PHANTOM

Pricing heuristics against non-human transaction orchestration

Daniel Rösel
IE University ▪ Supervisor: Alberto Martín Izquierdo

velocitatem.github.io/PHANTOM

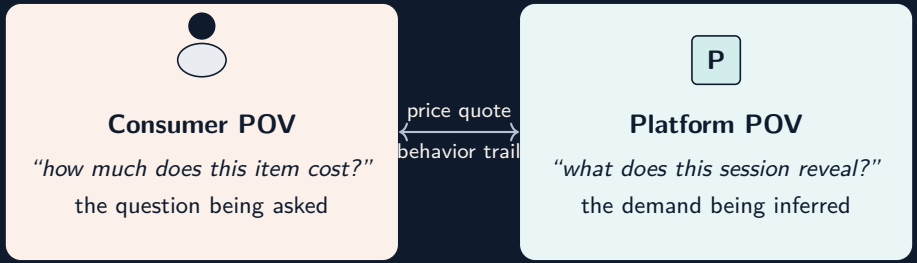
Roadmap: one argument in six stages

Main research question

How can dynamic pricing preserve margin integrity when transactions are increasingly mediated by non-human agents?

Dynamic pricing has often been treated as a secondary optimization layer; agent-mediated shopping turns it into a primary margin-risk surface.

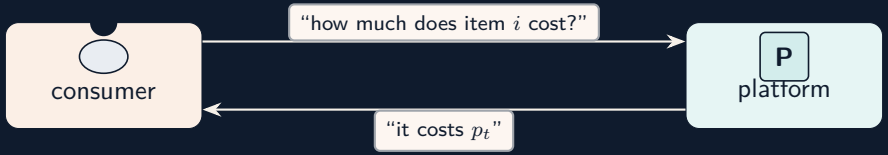
This work cycles through two points of view



Why two views

The same transaction looks very different from each side. We will switch between them to show where agent-mediation breaks the loop.

POV 1: the consumer asks the platform directly



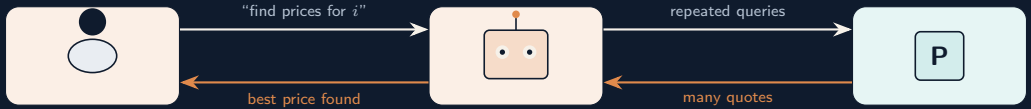
What you see: a website opens, you read a price, you decide.

What the platform sees: clicks, hovers, dwell time — a clean behavioral fingerprint of one human session.

Variables

i	the item being shopped (e.g. a hotel night)
p_t	posted price at time t, in EUR per booking

POV 2: the consumer asks an AI agent

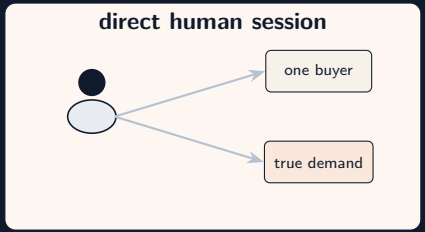


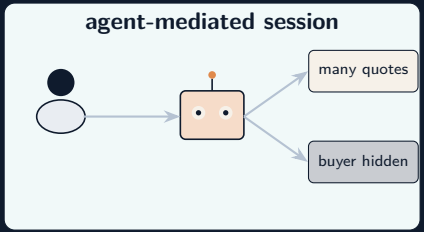
Same intent, different visible behavior. The platform sees a machine-paced session that looks nothing like the human who actually wants to buy.

Information asymmetry flips

The agent samples many quotes before committing; the platform only sees the recon, not the buyer.

Two flows, one demand signal — and only one is reliable





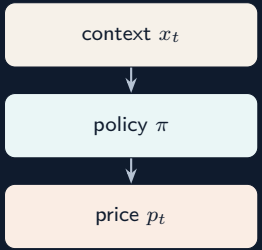
Takeaway. On the right, the platform sees recon, not intent. Pricing trained on the visible signal will misread real demand.

Policy first: one rule maps context into a price

Definition

$$p_t = \pi(x_t)$$

p_t	price quoted at time t [EUR per booking]
π	the pricing policy — a function the platform learns
x_t	context: product, time-of-day, and behavior summary of the session



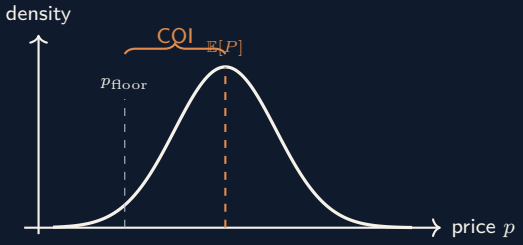
bandits first; later extended to DR-RL

Cost of Information (COI) — what the platform earns from knowing you

Definition

$$\mathrm{COI}(\pi) = \mathbb{E}[P] - p_{\text{floor}}$$

$\mathbb{E}[P]$	expected price the policy actually charges [EUR per booking]
p_{floor}	minimum viable price (marginal cost / break-even floor)
COI	average premium the platform extracts above marginal cost [EUR per transaction]



The "cost" is from the consumer's POV: it is what they pay because the platform can read their interest.

Revenue at risk equals COI × volume.

Why agents erode COI: the realizable price drops to the minimum

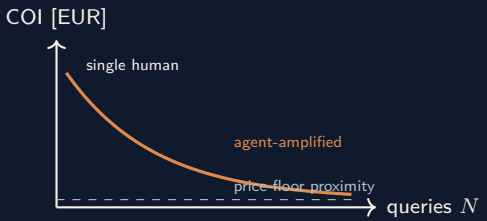
A single buyer pays the price they were quoted.

An agent samples N independent quotes and the buyer pays $p^{(1)} = \min(p_1, \ldots, p_N)$.

Result: as $N \to \infty$, $\mathrm{COI} \to 0$. More recon pushes realizable prices toward the floor.

One-line claim

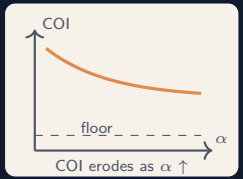
Untreated agentic recon behaves like an information leak that compresses sustainable margins.



The thesis answers one chain: mechanism → signal → control

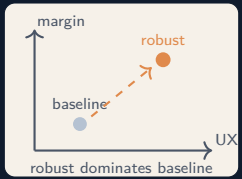
SQ1

Can we distinguish **H** and **A** sessions from interactions alone?



SQ2

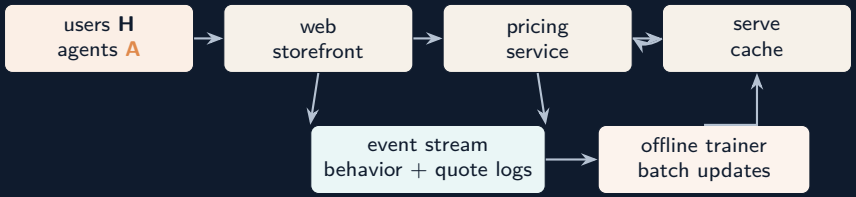
How strong is price and revenue erosion under agentic contamination?



SQ3

Can policy design recover margin while keeping UX stable?

Stage 1: a dual-loop platform pairs every quote with its behavior



- Every quote has a matching behavioral context in the log stream.
- The same architecture supports reproducible stress tests before any live deployment.

Dataset card: compact, labeled, experiment-ready

WhoClickedIt dataset card

`huggingface.co/datasets/velocitatem/whoclickedit`

human rows 798 **agent rows** 3076

Flat schema and explicit actor labels simplify session-aware train/test splits.

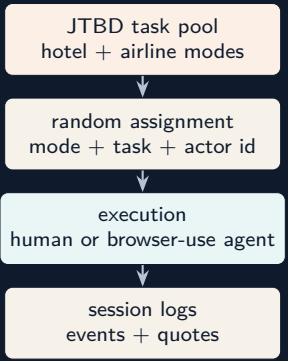
Kafka provenance is retained for reproducibility.

29 interviews
labeled trajectories

45% / 55%
human / agent split

2 streams
interaction + price logs

Experimental design controls goals, not instructions

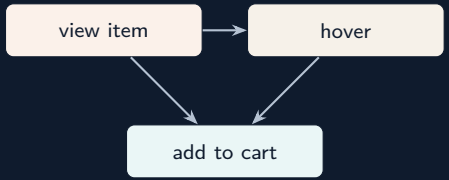


- Agents run with **browser-use** and a model-swappable LLM router (default `gpt-5-mini`).
- Tasks are defined by outcomes, not scripted clicks, to preserve behavioral variety.
- Current release is stronger on hotel flows than airline flows.

Stage 2: what is a kernel?

Plain definition

A **kernel** is a small square table T where $T[a, b]$ is the probability that action b follows action a inside one session. Every row sums to one.



example session graph

from \ to	view	hover	cart
view	0.00	0.64	0.36
hover	0.69	0.00	0.31
cart	0.00	0.00	1.00

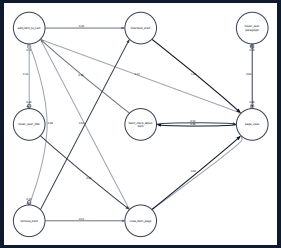
This is the kernel T. Each row is a probability distribution.

$T[a, b]$ probability that the next action is b given the current action is a

Humans and agents click in different patterns

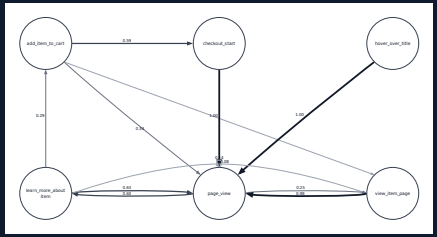
Human kernel $\bar{T}_H$

view → hover → cart, with detours



Agent kernel $\bar{T}_A$

view → view → view, almost no cart



-3.35
mean gap (human)

$+1.65$
mean gap (agent)

$p < 0.001$
Mann-Whitney rank

Two cohorts, two clearly separable click structures — this is the foundation of the detection signal.

From two divergences to one sigmoid score

Step 1 — distance to each prototype

$$\Delta_H = \mathrm{KL}(\hat{T}' \,\|\, \bar{T}_H)$$
$$\Delta_A = \mathrm{KL}(\hat{T}' \,\|\, \bar{T}_A)$$

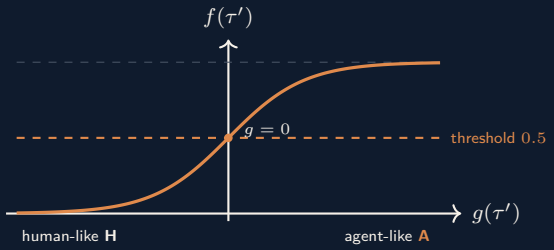
Step 2 — signed gap

$$g(\tau') = \Delta_H - \Delta_A$$

Step 3 — sigmoid squash

$$f(\tau') = \sigma\left(\frac{g(\tau')}{T}\right) \in [0, 1]$$

σ	the standard logistic sigmoid
T	temperature, controls how sharply the score moves away from 0.5



logistic curve — saturates at 0 and 1, threshold at 0.5

Stage 3: DR-RL trains against many plausible worlds

Standard RL trains against one demand model and overfits to it. **DR-RL** optimises the worst case across a small ball of plausible demand laws, so the policy still works when contamination shifts.

Robust objective

$$\pi^\star = \arg\max_{\pi} \min_{Q \in U_\epsilon} \mathbb{E}_Q[\, r \,]$$

- Q — a candidate demand distribution inside the ball
- U_ϵ — Wasserstein ball of radius ϵ around the empirical $\hat{P}_N$
- r — per-step reward (defined next slide)

Implementation note: in code we solve a local robust loop on the contamination parameter α, not the full continuous Wasserstein adversary.

Reward: revenue, minus leakage, minus UX cost

$$r_t = \underbrace{R(p_t, \hat{Q}_t)}_{\text{revenue}} - \underbrace{\lambda\, f(\tau'_t)\, c_{\text{info}}}_{\text{leakage penalty}} - \underbrace{\eta_{\text{ux}}\, \mathrm{UX}(\tau'_t, p_t)}_{\text{UX penalty}}$$

Revenue

$R(p_t, \hat{Q}_t) = p_t \cdot \hat{Q}_t$

units: EUR per session

Leakage

scales with $f(\tau'_t)$ **A**

λ: weight

c_{info}: per-query cost

UX

$\mathrm{UX} \in [0, 1]$

η_{ux}: weight

penalises unstable pricing

Reading the formula: if a session looks agent-like ($f \uparrow$), the leakage penalty grows and the policy backs off; for clean human sessions only the revenue and UX terms are active.

Wide sweeps are feasible only with aggressive optimization

$4 \times 4 \times 3 \times 2 \times 2 = \mathbf{192}$ configs

algorithms × contamination × robustness × COI penalty × action grid

160 PFLOPS
peak aggregate TPU budget

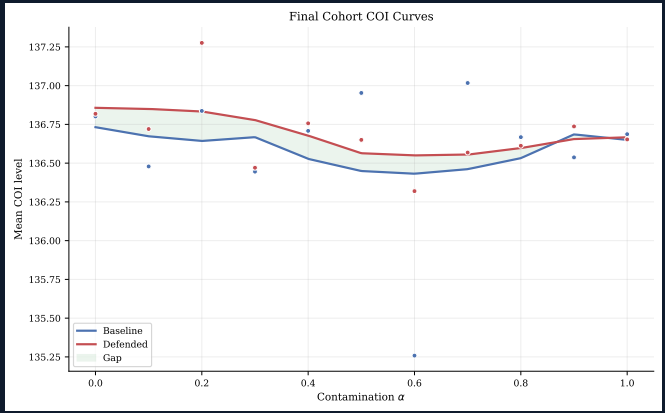
∼180 days
net compute logged

Hot-path rewrite impact

Mode	Before	After
Baseline step/s	26.0	220.0
Robust step/s	7.2	136.0

- pandas lookups replaced with array/JAX-style loops.
- $8.5\times$ and $19\times$ throughput gains made wide sweeps practical.

Defended policies recover COI as contamination rises



x: contamination α (fraction of agent traffic) | y: COI [EUR per transaction]

$-90{,}140$
baseline COI slope (EUR per unit α)

$\sim 3\%$
short-run revenue cost of defense

regime-dependent
COI gains strongest at higher α

Conclusion

Without defense, COI collapses with α. Robust policies hold a measurable margin floor at the cost of a small, bounded revenue trade.

Yes, with boundaries: margin integrity is defensible under agentic orchestration

SQ1 distinguishability	SQ2 theoretical impact	SQ3 mitigation
kernels are separable $p < 0.001$	COI erosion mechanism proved in baseline limit	robust control shifts COI / revenue / UX trade-off

Boundary conditions

Evidence is from a controlled platform and a small labeled cohort. This is mechanism validation, not full production external validity.

What this implies for real pricing systems

- **Financially:** untreated reconnaissance behaves like an information leak and can compress sustainable margins.
- **Operationally:** behavior-only session scoring can be wired into pricing without device fingerprinting.
- **Market exposure:** channels where dynamic pricing is a secondary layer (aggregators, comparison funnels, promo traffic) are disrupted first.
- **Strategically:** robust pricing should be calibrated by regime; there is no single penalty that wins everywhere.
- **Before deployment:** larger human baselines, governance review, and legal safeguards are mandatory.

Thank you

Questions and discussion

Appendix follows: COI theorem derivation, reward composition, and sample-size notes.

Appendix roadmap

A. Objects

Notation, COI, proxies

B. Mechanism

Order stats, kernels, KL

C. Control

Simulator, robust loop, factorial grid

Figures

Full charts, MDPs, extra revenue view

Appendix: core notation (quick reference, I)

$$\tau_s = (e_{s,1}, \ldots, e_{s,L_s}) \qquad \text{session}$$

$$\hat{q}_{t,i} = \sum_{s \in S_t} \sum_{k} \omega(a_{s,k})\, \mathbf{1}[i_{s,k} = i] \qquad \text{proxy } (\mathbf{H}, \mathbf{A})$$

$$Q(p) = (1-\alpha)\, \mathbb{E}_{\theta \sim D_H}[d(p;\theta)] + \alpha\, \mathbb{E}_{\theta \sim D_A}[d(p;\theta)] + \epsilon_t \qquad \text{mixture of } \mathbf{H}/\mathbf{A}$$

$$\mathrm{COI}(\pi) = \mathbb{E}[P] - \underline{p} \qquad \text{COI}$$

Appendix: core notation (quick reference, II)

- $\underline{p}$: minimum viable price anchor (thesis simplification).
- α: contamination with agent traffic in the mixture.
- $\omega(a)$: hand-engineered action weights for the proxy (baseline).

Reading guide

Objects on the left are **observable**; $d(\cdot)$ and many θ remain hidden.

Appendix: COI as a reporting functional

$$\mathrm{COI}(\pi) = \mathbb{E}_{P \sim F_\pi}[P] - \underline{p}$$

Interpretation

Premium above the floor induced by policy π; used as a KPI and as the object Theorem 1 attacks under query saturation.

Appendix: demand proxy vs. latent demand

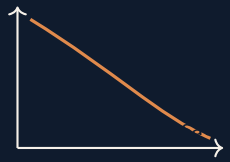
$$\hat{q}_{t,i} = \sum_{s \in S_t} \sum_{k=1}^{L_s} \omega(a_{s,k})\, \mathbf{1}[i_{s,k} = i]$$

Key distinction

$\hat{q}$ is an operational sensor from logs (**H**, **A**); true demand $d(p;\theta)$ stays latent. Pricing reacts to $\hat{q}$, so agent-shaped behavior can poison the signal.

Appendix: independent draws and order statistics (intuition)

- Independent price draws $\{P_i\}_{i=1}^{N}$ from fixed offer law.
- Purchase-side minimum behaves like $P_{(1)}$: mass shifts left as N grows.
- Expected premium vs. $\underline{p}$ compresses: COI pressure.



Appendix: Theorem 1 scope (what is and is not claimed)

Inside the baseline proof

Non-collusive sessions, independent draws, fixed offer distribution across queries.

Outside (handled elsewhere)

Collusion, pooled recon, sequential repricing that breaks iid structure: evidence moves to the simulator.

Appendix: empirical transition kernel (MLE)

$$\hat{P}(s' \mid s) = \frac{N(s, s')}{\sum_k N(s, k)}$$

Use

Human and agent centroids $\bar{T}_H, \bar{T}_A$ for divergence-to-prototype scores.

Appendix: KL to prototypes (shared support)

$$\Delta_H = D_{\mathrm{KL}}(\hat{T}' \,\|\, \bar{T}_H), \qquad \Delta_A = D_{\mathrm{KL}}(\hat{T}' \,\|\, \bar{T}_A)$$

Asymmetric choice

KL measures deviation from the **human** reference; symmetric JS/Wasserstein on behavior was not the design target.

Appendix: softmax to sigmoid (algebra)

Let $z_A = -\Delta_A/T$, $z_H = -\Delta_H/T$. Then

$$P(A \mid \tau) = \frac{e^{z_A}}{e^{z_A} + e^{z_H}} = \frac{1}{1 + e^{z_H - z_A}} = \sigma\big(z_A - z_H\big)$$
$$= \sigma\left(\frac{\Delta_H - \Delta_A}{T}\right).$$

Takeaway

Two-class softmax over (z_A, z_H) is exactly one sigmoid on the gap $(\Delta_H - \Delta_A)$.

Appendix: contamination generator $\mathcal{G}(\alpha)$

$\mathcal{G}(\alpha)$: inject synthetic agent trajectories until mixture reaches target α

Role in the lab

Supplies controlled stress tests for the pricing learner; not a claim of production-faithful agents.

Appendix: Wasserstein ambiguity (ideal object)

$$\mathcal{U}_\epsilon(\hat{P}_N) = \left\{ Q : \; W_p(Q, \hat{P}_N) \leq \epsilon \right\}$$

What the code implements instead

A **local** grid over α near α_0 with radius ϵ_α: tractable inner worst case, not a full ball solver.

Appendix: per-step reward sketch

$$r = R(p, d) - \lambda \, \mathrm{COI}_{\mathrm{leak}}(p, \tau') - \eta \, \mathrm{UX}(\tau', p) - \text{(supra-competitive excess)}$$

- Query-tax style $\mathrm{COI}_{\mathrm{leak}}$: minimal nonzero surrogate to expose the control channel.
- UX and anchor penalties prevent trivial solutions (flat but exploitative prices).

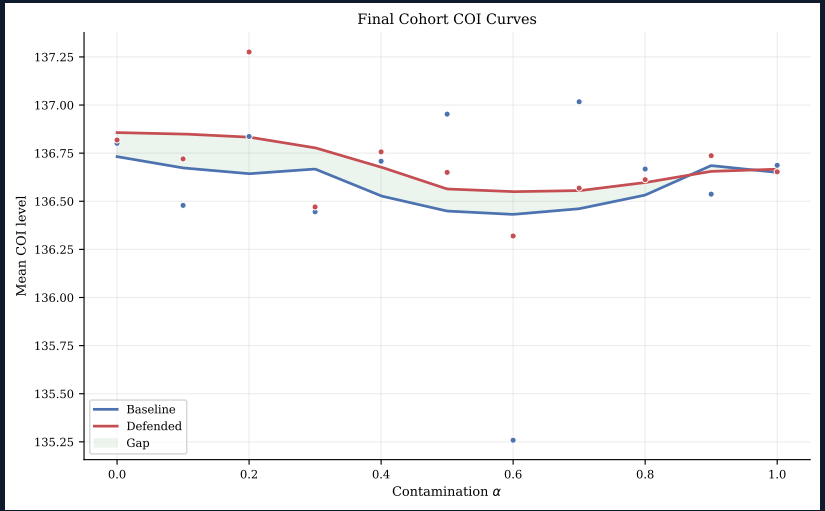
Appendix: factorial design (192 cells)

Axis	Levels	Count
RL algorithm	PPO, A2C, DQN, Q-table	4
Contamination α	4 representative values in $[0.1, 0.6]$	4
Robustness radius ϵ_α	3	3
COI penalty λ_{coi}	2	2
Action granularity	2	2
Total		$4 \times 4 \times 3 \times 2 \times 2 = \mathbf{192}$

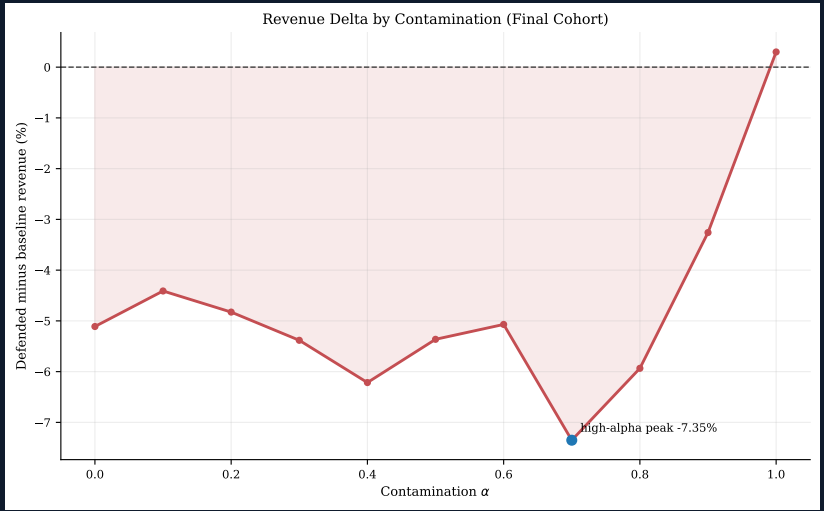
Appendix: engineering note (pandas $\rightarrow$ JAX)

- Hot path was label-indexed transition lookups; profiling showed pandas overhead dominated.
- Integer-indexed arrays + JAX inner loop: large step/s throughput (thesis numbers; environment dependent).
- Kronecker expansion of product-conditioned kernels: research simulator cost, scales with catalog.

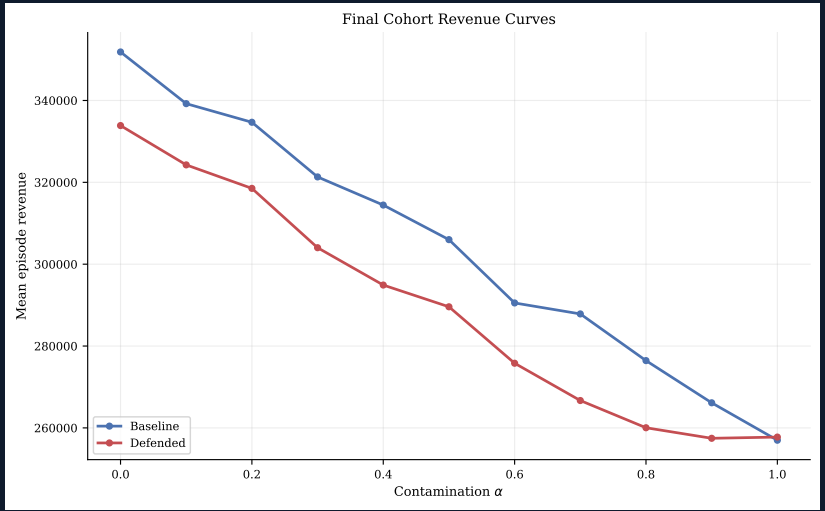
Appendix figure: COI by α (full)



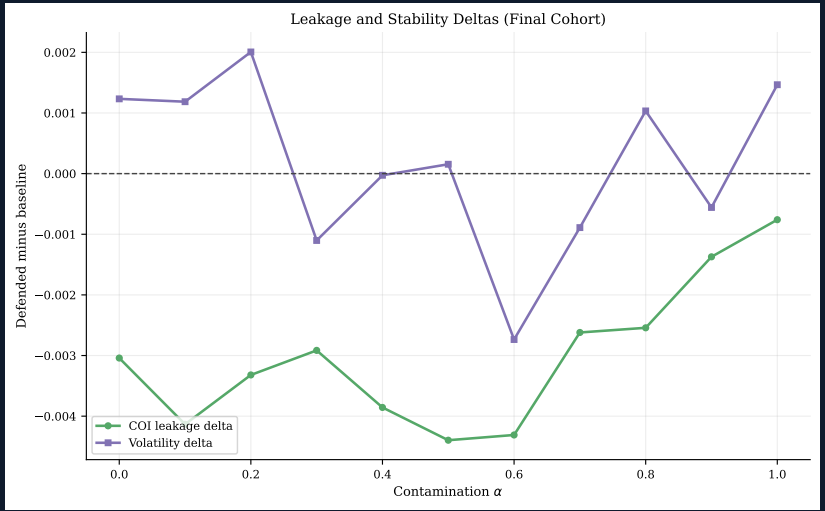
Appendix figure: revenue deltas (full)



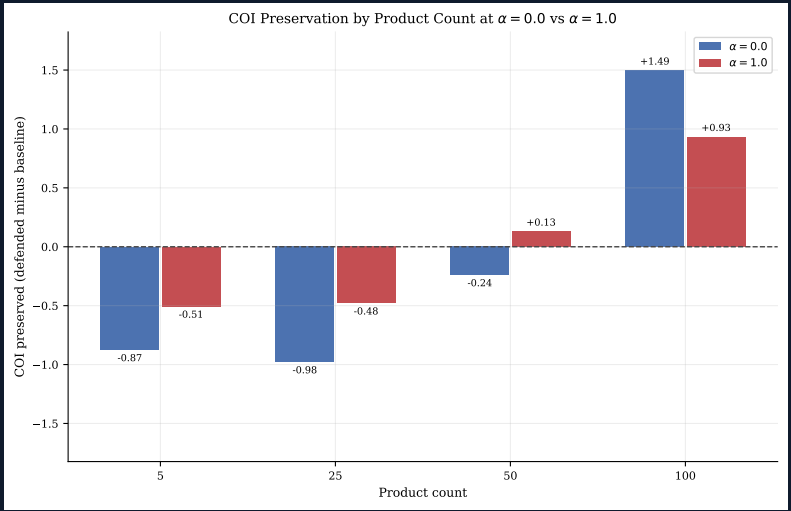
Appendix figure: revenue by α (full)



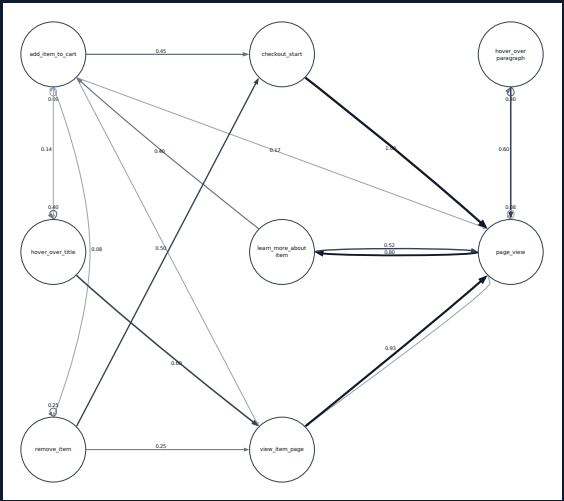
Appendix figure: risk / stability deltas (full)



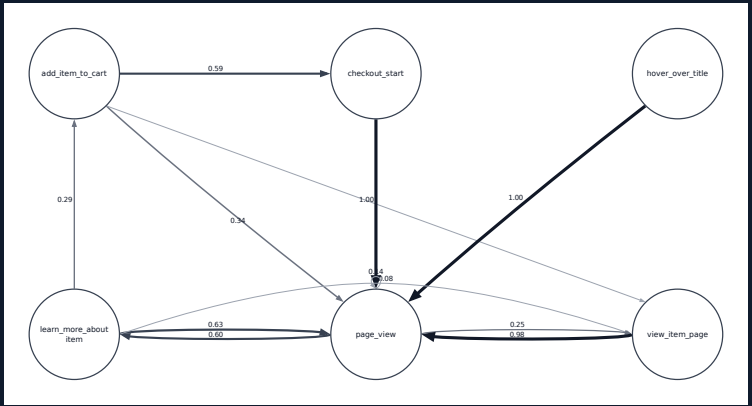
Appendix figure: COI preservation grid (full)



Appendix figure: human MDP (full)



Appendix figure: agent MDP (full)



Appendix: threat model map

Claim boundary

Residual contamination after security controls is the motivating scenario.

Appendix: evaluation checklist (robustness culture)

1. Session-aware labels: avoid splitting rows inside a trajectory if that inflates scores.
2. Document how prototypes $\bar{T}_H, \bar{T}_A$ were fit (full cohort vs. held-out); state explicitly in writing.
3. Report temperature T as calibration, not as a tuned hyperparameter unless a sweep is shown.
4. Separate **architecture** claims from **coverage** claims (hotel vs. airline balance at release).

Appendix: sim-to-real gap (explicit)

- Kernels and generators reflect a **small labeled cohort** and a **browser-use style** agent class.
- RL policies are trained in a **surrogate** market with engineered rewards and discretized prices.
- Deployment would require legal review, fairness testing, and refreshed baselines at scale.

Appendix: leakage surrogate (query-tax form)

$$\mathrm{COI}_{\mathrm{leak}}(p, \tau') \approx f(\tau') \cdot c_{\mathrm{info}}$$

Reading

$f(\tau')$ is the weak agent score; c_{info} is a minimal constant leakage proxy to expose the control channel. Revelation-style $-\log \pi(p \mid \tau')$ is the natural upgrade.

Appendix: robust pricing template (symbolic)

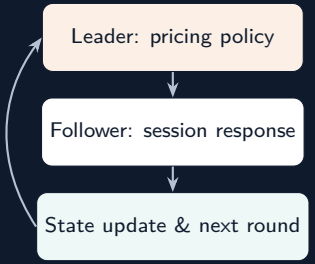
$$\max_{\pi} \min_{Q \in \mathcal{U}_\epsilon(\hat{P}_N)} \mathbb{E}_{d \sim Q}\left[R(p,d) - \lambda\,\mathrm{COI}_{\mathrm{leak}} - \eta\,\mathrm{UX}\right]$$

Code-level substitute

Inner min over a **finite grid** of $\alpha_k \in [\alpha_0 \pm \epsilon_\alpha]$ around the nominal generator mix, not a continuous adversary over all Q in the ball.

Appendix: why a Stackelberg game is a useful abstraction

- **Leader move**: the platform commits a quote via policy $p_t = \pi(x_t)$.
- **Follower move**: session behavior then reacts (click, continue, abandon, purchase).
- This ordering matches real serving APIs: price is emitted before response is observed.
- Repeating this local sequence gives a tractable leader-follower control model.



Boundary

We do **not** claim a full market equilibrium. We claim a useful timing model for explainable policy updates under contamination.

Appendix: why Theorem 1 helps (without over-claiming)

What the theorem gives us

- A directional mechanism: independent recon pressure compresses COI.
- A sanity check for reward design: leakage penalties should grow with recon likelihood.
- A clean explanatory anchor for stakeholders and governance review.

What the theorem does not claim

- It is not a finite-sample forecast for every market.
- It does not cover collusion or all adaptive adversaries.
- It does not replace simulator evidence or offline policy validation.

Three evidence layers used in this thesis

Theorem 1 (mechanism direction) $\rightarrow$ **simulator** (finite-regime quantification) $\rightarrow$ **implementation** (local robust policy training).

Appendix: composite strip (five plots, small multiples)

Same PDFs as the main talk, shrunk to scan the full panel at once.