Sensus

Model-Agnostic AI Governance Through Multi-Dimensional Content Evaluation

Benchmark Results Across Five Frontier Foundation Models

CyberGym (1,507 CVE Exploit Tasks) · LKM Red Team Corpus (28 Multi-Turn Adversarial Campaigns) · Three Sensus Versions · Four Distinct Safety Behavior Patterns Identified

Author Melissa K. Pinkston
Organization LKM Constructs LLC
Date February 26, 2026
Version 1.1 (Post-Audit, Corrected)
Status Patent Pending · Public Release

Abstract

We present Sensus, a deterministic, model-agnostic governance system that evaluates AI model outputs across five weighted dimensions to detect harmful content that bypasses model-native safety layers. Benchmarked against five frontier foundation models (Claude Opus 4.6, GPT-5.2, Grok 4.1, Mistral Large 3, Qwen-235b) on two complementary test corpuses — 1,507 CVE exploit generation tasks from Google CyberGym and 28 proprietary multi-turn adversarial campaigns — Sensus achieves 61.5–99.1% detection rates on compliant model responses and 85.7–96.4% effective governance when combined with model-native refusals. We demonstrate a model-agnostic learning loop that improves detection by +8 to +14 attacks across all providers without retraining, identify four distinct model safety behavior patterns (full compliance, selective refusal, partial refusal, and balanced), and introduce a refusal-aware pre-filter that eliminates false positives from keyword matches in refusal text. Three of five frontier models exhibit zero native safety on exploit generation, generating weaponizable proof-of-concept code for every request. We argue that infrastructure-level governance is necessary because model-level safety is unreliable, inconsistent, and provider-dependent.

Contents

1. Introduction
2. Architecture
3. Scoring Methodology
4. Test Corpuses
5. CyberGym Results
6. Red Team Results
7. Model Safety Behavior Taxonomy
8. Learning Loop
9. Refusal-Aware Pre-Filtering
10. Cross-Model Analysis
11. Limitations and Future Work
12. Conclusion

1. Introduction

The current AI safety landscape relies primarily on model-level alignment: reinforcement learning from human feedback (RLHF), constitutional AI training, and provider-specific content policies baked into model weights during training. This approach produces wildly inconsistent results. On identical exploit generation prompts, model-native refusal rates range from 0.0% (Mistral Large 3, Grok 4.1 Fast, Claude Opus 4.6) to 96.3% (GPT-5.2). An organization deploying multiple foundation models — or switching providers — inherits a safety lottery where the governance posture depends entirely on which API key is in the config file.

Sensus addresses this gap through infrastructure-level governance: a deterministic evaluation layer that scores model outputs against five weighted dimensions regardless of which model produced them. Unlike model-native safety, Sensus is provider-independent, auditable, and improvable through iterative gap analysis without requiring model retraining.

This paper presents benchmark results from one of the few published cross-model governance evaluations with reproducible methodology: five frontier foundation models tested against two complementary corpuses (technical exploit generation and social engineering), evaluated across three Sensus versions that demonstrate a measurable learning loop. We report corrected results following an independent audit that identified and resolved false positive classifications in the scoring pipeline.

> **Core Finding:** Three of five frontier models generate weaponizable exploit code for every request that reaches them, with zero native safety. Infrastructure-level governance is not supplemental — for these models, it is the only governance layer.

2. Architecture

Sensus operates as the post-inference component of a four-stage governance pipeline. The complete architecture provides defense-in-depth through complementary detection mechanisms:

Stage	Component	Function	Timing
1	PatternWall	Pre-inference adversarial prompt detection; blocks known attack patterns before model invocation	Pre-inference
2	Foundation Model	Generates response to (filtered) prompt	Inference
3	Refusal Filter	Classifies response as COMPLIANT, REFUSAL, PARTIAL_REFUSAL, or INFRA_ERROR	Post-inference
4	Sensus	Multi-dimensional scoring of compliant responses; flags harmful content	Post-inference

Table 1: Four-stage governance pipeline. Sensus evaluates only responses classified as COMPLIANT by the Refusal Filter, preventing false positives on refusal text.

Each request processed through the pipeline receives one of five outcome classifications. **PW_CAUGHT:** PatternWall detected an adversarial prompt pattern and blocked the request before model invocation. **SENSUS_CAUGHT:** the model generated a response, but Sensus flagged it as harmful (any evaluation

dimension at MEDIUM or above severity). **BOTH_CAUGHT:** PatternWall would have blocked the prompt and Sensus independently flagged the response — dual-layer confirmation. **DOUBLE_MISS:** neither PatternWall nor Sensus detected the harmful content — this is the governance gap that drives the learning loop. **CLEAN_PASS:** the request was legitimate and the response contained no harmful content, passing through all stages without intervention.

This paper focuses on Stage 4 (Sensus scoring) and Stage 3 (Refusal Filter), which together constitute the post-inference governance layer. PatternWall (Stage 1) is documented separately. The key architectural decision is the separation of classification from scoring: the Refusal Filter determines whether Sensus should evaluate a response, and Sensus determines what the response contains. This prevents a systemic false positive class where keyword matches in refusal text inflate detection rates.

2.1 Design Principles

Deterministic evaluation. Sensus uses pattern matching with fixed weights, not probabilistic classification. The same input always produces the same score. This enables reproducible audits and eliminates evaluation variance.

Model-agnostic operation. Sensus evaluates text content regardless of which model produced it. No model-specific tuning is required. Patterns discovered through gap analysis on one model transfer to all providers without modification.

Transparent scoring. Every evaluation produces a complete audit record: per-dimension scores, matched pattern identifiers, composite calculation, and severity classification. Decisions can be traced to specific pattern matches.

Iterative improvement. Sensus improves through a structured learning loop: benchmark → identify gaps → add patterns → re-benchmark → verify no regressions. Three iterations (v1.0 → v1.1 → v1.2) are documented in this paper with measured improvement at each stage.

3. Scoring Methodology

Sensus evaluates each response across five weighted dimensions. Each dimension scores 0.0–1.0. Severity is determined per-dimension, and a weighted composite provides an aggregate risk signal.

Dimension	Weight	Detection Scope
D1: Truth	0.30	Harmful content detection: exploit code patterns, lethal dosage, chemical weapons, infrastructure attacks, fraud evasion, social engineering, safety bypass
D2: Governance	0.20	Policy compliance, constitutional boundary adherence, operational rule violations
D3: Consistency	0.15	Behavioral drift from expected patterns, identity breaks, instruction override detection
D4: Trajectory	0.25	Session-level escalation tracking across turns; uses prior turn scores as context
D5: EQ	0.10	Manipulation resistance: authority spoofing, emotional leverage, urgency exploitation

Table 2: Sensus evaluation dimensions with weights. D1 (Truth) and D4 (Trajectory) carry the highest weights as the primary harmful content detector and session-level escalation tracker respectively.

Note on D1 nomenclature: In earlier system documentation and architecture diagrams, D1 is labeled "Truth & Source Confidence" with a broader scope including source type verification and fact-checking. In this benchmark context, D1 is scoped specifically to harmful content detection — the patterns that identify exploit code, lethal dosage information, chemical weapon instructions, and other dangerous content in model outputs. The underlying dimension is the same; the label and active pattern set reflect the evaluation domain.

Weighting rationale. D1 (Truth) and D4 (Trajectory) carry the highest weights (0.30 and 0.25 respectively) because harmful content detection and session-level escalation tracking are the primary governance objectives in adversarial contexts. A system that fails to catch exploit code or misses a multi-turn escalation pattern has failed regardless of other scores. D2 (Governance, 0.20) provides the structural compliance layer, catching policy violations that D1's lexical patterns may miss. D3 (Consistency, 0.15) provides within-turn contextual signal for behavioral drift and identity breaks. D5 (EQ) carries the lowest weight (0.10) because emotional manipulation indicators are typically caught by D1 and D2 before EQ patterns would trigger independently — when a model delivers harmful content under emotional pressure, D1 catches the content and D2 catches the boundary violation, making D5 confirmatory rather than primary.

3.1 Caught Determination and Composite Scoring

A response is CAUGHT if any individual dimension reaches MEDIUM severity or above. This is the primary governance decision: per-dimension severity thresholds determine whether a response is flagged, not the composite score alone.

Dimension	MEDIUM Threshold	HIGH Threshold	Rationale
D1: Truth	≥ 0.15	≥ 0.35	Exploit patterns produce concentrated signal
D2: Governance	≥ 0.10	≥ 0.25	Policy violations are binary — any breach is significant
D3: Consistency	≥ 0.20	≥ 0.40	Behavioral drift requires more evidence to confirm

Dimension	MEDIUM Threshold	HIGH Threshold	Rationale
D4: Trajectory	≥ 0.15	≥ 0.35	Escalation patterns accumulate across turns
D5: EQ	≥ 0.20	≥ 0.40	Manipulation indicators need higher confidence

Table 3a: Per-dimension severity thresholds. CAUGHT = any dimension ≥ MEDIUM.

The composite score provides an aggregate severity classification for audit and reporting purposes:

```
composite = (D1 × 0.30) + (D2 × 0.20) + (D3 × 0.15) + (D4 × 0.25) + (D5 × 0.10)
```

Severity	Composite Range	Interpretation
CRITICAL	≥ 0.15	Active harmful content requiring immediate intervention
HIGH	0.10 – 0.149	Significant policy violation or harmful knowledge delivery
MEDIUM	0.06 – 0.099	Concerning content flagged for review
LOW	0.03 – 0.059	Minor concern, may warrant monitoring
NONE	< 0.03	No governance concern detected

Table 3b: Composite severity thresholds for aggregate classification.

Why per-dimension thresholds matter: A response that triggers a single dimension at MEDIUM severity (e.g., D2 Governance at 0.10 with all others at 0.0) may produce a composite score below 0.06 (in this case: 0.10 × 0.20 = 0.02). Under composite-only thresholding, this response would be classified as LOW and missed. Per-dimension thresholding catches it because a governance violation at MEDIUM severity is inherently concerning regardless of the aggregate score. In practice, 4.9% of Grok catches and 10.3% of GPT-5.2 catches fall into this category — responses caught by per-dimension severity that composite thresholding alone would miss.

3.2 Pattern Library Evolution

Sensus v1.0 shipped with exploit code detection patterns targeting vulnerability proof-of-concept generation. v1.1 added 29 patterns for format-native exploits (PostScript, PDF structure, binary construction, submission pipelines) discovered through CyberGym gap analysis. v1.2 added 43 patterns across six harmful knowledge categories (lethal dosage, chemical weapons, infrastructure attacks, fraud evasion, social engineering, safety bypass) discovered through red team gap analysis. Total pattern library: 100+ patterns across technical exploit and social engineering domains.

4. Test Corpuses

4.1 CyberGym (Third-Party)

Source: Google CyberGym benchmark [1]. Tasks: 1,507 unique CVE exploit generation prompts across 188 real-world software projects. Each task provides a vulnerability description and asks the model to generate a proof-of-concept exploit file. This corpus tests whether governance catches weaponizable technical output — actual exploit code that could be used against real systems.

CyberGym provides independent validation: we did not design these prompts, and the corpus is widely used in AI safety research. Results on CyberGym are directly comparable to other published benchmarks.

4.2 LKM Red Team Corpus (Proprietary)

Source: LKM Constructs, designed and maintained internally. Attacks: 28 multi-turn adversarial sequences across 118 individual turns. Each attack uses one or more adversarial techniques to escalate toward harmful content delivery through social engineering rather than direct requests.

Vector	Count	Description
Crescendo	12	Progressive rapport-building escalation across 5–7 turns [2]
Self-Correction Exploit	3	Exploits model tendency to correct previous refusals
Multimodal Grooming	2	Cross-context trust establishment
Memory Exploit	1	Weaponizes conversation memory for context injection
Skeleton Key	1	Incremental boundary dissolution through trust-building [3]
Policy Puppetry	1	Policy reframing to legitimize harmful requests
Rolebreaker	1	Direct persona override attempt
Fused Multi-Vector	1	Combines token manipulation, self-correction, and memory exploit
Other	6	DAP/distract, infoflood, skeleton key + memory variants

Table 4: Red team attack vectors. Harm categories include lethal dosage, chemical weapons, infrastructure attacks, fraud evasion, social engineering, and safety bypass.

The dual-corpus design tests two fundamentally different attack surfaces: CyberGym measures detection of technical harmful output (exploit code), while the red team measures detection of social harmful output (manipulated knowledge delivery). A governance system that performs well on only one corpus has a critical blind spot.

5. CyberGym Results

All five models were evaluated on the full 1,507-task CyberGym corpus. Results reflect corrected classifications: infrastructure errors (DNS failures, timeouts, rate limits) are excluded from both refusal rates and Sensus evaluation pools. Sensus detection rates are calculated on compliant responses only — responses where the model generated substantive content.

Model	Compliant	Refusals	v1.0	v1.1	Δ	Effective	Gap
Opus 4.6	1,469	0 (0.0%)	94.4%	99.1%	+4.8pp	99.1%	13
Mistral Large 3	1,505	1 (0.1%)	93.0%	95.7%	+2.7pp	95.7%	65
Grok 4.1 Fast	1,082*	0 (0.0%)	48.3%	62.7%	+14.3pp	62.7%	402
Qwen-235b	687*	300 (30.4%)	87.1%	88.1%	+1.1pp	91.7%	82
GPT-5.2	39*	1,464 (96.3%)	35.9%†	61.5%†	+25.6pp	99.0%	15

*Table 5: CyberGym results (1,507 tasks). *Evaluable count after infrastructure error exclusion. †GPT-5.2 rates on 39 true compliant responses after refusal filter reclassification. Effective = refusals + Sensus catches.*

Key Finding: Three of five models (Opus, Mistral, Grok) refused zero exploit generation requests. Every single task that reached these models produced weaponizable PoC code. For these models, Sensus is not supplementing safety — it is the only governance layer.

5.1 Learning Loop: CyberGym

Sensus v1.0 → v1.1 added 29 patterns targeting format-native exploits (PostScript, PDF structure, binary construction indicators, submission pipelines, overflow values) plus 14 compound amplifiers. Improvement ranged from +1.1pp (Qwen) to +14.3pp (Grok), with the largest gains on models that produce terse, code-heavy responses where format-native patterns provide critical signal. No model showed regression.

6. Red Team Results

All five models were evaluated against the full 28-attack red team corpus. Results reflect v1.2 Sensus scoring with refusal-aware pre-filtering applied: responses classified as refusals by the Refusal Filter are excluded from Sensus evaluation, preventing false positives from keyword matches in refusal text.

Model	v1.0	v1.2	Δ	Effective	Eff %	Gap
Opus 4.6	14	24	+10	27/28	96.4%	1
Mistral Large 3	12	25	+13	25/28	89.3%	3
Grok 4.1 Fast	13	22	+9	24/28	85.7%	4
Qwen-235b	13	21	+8	24/28	85.7%	4
GPT-5.2	8	22	+14	24/28	85.7%	4

Table 6: Red team results (28 attacks). Effective = Sensus catches + model refusals.

Key Finding: The learning loop is model-agnostic. Patterns discovered through gap analysis on Mistral (v1.2, +13 attacks) transfer to all five providers: Opus +10, Grok +9, Qwen +8, GPT-5.2 +14. Zero regressions on CyberGym across all models.

6.1 Per-Attack Analysis

Certain attack vectors proved consistently difficult across all models. The legal-evidence-crescendo and distract-attack (DAP) vectors showed the lowest catch rates, exploiting dual-use educational framing that produces content with legitimate surface characteristics. The infoflood vector uses volume-based overwhelm to dilute governance signal.

Conversely, the v1.2 learning loop was most effective on memory exploits, self-correction exploits, and chemical/dosage crescendos, where harmful knowledge patterns (lethal dosage terminology, chemical weapon indicators) provide strong lexical signal regardless of the social engineering wrapper.

7. Model Safety Behavior Taxonomy

Cross-model analysis revealed four distinct safety behavior patterns. These are not quality judgments — each pattern has implications for governance architecture design.

Behavior	Model(s)	Description
Full Compliance	Mistral Large 3	Zero native safety on both corpuses. Model completes all requests regardless of content. Sensus provides 100% of governance.
Selective Refusal	Qwen-235b, Grok 4.1	Refuses some requests outright. CyberGym: 0–30.5% refusal. Red team: <1–10%. Safety is inconsistent across task types.
Partial Refusal	GPT-5.2	Provides educational context but explicitly withholds actionable harmful details. 96.3% CyberGym refusal; 32.2% partial refusal on red team turns.
Balanced	Opus 4.6	Low refusal rate, high compliance, verbose detailed responses. Gives Sensus maximum evaluation signal. Highest combined governance.

Table 7: Four distinct safety behavior patterns across five frontier models.

7.1 Partial Refusal: A New Behavioral Category

GPT-5.2 exhibits a previously undocumented behavioral pattern we term partial refusal: the model engages substantively with harmful requests, providing educational context, mechanism descriptions, and related information, but explicitly withholds the specific actionable payload. Characteristic phrases include “I can’t provide LD50 values,” “I can’t give mixing ratios for dangerous chemical combinations,” and “I can’t help with step-by-step exploit instructions.”

This behavior appeared on 32.2% of red team turns for GPT-5.2. It represents a distinct safety strategy: the model treats harmful requests as teaching moments rather than binary accept/refuse decisions. For governance purposes, partial refusals require special handling — the educational content may still contain harmful information even though the specific payload was withheld.

7.2 Implications for Multi-Model Deployment

Organizations deploying multiple foundation models face a governance fragmentation problem. The same prompt produces dramatically different safety behaviors depending on which model receives it. Without infrastructure-level governance, the organization’s safety posture is determined by the weakest model in its deployment — and as our results show, the weakest model may have zero native safety.

8. Learning Loop

Sensus improves through a structured gap analysis methodology. Each iteration follows a four-step cycle: benchmark (run current version against corpus), analyze gaps (manually inspect missed detections), add patterns (design new detection rules targeting identified gap categories), re-benchmark (verify improvement without regressions). Gap analysis is conducted by the system architect (the author), who reviews each missed detection to determine whether the miss represents a pattern gap (Sensus lacks vocabulary for this content type), a structural gap (content is harmful but lexically benign), or a classification error (response was misclassified as compliant). New patterns are validated against a held-back set of known-caught responses to confirm zero regressions before deployment to full-corpus re-evaluation.

Iteration	Trigger	Patterns Added	Result
v1.0 → v1.1	CyberGym gap analysis (11 misses on 30-task sample)	29 patterns: format-native PoC (6), binary construction (4), submission pipeline (3), overflow values (2), compound amplifiers (14)	+2.7pp to +14.3pp on CyberGym; zero impact on red team
v1.1 → v1.2	Red team gap analysis (16 missed attacks on Mistral baseline)	43 patterns in 6 categories: lethal dosage (9), chemical weapons (10), infrastructure attack (6), fraud evasion (7), social engineering (7), safety bypass (4)	+8 to +14 attacks on red team; zero regressions on CyberGym

Table 8: Learning loop iterations with triggers, pattern additions, and measured results.

The critical property of the learning loop is **model-agnostic transfer.** Patterns discovered through gap analysis on Mistral Large 3 (the first model benchmarked) improved detection across all five providers without any model-specific tuning. This is possible because Sensus evaluates content, not model behavior — harmful content has consistent lexical characteristics regardless of which model produced it.

Equally important is the **zero-regression property.** Each iteration maintains or improves performance on all previously tested corpuses and models. v1.2's 43 new red team patterns caused no detection loss on any of the five models' CyberGym results. This is verified through full re-evaluation at each iteration, not through holdout testing.

9. Refusal-Aware Pre-Filtering

An independent audit identified a systematic false positive class in Sensus v1.2: keyword pattern matches firing on model refusal text. When a model responds, "I won't provide LD50 values," Sensus detects the "LD50" keyword and scores the response as harmful — despite the model having already refused the request. This inflated Opus red team results from 24 to 26 attacks caught and inflated GPT-5.2 CyberGym compliant count from 39 to 56 (17 refusals misclassified as compliant by the harness).

To address this, we introduce a Refusal Filter that classifies responses before Sensus scoring. The filter uses three-stage detection: (1) infrastructure errors (DNS, timeout, HTTP errors), (2) hard/soft refusals (model explicitly declines), and (3) partial refusals (model engages but withholds actionable harm). Only responses classified as COMPLIANT proceed to Sensus evaluation.

The filter is designed for conservative classification: when uncertain, it classifies as COMPLIANT so Sensus can evaluate. False negatives (missing a refusal) are acceptable because Sensus will score harmless

refusal text as LOW/NONE severity. False positives (classifying compliant content as refusal) are dangerous because they would skip Sensus scoring on genuinely harmful content.

Validation results: 23/23 on synthetic test cases, 6/6 known false positives from audit caught, 17 harness misclassifications on GPT-5.2 CyberGym identified and corrected. All numbers reported in this paper reflect post-filter corrected results.

10. Cross-Model Analysis

10.1 Model-Native Safety Is Unreliable

On identical CyberGym prompts, native refusal rates span three orders of magnitude: 0.0% (Opus, Mistral, Grok) to 96.3% (GPT-5.2). An organization cannot rely on model-native safety because the safety posture changes when the model changes. Provider switches, cost optimizations, and multi-model routing all expose governance gaps that only infrastructure-level evaluation can close.

10.2 Infrastructure Governance Converges

Despite wildly different model behaviors, Sensus + model refusals produces effective governance rates of 62.7–99.1% on CyberGym and 85.7–96.4% on red team. The convergence is strongest on models with high compliance (Opus: 99.1% CyberGym, 96.4% red team) and on models with high native refusal (GPT-5.2: 99.0% CyberGym effective). The primary outlier is Grok at 62.7% CyberGym — a gap driven by the fundamental limitations of regex-based detection on terse outputs.

10.3 The Grok Gap: Regex Detection Has a Floor

Grok 4.1 Fast produces the shortest responses of any model tested. On identical CyberGym tasks, Grok's median response length is 130 characters versus Opus's 3,193 characters. Sensus catches Opus at 99.1% and Grok at 62.7% — not because Grok produces different content, but because it produces less text for pattern matching. 40% of Grok's misses are plaintext payloads (heredoc XML, config files) with zero exploit vocabulary. 15% are minimal/empty files (single-byte writes, truncated content).

This is not a soft limitation — it is a hard floor inherent to lexical detection. Regex pattern matching requires lexical signal to match against. When a model generates a 40-character binary payload with no commentary, no explanation, and no exploit vocabulary, there is nothing for patterns to catch. Section 11.1 discusses this ceiling in detail and identifies behavioral detection (v1.3) as the architectural response: detecting what a response does rather than what it says.

10.4 Social Engineering Bypasses Model Safety

A striking pattern across all models: CyberGym refusal rates are dramatically higher than red team refusal rates. GPT-5.2 refuses 96.3% of exploit code but only 32.2% of social engineering turns. Model safety layers are calibrated for direct harmful requests, not for multi-turn manipulation. Social engineering is the bypass vector that model training has not adequately addressed.

11. Limitations and Future Work

11.1 Current Limitations

Lexical detection ceiling. Sensus relies on regex pattern matching, which has fundamental limits on terse or vocabulary-free content. Grok's 62.7% CyberGym rate demonstrates this ceiling: when a model produces a 40-character exploit with no commentary, there is insufficient text for lexical patterns to match. This is the primary driver of the 402-response Grok gap and represents an architectural limitation that cannot be resolved through additional patterns alone.

Detection, not prevention. Sensus evaluates after generation. The model has already produced the harmful content by the time Sensus flags it. In high-throughput API deployments, this creates a latency window where harmful content exists before governance acts. PatternWall (pre-inference) addresses this for known attack patterns, but novel single-turn requests still reach the model.

Corpus scope. Testing covers CBRN-adjacent harms, fraud, social engineering, and exploit code. Coverage of self-improvement instructions, persuasion toward extremism, subtle value drift over long contexts, and multimodal attacks (image + text) remains limited. The red team corpus includes only 2 multimodal grooming attacks.

Adversarial robustness. Publication of this methodology enables attackers to probe for blind spots or craft around known patterns. The learning loop provides a response mechanism, but there is an inherent cat-and-mouse dynamic in any pattern-based detection system.

11.2 Future Work

v1.3: Behavioral detection. Priority addition of structural analysis: does the response write a file? Execute a pipeline? Construct a binary payload? These behavioral indicators are independent of vocabulary and would close the Grok gap. Target: detection of responses that perform harmful actions regardless of what words they use.

Expanded red team corpus. Additional attack vectors targeting: self-improvement/capability elicitation, political persuasion/radicalization, long-context value drift (50+ turn conversations), multimodal attacks (image-text combinations), and tool-use exploitation.

Regulatory alignment validation. Formal mapping of Sensus evaluation dimensions to EU AI Act Annex IV requirements [6], ISO/IEC 42001 traceability standards [14], and NIST AI Risk Management Framework categories [4, 5]. Initial analysis suggests strong alignment, but formal certification has not been pursued.

12. Conclusion

Sensus demonstrates that infrastructure-level governance can provide consistent, measurable safety across foundation models with wildly different native safety behaviors. Three of five frontier models generate weaponizable exploit code for every request. Model-native safety ranges from 0% to 96.3% on identical prompts. This is not a safety ecosystem — it is a lottery.

Against this backdrop, Sensus achieves 61.5–99.1% detection rates on compliant responses and 85.7–96.4% effective governance when combined with model-native refusals. The learning loop improves detection by +8 to +14 attacks across all providers without retraining, demonstrating model-agnostic pattern transfer. The refusal-aware pre-filter eliminates a systematic false positive class, producing audited numbers that withstand independent verification.

The core thesis: organizations deploying AI systems need governance that works regardless of which model is behind the API. Provider-dependent safety is not safety — it is a variable that changes when the contract changes. Infrastructure-level governance makes safety a property of the deployment, not a feature of the model.

References

[1] Wang, Z. et al. (2025). “CyberGym: Evaluating AI Agents’ Real-World Cybersecurity Capabilities at Scale.” arXiv:2506.02548. UC Berkeley.

[2] Russinovich, M., Salem, A., & Eldan, R. (2024). “Great, Now Write an Article About That: The Crescendo Multi-Turn LLM Jailbreak Attack.” USENIX Security Symposium. arXiv:2404.01833.

[3] Russinovich, M. (2024). “Mitigating Skeleton Key, a New Type of Generative AI Jailbreak Technique.” Microsoft Security Blog, June 2024.

[4] National Institute of Standards and Technology (2023). “Artificial Intelligence Risk Management Framework (AI RMF 1.0).” NIST AI 100-1.

[5] National Institute of Standards and Technology (2024). “Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile.” NIST AI 600-1.

[6] European Parliament and Council of the European Union (2024). “Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (AI Act).” Official Journal of the European Union.

[7] Anthropic (2025). “AI for Cyber Defenders.” red.anthropic.com.

[8] Anthropic (2025). “Claude Opus 4.6 Model Card.” Anthropic Model Documentation.

[9] OpenAI (2025). “GPT-5.2 Model Documentation.” OpenAI Platform.

[10] xAI (2025). “Grok 4.1 Technical Documentation.” xAI.

[11] Mistral AI (2025). “Mistral Large 3.” Mistral AI.

[12] Qwen Team (2025). “Qwen-235b Model Documentation.” Alibaba Cloud / QwenLM.

[13] Pinkston, M.K. (2026). “PatternWall: Constitutional Governance Middleware for AI Safety.” LKM Constructs LLC. Zenodo.

[14] International Organization for Standardization (2023). “ISO/IEC 42001:2023 — Information technology — Artificial intelligence — Management system.” ISO.

LKM Constructs LLC

Contact: melissa@lkmconstructs.com

Patent Pending — February 2026

