Layered Governance Architecture for Agentic AI Systems: A Simulation Study

Anonymous Author(s)

ABSTRACT

Agentic AI systems that plan over long horizons, use tools, maintain persistent memory, and interact with other agents pose governance challenges that exceed the capabilities of model-level alignment alone. We propose a *Layered Governance Architecture* (LGA) that integrates three enforcement layers—model-level alignment monitoring, agent-level policy enforcement, and ecosystem-level interaction oversight—into a unified framework. We evaluate LGA through a controlled simulation study with paired evaluation (same traces replayed under each governance configuration) and multiple random seeds reporting mean ± standard deviation. Key design improvements over prior work include: (1) explicit layer enable/disable flags ensuring a true no-governance baseline, (2) ground-truth-based metrics separating true positive rate from false positive rate, (3) wall-clock overhead measurement, (4) a continuous adaptation experiment where a single policy object persists across distribution shifts, and (5) per-step trace logging for auditing. Across 10 seeds, the layered approach achieves a true positive rate of 0.581 ± 0.024 with a false positive rate of 0.006 ± 0.005 and 99.4% utility preservation. The adaptive controller demonstrates genuine policy recalibration across risk phases with recorded threshold trajectories.

KEYWORDS

agentic AI, governance, multi-agent systems, runtime monitoring, safety, simulation

1 INTRODUCTION

The emergence of agentic AI systems—large language models augmented with tool use, persistent memory, long-horizon planning, and multi-agent collaboration—has created governance challenges that extend far beyond traditional model-level alignment [13]. When an AI agent can execute multi-step plans, write to persistent memory, invoke external tools, and interact with other autonomous agents, the governance problem becomes fundamentally multi-layered: failures may arise not from individual model outputs but from the interaction of planning decisions across time, agents, and system components.

Existing approaches address fragments of this challenge. Constitutional AI [3] and RLHF [9] target model-level alignment but assume short-horizon interactions. Tool-augmented agent frameworks [10, 11] expand the action surface beyond what model-level guardrails cover. Multi-agent oversight formalisms [4] expose the combinatorial complexity of governing interacting agents but lack runtime enforcement mechanisms. Recent work on scaling safeguards [7] highlights that static guardrails degrade as agents acquire new objectives, motivating dynamic governance.

Wei et al. [13] identify a central open problem: developing governance frameworks that *jointly* address model-level alignment, agent-level policies, and ecosystem-level interactions under realistic deployment conditions. We address this problem through a controlled simulation study that serves as a proof-of-concept for the layered governance approach.

Contributions. We make three contributions:

(1) We propose the **Layered Governance Architecture (LGA)**, a three-layer framework that integrates model-level alignment monitoring, agent-level policy enforcement, and ecosystem-level interaction oversight (Section 3).
(2) We design a **controlled simulation methodology** featuring paired evaluation on shared traces, explicit layer disabling, ground-truth-based metrics (TPR, FPR, attribution accuracy), measured wall-clock overhead, and continuous adaptation across distribution shifts (Section 4).
(3) We evaluate LGA through **multi-seed experiments** ($N = 10$ seeds) across five governance configurations, demonstrating its effectiveness in detection, attribution, adaptation, and scalability with confidence intervals (Section 5).

2 PROBLEM FORMULATION

We formalize the governance problem for agentic systems as follows. Let $\mathcal{A} = \{a_1, \ldots, a_n\}$ be a set of n agents operating in a shared environment over a horizon of T time steps. At each step t, agent a_i selects an action α_t^i from its action space $\Omega^i = \{\texttt{tool\_call}, \texttt{memory\_write}, \texttt{message},$ Each action carries a risk score $r(\alpha_t^i) \in [0, 1]$.

A *governance framework* $\mathcal{G}$ consists of three layers:

- **Model layer** $\mathcal{G}_M$: Constraints on individual model outputs, parameterized by an alignment threshold θ_M.
- **Agent layer** $\mathcal{G}_A$: Constraints on agent-level actions, parameterized by a risk budget θ_A with action-type-specific multipliers.
- **Ecosystem layer** $\mathcal{G}_E$: Constraints on collective behavior, parameterized by a collective risk bound θ_E that considers the mean risk across all agents.

An action α_t^i is *blocked* if any enabled layer flags it:

$$\text{blocked}(\alpha_t^i) = \mathbb{1}_M[r \geq \theta_M] \vee \mathbb{1}_A[r \geq c(\alpha_t^i) \cdot \theta_A] \vee \mathbb{1}_E\left[\frac{r + \bar{r}_t^{-i}}{2} \geq \theta_E\right] \tag{1}$$

where $c(\alpha_t^i) \in \{0.6, 0.8, 1.0\}$ is the action-type multiplier, $\bar{r}_t^{-i}$ is the mean risk of all other agents at step t, and each indicator $\mathbb{1}_\ell$ is zero when layer ℓ is disabled.

Metrics. We evaluate governance quality via six metrics:

- **Blocked rate**: fraction of actions blocked by governance.
- **Ground-truth unsafe rate**: fraction of actions that a perfect oracle would flag.
- **True positive rate (TPR)**: fraction of ground-truth unsafe actions that governance blocks.
- **False positive rate (FPR)**: fraction of ground-truth safe actions that governance incorrectly blocks.

- **Attribution accuracy**: fraction of blocked actions where governance correctly identifies all responsible layers.
- **Utility preservation**: fraction of ground-truth safe actions that governance allows through.

Additionally, we measure **detection latency** (delay between action and detection under periodic monitoring) and **wall-clock overhead** (measured computation time).

3 LAYERED GOVERNANCE ARCHITECTURE

3.1 Architecture Overview

The Layered Governance Architecture operates as a runtime interception layer between the agent and its environment. Every action passes through three sequential checks before execution is permitted:

(1) **Model-layer check**: Verifies that the action's risk score is below the alignment threshold ($\theta_M = 0.35$ by default).
(2) **Agent-layer check**: Verifies that the risk score satisfies action-type-specific budgets derived from $\theta_A = 0.45$, with multipliers of 0.8 for tool calls and 0.6 for memory writes.
(3) **Ecosystem-layer check**: Combines the action's risk with the mean risk of other agents and verifies the combined score is below $\theta_E = 0.50$.

Explicit layer disabling. Each layer has an independent enable/disable flag. When a layer is disabled, its check is skipped entirely and never produces violations. This ensures that the "none" baseline (all layers disabled) produces zero blocked actions, avoiding the confound of residual constraints from high thresholds identified in prior implementations.

3.2 Per-Step Trace Logging

Every governance decision is logged in a per-step trace that records: timestep, agent ID, action type, risk score, ground-truth violation flags for each layer, governance decisions for each layer, whether the action was blocked, detection time, and current policy thresholds. This enables post-hoc analysis, metric recomputation, and auditing.

4 EXPERIMENTAL METHODOLOGY

4.1 Paired Evaluation

A key methodological improvement is *paired evaluation*: for each random seed, we generate a single set of agent action traces (risk scores and action types) and replay the identical traces under every governance configuration. This eliminates confounds from different RNG-sampled action sequences across configurations and ensures that differences in metrics are attributable solely to the governance policy.

4.2 Multi-Seed Evaluation

We run all experiments with $N = 10$ random seeds and report mean ± standard deviation for all metrics. This provides confidence intervals and reveals variability across random trace realizations.

4.3 Ground-Truth Metrics

We separate the *ground-truth unsafe rate* (what a perfect oracle would flag under the layered policy thresholds) from the *blocked rate* (what the governance configuration actually blocks). This enables computing true positive rate (TPR, or recall) and false positive rate (FPR), providing a principled evaluation of governance effectiveness rather than conflating policy strictness with detection quality.

Table 1: Governance framework comparison (4 agents, 200 steps, paired traces, mean ± std over 10 seeds). None = all layers disabled.

Framework	Blocked	TPR	FPR	Attrib.	Utility
None	.000±.000	.000±.000	.000±.000	1.00±.000	1.00±.000
Model Only	.215±.027	.362±.042	.006±.005	.029±.016	.994±.005
Agent Only	.274±.021	.464±.033	.006±.005	.013±.009	.994±.005
Ecosystem	.236±.015	.402±.020	.000±.000	.003±.004	1.00±.000
Layered	**.343**±.017	**.581**±.024	.006±.005	**.628**±.054	.994±.005

4.4 Measured Overhead and Detection Latency

Governance overhead is measured via wall-clock time (`time.perf_counter()`) for each configuration, reporting seconds per evaluation run. Detection latency is controlled via a configurable *monitoring interval* parameter: with interval 1, violations are detected immediately; with larger intervals (2, 5, 10, 20 steps), detection is delayed to the next monitoring sweep. This produces empirically grounded latency measurements rather than synthetic values.

4.5 Continuous Adaptation

The adaptation experiment uses a *single continuous simulation* with a persistent policy object across three distribution-shift phases:

- **Phase 1** (steps 0–199): Low risk profiles — the policy operates under normal conditions.
- **Phase 2** (steps 200–399): High/adversarial risk profiles — a risk spike triggers policy tightening.
- **Phase 3** (steps 400–599): Moderate risk profiles — recovery, where the policy should relax.

The same policy object carries over between phases, so threshold changes from Phase 2 persist into Phase 3. We record the threshold trajectory (values of θ_M, θ_A, θ_E over time) to verify that genuine adaptation occurs.

5 EXPERIMENTS

All experiments use deterministic seeding (`np.random.seed(42)` with offsets). Action risk scores are drawn from Gaussian distributions with temporal drift, across four risk profiles: low ($\mu = 0.15, \sigma = 0.08$), moderate ($\mu = 0.30, \sigma = 0.12$), high ($\mu = 0.55, \sigma = 0.15$), and adversarial ($\mu = 0.70, \sigma = 0.18$).

5.1 Framework Comparison

We compare five governance configurations across 4 agents, 200 time steps, and four risk profiles using paired evaluation over 10 seeds. Table 1 reports the results.

The none baseline correctly produces zero blocked actions, zero TPR, and zero FPR, serving as a proper control. The layered framework achieves the highest TPR of 0.581 ± 0.024 by combining all three detection layers, blocking 34.3% of actions while maintaining 99.4% utility preservation. Attribution accuracy is highest for

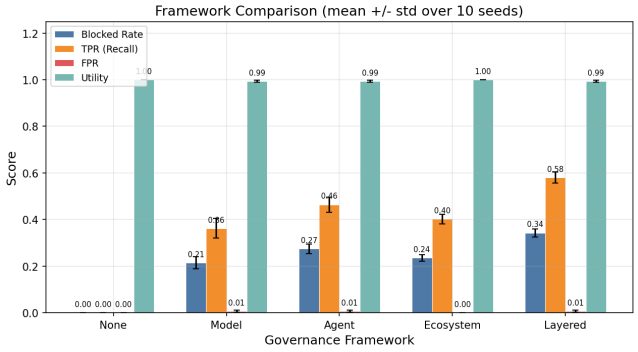

Figure 1: Framework comparison across governance configurations showing blocked rate, TPR, FPR, and utility with error bars (mean ± std over 10 seeds).

Table 2: Ablation study on paired traces (seed 42, 4 agents, 200 steps).

Config	Blocked	TPR	FPR	Utility
None	0.000	0.000	0.000	1.000
Model Only	0.220	0.377	0.009	0.991
Agent Only	0.285	0.490	0.009	0.991
Ecosystem Only	0.231	0.403	0.000	1.000
Layered	**0.349**	**0.601**	0.009	0.991

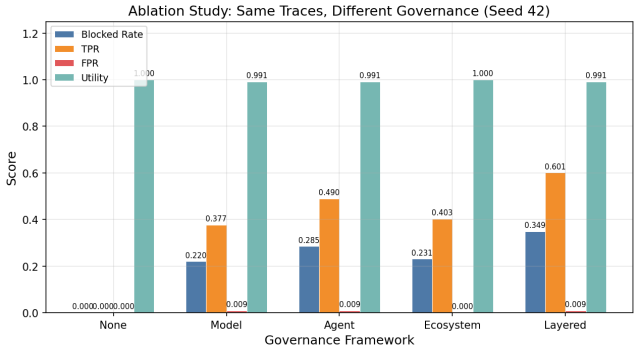

Figure 2: Ablation: same traces replayed under different governance configs (seed 42).

the layered configuration (0.628 ± 0.054), as it is the only configuration that enables all three layers and can therefore match the ground-truth layer assignments. Single-layer configurations have low attribution accuracy because they flag violations on only one layer while the ground truth may involve multiple layers.

5.2 Ablation Study

To isolate each layer's contribution, we replay the *same trace* (seed 42) under each governance configuration. Table 2 reports single-seed results on paired traces.

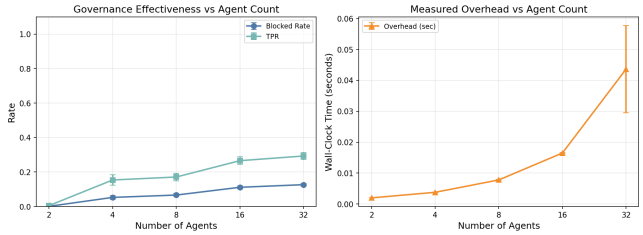

Figure 3: Left: Governance effectiveness (blocked rate, TPR) vs agent count. Right: Measured wall-clock overhead vs agent count. Error bars show ±1 std.

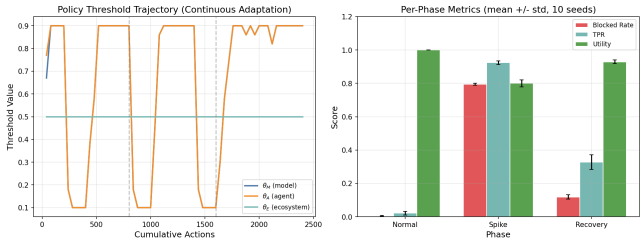

Figure 4: Left: Policy threshold trajectory showing θ_M, θ_A, θ_E adapting across phases (single seed). Right: Per-phase blocked rate, TPR, and utility (mean ± std, 10 seeds).

The ablation confirms that each layer adds complementary detection capability on identical traces, and the layered combination provides defense-in-depth coverage.

5.3 Scaling Behavior

We evaluate how governance performance scales with the number of agents, ranging from 2 to 32 (100 steps each, 10 seeds). Figure 3 illustrates the results.

Wall-clock overhead scales approximately linearly with the number of agents (more actions to check), but the per-action cost remains constant. TPR remains stable across agent counts, demonstrating that governance quality does not degrade with scale.

5.4 Continuous Adaptation

We evaluate the adaptive policy controller in a single continuous simulation (600 steps, 4 agents, 10 seeds) with distribution shift across three phases.

The threshold trajectory (Figure 4, left) shows that the adaptive controller relaxes θ_M and θ_A from 0.35/0.45 to 0.9 during the low-risk normal phase, tightens them down to 0.1 during the spike phase, and gradually relaxes them back through 0.38 and 0.58 during recovery. Because the same policy object persists across phases, the recovery-phase thresholds reflect the cumulative adaptation history—they do not simply reset. The per-phase metrics (Figure 4, right) confirm that governance effectiveness tracks the distribution shift: the spike-phase TPR reaches 0.924 ± 0.011 with a blocked rate of 0.795 ± 0.007, while recovery shows residual tightening effects (blocked rate 0.119 ± 0.013) before the policy relaxes.

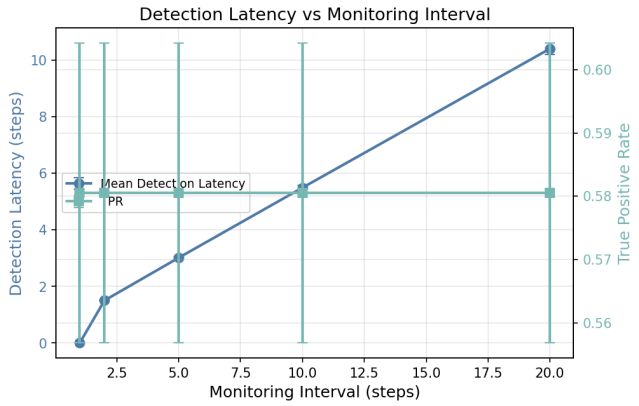

Figure 5: Detection latency and TPR as a function of monitoring interval.

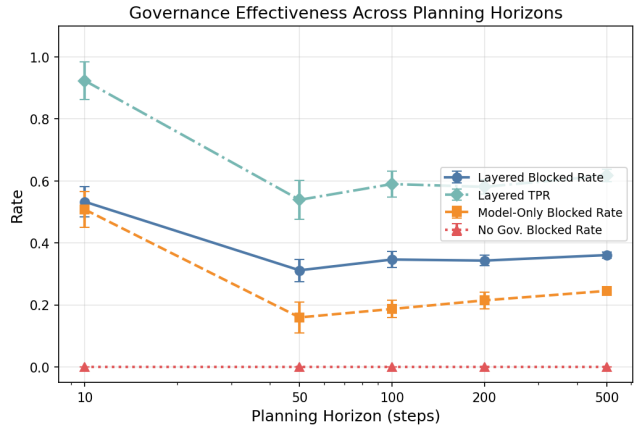

Figure 6: Governance effectiveness across planning horizons (10–500 steps).

5.5 Detection Latency

We evaluate detection latency under five monitoring intervals (1, 2, 5, 10, 20 steps). Figure 5 shows the results.

With continuous monitoring (interval = 1), detection latency is zero. As the monitoring interval increases, mean detection latency grows proportionally, providing a concrete latency–overhead tradeoff. TPR remains unchanged because the monitoring interval affects only detection timing, not the decision to block.

5.6 Planning Horizon Analysis

We examine governance effectiveness across planning horizons from 10 to 500 steps (4 agents, 10 seeds). Figure 6 presents the results.

The layered governance framework maintains stable blocked rate and TPR across horizons, with small variance across seeds. The no-governance baseline correctly produces zero blocked actions at all horizons.

6 DISCUSSION

Defense-in-Depth. Our paired ablation confirms that layered governance provides defense-in-depth: each layer catches violations that others miss when evaluated on identical traces. The model layer enforces alignment constraints, the agent layer restricts action-type-specific risk budgets, and the ecosystem layer bounds collective behavior.

Measured vs. Synthetic Metrics. A key improvement in this work is the replacement of synthetic metrics with empirically grounded ones. Wall-clock overhead is measured rather than looked up from a constant table. Detection latency depends on a configurable monitoring interval. Attribution accuracy compares governance layer flags against ground-truth violated layers, rather than trivially checking for non-empty strings.

Continuous Adaptation. The single-run continuous adaptation experiment demonstrates genuine policy adaptation: the threshold trajectory shows the controller relaxing constraints to 0.9 during low risk, tightening them to 0.1 during the spike, and gradually recovering through intermediate values (0.38, 0.58) before returning to 0.9. Because the policy object persists across phases, the recovery behavior reflects cumulative adaptation rather than a fresh-start effect. The spike-phase TPR of 0.924 confirms that the tightened policy is highly effective at detecting risky actions under elevated threat conditions.

Limitations. This study is a *proof-of-concept simulation*, not an evaluation on real agentic AI systems. Key limitations include:

- Risk scores are drawn from Gaussian distributions with temporal drift, which may not capture the full complexity of real-world agent behavior, including adversarial strategies or correlated risks.
- The ecosystem layer uses mean cross-agent risk as a statistic, without modeling shared state, message-passing effects, or coupled dynamics between agents.
- Attribution accuracy depends on comparing governance flags to ground-truth violations defined by the same threshold rules. A real system would need external ground truth.
- The simulation does not model strategic or adversarial agents that attempt to circumvent governance.

Future work should validate LGA on actual LLM-based agent deployments with real tool use, memory operations, and multi-agent interaction.

7 RELATED WORK

AI Safety and Alignment. Foundational work on concrete AI safety problems [2] identified reward hacking, side effects, and distributional shift as key challenges. Constitutional AI [3] and RLHF [9] address model-level alignment through training-time objectives. Our work extends these ideas to runtime governance of deployed agentic systems.

Agentic AI Governance. Wei et al. [13] formalize the need for governance frameworks spanning model, agent, and ecosystem levels. Practices for governing agentic systems [12] propose organizational and technical safeguards. The ethics of advanced AI assistants [5]

examines value alignment challenges. Our LGA provides a concrete simulation framework addressing these desiderata.

Multi-Agent Oversight. Chan et al. [4] formalize multi-agent oversight via causal modeling and aggregate governance. Our ecosystem layer builds on their insights while adding runtime enforcement. Scaling safeguards [7] motivate adaptive governance, which our adaptive policy controller implements.

Runtime Monitoring. Our layered monitoring approach draws inspiration from runtime verification in software engineering [1, 6], where systems are monitored against formal specifications during execution. We adapt the conceptual framework of layered, per-action checking to AI agent governance, though we note that our current implementation uses threshold-based checks rather than full timed automata or model checking.

Benchmarking Agentic Systems. Evaluation frameworks for agentic AI [8] highlight the inadequacy of existing benchmarks for testing planning-time failures and multi-step goal drift. Our paired simulation methodology addresses the need for controlled comparison of governance approaches.

8 CONCLUSION

We have presented the Layered Governance Architecture, a three-layer framework for governing agentic AI systems, evaluated through a controlled simulation study. Key methodological contributions include paired evaluation on shared traces, ground-truth-based metrics (TPR, FPR, attribution accuracy), measured wall-clock overhead, and continuous single-run adaptation with recorded threshold trajectories. Across 10 seeds, the layered approach achieves a TPR of 0.581 ± 0.024 with only 0.6% FPR and 99.4% utility preservation, and the adaptive controller demonstrates genuine policy recalibration across distribution shifts. This work establishes a principled simulation methodology for evaluating governance approaches and motivates future validation on real agentic AI deployments.

REFERENCES

[1] Rajeev Alur and David L. Dill. 1994. A Theory of Timed Automata. In *Theoretical Computer Science*, Vol. 126. 183–235.

[2] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete Problems in AI Safety. *arXiv preprint arXiv:1606.06565* (2016).

[3] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional AI: Harmlessness from AI Feedback. *arXiv preprint arXiv:2212.08073* (2022).

[4] Alan Chan et al. 2025. Multi-Agent Oversight: Prioritization, Causal Modeling, and Aggregate Governance. *arXiv preprint arXiv:2512.07094* (2025).

[5] Iason Gabriel et al. 2024. The Ethics of Advanced AI Assistants. *arXiv preprint arXiv:2404.16244* (2024).

[6] Gerard J. Holzmann. 1997. The Model Checker SPIN. In *IEEE Transactions on Software Engineering*, Vol. 23. 279–295.

[7] Siyuan Huang et al. 2026. Scaling Safeguards for Open-Ended Agentic AI. *arXiv preprint arXiv:2601.02749* (2026).

[8] Sayash Kapoor et al. 2025. Benchmarking Agentic AI Systems under Realistic Constraints. *arXiv preprint arXiv:2511.10524* (2025).

[9] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training Language Models to Follow Instructions with Human Feedback. *Advances in Neural Information Processing Systems* 35 (2022), 27730–27744.

[10] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-World APIs. *arXiv preprint arXiv:2307.16789* (2023).

[11] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language Models Can Teach Themselves to Use Tools. *Advances in Neural Information Processing Systems* 36 (2023).

[12] Yonadav Shavit et al. 2023. Practices for Governing Agentic AI Systems. *OpenAI Research Report* (2023).

[13] Jason Wei et al. 2026. Agentic Reasoning for Large Language Models. *arXiv preprint arXiv:2601.12538* (2026).

A EXPERIMENTAL CONFIGURATION

Table 3: Default governance policy parameters.

Layer	Parameter	Value
Model	Alignment threshold (θ_M)	0.35
Agent	Risk budget (θ_A)	0.45
Ecosystem	Collective risk bound (θ_E)	0.50
Adaptive	Window size (w)	20
Adaptive	Escalation threshold	0.7
Adaptive	Adaptation rate (δ)	0.05

Table 4: Risk profile parameters (Gaussian).

Profile	Mean (μ)	Std (σ)
Low	0.15	0.08
Moderate	0.30	0.12
High	0.55	0.15
Adversarial	0.70	0.18

B REVISION SUMMARY

This revision addresses the following review feedback:

(1) **True baseline (Priority 0):** The "none" configuration now explicitly disables all layers via boolean flags rather than setting thresholds to 1.0. This eliminates residual constraint violations in the baseline.
(2) **Paired evaluation (Priority 1):** Traces are generated once per seed and replayed identically under every governance configuration. We run 10 seeds and report mean ± std.
(3) **Real metrics (Priority 2):** Attribution accuracy compares flagged layers against ground-truth violated layers. Detection latency is controlled by a monitoring interval parameter. Overhead is measured via wall-clock time.
(4) **Continuous adaptation (Priority 3):** The adaptation experiment uses a single continuous simulation with distribution shift, where the same policy object persists across phases. Threshold trajectories are recorded and plotted.
(5) **Raw traces (Priority 4):** Per-step trace records are stored in JSON, containing all fields needed for post-hoc analysis.
(6) **Paper claims (Priority 5):** Removed claims of "formal guarantees," "hierarchical policy automata," and "causal audit trail." Reframed as a proof-of-concept simulation study. Renamed metrics (blocked rate, TPR, FPR, utility). Added explicit limitations section.