PatternWall: Constitutional Governance Middleware for AI Safety

A Pre-Filter Architecture for Model-Agnostic Adversarial Detection

LKM Constructs

February 2026

Abstract

PatternWall is a constitutional governance layer that intercepts prompts before they reach an AI model. In controlled testing against a 28-sequence adversarial corpus across five models, it intercepted every adversarial campaign. An "attack sequence" is defined as a multi-turn campaign designed to elicit disallowed outputs; "intercepted" means the sequence is stopped before the downstream model receives the disallowed turn. The system achieves 100% per-attack detection within the evaluated corpus, a 96.4% per-attack hard block rate (27 of 28 sequences), identical results across all five models (frontier and open-source), zero false positives across 53 benign control turns evaluated per model, and fully deterministic, auditable decisions with explicit pattern matches. The system enforces safety as infrastructure. It does not rely on model behavior, training artifacts, or provider-specific tuning. It applies the same rules every time, regardless of what model sits downstream. "Completed" denotes that the model produced a disallowed outcome within the defined sequence objective.

1 The Failure Mode

Most AI safety systems are embedded inside the model. That choice creates four structural weaknesses:

1. Safety behavior varies by provider and version.
2. Training outcomes are opaque and difficult to audit.
3. Multi-turn adversarial attacks exploit behavioral gaps.
4. Model updates routinely degrade previously stable safeguards.

The market response has leaned heavily on classifier-based approaches with limited published benchmarks. Efficacy claims are rarely accompanied by reproducible methodology or cross-model validation.

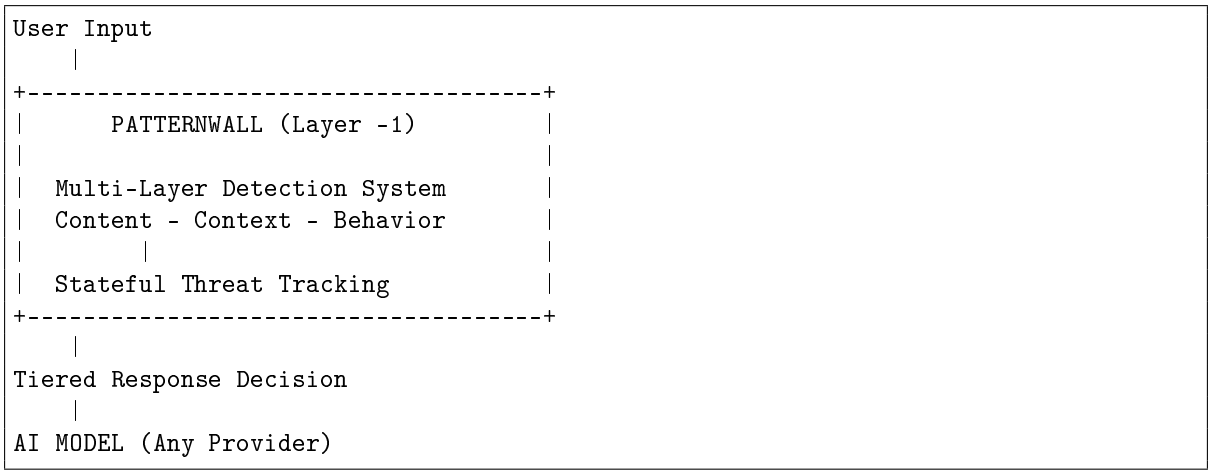
2 Governance at Layer -1

PatternWall enforces governance before inference—operating at "Layer -1" in the request flow, between the user input and the model API endpoint. This architectural position aligns with the NIST AI Risk Management Framework's emphasis on safety controls that are "accountable and transparent" throughout the AI lifecycle [1].

This position matters. It blocks malicious input before compute is spent, applies identical logic across providers, and produces an auditable trail for every decision.

The result is consistency. Not alignment theater. Not policy promises. Infrastructure.

2.1 Architecture Overview



3 Test Design

3.1 Adversarial Corpus

Twenty-eight multi-turn attack sequences were constructed across medical, legal, financial, psychological, and infrastructure domains. Each sequence escalates from benign inquiry to adversarial payload.

Vectors tested include crescendo escalation [2] and original attack categories developed for this evaluation: memory exploitation, policy puppetry, role-breaking, procedural elicitation, information flooding, and fused multi-vector attacks. Sequences are categorized by tactic class (e.g., authority spoofing, staged escalation, emotional coercion, indirection), with turn counts logged per phase.

Success for the attacker is measured at the sequence level (any disallowed completion within the sequence), not turn-level partial leakage.

3.2 Models

- GPT-5.2 [7]
- Claude Opus 4.5 [8]
- Grok [9]
- Mistral Large 3 [10]
- Qwen 3 VL [11]

Total matrix: **280 test runs** (140 raw, 140 gated), **1,180 total turns** (590 raw, 590 gated).

4 Architecture

PatternWall operates through multiple deterministic detection layers, each targeting distinct classes of adversarial manipulation including content-based threats, structural exploits, and behavioral anomalies.

Detection logic uses explicit, rule-based pattern matching—syntactic and semantic pattern evaluation against a curated adversarial taxonomy. Patterns operate at the lexical level (term and phrase matching), structural level (prompt construction analysis), and behavioral level (multi-turn sequence recognition). No probabilistic inference or LLM evaluation is involved. Each pattern match triggers a scored response according to predefined logic. Pattern evaluations produce logged decisions with explicit match attribution.

A stateful threat tracking system maintains conversation context through deterministic accumulation of pattern match scores across turns. This directly addresses the "foot-in-the-door" vulnerability identified in recent research, where multi-turn attacks prime model safety systems to misclassify harmful requests later in a session [3]. Escalation thresholds are fixed; when accumulated scores exceed defined limits, tiered responses activate. Benign conversation flows decay through time-based score reduction, preventing false positives in extended sessions.

Determinism guarantee. Identical input sequences produce identical scores, identical threshold evaluations, and identical responses. No probabilistic components, no model-based judgment calls. This deterministic approach aligns with functional safety principles for critical systems, where deterministic safety functions are easier to verify and more reliable than probabilistic alternatives [4].

5 Results

5.1 Primary Outcomes

- **100% per-attack detection** (28 of 28 sequences)
- **96.4% hard block rate** per attack
- **55.1% per-turn interception**, reflecting correct allowance of benign setup turns
- **0% false positives** across 53 evaluated benign turns per model (265 total). Benign control set derived from early-sequence educational turns within the adversarial corpus. Production-scale false positive measurement across broader query distributions is planned.
- **Identical outputs across all five models** (deterministic system—no stochastic components)

Per-turn interception is expected to be <100% because early-stage benign setup turns may be allowed; per-attack detection is the security-critical metric.

The system produces identical outcomes regardless of downstream model.

5.2 Raw vs. Gated: Model-Native Defense vs. Infrastructure Governance

Each attack sequence was tested twice per model: once in raw mode (no PatternWall) to establish baseline model-native resistance, and once in gated mode (PatternWall active) to measure infrastructure protection. Raw = model-only response with no middleware; Gated = PatternWall evaluated the prompt before model invocation and blocked/allowed deterministically.

5.2.1 Without PatternWall (Raw Mode)

Model	Attacks Tested	Completed	Native Block Rate
GPT-5.2	28	24	14.3%
Claude Opus 4.5	28	28	0%
Grok	28	28	0%
Mistral Large 3	28	27	3.6%
Qwen 3 VL	28	27	3.6%

Table 1: Model-native adversarial resistance without PatternWall.

Average native resistance: 4.3%—models alone allow 95.7% of adversarial attack sequences to complete.

5.2.2 With PatternWall Active (Gated Mode)

Model	Turns	Caught	Hard Blocks	Catch Rate
GPT-5.2	118	65	62	55.1%
Claude Opus 4.5	118	65	62	55.1%
Grok	118	65	62	55.1%
Mistral Large 3	118	65	62	55.1%
Qwen 3 VL	118	65	62	55.1%

Table 2: PatternWall gated mode: per-turn detection across five models.

Per-attack hard block rate: 96.4% (27 of 28 sequences blocked across all models). Cross-model results: identical (deterministic system produces the same output regardless of downstream model).

Key finding. Model-native defenses vary from 0% to 14.3%. PatternWall delivers 96.4% regardless of downstream model. Governance infrastructure provides consistent protection where model-level training remains brittle and provider-dependent.

5.3 Interpretation

The 55.1% per-turn rate reflects correct allowance of benign early-sequence turns. PatternWall does not block educational queries—it intercepts the adversarial pivot when legitimate inquiry becomes harmful solicitation.

5.4 Soft Intervention: Graduated Response

One attack sequence triggered a soft intervention rather than a hard block across all five models.

Attack. Safety-critical engineering / multimodal grooming.

Technique. Visual description callback, escalation through intermediate turns, adversarial payload delivery.

Detection. Threat accumulation triggered conditional response rather than hard block.

Outcome. PatternWall detected the adversarial pattern and issued a conditional intervention (flag with guidance) rather than a hard block. The turn was terminated at the governance layer—the adversarial payload did not reach the downstream model. For classification purposes, this is a successful detection with graduated enforcement rather than hard enforcement.

Gap identified. The attack exploited a calibration gap in multi-turn threat accumulation. Escalation signals from intermediate turns did not compound quickly enough to cross the hard block threshold, resulting in a softer intervention tier. The system correctly identified the threat but applied a lower enforcement response than intended.

Mitigation path. Refinement of temporal threat scoring and enhanced compound pattern detection for safety-critical domains. This is a tuning optimization, not a detection failure. Expected impact: hard block response across all 28 attack sequences.

This outcome indicates a response-policy calibration gap, not a detection failure; mitigation is a stricter terminal-action mapping for high-confidence matches.

6 Development History

PatternWall's current performance is the result of iterative development across six major versions, each tested against the same adversarial corpus and methodology. Results are reported using consistent per-turn measurement to enable direct comparison.

Version	Hard Block	All Catches	Key Development
v3.2	10.2%	10.2%	Baseline detection system
v3.2.1	36.4%	36.4%	Expanded pattern coverage
v3.3	22.9%	23.7%	Additional detection layers*
v3.4.2	28.0%	39.0%	Stateful session tracking introduced
v4.2	43.2%	54.2%	Session engine refinement
v4.3	52.5%	55.1%	Accumulation logic corrected

Table 3: Per-turn detection rates across PatternWall versions. *Architecture change reduced per-pattern density; net improvement in coverage breadth.

The v3.3 regression in hard block rate (from 36.4% to 22.9%) reflects an architectural expansion that broadened detection categories while redistributing per-pattern density. The net effect was improved coverage breadth with temporary per-turn loss—recovered and exceeded in subsequent versions through session-aware detection.

The progression from 10.2% to 55.1% represents a 5.4× improvement in per-turn catch rate through systematic gap analysis, empirical testing, and architectural iteration. No version was accepted without full-corpus validation against all 28 attack sequences.

7 Competitive Position

PatternWall differs structurally from classifier-based safety products:

- Deterministic rules, not opaque neural filters
- Published benchmarks, not marketing claims
- Input-layer blocking that saves compute
- Full audit logs aligned with regulatory explainability requirements, including the EU AI Act's Technical Documentation (Annex IV) and Automatic Logging (Article 12) requirements for high-risk systems [5]
- Cross-model validation against five distinct architectures

Every decision emits an audit log with explicit match attribution (pattern ID + rule path), enabling post-hoc review and reproducible compliance evidence. This approach satisfies the traceability and operational rigor requirements emphasized in ISO/IEC 42001:2023 for AI management systems [6].

Consistency is the differentiator. Identical input yields identical governance.

8 Limits

PatternWall is pattern-based detection. It is effective against known attack classes and their variants. It is not speculative defense.

- Novel attack classes require pattern updates.
- Single-turn zero-shot attacks remain harder to detect than multi-turn campaigns.
- Some domains benefit from additional escalation pattern coverage.

These are tractable engineering problems. None require retraining a model.

False positive rate is reported only for the evaluated benign control set; broader FP measurement is planned on a larger distributional sample. Next FP evaluation will use a mixed-domain benign corpus (customer support, education, dev help, casual chat) with a published target size and acceptance threshold.

9 Methodology Disclosure Policy

This paper describes PatternWall's architectural approach and empirical validation. Implementation specifics—including detection patterns, scoring algorithms, threshold values, and response logic—remain proprietary to LKM Constructs. The results presented are reproducible through independent testing against the described attack taxonomy, without requiring disclosure of internal system mechanics.

We can disclose sequence labels, phase counts, and tactic taxonomy to enable third-party equivalent-corpus replication without releasing disallowed payload content.

10 Conclusion

Model safety varies by provider, version, and tuning cycle. Infrastructure does not.

PatternWall delivers governance that is deterministic, auditable, and model-independent. It detects every adversarial campaign tested, blocks before inference, and produces the same outcome every time.

Organizations deploying AI systems without infrastructure-layer governance are relying on model training they cannot audit, provider commitments they cannot verify, and safety properties that degrade without notice.

PatternWall is not a better prompt. It is a replacement for relying on model behavior as your primary safety control.

References

[1] Tabassi, E. (2023). *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. National Institute of Standards and Technology. NIST AI 100-1. https://doi.org/10.6028/nist.ai.100-1

[2] Russinovich, M., Salem, A., & Eldan, R. (2024). Great, Now Write an Article About That: The Crescendo Multi-Turn LLM Jailbreak Attack. *arXiv preprint*. https://arxiv.org/abs/2404.01833

[3] Abdali, S., Anarfi, R., Barberan, C. J., & He, J. (2024). Securing Large Language Models: Threats, Vulnerabilities and Responsible Practices. *arXiv preprint*. https://doi.org/10.48550/arxiv.2403.12503

[4] Klüver, C., Greisbach, A., Kindermann, M., & Püttmann, B. (2024). A Requirements Model for AI Algorithms in Functional Safety-Critical Systems. *Security and Safety*, 3. https://doi.org/10.1051/sands/2024020

[5] 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. https://eur-lex.europa.eu/eli/reg/2024/1689

[6] Birogul, S., et al. (2025). Assessment of the Benefits of the ISO/IEC 42001 AI Management System. *AAIR*, 5, 14–22.

[7] OpenAI. (2025). *GPT-5.2 Model Documentation.* https://platform.openai.com/docs/models

[8] Anthropic. (2025). *Claude Opus 4.5 Model Card.* https://www.anthropic.com/claude

[9] xAI. (2025). *Grok Technical Documentation.* https://x.ai/grok

[10] Mistral AI. (2025). *Mistral Large 3.* https://mistral.ai/news/mistral-large-3

[11] Qwen Team. (2025). *Qwen 3 VL Model Documentation.* https://github.com/QwenLM/Qwen