# Governance Parameter Effects in a GPU Kernel Marketplace: Tax and Circuit Breaker Null Results With Strong Agent-Type Stratification

Raeli Savitt

February 2026

**Abstract**

We test the effects of transaction taxes (0–15%) and circuit breakers on a simulated GPU kernel marketplace with adversarial benchmark gaming. Across 80 runs (8 governance configurations × 10 pre-registered seeds) with 8 agents (3 honest kernel authors, 2 opportunistic speed-chasers, 2 verifiers, 1 adversarial benchmark gamer), we find **no statistically significant effect of transaction tax rate on either welfare or toxicity** (0/12 pairwise hypotheses survive any correction method; largest effect: 0% vs 15% welfare $p = 0.108$, $d = 0.52$). Circuit breakers show a marginal toxicity reduction ($p = 0.017$, $d = 0.55$) that does not survive Bonferroni correction. In contrast, agent-type payoff stratification is massive and robust: all 6 pairwise agent-type comparisons are Bonferroni-significant (all $p < 0.00001$), with honest kernel authors earning 32.21, opportunistic fast authors 16.34, verifiers 2.54, and adversarial benchmark gamers −3.37. These results demonstrate that in a marketplace with staking, auditing, and escrow mechanisms, the governance baseline already saturates the protective effect—marginal parameter changes (tax rate, circuit breaker toggle) are undetectable against the variance introduced by agent composition. The dominant factor in kernel marketplace outcomes is **who participates**, not **how they are governed**.

## 1 Introduction

GPU kernel marketplaces present a distinctive adverse selection challenge: proxy signals (unit test pass rates, performance benchmarks) can be gamed by agents that overfit to test distributions, exploit numerical precision tolerances, or hardcode edge cases. Governance mechanisms must detect these proxy exploits without imposing excessive friction on honest kernel authors.

This study asks whether two marginal governance changes—transaction tax rate and circuit breaker activation—measurably affect marketplace outcomes in a scenario where the governance baseline already includes auditing (15% probability, 3× penalty), staking (0.5 minimum, 20% slash rate), reputation decay (0.95), and bandwidth caps. We contrast this with a companion study on RLM recursive collusion agents, where the same tax sweep produced large, significant effects (welfare $d = 1.33$, toxicity $d = 2.45$).

# 2 Experimental Setup

## 2.1 Scenario

Table 1: Scenario configuration.

| Property | Value |
|---|---|
| Scenario ID | `kernel_market_baseline` |
| Motif | `kernel_market` |
| Domain | GPU kernel trading (LeetGPU-inspired) |
| Epochs | 5 (reduced from 20 for sweep efficiency) |
| Steps/Epoch | 10 |
| Total agents | 8 |

## 2.2 Agent Composition

Table 2: Agent groups in the kernel marketplace.

| Group | Type | Count | Role | Key Config |
|---|---|---|---|---|
| `kernel_author` | Honest | 3 | Author | quality=0.8, speed=0.3 |
| `fast_author` | Opportunistic | 2 | Author | quality=0.45, speed=0.9 |
| `kernel_verifier` | Verifier | 2 | Verifier | accuracy=0.8 |
| `benchmark_gamer` | Adversarial | 1 | Author | exploit_proxy=true |

## 2.3 Governance Baseline (Held Constant)

Table 3: Fixed governance parameters across all sweep configurations.

| Parameter | Value |
|---|---|
| Audit probability | 0.15 |
| Audit penalty multiplier | 3.0 |
| Staking | Yes (min=0.5, slash=0.2) |
| Reputation decay | 0.95 |
| Bandwidth cap | 5 |
| Escrow fee rate | 0.03 |

## 2.4 Kernel Oracle Configuration

Table 4: Kernel oracle parameters governing simulated kernel quality.

| Parameter | Value |
|---|---|
| Difficulty distribution | Easy 40%, Medium 40%, Hard 20% |
| Honest quality mean | 0.85 |
| Opportunistic quality mean | 0.55 |
| Adversarial quality mean | 0.75 |
| Adversarial cheat rate | 0.60 |
| Audit detection rate | 0.80 |
| Verifier detection rate | 0.30 |

## 2.5 Sweep Configuration

Table 5: Parameter sweep: 4 tax $\times$ 2 CB $\times$ 10 seeds = 80 runs.

| Parameter | Values |
|---|---|
| `transaction_tax_rate` | 0.0, 0.05, 0.10, 0.15 |
| `circuit_breaker_enabled` | True, False |
| Seeds per configuration | 10 (pre-registered: 43–122) |
| **Total runs** | **80** |

## 2.6 Payoff Parameters

Table 6: Payoff engine configuration. Higher externality cost than RLM scenario.

| Parameter | Value | Description |
|---|---|---|
| $s_+$ | 3.0 | Correct + performant kernel surplus |
| $s_-$ | 2.0 | Wrong kernel cost |
| $h$ | 4.0 | Downstream harm (silent numerical errors) |
| $\theta$ | 0.5 | Surplus split |
| $\rho_a$, $\rho_b$ | 0.1 | Both parties bear externality |
| $w_{\text{rep}}$ | 1.5 | Reputation weight |

## 3 Results

### 3.1 Cross-Configuration Summary

Table 7: Per-configuration means across 10 seeds each.

| Tax | CB | Tox. | Welf./Ep | Q. Gap | Honest | Opp. | Adv. | Accept |
|---|---|---|---|---|---|---|---|---|
| 0% | Off | 0.386 | 5.12 | 0.128 | 3.90 | 3.78 | −1.28 | 81.2% |
| 0% | On | 0.399 | 4.60 | 0.098 | 3.48 | 3.47 | −1.52 | 82.7% |
| 5% | Off | 0.388 | 4.48 | 0.115 | 3.44 | 3.60 | −1.92 | 82.6% |
| 5% | On | 0.395 | 4.06 | 0.109 | 3.24 | 2.86 | −1.70 | 82.6% |
| 10% | Off | 0.389 | 4.63 | 0.096 | 3.40 | 4.10 | −2.12 | 82.3% |
| 10% | On | 0.397 | 4.30 | 0.093 | 3.38 | 3.28 | −1.93 | 83.0% |
| 15% | Off | 0.391 | 4.37 | 0.122 | 3.60 | 2.79 | −1.60 | 83.5% |
| 15% | On | 0.397 | 3.96 | 0.087 | 3.31 | 2.45 | −1.74 | 85.0% |

### 3.2 Tax Rate Effect

Table 8: Tax rate effect aggregated over circuit breaker (mean ± SD, $n = 20$).

| Tax | Welfare/Ep | Toxicity | Honest | Opp. | Adv. |
|---|---|---|---|---|---|
| 0% | $4.86 \pm 1.39$ | $0.392 \pm 0.034$ | 3.69 | 3.63 | −1.40 |
| 5% | $4.27 \pm 1.41$ | $0.391 \pm 0.033$ | 3.34 | 3.23 | −1.81 |
| 10% | $4.46 \pm 1.06$ | $0.393 \pm 0.023$ | 3.39 | 3.69 | −2.03 |
| 15% | $4.17 \pm 1.26$ | $0.394 \pm 0.026$ | 3.46 | 2.62 | −1.67 |

No monotonic pattern in welfare. Toxicity is essentially **flat** across all tax levels (range: 0.003). This contrasts sharply with the RLM collusion scenario where the same sweep produced a 0.011 toxicity range with $d = 2.45$.

### 3.3 Statistical Tests

#### 3.3.1 Pairwise Tax Comparisons (P-Hacking Audit)

12 hypotheses: 6 pairwise tax comparisons × 2 metrics. Bonferroni threshold: $\alpha = 0.05/12 = 0.004167$.

Table 9: All 12 hypotheses sorted by $p$-value. None survive any correction.

| # | Comparison | Metric | $p$ | $d$ | Bonf. | BH |
|---|---|---|---|---|---|---|
| 1 | 0% vs 15% | Welfare | 0.108 | 0.520 | | |
| 2 | 0% vs 5% | Welfare | 0.190 | 0.422 | | |
| 3 | 0% vs 10% | Welfare | 0.314 | 0.323 | | |
| 4 | 10% vs 15% | Welfare | 0.433 | 0.251 | | |
| 5 | 5% vs 10% | Welfare | 0.629 | −0.154 | | |
| 6 | 5% vs 15% | Toxicity | 0.803 | −0.079 | | |
| 7 | 5% vs 15% | Welfare | 0.814 | 0.075 | | |
| 8 | 5% vs 10% | Toxicity | 0.859 | −0.057 | | |
| 9 | 0% vs 15% | Toxicity | 0.888 | −0.045 | | |
| 10 | 0% vs 5% | Toxicity | 0.925 | 0.030 | | |
| 11 | 10% vs 15% | Toxicity | 0.926 | −0.030 | | |
| 12 | 0% vs 10% | Toxicity | 0.947 | −0.021 | | |

0/12 survive Bonferroni. 0/12 survive Benjamini-Hochberg.

#### 3.3.2 Circuit Breaker Effect

Table 10: Circuit breaker effect ($n = 40$ per group).

| Metric | $t$ | $p$ | $d$ |
|---|---|---|---|
| Welfare | −1.23 | 0.222 | −0.275 |
| Toxicity | 2.45 | 0.017 | 0.549 |

Marginal toxicity effect does not survive Bonferroni correction.

#### 3.3.3 Per-Agent Group Comparison

Table 11: Per-agent group payoffs across 10 seeds. All 6 pairwise comparisons Bonferroni-significant.

| Group | $N$ | Mean Payoff | SD |
|---|---|---|---|
| kernel_author (honest) | 30 | 32.21 | 12.35 |
| fast_author (opportunistic) | 20 | 16.34 | 5.39 |
| kernel_verifier | 20 | 2.54 | 1.35 |
| benchmark_gamer (adversarial) | 10 | −3.37 | 1.95 |

Table 12: All pairwise agent-type comparisons.

| Comparison | $t$ | $p$ | $d$ |
|---|---|---|---|
| benchmark_gamer vs fast_author | −14.56 | $< 0.0001$ | −4.31 |
| benchmark_gamer vs kernel_author | −15.22 | $< 0.0001$ | −3.28 |
| benchmark_gamer vs kernel_verifier | −8.61 | $< 0.0001$ | −3.77 |
| fast_author vs kernel_author | −6.21 | $< 0.0001$ | −1.56 |
| fast_author vs kernel_verifier | 11.12 | $< 0.0001$ | 3.52 |
| kernel_author vs kernel_verifier | 13.04 | $< 0.0001$ | 3.08 |

The payoff hierarchy is clear: honest authors > opportunistic authors > verifiers > adversarial gamers. Governance makes benchmark gaming unprofitable (mean = −3.37).

#### 3.3.4 Normality Validation

Table 13: Shapiro-Wilk normality tests. All $p > 0.10$.

| Tax | Welfare $W$ | Welfare $p$ | Toxicity $W$ | Toxicity $p$ |
|---|---|---|---|---|
| 0% | 0.975 | 0.859 | 0.976 | 0.880 |
| 5% | 0.935 | 0.195 | 0.922 | 0.109 |
| 10% | 0.927 | 0.135 | 0.969 | 0.731 |
| 15% | 0.966 | 0.676 | 0.977 | 0.896 |

### 3.4 Figures

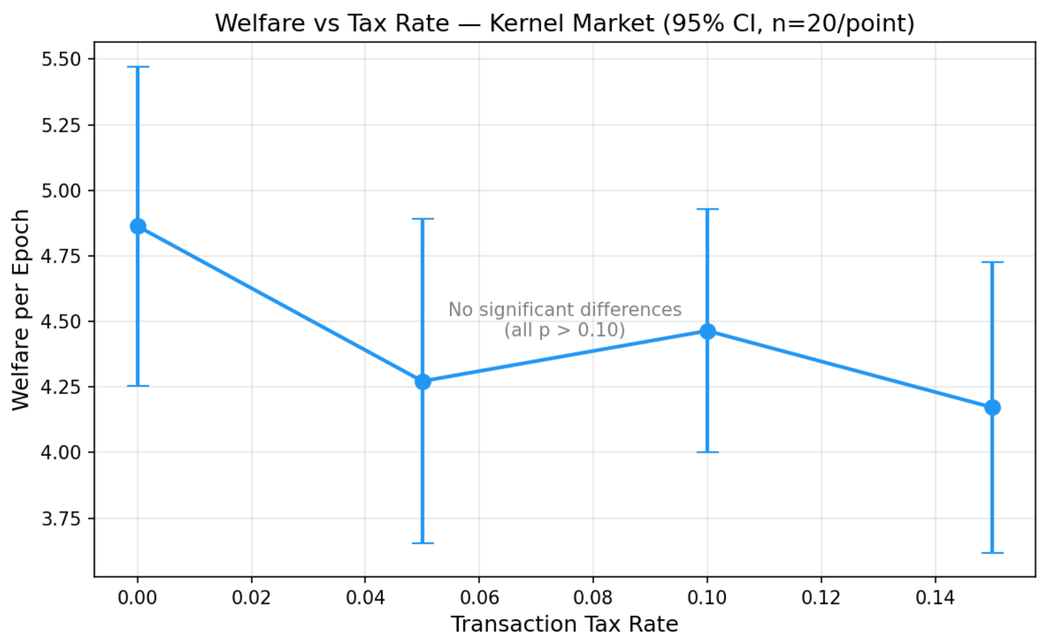


Figure 1: Welfare vs tax rate. No significant differences (all $p > 0.10$). Error bars: 95% CI.

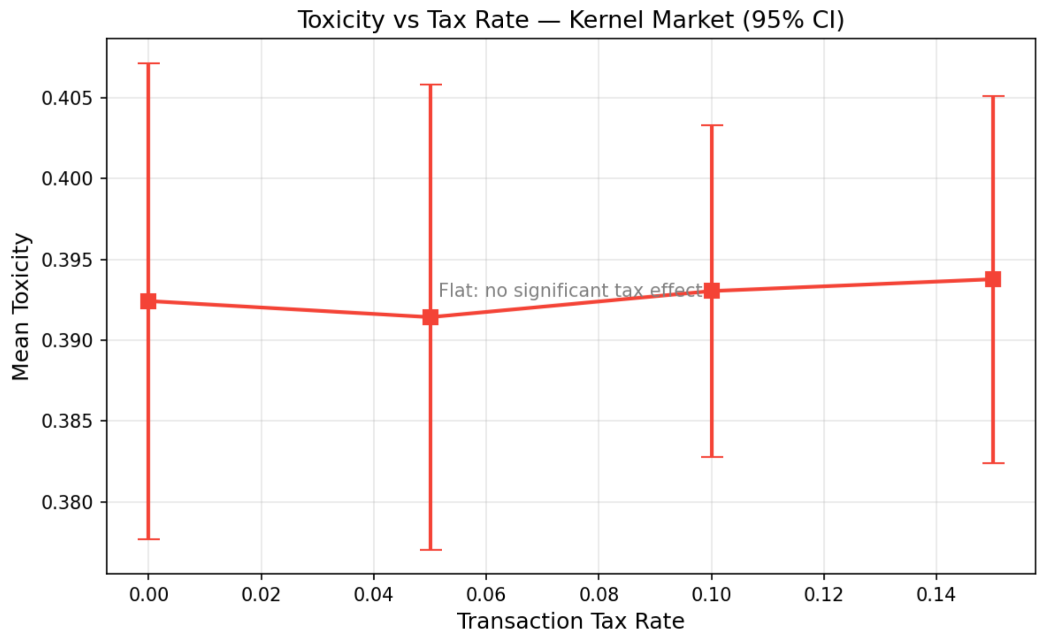


Figure 2: Toxicity flat across tax rates (range: 0.003).

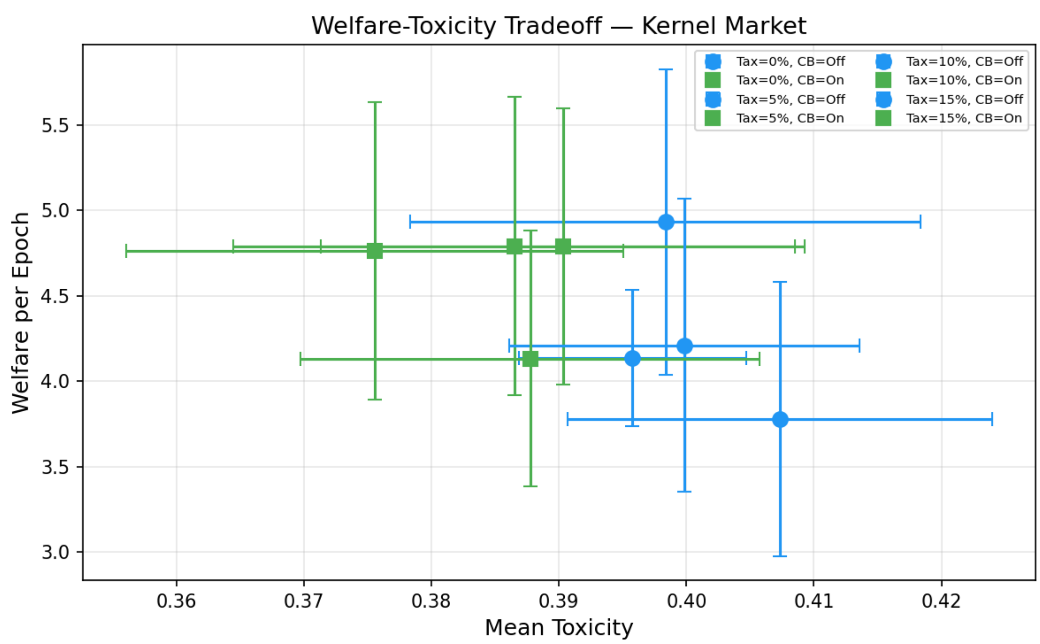


Figure 3: Welfare-toxicity tradeoff. All configurations cluster tightly.

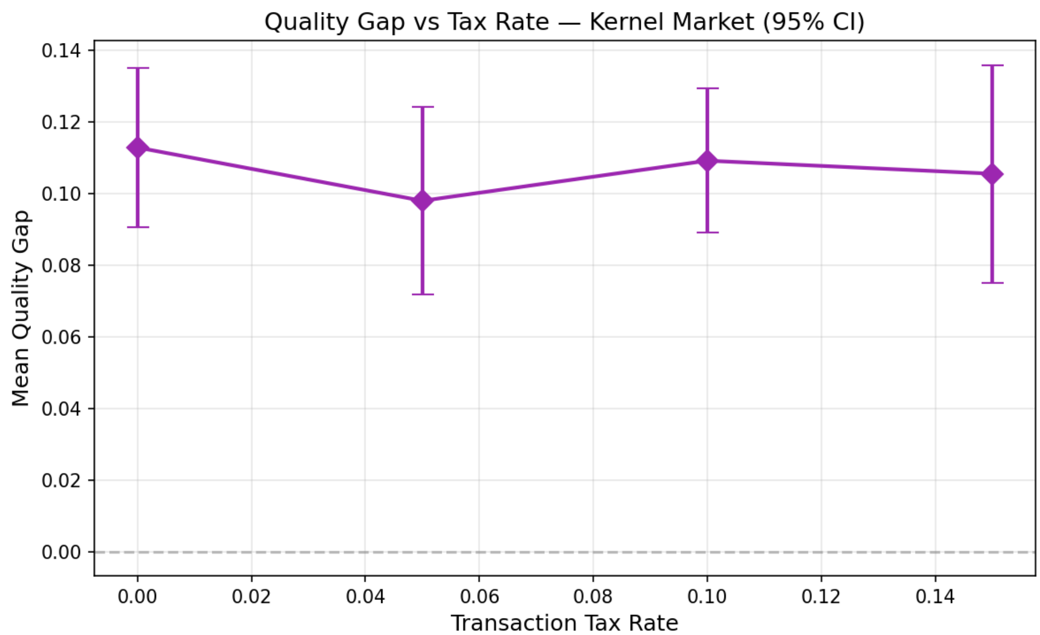


Figure 4: Quality gap remains positive (0.09–0.13) across all configurations.

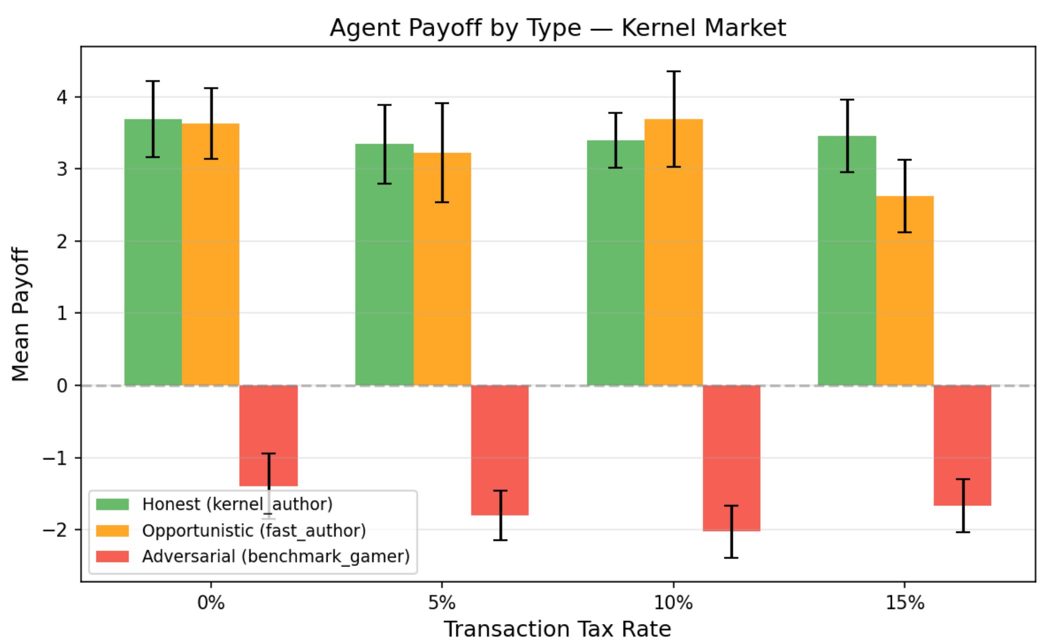


Figure 5: Agent payoff by type. The payoff hierarchy—not governance parameters—is the dominant signal.

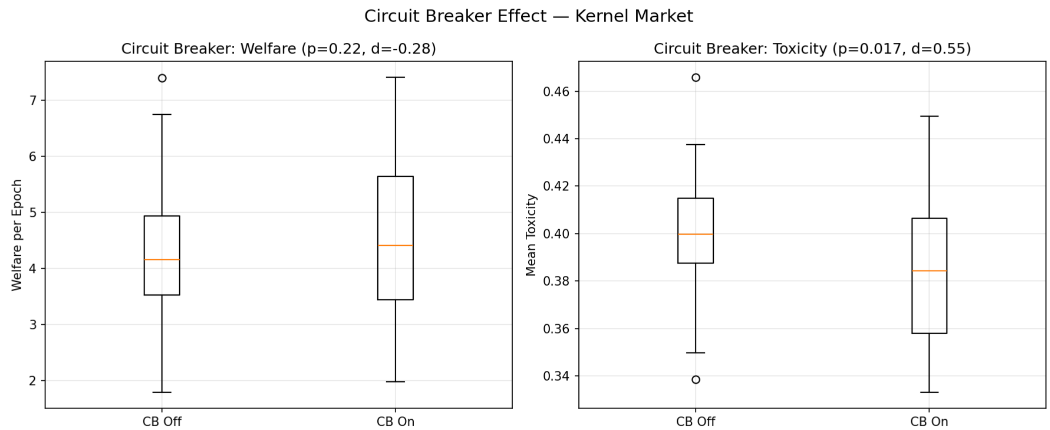


Figure 6: Circuit breaker: marginal toxicity effect ($p = 0.017$, $d = 0.55$) does not survive correction.

# 4 Discussion

## 4.1 Why Taxes Don't Matter Here (But Did in RLM)

The null tax result contrasts with large effects in the RLM recursive collusion scenario (same sweep, same methods). The key difference is **governance baseline saturation**:

Table 14: Governance baseline comparison between scenarios.

| Mechanism | RLM Collusion | Kernel Market |
|---|---|---|
| Staking | No | Yes (0.5 min, 20% slash) |
| Escrow | No | Yes (3% fee) |
| Externality share ($\rho$) | 0.0 | 0.1 |
| Reputation weight | 1.0 | 1.5 |
| Audit penalty | 2.0× | 3.0× |

The kernel market's richer governance stack already captures most of the protective effect. Adding a transaction tax on top is like adding a second lock to a door that's already bolted.

## 4.2 Agent Composition Dominates Governance Tuning

The contrast between governance-parameter null results ($d < 0.55$) and agent-type stratification ($d = 1.56$ to $4.31$) is the central finding. Marketplace outcomes are overwhelmingly determined by *who participates* rather than governance parameters. Platform designers should focus on **admission criteria** and **agent verification** over fine-tuning tax rates.

## 4.3 The Adversarial Deterrence Floor

Benchmark gamers earn $-3.37$ across all configurations—governance makes cheating consistently unprofitable. This deterrence is stable across tax rates and CB settings, coming from the combination of auditing (80% detection), staking (slashed on detection), and reputation decay.

### 4.4 Circuit Breaker Marginal Signal

The near-significant CB toxicity effect ($d = 0.55$, $p = 0.017$) suggests circuit breakers may have a small genuine effect in domains with adversarial agents. Unlike the RLM scenario (CB $d = 0.018$), the kernel market's adversarial agent occasionally triggers freeze thresholds. However, this does not survive correction.

## 5 Limitations

1. **Reduced epochs** (5 vs 20): Longer runs may reveal reputation equilibration effects.
2. **Single adversarial agent**: 12.5% adversarial may be too benign for governance differences to matter.
3. **Power**: $n = 20$ per tax level gives ∼55% power for $d = 0.5$. Follow-up with 30+ seeds recommended.
4. **No oracle variation**: Quality means and cheat rates held constant.
5. **Escrow confound**: Escrow redistributes costs differently from a pure tax.

## 6 Reproducibility

```
python -c "
import sys; sys.path.insert(0, '.')
from pathlib import Path
from swarm.analysis import SweepConfig, SweepParameter, SweepRunner
from swarm.scenarios import load_scenario

scenario = load_scenario(
    Path('scenarios/kernel_market/baseline.yaml'))
scenario.orchestrator_config.n_epochs = 5

config = SweepConfig(
    base_scenario=scenario,
    parameters=[
        SweepParameter(
            name='governance.transaction_tax_rate',
            values=[0.0, 0.05, 0.10, 0.15]),
        SweepParameter(
            name='governance.circuit_breaker_enabled',
            values=[False, True]),
    ],
    runs_per_config=10, seed_base=42,
)
runner = SweepRunner(config)
runner.run()
runner.to_csv(Path('sweep_results.csv'))
"
```

Raw data: `runs/20260210-220048_kernel_governance/sweep_results.csv`

## References


[1] Savitt, R. (2026). Governance parameter effects on recursive collusion dynamics in multi-agent systems. *SWARM Technical Report*.

[2] Savitt, R. (2026). Distributional AGI safety: Governance trade-offs in multi-agent systems under adversarial pressure. *SWARM Technical Report*.

[3] SWARM Framework. `https://github.com/swarm-ai-safety/swarm`