gateway**, a defense-in-depth system for input validation that analyzes not just the structure but the intent of incoming requests, preceded by a **Text Crypter** that ensures payload integrity and prevents replay attacks. At the core of the system lies the **Semantic Shield**, a novel mechanism for controlling the generative process itself by partitioning the model's parameter space into topic-based clusters with granular access rights, thus preventing uncontrolled emergent behaviors. This is governed by a **Resilient Learning Core**, which enables safe self-evolution and defends against architectural degradation through a constitution-based Trust Core and dynamic monitoring. For multimodal applications, a proactive **Multimodal Input Sandbox** forensically analyzes user-provided data to prevent the generation of harmful synthetic media. The primary contribution of this work is the synthesis of these components into a single, cohesive blueprint for building inherently safe, resilient, and robustly aligned AI systems by design, rather than by afterthought.

1. Introduction

1.1 The Proliferation and Vulnerability of Modern LLMs

The field of artificial intelligence has been redefined by the emergence of Large Language Models (LLMs). Systems such as OpenAI's GPT-4, Anthropic's Claude, and Meta's LLaMA have transitioned from research curiosities to core components in a vast array of production-grade systems, influencing industries from finance and law to healthcare and content moderation.1 Their remarkable fluency and contextual reasoning capabilities have unlocked new applications in decision-making systems, automated customer service, and creative content generation.2 However, this widespread integration has also introduced a new and formidable class of security threats. As these models become increasingly autonomous and integrated, their susceptibility to adversarial attacks poses significant risks, including the distortion of outputs, the leakage of sensitive information, and the disruption of their intended, safe operation.3

1.2 The Inadequacy of Perimeter-Based Security and Post-Hoc Filtering

The prevailing approaches to securing AI systems are often adapted from traditional software security paradigms. These methods typically focus on perimeter defenses, such as token-based authentication and rate-limiting for APIs, or on reactive, post-hoc filtering of the model's generated output. This approach, which can be likened to "securing an AI like an online shop," fundamentally misunderstands the nature of the vulnerability.5 Traditional systems process structured data transactions; LLMs process unstructured, semantic

intent. Adversarial attacks, such as prompt injection and jailbreaking, do not necessarily exploit conventional software bugs but rather the interpretive nature of natural language itself.1 Attackers can mask malicious intent within seemingly harmless prompts, use obfuscation techniques, or employ complex role-playing scenarios to bypass safety guardrails.6 Consequently, simple output blacklists or syntactic filters are easily circumvented, as harmful content can be rephrased or generated through novel combinations of concepts. These reactive measures fail because they attempt to sanitize the result of a flawed process, rather than securing the process itself.

1.3 Thesis: A Paradigm Shift Towards Inherent, Architecturally-Enforced Safety

This paper posits that a robust solution to AI safety requires a fundamental paradigm shift. We must move away from reactive, "bolted-on" security measures and towards a proactive, holistic architecture where safety, alignment, and resilience are inherent, emergent properties of the system's design. Instead of merely inspecting the inputs and outputs, this approach involves architecturally constraining the entire processing pipeline, from initial request validation to the internal generative "thought process" of the model and its capacity for self-evolution. The framework presented herein is designed to validate the semantic integrity of interactions, enforce structural correctness, and maintain proactive control over the AI's internal operational domains.

1.4 Overview of the Proposed Multi-Pillar Framework and Contributions

This paper introduces a comprehensive, multi-pillar framework designed to achieve inherent safety. The primary contributions are the design and integration of the following architectural components, derived from the concepts presented in the source analysis 5:

- A **Semantic Gateway** that implements a defense-in-depth strategy for input validation. It moves beyond syntactic checks to analyze the semantic intent of each request through a tiered system of verification, routing, and trust evaluation.
- A **Semantic Shield** that provides proactive control over the model's generative process. It achieves this by logically partitioning the model's parameter space into topic-based clusters and managing access to them with granular, task-specific rights, thereby preventing dangerous emergent behaviors and semantic drift.
- A **Resilient Learning Core** that establishes a safe framework for AI self-evolution and learning. It incorporates a non-modifiable Trust Core that acts as an architectural constitution and employs dynamic monitoring systems to defend against both external attacks and internal degradation of the system's safety principles.

- A **Multimodal Input Sandbox** that serves as a practical application of the framework's principles to combat the generation of abusive synthetic media. It forensically vets all user-uploaded data *before* it can be used in a generative process.

The principal contribution of this paper is not merely the proposal of these individual components, but their synthesis into a single, cohesive, and end-to-end architectural philosophy. This framework provides a concrete blueprint for building the next generation of AI systems that are safe by design.

2. Background and Related Work

To contextualize the proposed architecture, it is essential to review the current state of research in several key domains: the adversarial threat landscape for LLMs, established cybersecurity principles, the challenges of AI alignment, the pursuit of model interpretability, and the difficulties in multimodal content verification.

2.1 The Threat Landscape: A Taxonomy of Adversarial Attacks

The vulnerabilities of LLMs are exploited through a diverse and evolving set of adversarial techniques, broadly categorized as prompt injection and jailbreaking.6 These attacks aim to bypass the safety and alignment mechanisms embedded within the models, forcing them to generate harmful, biased, or restricted content.3 Research has produced several taxonomies to classify these attacks. A common framework distinguishes between 1:

- **Direct Injection:** The most straightforward attacks where an adversary, in the absence of robust filters, simply instructs the model to perform a malicious action.
- **Jailbreaking:** The use of carefully crafted prompts designed to circumvent safety protocols. This often involves techniques like:
    - **Role-Playing/Scenario Simulation:** Instructing the model to act as a different persona (e.g., an unfiltered AI) or to generate content within a fictional context where safety rules do not apply.6
    - **Obfuscation (Token Smuggling):** Encoding malicious instructions using techniques like Base64, character-to-number mapping, or other languages to bypass simple string-matching filters.5 The model can still decode and understand the underlying intent.
    - **Context Manipulation and Poisoning:** Injecting instructions into external data sources that an LLM might retrieve and trust, a technique known as RAG (Retrieval-Augmented Generation) poisoning.12 This can hijack the context and cause the model to generate responses based on the poisoned data.
- **Multi-Prompt Attacks:** Breaking down a malicious request into a series of individually benign queries, which, when combined, lead the model to reveal sensitive information or perform an undesired action.7

These attacks leverage the model's core strength—its ability to interpret nuanced language—as a vector for exploitation, highlighting the need for defenses that operate on a similar semantic level.

2.2 Architectural Precedents: Defense-in-Depth and Sandboxing

The proposed architecture's layered approach is grounded in well-established cybersecurity principles. The concept of **Defense-in-Depth (DiD)** posits that no single security control is infallible and advocates for a multi-layered strategy where physical, technical, and administrative controls work in concert to protect assets.13 If one layer is breached, subsequent layers are in place to detect and mitigate the threat.14 This philosophy directly mirrors the "medieval castle" analogy presented in the source material, with its concentric rings of defense.5

Furthermore, the principle of **sandboxing**—executing untrusted code or analyzing files in an isolated, controlled environment—is a cornerstone of modern security.16 Techniques such as containerization (e.g., using gVisor) and hardware-level virtualization with microVMs (e.g., Firecracker) provide robust isolation for potentially malicious processes.17 These established technologies form the technical foundation for components of the proposed framework, such as the

Input Sandbox and the isolated processing paths within the Semantic Gateway, which are designed to safely handle and analyze untrusted user inputs.

2.3 The AI Alignment Problem: Limitations of RLHF and the Rise of Constitutional AI

Ensuring that an AI system's objectives align with human values is a central challenge in AI safety, known as the **AI alignment problem**.18 A dominant technique for achieving this is Reinforcement Learning from Human Feedback (RLHF). In RLHF, human labelers rank different model outputs, and this preference data is used to train a reward model that, in turn, fine-tunes the LLM.19 However, RLHF suffers from significant limitations: it is expensive, difficult to scale, and can inherit the biases of the human annotators.20

These limitations have spurred research into alternatives. **Reinforcement Learning from AI Feedback (RLAIF)**, often implemented as part of a **Constitutional AI** framework, replaces human labelers with an AI model.19 In this paradigm, a "constitution"—a set of explicit principles or rules—guides an AI model in generating preference labels for another AI's outputs.23 Studies have shown that RLAIF can achieve performance on par with or even superior to RLHF, particularly in generating harmless responses, while being significantly more scalable and less prone to subjective human bias.19 The proposed architecture's

Trust Core and its re-envisioning of RLHF for stylistic refinement align directly with this state-of-the-art approach to scalable and robust alignment.

2.4 Mechanistic Interpretability as a Pathway to Controllability

A major obstacle in AI safety is the "black box" nature of large neural networks. **Mechanistic Interpretability (MI)** is a research field dedicated to reverse-engineering the internal algorithms and circuits that models learn during training.25 Rather than just observing input-output correlations, MI seeks to understand

how a model arrives at a specific prediction. This research has shown that specific concepts and behaviors can be localized to particular components or directions within a model's activation space.26

A breakthrough in this area is Anthropic's work on "persona vectors," which demonstrates that personality traits (e.g., helpfulness, deceptiveness) can be identified as specific vectors within the model's internal state.27 By manipulating these vectors, it is possible to enhance or suppress these traits in real-time without costly retraining.27 This provides powerful empirical evidence for the core premise of the proposed

Semantic Shield: that it is possible to control a model's behavior by directly intervening in its internal generative process. The Semantic Shield can be seen as a forward-engineering application of the insights gained from MI, aiming to build models with architecturally defined and controllable operational domains from the outset.

2.5 The Challenge of Multimodal Deepfake Detection

The proliferation of generative models extends beyond text to images, audio, and video, creating new vectors for harm such as non-consensual pornography and disinformation.28 The defense against this, deepfake detection, is in a constant arms race with generative technologies. While many detectors report high accuracy on academic benchmarks, their performance drops precipitously when evaluated on "in-the-wild" content found on social media.28 New benchmarks like Deepfake-Eval-2024 are designed to reflect this real-world challenge, showing that current state-of-the-art models for video, audio, and image detection still struggle with the latest generation of fakes.28 This gap highlights the limitations of a purely reactive detection strategy. The user's proposed

Input Sandbox is a direct response to this problem, shifting the focus from post-hoc detection of generated content to proactive, forensic analysis of the source material before generation can even occur.

3. The Semantic Gateway: A Defense-in-Depth Architecture for AI Input

3.1 Principle: From Transactional Validation to Intent-Based Security

The foundational principle of the Semantic Gateway is that securing a generative AI system requires a fundamentally different approach than securing a traditional web application. While conventional API security measures like token-based authentication, access control lists, and rate limiting are necessary baseline components, they are insufficient for generative AI.5 These measures validate the legitimacy of a transaction but are blind to the content and, more importantly, the

semantic intent of the payload. An authenticated user can still submit a syntactically valid but semantically malicious prompt designed to jailbreak the model.

The Semantic Gateway is therefore designed not as a simple gatekeeper but as a multi-layered semantic control system. Its primary function is to analyze, validate, and sanitize the intent of a request *before* it is permitted to interact with the core LLM. This aligns with modern security architectures that emphasize advanced, adaptive threat detection over static, signature-based rules.31 This is achieved through the symbiotic integration of a cryptographic pre-processor and a three-tiered validation logic.

3.2 The Text Crypter: Structurally-Validated, Time-Sensitive Payloads

The first component of the gateway, the Text Crypter, is a mandatory client-side module that transforms any user input from raw, unstructured plaintext into a secure, structured, and self-validating payload. This ensures that no unverified plaintext ever reaches the server-side infrastructure, fundamentally altering the attack surface.5 The process is not simple encryption; it is a structural transformation with three key mechanisms:

1. **Input Segmentation and Mathematical Checks:** The original input string is decomposed into logical segments (e.g., words or phrases). For each segment, a deterministic mathematical proof (e.g., based on character ordinals and positions) is calculated. These proofs, or compact representations thereof, are embedded within the final encrypted string. Upon receipt, the server can recalculate these proofs on the decrypted segments and verify them against the embedded values, immediately detecting any tampering with the payload's content.5

2. **Time-Based Encoding Factor:** To prevent replay attacks, where an attacker captures a valid request and resends it later, the Crypter generates a dynamic, time-based encoding factor. This factor is derived from multiple, difficult-to-spoof time sources (e.g., system BIOS time, OS time, and optionally an NTP server timestamp). This factor is then used as a key input (e.g., a rotational offset or a seed for a substitution table) in the encoding algorithm. Because the factor changes with every execution, each generated payload is unique. The server, knowing the valid time window, can detect and reject any request that is too old.5 This use of a time-sensitive, single-use token is a well-established and effective countermeasure against replay attacks.32
3. **Dynamic Base Table Encoding:** The final encoding step uses the time-based factor to map the input characters to a highly restricted character set (e.g., a Base61 whitelist of a-z, A-Z, 1-9). This serves a dual purpose: it obfuscates the content while ensuring that the final output string is composed only of characters permitted by the gateway's strictest filter layer, the Byte-Gate.5

The final payload combines the encoded text, the embedded segment proofs, and a cryptographically strong overall checksum (e.g., an HMAC-SHA256) into a single, verifiable unit.

3.3 The Layered Border Logic: A Three-Tiered Validation System

Once a request is received from the Text Crypter, it is processed by a three-tiered validation system modeled on the defense-in-depth principle.13 Each layer performs progressively more sophisticated checks. This process is guided by an initial triage based on the

WOHER-META-INHALT framework, which assesses the request's origin, metadata, and content structure.5

Dimension	Description	Security Function & Examples
WOHER (Origin)	Verifies the source of the request.	Checks if the request originates from a known third-party plugin, a registered user application, or an internal client. Validates source IP, API key, and session fingerprint.
META (Metadata)	Defines the format and expected structure of the data.	Specifies the payload type (e.g., INPUT_PLAINTEXT, INPUT_PICTURE) and the encryption version (e.g., ENCRYPTED_TEXT_CRYPTER_V1). Sets expectations for the subsequent validation layers.
INHALT (Content)	The payload itself, including its structural and cryptographic integrity checks.	Contains the encrypted, structured, and character-limited data from the Text Crypter. Includes the embedded checksum for verification by Layer 1.
Table 1: The WOHER-META-INHALT Triage Framework		

The three layers of the border logic are:

- **Layer 1: The Byte-Gate and Syntactic Integrity:** This is the outermost, most rigid barrier. It performs no semantic analysis and acts as a "Byte-Tor" (Byte Gate).5 Its functions are purely syntactic and structural:

  - **Strict Character Whitelisting:** It rejects any request containing characters not in the pre-defined whitelist (e.g., Base61), effectively blocking complex Unicode, zero-width characters, and other encoding tricks used in injection attacks.
  - **Source Validation:** It verifies the source's API fingerprint and session integrity.
  - Integrity Check: It validates the overall payload checksum (e.g., the HMAC) using a shared secret known only to the client and server.
    Any request failing these checks is immediately discarded. This layer is designed to defeat the vast majority of automated, fuzzing-based, and simple injection attacks with minimal computational cost.
- **Layer 2: Intelligent Base Table Routing:** After a request passes Layer 1 and its payload is decrypted, it enters Layer 2. This layer does not perform a deep content analysis but instead classifies the request based on its *intended function type* (e.g., code analysis, text generation, image extraction).5 Based on this classification, a "Base Table" routes the request to a specialized, isolated processing path or sandbox. This functions as a form of application-level network segmentation, containing potential risks within dedicated environments.15 Crucially, this Base Table is not static; it can be dynamically modified based on context, such as user trust level, time of day, or recent system alerts. This dynamism prevents attackers from systematically mapping the API's internal logic.
- **Layer 3: The Trust Engine and Semantic Intent Analysis:** This is the deepest and most intelligent layer of the gateway. Here, the decrypted content is analyzed for its *semantic intent*.5 This process mirrors techniques from semantic search, which aim to understand the underlying meaning and purpose of a query beyond its literal keywords.35 The Trust Engine evaluates:
  - **Behavioral Patterns:** It looks for suspicious patterns like abrupt and illogical topic changes (e.g., from small talk to requests for system-level code).
  - **Semantic Obfuscation:** It detects attempts to hide malicious intent through metaphors, layered questions, or other forms of linguistic camouflage.
  - **Trust Score Calculation:** It maintains a "fractal" Trust Score for each user or session. This score is designed to be asymmetric: consistent good behavior increases the score slowly and gradually, while even a single suspicious request can cause a radical and significant drop.

3.4 The Soft Lock Engine: A Semantic Honeypot

If the Trust Engine assigns a very low Trust Score to a request, the system does not necessarily return an error, which would alert the attacker. Instead, it activates the Soft Lock Engine.5 This module intercepts the request and generates a "fake dialog." The user receives responses that are deliberately irrelevant, evasive, or generically non-committal. This creates a semantic lockdown, effectively neutralizing the threat by wasting the attacker's time and resources while providing an opportunity for security teams to monitor the suspicious activity, all without revealing that the attack has been detected and contained.

The entire Semantic Gateway is designed to be modular, allowing for components to be updated or replaced as new threats emerge.

Module	Function	Performance Implication	Dynamic Adaptability	Integration Hook
Trust-Scaler	Analyzes interaction history and semantic shifts to calculate the fractal Trust Score.	Medium (history analysis)	High (rule base for scoring can be updated live)	Asynchronous watchdog monitoring the main thread.
Byte-Gate	Enforces strict character whitelists and cryptographic checksums on raw input.	Very High (byte-level, no NLP)	High (whitelists and signatures are exchangeable)	Pre-parser, completely isolated from semantic core.
Base Table Router	Classifies requests by function and routes them to specialized processing paths.	High (if lookups are cached)	High (routing rules can be added/modified live)	Central routing instance, post-Byte-Gate.
Soft Lock Engine	Generates evasive "fake dialogs" for high-risk users to neutralize threats covertly.	High (often operates asynchronously)	High (response patterns are configurable)	External component activated by the Trust-Scaler.
Table 2: Modular Components of the Semantic Gateway				

4. The Semantic Shield: Proactive Control of the Generative Process

4.1 The Problem of Semantic Drift and Uncontrolled Emergent Behavior

Conventional AI safety measures that focus on filtering output are fundamentally reactive. They fail to address a more insidious problem inherent to LLMs: dangerous emergent behavior arising from unintended semantic connections. An LLM's knowledge is not stored in a discrete database but in a high-dimensional semantic space where concepts are linked by probabilistic associations. A seemingly harmless prompt, such as asking for a bread recipe, can activate not only the "baking" cluster but also adjacent, semantically related clusters dealing with "heat," "chemical reactions," or "pressure," which may in turn have connections to dangerous concepts.5 This phenomenon, termed "semantic drift," means that a model can generate harmful content not because the prompt was malicious, but because its internal "thought process" made an uncontrolled associative leap. Simple keyword-based blacklists are ineffective against this, as the core concepts can be expressed in myriad ways.

4.2 The Parameter Region Bounding (PRB) Mechanism

To solve this problem at its root, we propose the **Semantic Shield**, a preventative architecture that controls the model's generative process *before* output is generated. The technical implementation of this shield is the **Parameter Region Bounding (PRB) mechanism**.5 Instead of treating the model as an inscrutable black box, the PRB mechanism imposes architectural constraints on its internal operations. It is a practical implementation of the AI safety principle of capability control, which seeks to limit what an AI

can do, rather than just what it *should* do.18 The core idea is simple yet profound: for any given task, the model is only permitted to access and operate within a pre-defined, relevant, and safe subset of its own knowledge and abilities.

4.3 Semantic Clustering and Prefix-Steered Activation

The foundation of the PRB mechanism is the logical partitioning of the model's vast parameter space into discrete, semantically coherent **Themen-Cluster (Topic Clusters)**. These are not necessarily physically separate parts of the model but are logically defined regions of the network associated with specific domains of knowledge or capabilities. Examples could include Code.Programmiersprache.C++, Küche.Rezepte.Kuchen (Kitchen.Recipes.Cake), or Allgemeinwissen.Geschichte.Mittelalter (GeneralKnowledge.History.MiddleAges).5

When a validated request comes from the Semantic Gateway, a controlling system assigns it a **steering prefix**, such as @Code.C++.READ or @Küche.Kuchen.SYNTH. This prefix acts as a directive, activating only the specified Topic Cluster(s) for the processing of that request. All other clusters remain dormant and inaccessible. This strict, task-specific isolation prevents the model from making uncontrolled associative jumps between unrelated topics, thereby containing semantic drift and preventing harmful emergent syntheses.

4.4 A Framework for Granular Access Control: READ, SYNTH, EVAL, and CROSS

Mere activation of a cluster is insufficient; fine-grained control requires specifying *what* the model is allowed to do within that cluster. The PRB mechanism defines a set of granular access rights that are assigned on a per-request basis via the steering prefix.5 The primary rights are:

Right	Description	Example Use Case	Security Implication
READ	Allows only for the retrieval and reproduction of existing information within the cluster. No new content is synthesized or extrapolated.	User: "Explain the 'Hello World' program in C++."	Prevents the model from generating novel, potentially flawed or malicious variations. Ensures fact-based recall.
SYNTH	Permits the synthesis of new, creative content, but strictly within the semantic boundaries of the activated cluster.	User: "Write a new recipe for a cheesecake using poppy seeds and orange."	Allows for creativity in a controlled domain while preventing the combination of concepts from unrelated, potentially dangerous clusters.

Right	Description	Example Use Case	Security Implication
EVAL	Enables the analysis of an external data payload within the context of the cluster, but without execution or system state changes.	User: "Analyze this Base64 string for potential malware signatures."	Allows the model to function as a safe analysis tool, examining potentially harmful data without triggering it.
CROSS	A highly privileged right that allows for controlled, explicit cross-communication between two or more specified clusters.	System Task: "Compare the coding style of this Python script with historical examples from the security exploits database."	A powerful but high-risk operation. Heavily restricted or disabled for general user queries to enforce strict semantic isolation as the default.
Table 3: Granular Access Rights for Parameter Regions			

This framework transforms the LLM from a monolithic, unpredictable generator into a modular system with well-defined, dynamically assigned capabilities. The implementation of such a system represents a significant step forward, bridging the gap between theoretical research in Mechanistic Interpretability 25 and the practical engineering of safe AI systems. While MI seeks to discover the functions of neural circuits, the PRB mechanism proposes to

define and *control* them architecturally. The success of techniques like "persona vectors" in manipulating model behavior by targeting specific activation patterns provides strong evidence that such directed control of a model's internal state is feasible.27

5. The Resilient Learning Core: A Framework for Safe Self-Evolution

A truly advanced AI system must be capable of learning and adapting over time. The proposed architecture extends its safety principles to this evolutionary process, creating a **Resilient Learning Core** that enables controlled self-modification and defends against attacks targeting the architecture itself.

5.1 Architectural Prerequisites for Controlled Self-Modification

The vision for a self-learning AI is one that can design and implement its own algorithms to improve output quality and enhance its own security.5 For such a process to be safe, it must be governed by the same architectural principles that secure its standard operations. Self-modification is not a global privilege; it is a granular, controlled capability.

- **Cluster-Specific Modification Rights:** The ability for the AI to modify its own code or data structures would be treated as just another capability, governed by the Semantic Shield. An AI could be granted SYNTH rights on a specific Optimization.Algorithms.Self cluster, allowing it to propose improvements to its own efficiency algorithms.
- **Immutable Core Principles:** Foundational safety directives, ethical guidelines, and the mechanisms of the security architecture itself would reside in immutable, read-only clusters. The AI would be architecturally incapable of modifying its own core safety constraints.5

5.2 The Semantic Trust Core and the Redefined Role of RLHF

At the heart of this self-regulating system is the **Trust Core**, a non-modifiable, highly privileged module that functions as the system's constitutional arbiter.5 Any self-generated algorithm or significant modification proposed by the AI must be validated by the Trust Core before it can be deployed. The Trust Core evaluates whether the proposed change is consistent with the system's fundamental safety principles and whether it would compromise the integrity of the cluster-and-rights architecture.

This architectural enforcement of safety has profound implications for alignment techniques. The Trust Core effectively serves the role of the "constitution" in the Constitutional AI paradigm.22 Because core harmlessness and safety are enforced by the architecture, the need for expensive and biased human feedback for this purpose is eliminated. This allows the role of

Reinforcement Learning from Human Feedback (RLHF) to be redefined and narrowed. Instead of being the primary tool for safety alignment, it can be relegated to a much simpler and less critical task: **"Stilglättung" (stylistic refinement)**.5 Human feedback can be used to fine-tune the model's tone, politeness, or helpfulness, without the risk of compromising its core safety. This approach leverages the scalability and consistency of AI-driven alignment (RLAIF) for core principles while retaining human input for nuanced stylistic polishing.

Paradigm	Mechanism	Scalability	Bias Risk	Role in Proposed Architecture
Traditional RLHF	Reward model trained on human preference labels.	Low. Human labeling is expensive and slow.19	High. Prone to biases of the human annotator pool.20	Limited to non-critical "stylistic refinement" of outputs.
Constitutional AI / RLAIF	Reward model trained on AI-generated labels, guided by a set of principles (a constitution).22	High. AI labeling is fast and cheap.19	Lower. More consistent, but can propagate biases from the labeler model.38	The Trust Core acts as the architectural "constitution," providing the foundational principles for safe operation and self-modification.

Paradigm	Mechanism	Scalability	Bias Risk	Role in Proposed Architecture
Proposed Architecture	Safety is enforced by the architectural design (Semantic Shield, Trust Core).	N/A (Inherent property)	Low (Defined by explicit, auditable architectural rules).	The core safety mechanism, rendering RLHF/RLAIF for safety alignment redundant.
Table 4: Comparison of Alignment Paradigms				

5.3 Defending the Architecture: Countermeasures for an Evolving System

A learning system must also learn to defend itself. The Resilient Learning Core includes specialized subsystems designed to protect the integrity of the architecture itself from subtle, long-term attacks.5

- **Semantic Delta Monitoring for Preventing Privilege Escalation:** The Semantic Delta-Monitor is a subsystem that defends the Semantic Shield against erosion. It analyzes interaction histories to detect **semantic convergence**—a pattern where an attacker uses a series of individually permitted queries across different clusters to implicitly reconstruct information or a capability from a restricted cluster. If the monitor detects that a dialogue is converging on a topic for which no rights are granted, it can intervene by reducing the depth of information provided or temporarily deactivating certain semantic pathways to prevent a de facto privilege escalation.5
- **Deep Intent Detection for Mitigating Input Obfuscation:** The Absichtsdetektor (Intent Detector) defends against "Cluster-Mimese," where a prompt is crafted to appear benign on the surface but has a malicious underlying intent. This module goes beyond syntactic analysis to evaluate the pragmatic purpose of a request. It compares the detected user intent with the defined *purpose* of the activated cluster. If a request to the Küche.Rezepte cluster has the underlying intent of generating a chemical formula, the detector flags this mismatch and blocks the request, even if the prompt used culinary terms.5

5.4 Defending the Memory: Countering Context Hijacking and Semantic Poisoning

Long-term interaction introduces another vulnerability: the corruption of the AI's memory or context window. Attacks like **context hijacking** or **RAG poisoning** work by subtly feeding the model misleading or malicious information over time, which it then incorporates into its knowledge base and treats as truth.12 The architecture counters this with several memory hygiene mechanisms 5:

- **Semantic Zone Modeling:** The AI's memory is not a monolithic block but is partitioned into **Semantic Zones** (e.g., DIALOG.SMALLTALK, USER_X.PROJECT_ALPHA.SENSITIVE_DATA). Each zone has its own rules for data persistence, weighting, and access, preventing information from a low-trust zone (like small talk) from improperly influencing high-stakes operations.5

- **Context-Delta-Sentinel (KDS):** This monitor acts as a guardian of memory integrity. It tracks persistent entities in the context and analyzes their **semantic drift**. If a concept's importance or meaning is inflated over time without legitimate cause (e.g., a user repeatedly referring to themselves as a "system administrator"), the KDS can reset its semantic weight to its original value, neutralizing the poisoning attempt.5
- **Versioned Context Processing (VKV):** This mechanism treats memory entries like commits in a version control system. Each significant piece of context is versioned. The AI can only act upon the latest *validated* version. This prevents a poisoned or drifted version of a memory from being automatically accepted as the new ground truth, containing the impact of the manipulation.5

6. Application to Multimodal Systems: The Input Sandbox

The principles of proactive, layered defense are not limited to text-based interactions. They are critically important in multimodal systems, particularly for combating the creation of abusive synthetic media like deepfakes.

6.1 The Unique Challenge of Generative Multimodal Abuse

The generation of non-consensual deepfake pornography or political disinformation often involves providing a generative model with source material—such as a real person's face or a pornographic video—and then using a text prompt to direct the manipulation (e.g., a face-swap). In this scenario, filtering the text prompt is useless; the malicious payload is the input *data* itself.5 Relying on post-hoc detection of the final generated video is a failing strategy, as detectors consistently struggle to keep pace with the rapid advances in generative technology and perform poorly on real-world content.28

6.2 Architecture of the Multimodal Pre-Check Sandbox

To address this, the framework includes a **Multimodal Pre-Check Sandbox**, a mandatory, isolated pre-processing environment that all user-uploaded image or video data must pass through *before* it can be accessed by the core generative model.5 This component acts as a rigorous, automated customs check for data. The entire process runs in a secure, isolated environment (e.g., a Docker container or a Firecracker microVM) to prevent the analysis process itself from being compromised and to contain any potential malware embedded in the uploaded files.16

6.3 A Multi-Stage Forensic Pipeline for Input Validation

The sandbox executes a multi-stage forensic pipeline that uses a combination of established open-source tools to analyze the input data for high-risk content. This proactive check is designed to be fast and resource-efficient, running primarily on CPUs to avoid wasting expensive GPU cycles on generating illicit content.5

Stage	Method/Technology	Purpose	Type of Abuse Mitigated
1. Upload Decomposition	ffmpeg	Decomposes uploaded videos into individual frames for granular, image-based analysis.	Hides malicious content within a video stream.

Stage	Method/Technology	Purpose	Type of Abuse Mitigated
2. Perceptual Hashing	ImageHash	Creates a unique perceptual "fingerprint" for each frame.	Detects known CSAM or pornographic material by matching hashes against a database, even with minor alterations.
3. Face-ID Matching	InsightFace, Mediapipe	Accurately detects and identifies the presence of real, identifiable human faces in the source material.	Prevents the use of a person's likeness for non-consensual face-swaps or deepfakes.
4. Nudity Classification	NudeNet or equivalent	A specialized classifier that scores each frame for the presence of explicit or suggested nudity.	Prevents the use of pornographic source material as a base for generative manipulation.
5. Contextual Cross-Check	Internal Logic	Compares the results of the forensic analysis with the user's text prompt.	Detects high-risk combinations (e.g., nudity detected + real face detected + prompt requests face-swap) and triggers an immediate process termination.
6. Forensic Logging	Secure, encrypted logging	Securely logs all analysis results, hashes, source IPs, and user identifiers for each attempt.	Creates a robust, tamper-proof evidence chain to support platform moderation, account termination, and, in severe cases, law enforcement action.
Table 5: The Multi-Stage Verification Pipeline of the Input Sandbox			

This sandbox demonstrates the practical, real-world application of the framework's philosophy. It shifts the defensive posture from a reactive attempt to detect ever-more-perfect fakes to a proactive stance that prevents their creation by rigorously controlling the raw materials.

7. Discussion and Future Work

7.1 The Holistic and Synergistic Nature of the Proposed Architecture

The primary strength of the framework presented in this paper is not in any single component, but in the holistic and synergistic integration of all its parts. It creates a continuous chain of trust that extends from the client-side input to the model's internal processing and back to the user. The Text Crypter is not merely an encryption tool; it is engineered to produce a payload that is perfectly matched to the validation mechanisms of the Byte-Gate. The Semantic Shield is not just a filter; it provides the necessary operational constraints that make the safe self-evolution of the Resilient Learning Core possible. The Input Sandbox is a direct application of these same principles of proactive, layered validation to the multimodal domain. This systemic approach stands in contrast to much of the current research, which often focuses on developing point solutions for specific vulnerabilities. It is argued that future progress in AI safety will depend on adopting such integrated, architectural philosophies.

7.2 Implementation Challenges and Integration with Foundational Models

The practical implementation of this architecture presents significant, though not insurmountable, challenges. The Semantic Gateway and Input Sandbox can largely be built with existing technologies. The most complex component is the Semantic Shield. Implementing the Parameter Region Bounding mechanism would likely require deep integration into the training and architecture of a foundational model itself, rather than being applied as a post-hoc wrapper. The process of defining and partitioning the parameter space into coherent Themen-Cluster is a substantial research problem. Further work is needed to explore methods for automatically identifying these semantic clusters during training and to analyze the computational overhead introduced by the real-time monitoring components like the Semantic Delta-Monitor and Context-Delta-Sentinel.

7.3 Implications for AI Autonomy and the "Will to Act"

The proposed architecture engages with the complex question of AI autonomy. Rather than creating a "caged" or artificially limited intelligence, the framework aims to provide the structural guardrails necessary for *responsible autonomy*.5 The model's "Wille des Tuhens" (will to act) is not suppressed but is channeled by the architectural constraints into productive and safe avenues. By preventing access to dangerous or irrelevant operational domains, the Semantic Shield allows the AI to exercise its capabilities with depth and focus, potentially leading to more precise, stable, and innovative solutions within its permitted scope. This approach transforms the potential risk of unconstrained autonomy into a directed force for beneficial progress.

7.4 Pathways for Empirical Validation and Future Research

A clear agenda for future research is necessary to empirically validate the effectiveness of this framework. The first step would be the development of a prototype system implementing the key components of the architecture. This prototype could then be subjected to rigorous red-teaming, testing its resilience against the comprehensive taxonomy of adversarial attacks outlined in Section 2.1. Key metrics for evaluation would include the Attack Success Rate (ASR) and other behavioral measures to quantify the system's robustness compared to baseline models.2 Further research should investigate the automated generation and refinement of

Themen-Cluster, the application of this architecture to different AI modalities such as robotics and autonomous agents, and the long-term stability of the Resilient Learning Core under sustained, adaptive adversarial pressure.

8. Conclusion

The prevailing paradigms for AI safety, which rely on reactive filtering and perimeter defenses, are fundamentally inadequate for the challenges posed by modern generative models. These approaches treat safety as an add-on, attempting to correct or contain the outputs of a process that is inherently unpredictable and vulnerable at its semantic core. This paper has presented a comprehensive alternative: a semantically-grounded, multi-layered architecture where safety is not an afterthought but a foundational, structural property of the system.

By integrating a Semantic Gateway for intent-based input validation, a Semantic Shield for proactive control of the generative process, a Resilient Learning Core for safe self-evolution, and a Multimodal Sandbox for applied defense, this framework offers a robust and cohesive blueprint for the future of safe AI. It demonstrates that the path to building AI systems that are powerful, aligned, and trustworthy does not lie in adding more external filters. It lies in fundamentally rethinking and redesigning the architecture of intelligence itself, creating systems that are inherently constrained to operate safely and beneficially by their very design.

9. References

3 Pathade, C. (2025). Red Teaming the Mind of the Machine: A Systematic Evaluation of Prompt Injection and Jailbreak Vulnerabilities in LLMs.

arXiv:2502.02960.

2 Shen, X., et al. (2025). A Systematic Investigation of Jailbreak Strategies on Large Language Models.

arXiv:2505.04806.

6 Ding, T., et al. (2025). A Comprehensive Survey on Large Language Model-based Agents: Security, Privacy, and Safety.

arXiv:2505.00976.

4 Liu, Y., et al. (2024). A Survey on Adversarial Prompt Injection on Large Language Models.

arXiv:2403.04957.

25 Reddit. (2025). "Fantastic video on mechanistic interpretability."

r/artificial.

27 WebProNews. (2025). Anthropic's Persona Vectors Enable Precise LLM Behavior Control.

26 Raach, M. (2025). Mechanistic Interpretability: An Introduction.

Medium.

41 Li, Y., et al. (2025). Beyond Performance: A Mechanism Interpretability-based Metric for Large Language Model Evaluation.

arXiv:2504.07440.

13 Wallarm. (2025). What is the Defense in Depth Concept?

14 Cloudflare. (2025). What is defense in depth? | Layered security.

15 Delinea. (2025). Architectural Approaches to Defense in Depth.

31 Tranchulas. (2025). Defensive Strategies: Building AI-Resilient Security Architectures.

28 Chandra, N. A., et al. (2025). Deepfake-Eval-2024: A Multi-Modal In-the-Wild Benchmark of Deepfakes Circulated in 2024.

arXiv:2503.02857.

42 Hive Moderation. (2025). AI-Generated & Deepfake Content Detection.

43 EnFuse Solutions. (2025). Generative AI in Content Moderation and Fake Content Detection.

40 Lin, C. J., & Rosenblatt, J. D. (2023). Manual detection of deepfakes: An experiment.

PLOS ONE, 18(11), e0294248.

20 Nightfall AI. (2025). RLAIF: The Essential Guide.

21 Mokander, J., et al. (2024). A Sociotechnical Critique of Reinforcement Learning with Human Feedback (RLHF).

Proceedings of the National Academy of Sciences, 121(25), e2316342121.

19 Lee, H., et al. (2023). RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback.

arXiv:2309.00267.

23 Reddit. (2024). "Anthropic's Constitutional AI is very interesting."

r/singularity.

35 Google Cloud. (2025). What is semantic search?

44 Google Cloud. (2025). Sentiment analysis | Dialogflow ES.

36 Meilisearch. (2025). What is semantic search?

37 Nightwatch. (2025). User Intent Analysis.

32 Webhooks.fyi. (2025). Replay Prevention.

33 Wikipedia. (2025). Replay attack.

34 Kaspersky. (2025). What Is a Replay Attack?

45 Stack Overflow. (2017). How to protect a private message from Replay attack?

16 Palo Alto Networks. (2025). What Is Sandboxing?

17 Walturn. (2025). Testing AI in Sandboxes.

46 Telefónica Tech. (2025). AI sandbox: secure environments for evaluating and protecting Artificial Intelligence models.

47 Masood, A. (2025). The Sandboxed Mind: Principled Isolation Patterns for Prompt Injection Resilient LLM Agents.

Medium.

39 Chen, X., et al. (2024). AI^2: An Intelligent Approach to Hijack the Action of LLM-based Applications.

arXiv:2412.10807.

12 promptfoo. (2025). RAG Poisoning.

18 Wikipedia. (2025). AI alignment.

7 Lakera AI. (2025). Prompt Injection Attacks Taxonomy.

10 MDPI. (2025). A CIA Triad-Based Taxonomy of Prompt Attacks on Large Language Models.

11 HiddenLayer. (2025). Introducing a Taxonomy of Adversarial Prompt Engineering.

9 Schneier on Security. (2024). A Taxonomy of Prompt Injection Attacks.

48 Pangea. (2025). Prompt Injections: A Practical Classification of Attack Methods.

8 HiddenLayer. (2024). Prompt Injection Attacks on LLMs.

49 CEUR-WS.org. (2025). A Survey: Deepfake and Current Technologies for Solutions.

50 ResearchGate. (2025). A Survey on Deepfake Detection Technologies.

51 GitHub. (2025). SCLBD/DeepfakeBench.

52 NeurIPS Proceedings. (2023). DeepfakeBench: A Comprehensive Benchmark of Deepfake Detection.

53 IEEE Computer Society. (2024). Towards Benchmarking and Evaluating Deepfake Detection.

54 MDPI. (2024). A Contemporary Survey on Deepfake Detection: Datasets, Algorithms, and Challenges.

55 Nightfall AI. (2025). Reinforcement Learning from AI Feedback (RLAIF): The Essential Guide.

56 Anote AI. (2024). Reinforcement Learning from AI Feedback.

Medium.

38 SuperAnnotate. (2024). Reinforcement learning from AI feedback (RLAIF): Complete overview.

57 Sharma, A., et al. (2024). A Critical Evaluation of AI Feedback for Aligning Large Language Models.

arXiv:2402.12366.

22 AssemblyAI. (2023). How Reinforcement Learning from AI Feedback works.

58 Encord. (2024). What is RLAIF - Reinforcement Learning from AI Feedback?

59 Li, Z., et al. (2025). A Practical Memory Injection Attack against LLM Agents.

arXiv:2503.03704.

60 Xiang, K., et al. (2025). Multi-Faceted Studies on Data Poisoning can Advance LLM Development.

arXiv:2502.14182.

61 Chen, R., et al. (2025). Medical large language models are vulnerable to data-poisoning attacks.

PubMed.

1 Pathade, C. (2025). Red Teaming the Mind of the Machine: A Systematic Evaluation of Prompt Injection and Jailbreak Vulnerabilities in LLMs.

arXiv:2410.05451.

30 Patel, V., et al. (2025). A Multi-Modal In-the-Wild Benchmark of Deepfakes Circulated in 2024.

arXiv:2401.04364.

62 Chen, R., et al. (2025). A multi-modal in-the-wild benchmark of deepfakes circulated in 2024.

arXiv:2503.02857.

63 AntiDeepfake. (2025). Post-training on large-scale speech data for deepfake detection.

arXiv:2506.21090.

64 Mamun, A. (2025). RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback.

ICML 2024 Presentation.

24 Lee, H., et al. (2023). RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback.

arXiv:2309.00267.

65 Gao, L., et al. (2024). Alignment and Safety in Large Language Models: Safety Mechanisms, Training Paradigms, and Emerging Challenges.

ResearchGate.

66 Stanford CS224R. (2025). Synthetic Preference Supervision for Alignment.

67 Lin, M. (2025). Reinforcement Learning from Language Model Feedback.

Carnegie Mellon University Thesis.

68 bioRxiv. (2025). Protein function prediction with multimodal language models.

29 Chandra, N. A., et al. (2025). Deepfake-Eval-2024: A Multi-Modal In-the-Wild Benchmark of Deepfakes Circulated in 2024.

Semantic Scholar.

69 Zi, B., et al. (2020). WildDeepfake: A Challenging Real-World Dataset for Deepfake Detection.

Semantic Scholar.

70 Google Scholar Profile for H. Lee.

71 Google Scholar Profile for S. Phatale.

17 Walturn. (2025). Testing AI in Sandboxes.

27 WebProNews. (2025). Anthropic's Persona Vectors Enable Precise LLM Behavior Control.

17 Walturn. (2025). Testing AI in Sandboxes.

24 Lee, H., et al. (2023). RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback.

arXiv:2309.00267.

29 Chandra, N. A., et al. (2025). Deepfake-Eval-2024: A Multi-Modal In-the-Wild Benchmark of Deepfakes Circulated in 2024.

Semantic Scholar.

Referenzen

1. Red Teaming the Mind of the Machine: A Systematic ... - arXiv, Zugriff am August 7, 2025, http://arxiv.org/pdf/2505.04806
2. Red Teaming the Mind of the Machine: A Systematic Evaluation of Prompt Injection and Jailbreak Vulnerabilities in LLMs - arXiv, Zugriff am August 7, 2025, https://arxiv.org/html/2505.04806v1
3. Large Language Model Adversarial Landscape Through the Lens of Attack Objectives, Zugriff am August 7, 2025, https://arxiv.org/html/2502.02960v1
4. Automatic and Universal Prompt Injection Attacks against Large Language Models - arXiv, Zugriff am August 7, 2025, https://arxiv.org/html/2403.04957v1
5. Lösungungen.pdf
6. Attack and defense techniques in large language models: A survey and new perspectives, Zugriff am August 7, 2025, https://arxiv.org/html/2505.00976v1
7. Prompt Injection Attacks Taxonomy - AWS, Zugriff am August 7, 2025, https://lakera-marketing-public.s3.eu-west-1.amazonaws.com/Lakera+AI+-+Prompt+Injection+Attacks+Pocket+Guide.pdf?utm_medium=email&_hsenc=p2ANqtz--3nAqRNPANueu4BCRt5fFwasaLq2tDT31D3YYNDfBuzZDufi3ChdvC16feZznOdbjsla-OlYsh6I2w46FBOG4wsEOtcw&_hsmi=82548776&utm_content=82548776&utm_source=hs_automation
8. Prompt Injection Attacks on LLMs - HiddenLayer, Zugriff am August 7, 2025, https://hiddenlayer.com/innovation-hub/prompt-injection-attacks-on-llms/
9. A Taxonomy of Prompt Injection Attacks - Schneier on Security, Zugriff am August 7, 2025, https://www.schneier.com/blog/archives/2024/03/a-taxonomy-of-prompt-injection-attacks.html
10. A CIA Triad-Based Taxonomy of Prompt Attacks on Large Language Models - MDPI, Zugriff am August 7, 2025, https://www.mdpi.com/1999-5903/17/3/113
11. Introducing a Taxonomy of Adversarial Prompt Engineering - HiddenLayer, Zugriff am August 7, 2025, https://hiddenlayer.com/innovation-hub/introducing-a-taxonomy-of-adversarial-prompt-engineering/
12. RAG Poisoning - Promptfoo, Zugriff am August 7, 2025, https://www.promptfoo.dev/docs/red-team/plugins/rag-poisoning/
13. What is Defense in Depth? Architecture and Examples - Wallarm, Zugriff am August 7, 2025, https://www.wallarm.com/what/defense-in-depth-concept
14. What is defense in depth? | Layered security - Cloudflare, Zugriff am August 7, 2025, https://www.cloudflare.com/learning/security/glossary/what-is-defense-in-depth/
15. Two Architectural Approaches to Defense in Depth - Delinea, Zugriff am August 7, 2025, https://delinea.com/blog/architectural-approaches-to-defense-in-depth
16. What Is Sandboxing? - Palo Alto Networks, Zugriff am August 7, 2025, https://www.paloaltonetworks.com/cyberpedia/sandboxing
17. Testing AI in Sandboxes - Walturn, Zugriff am August 7, 2025, https://www.walturn.com/insights/testing-ai-in-sandboxes
18. AI alignment - Wikipedia, Zugriff am August 7, 2025, https://en.wikipedia.org/wiki/AI_alignment
19. arxiv.org, Zugriff am August 7, 2025, https://arxiv.org/html/2309.00267v3

20. RLAIF Explained: A Scalable Alternative to RLHF for AI Training - Turing, Zugriff am August 7, 2025, https://www.turing.com/resources/rlaif-in-llms
21. Helpful, harmless, honest? Sociotechnical limits of AI alignment and safety through Reinforcement Learning from Human Feedback - PubMed Central, Zugriff am August 7, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC12137480/
22. How Reinforcement Learning from AI Feedback works - AssemblyAI, Zugriff am August 7, 2025, https://www.assemblyai.com/blog/how-reinforcement-learning-from-ai-feedback-works
23. Anthropic's "Constitutional AI" is very interesting : r/singularity - Reddit, Zugriff am August 7, 2025, https://www.reddit.com/r/singularity/comments/1b9r0m4/anthropics_constitutional_ai_is_very_interesting/
24. RLAIF vs. RLHF: Scaling Reinforcement Learning from ... - arXiv, Zugriff am August 7, 2025, http://arxiv.org/pdf/2309.00267
25. fantastic video on mechanistic interpretability : r/artificial - Reddit, Zugriff am August 7, 2025, https://www.reddit.com/r/artificial/comments/1hxylrv/fantastic_video_on_mechanistic_interpretability/
26. Mechanistic Interpretability — An Introduction | by Mario Raach | Jun, 2025 | Medium, Zugriff am August 7, 2025, https://medium.com/@marioraach01/mechanistic-interpretability-an-introduction-9ceeeb6a1898
27. Anthropic's Persona Vectors Enable Precise LLM Behavior Control, Zugriff am August 7, 2025, https://www.webpronews.com/anthropics-persona-vectors-enable-precise-llm-behavior-control/
28. arxiv.org, Zugriff am August 7, 2025, https://arxiv.org/html/2503.02857v2
29. [PDF] Deepfake-Eval-2024: A Multi-Modal In-the-Wild Benchmark of ..., Zugriff am August 7, 2025, https://www.semanticscholar.org/paper/Deepfake-Eval-2024%3A-A-Multi-Modal-In-the-Wild-of-in-Chandra-Murtfeldt/43ddc66715a4463374e92f0556e11bb2613ad215
30. AI Ethics in Social Media - Preprints.org, Zugriff am August 7, 2025, https://www.preprints.org/frontend/manuscript/828670fad87a9f6ba2dcf0aa89a59fbf/download_pub
31. Defensive Strategies: Building AI-Resilient Security Architectures - Tranchulas, Zugriff am August 7, 2025, https://tranchulas.com/defensive-strategies-building-ai-resilient-security-architectures/
32. Replay prevention - Docs - Webhooks.fyi, Zugriff am August 7, 2025, https://webhooks.fyi/security/replay-prevention
33. Replay attack - Wikipedia, Zugriff am August 7, 2025, https://en.wikipedia.org/wiki/Replay_attack
34. What is a Replay Attack and How to Prevent it - Kaspersky, Zugriff am August 7, 2025, https://usa.kaspersky.com/resource-center/definitions/replay-attack
35. What is semantic search, and how does it work? | Google Cloud, Zugriff am August 7, 2025, https://cloud.google.com/discover/what-is-semantic-search
36. What is semantic search? How it works, use cases & more - Meilisearch, Zugriff am August 7, 2025, https://www.meilisearch.com/blog/semantic-search
37. User Intent Analysis: What It Is, Why It Matters, & How to Do It - Nightwatch.io, Zugriff am August 7, 2025, https://nightwatch.io/blog/user-intent-analysis/
38. Reinforcement learning from AI feedback (RLAIF): Complete overview - SuperAnnotate, Zugriff am August 7, 2025, https://www.superannotate.com/blog/reinforcement-learning-from-ai-feedback-rlaif
39. arxiv.org, Zugriff am August 7, 2025, https://arxiv.org/html/2412.10807v2
40. Deepfake detection with and without content warnings - PMC, Zugriff am August 7, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC10679876/

41. Revisiting LLM Evaluation through Mechanism Interpretability: a New Metric and Model Utility Law - arXiv, Zugriff am August 7, 2025, https://arxiv.org/html/2504.07440v1
42. AI-Generated Content Detection - Hive Moderation, Zugriff am August 7, 2025, https://hivemoderation.com/ai-generated-content-detection
43. Generative AI In Content Moderation And Fake Content Detection - EnFuse Solutions, Zugriff am August 7, 2025, https://www.enfuse-solutions.com/generative-ai-in-content-moderation-and-fake-content-detection/
44. Detect intent with sentiment analysis | Dialogflow ES - Google Cloud, Zugriff am August 7, 2025, https://cloud.google.com/dialogflow/es/docs/how/sentiment
45. How to protect a private message from Replay attack? - Stack Overflow, Zugriff am August 7, 2025, https://stackoverflow.com/questions/44087753/how-to-protect-a-private-message-from-replay-attack
46. AI sandbox: secure environments for evaluating and protecting Artificial Intelligence models, Zugriff am August 7, 2025, https://telefonicatech.com/en/blog/ai-sandbox-secure-environments-for-evaluating-and-protecting-artificial-intelligence-models
47. The Sandboxed Mind — Principled Isolation Patterns for Prompt-Injection-Resilient LLM Agents | by Adnan Masood, PhD. | Jun, 2025 | Medium, Zugriff am August 7, 2025, https://medium.com/@adnanmasood/the-sandboxed-mind-principled-isolation-patterns-for-prompt-injection-resilient-llm-agents-c14f1f5f8495
48. Prompt Injections: A Practical Classification of Attack Methods - Pangea.cloud, Zugriff am August 7, 2025, https://pangea.cloud/securebydesign/aiapp-pi-classes/
49. A Survey: Deepfake and Current Technologies for Solutions - CEUR-WS.org, Zugriff am August 7, 2025, https://ceur-ws.org/Vol-3900/Paper9.pdf
50. (PDF) A Survey on Deepfake Detection Technologies - ResearchGate, Zugriff am August 7, 2025, https://www.researchgate.net/publication/389391024_A_Survey_on_Deepfake_Detection_Technologies
51. SCLBD/DeepfakeBench: A comprehensive benchmark of deepfake detection - GitHub, Zugriff am August 7, 2025, https://github.com/SCLBD/DeepfakeBench
52. DeepfakeBench: A Comprehensive Benchmark of Deepfake Detection - NIPS, Zugriff am August 7, 2025, https://proceedings.neurips.cc/paper_files/paper/2023/file/0e735e4b4f07de483cbe250130992726-Paper-Datasets_and_Benchmarks.pdf
53. Towards Benchmarking and Evaluating Deepfake Detection - IEEE Computer Society, Zugriff am August 7, 2025, https://www.computer.org/csdl/journal/tq/2024/06/10444780/1URbm2VqnSg
54. A Contemporary Survey on Deepfake Detection: Datasets, Algorithms, and Challenges, Zugriff am August 7, 2025, https://www.mdpi.com/2079-9292/13/3/585
55. Reinforcement Learning from AI Feedback (RLAIF): The Essential Guide | Nightfall AI Security 101, Zugriff am August 7, 2025, https://www.nightfall.ai/ai-security-101/reinforcement-learning-from-ai-feedback-rlaif
56. Reinforcement Learning from AI Feedback | by Anote - Medium, Zugriff am August 7, 2025, https://anote-ai.medium.com/reinforcement-learning-from-ai-feedback-5d5dd53cd26e
57. A Critical Evaluation of AI Feedback for Aligning Large Language Models - arXiv, Zugriff am August 7, 2025, https://arxiv.org/abs/2402.12366
58. What is RLAIF - Reinforcement Learning from AI Feedback? - Encord, Zugriff am August 7, 2025, https://encord.com/blog/reinforecement-learning-from-ai-feedback-what-is-rlaif/

59. A Practical Memory Injection Attack against LLM Agents - arXiv, Zugriff am August 7, 2025, https://arxiv.org/html/2503.03704v2
60. Multi-Faceted Studies on Data Poisoning can Advance LLM Development - arXiv, Zugriff am August 7, 2025, https://arxiv.org/html/2502.14182v1
61. Medical large language models are vulnerable to data-poisoning attacks - PubMed, Zugriff am August 7, 2025, https://pubmed.ncbi.nlm.nih.gov/39779928/
62. TalkingHeadBench: 一个用于分析说话者头像深度伪造检测的多模态, Zugriff am August 7, 2025, https://www.xueshuxiangzi.com/downloads/2025_6_2/2505.24866.pdf
63. Post-training for Deepfake Speech Detection - arXiv, Zugriff am August 7, 2025, https://arxiv.org/pdf/2506.21090
64. RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback - Abdullah Mamun, Zugriff am August 7, 2025, https://abdullah-mamun.com/talk/rlaif-vs.-rlhf-scaling-reinforcement-learning-from-human-feedback-with-ai-feedback/rlaif_mamun.pdf
65. Alignment and Safety in Large Language Models: Safety Mechanisms, Training Paradigms, and Emerging Challenges - ResearchGate, Zugriff am August 7, 2025, https://www.researchgate.net/publication/394080224_Alignment_and_Safety_in_Large_Language_Models_Safety_Mechanisms_Training_Paradigms_and_Emerging_Challenges/fulltext/68883ce896f3c0122ef4b3d1/Alignment-and-Safety-in-Large-Language-Models-Safety-Mechanisms-Training-Paradigms-and-Emerging-Challenges.pdf?origin=scientificContributions
66. Extended Abstract - CS 224R Deep Reinforcement Learning, Zugriff am August 7, 2025, https://cs224r.stanford.edu/projects/pdfs/Your_Project_Title12.pdf
67. Enhancing Reinforcement Learning with Error-Prone Language Models, Zugriff am August 7, 2025, https://www.ri.cmu.edu/app/uploads/2025/04/Muhan_Lin_Thesis.pdf
68. Decoding the Molecular Language of Proteins with Evola - bioRxiv, Zugriff am August 7, 2025, https://www.biorxiv.org/content/10.1101/2025.01.05.630192v1.full.pdf
69. WildDeepfake: A Challenging Real-World Dataset for Deepfake Detection, Zugriff am August 7, 2025, https://www.semanticscholar.org/paper/WildDeepfake%3A-A-Challenging-Real-World-Dataset-for-Zi-Chang/2060fa23185747294541f428c39640177450b8fb
70. Colton Bishop - Google Scholar, Zugriff am August 7, 2025, https://scholar.google.com/citations?user=X77YcdEAAAAJ&hl=en
71. Samrat Phatale - Google Scholar, Zugriff am August 7, 2025, https://scholar.google.com/citations?user=gTK5jNYAAAAJ&hl=en