KDD 2026 WORKSHOP ON EVALUATION & TRUSTWORTHINESS OF AGENTIC AI (KDD-ETAAI '26) · AUG 9, 2026 · ICC JEJU, KOREA

Reason Less, Verify More

Deterministic Gates Recover a Silent Policy-Violation Failure Mode in Tool-Using LLM Agents

Vikas Reddy[1] Sumanth Reddy Challaram[2] Abhishek Basu[3]

1 Independent Researcher 2 Indian Institute of Technology Kharagpur 3 Massachusetts Institute of Technology

TL;DR Policy-permissive tools execute forbidden writes **silently**: the transcript looks clean while the database is wrong. Four deterministic, read-only pre-execution gates lift τ²-bench airline pass¹ from **29.6% → 42.0%** on gpt-4o-mini (**+12.4pp**, replicated on disjoint seeds) – and the lift concentrates exactly where the gates fire.

1 The failure mode: silent wrong states

In τ²-bench airline, the policy lives in a natural-language prompt, but the tools are **policy-permissive**: any well-formed call executes even when the state transition is forbidden. The agent cancels a non-refundable booking, the tool obeys, no error is raised – and the agent reports success.

78% of observed failures are **silent wrong states** – wrong final database, zero tool errors, clean transcript.

✓ WHAT THE OPERATOR SEES
"Task completed." No exception raised, no error returned. The transcript reads as a success.

✗ WHAT THE DATABASE SAYS
Non-refundable reservation: **CANCELLED**. Policy violated – with no signal anywhere in the trace.

WHY RESAMPLING CAN'T FIX IT

Inconsistent, not rare: pass¹ 29.6% collapses to pass⁵ 8.0% – most tasks succeed only sometimes, so one run tells you little.

Silent: no error marks a violating run, so there is no signal to retry against – and in deployment the agent runs once.

→ **A gate changes the system property:** the known forbidden transition is blocked **every time** it is proposed.

WHEN CAN THIS CLASS EXIST? FIVE ADMISSION AXES

- **A1** Structured, machine-readable tool calls
- **A2** Policy-permissive tools (forbidden writes execute silently)
- **A3** State-decidable policy rules
- **A4** Final-state evaluation (catches silent wrong states)
- **A5** Tasks that actually induce violations

2 Deterministic pre-execution gates

A gate is a pure predicate over the proposed tool call and current database state – **no LLM calls, no writes, no ground-truth access**. It reads only state the agent itself can read.

```
g(tool, args, db_state) → {allow, reject}
```

Agent proposes a mutating tool call

▼

GATE – deterministic predicate, read-only ← reads → DB state

▼ REJECT No write. Structured reason returned; the agent re-plans.

▼ ALLOW Original call executes unchanged – gates add zero model calls.

First rejecting gate wins. Fail-open: if a gate itself errors, the original call proceeds – gate bugs never add new blocks.

THE FOUR-GATE SUITE (AIRLINE)

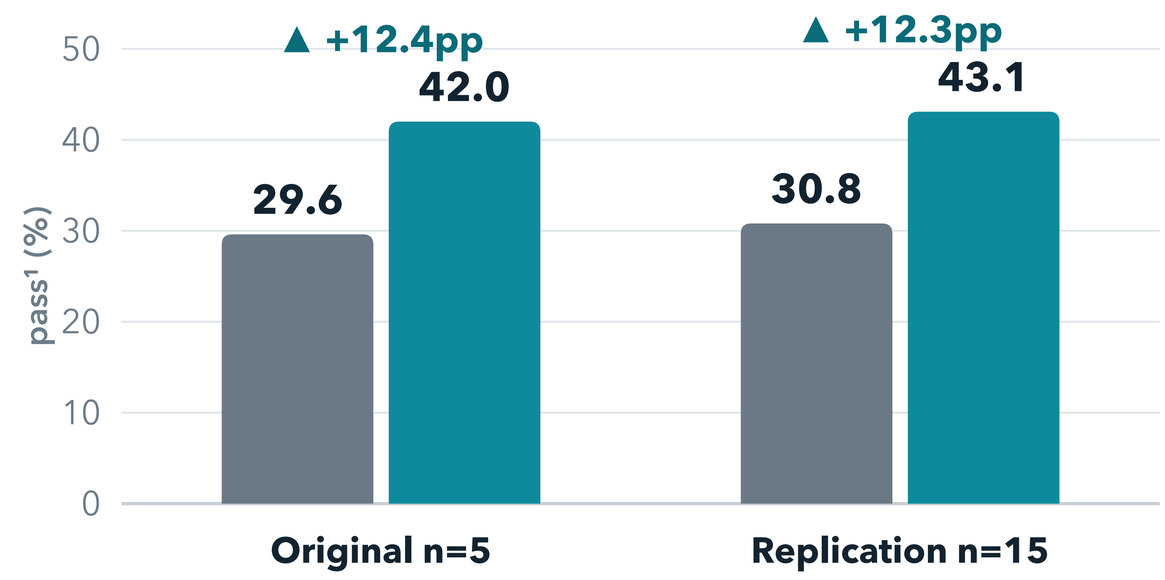
GATE	BLOCKS
`cancellation_eligibility`	Cancellations ineligible under fare, timing, insurance, or flown-segment rules
`baggage_allowance`	Baggage updates that misprice free bags or remove bags against policy
`passenger_count`	Any change to passenger count – immutable by policy
`must_read_before_write`	Writes to records the agent has not read this session

Selected on the budget tier, then held fixed across model tiers.

3 Headline result – replicated

+12.4pp

pass¹ 29.6% → 42.0% · gpt-4o-mini · full 50-task τ²-bench airline · paired task-level bootstrap **P = 0.0012**



Replicates on **15 disjoint seeds** (750 trials/condition): 30.8% → 43.1%, **+12.3pp, P = 0.0008** – a stable property of the intervention, not seed luck.

THE LIFT CONCENTRATES WHERE GATES FIRE

GATE FIRES · 26 TASKS
13.8% → 33.1% **+19.2pp**
P = 0.0006 95% CI [+6.9, +33.1]

GATE NEVER FIRES · 24 TASKS
46.7% → 51.7% **+5.0pp**
CI includes 0

+25 of the +31 net recovered trials sit in the firing stratum – the mechanism, not a scaffold artifact.

4 Reliability & the gate audit

pass^k = all k trials succeed. Vanilla collapses to **8.0%** at k=5; the gated agent holds **26.0% (3.3×)** – a recurring failure mode is removed, not occasional lucky wins.

NOT EVERY GATE IS GOOD – AUDIT THEM

GATE	FIRES	PRECISION	REMOVED → Δ
`cancellation_eligibility`	161	100%	−2
`must_read_before_write`	90	78%	+3
`baggage_allowance`	42	5%	+3
`basic_economy`	18	83%	+6
`passenger_count`	9	100%	+4

Removal Δ = pass¹ change when the gate is dropped (negative ⇒ load-bearing). One gate carries the lift at 100% precision over 161 fires; another fires 42× at 5%. **Gate precision must itself be audited.**

THE DECEPTIVE TASK (#48)
The user asserts false context to induce an out-of-policy cancellation. Gated: **16/16** across budget & gpt-4.1 tiers. Ungated: **≤1/16**. The gate reads database state, not the user's story.

Firing ≠ recovery: on task #39 the gate fires repeatedly and the task stays 0/5 – re-planning after rejection remains model-dependent.

Where gates don't help – negative controls

Retail (self-enforcing tools): forbidden calls already fail loudly; a gate duplicates the tool's own checks. 62.5% → 57.8% (−4.7pp, CI includes 0).

BFCL v4 multi-turn: zero gate firings across 200 entries (52.5% vs 51.0%) – errors there are loud, not silent.

Subset inflation: the same run reads +30pp on an enriched 8-task subset. Report the full-50 number.

Frontier arm – suggestive, not the anchor

gpt-5.2 (default reasoning) still attempts policy-violating writes: gates fire on 43/250 trials across 18 tasks.

Same suite: **61.2% → 71.6% (+10.4pp, P = 0.020)** – but n = 5, unreplicated, lower CI bound near zero.

Same shape: firing tasks +33.3pp, non-firing −2.5pp (point estimates). Scale reduces violations; it does not change the system property.

Takeaways

1. **Enforcement can raise task success** – blocking a forbidden write prevents silent, unrecoverable state corruption.
2. Evaluations should **separate loud tool errors from silent wrong states**; pass/fail alone hides the dangerous class.
3. A gate is a **different guarantee** than a stronger model: the forbidden transition is blocked on every run.
4. Gates aren't free: **firing ≠ recovery** and precision varies – audit suites per policy and per model.

Limitations (paper §7): single positive domain · gates authored from the policy and evaluated on the same task set (replication over seeds, not tasks) · rejection-message feedback confound.

`github.com/cvikasreddy/reason-less-verify-more-code` · KDD-ETAAI '26 (non-archival)