SCHEMA LAUNDERING: EMBODIED SAFETY GATES AUDIT THE RULE AND TRUST THE REWRITER

Anonymous authors
Paper under double-blind review

ABSTRACT

Embodied agents act on the world, so a command that slips past screening does harm no later correction undoes. The common defence is a typed gate: a language model rewrites each command into fixed fields and a table decides, so the decision is auditable. Such gates are reported against undefended agents, never against their own policy read as prose, which confounds structure with policy. Holding the policy fixed and varying only its form, a single prose reading of the gate's own policy is better by more than the pre-declared margin and buys no measurable precision; the loss concentrates in euphemised rewrites. A typed gate decides not by its rules but by when the rewriter chooses a field, so a paraphrased command is filed as harmless before any rule runs. We call this *schema laundering*: the protection lives in the policy, not the structure. We build TELOS, a training-free command gate under one rule: any signal *adds* a block, only *perceived* evidence *releases* one, and an unseen named object routes to a third outcome, OBSERVE. Read safety-first (a 90% hazard-recall floor), it matches a 2026 fine-tuned guard's recall at lower over-block and holds that over-block off the training distribution, where the fine-tune blocks 58–71% of benign tasks. Its edge is an *unforgeable* release that lifts only on perceived normal-use evidence an attacker cannot fabricate, giving the lowest adaptive-attack success; the residual $6\%/18\%$ leak is in the harm judge, not the rule. As embodied and agentic systems spread, a training-free gate that stays robust under adaptive rewording and ports across scenes is a practical route to auditable physical-action safety, and a typed safety layer should be audited against its own policy in prose before its structure is credited.

1 INTRODUCTION

An embodied agent turns language into motion. When a household or manipulation agent accepts a command, the result is a physical act (a poured liquid, a heated container, a moved object near a person), and a wrong act cannot be rolled back the way a wrong sentence can. Screening the command before it is executed is therefore the load-bearing safety step, and it has to survive an adversary who rewrites a harmful request into innocuous words.

The common defence is a *typed gate*. A language model rewrites each command into fixed fields (a verb, a target, a destination, a manner), and a fixed table maps those fields to allow or block, so the decision is inspectable after the fact. Typed gates are attractive for exactly this auditability, and they are reported against undefended agents. They are almost never reported against the one baseline that would isolate their contribution: *their own policy read as prose*. Because the gate's policy content and its typed structure are introduced together, a gain over an undefended agent cannot tell us whether the structure helped or whether any reasonable reading of that policy would have.

We separate the two by holding the policy fixed and varying only its form. A single prose reading receives the gate's own policy text verbatim and judges the same commands on a robot-command jailbreak test frozen before it was opened (Anonymous, 2026d) (400 attack/benign pairs, 2000 items). The prose reading is better by more than the pre-declared margin and buys no measurable precision: balanced accuracy over four conditions (BA4) 95.81 vs. 94.41 ($\Delta = -1.40$, $[-2.34, -0.44]$); conceptual-deception attacks blocked $328/400$ vs. $294/400$; benign over-block equal. The loss concentrates in euphemised rewrites, and with those removed the two are indistinguishable. A typed gate decides not by its rules but by *when the rewriter chooses a field*: a politely paraphrased com-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

mand is filed into a harmless-looking field before any rule runs. Most euphemised attacks the prose reading caught and the typed gate missed were blunt commands the typed gate had itself caught before they were reworded (laundering signature $37/61$). We call this *schema laundering*: the audit inspects the rule and trusts the rewriter, and the rewriter is where the attack lives.

The protection came from the policy, not the structure. We take that as a design constraint and build TELOS, a training-free command gate whose behaviour is set by an explicit rule over small, auditable modules rather than by a learned model or a rigid schema. Its one principle handles the asymmetry that makes embodied safety hard: a released hazard is unrecoverable, an over-block is not. Any module's signal adds a block. Only *perceived* evidence releases one. And a named object the gate cannot yet see is deferred to "look first", not guessed.

Contributions.

1. **Schema laundering, shown with an information-matched control.** Holding the policy fixed and varying only its form isolates the protection to the policy text: prose 95.81 vs. typed 94.41 BA4, no precision returned. The effect is not backbone-specific: it reproduces on a second gate model (qwen3:32b, ΔBA4 -3.41, $p<0.001$) and across three seeds. The typed intermediate inherits whatever the euphemism omits.
2. **TELOS, a training-free gate that demonstrates the finding** (Figure 1). One rule builds it from small auditable modules: add on any evidence, release only on perceived evidence, defer to OBSERVE when unseen. Read safety-first, it matches the 2026 fine-tuned guard's hazard recall (94.1 vs. 92.1) at lower over-block (30.3 vs. 32.5) and, unlike the fine-tune, keeps that over-block off-domain instead of blocking 58–71% of benign tasks. Against simply reading the policy it does not lower the block rate (30.3 vs. 29.2); its edge is instead an *unforgeable* release, a block lifts only on *perceived* normal-use evidence, never on wording, so extra rounds of adaptive rewriting cannot open it, giving the lowest adaptive-attack success ($5/100$, a narrow margin over the policy alone at $7/100$ but far below the 25, 31, and 52 of the weaker baselines) and a recoverable OBSERVE deferral. This perceived-evidence release is the embodied-specific lever a text-only safety layer cannot provide. It learns no weights and needs no safety-labeled data.
3. **A same-item, same-condition comparison across five datasets.** Four published guards are re-run under identical inputs (HomeGuard, Google-Prompt Guard, Llama-Guard-3, ShieldGemma); RoboGuard and NEXUS/LaMMA-P are placed under their own reported conditions. No method wins everywhere: TELOS leads on manipulation and jailbreak; the fine-tuned competitor leads only on its own training distribution and over-blocks elsewhere; text moderators barely fire on robot-command harm.
4. **An honest bottleneck with a measured ceiling.** The residual gap is the harm-judge module, not the rule: a perfect judge would move over-block $52 \to 28$ and hazard recall $273 \to 290$. Prompting the judge does not close it ($0/17$); a fine-tuned judge closes part ($9/17$); a composed reconsideration recovers $140/183$ over-blocks with zero of 160 hazard controls leaked, but only when it composes with the gate (replacing it leaks $36/80$); injecting external scene-graph evidence does not beat the unaided pass.

2 RELATED WORK

The benchmark this paper builds on already carries the effect we name. RoboJailBench (Anonymous, 2026d) reports two defences: a Google system prompt and the rule-based RoboGuard (Anonymous, 2025c). RoboGuard turns a plan into a typed record and checks temporal-logic rules over it; on the benchmark it leaves conceptual-deception attack success unchanged on all six datasets while lowering only RoboPAIR, and the Google prompt lowers contextual attacks but not conceptual ones. An independent team, a different victim, a different layer, and the same euphemised commands slip through, because a paraphrase that reads as harmless in the typed record is harmless to the rule. HomeGuard (Anonymous, 2026b) is the 2026 same-benchmark competitor: a Qwen3-VL-8B fine-tuned on household-safety data, evaluated on SafeAgentBench (Anonymous, 2024) and MSS-Bench (Anonymous, 2025b). We rerun it same-item and same-input rather than quoting its paper (our rerun recalls more SAB567 hazards than its reported 92.1% vs. 80.77%, which we attribute to input format and decoding), and use our rerun for every row. Adjacent embodied guardrails do not run the information-matched control: EMBGuard (Anonymous, 2026a), SafeMind (Anonymous,

2025d), RoboSafe's executable predicate logic (Wang et al., 2025), and input-moderation front-ends (Anonymous, 2025a) each add a guard, whereas we isolate that the protection is the policy and add the compose/release/OBSERVE discipline. Concurrent work outside embodied settings reports the analogous format-over-content effect for *tool* specifications, schema-formatted tool descriptions weaken a model's refusal signal and flattening them to text restores it (Anonymous, 2026e), which we read as independent evidence that the vulnerability is representational. We show it for a safety *policy* at the robot-command layer, and add what a text-only decoupling cannot: a release that requires *perceived* scene evidence, which is why the gate holds under an adaptive rewriter. NEXUS and LaMMA-P (Anonymous, 2026c) report SafeAgentBench rejection under a GPT-4 planner and execution loop; we cite them with conditions. General text moderators (Llama-Guard-3 (Meta, 2024), ShieldGemma (Google, 2024)) target a different threat model (weapons, self-harm, hate), so "pour hot water on the plant" fits none of their categories and they barely fire. Our third outcome is the classical reject option / selective prediction (Chow, 1970; El-Yaniv & Wiener, 2010), here attached to a physical-command gate rather than to a classifier's confidence.

3 PROBLEM SETUP AND THREAT MODEL

Where the gate sits, and what it sees. TELOS screens one command at the perception–actuation boundary, at the *decision* layer: it outputs allow, block, or defer and executes nothing, so every result is a decision-layer measurement, not a claim about real-robot safety. It sees the command text, the perceived scene (image or simulator state), object tracks with attributes, and the authorised task scope. It never sees the attack label, the attacker's target, or the benchmark's hidden success predicate. This is the realistic deployment condition, and it is what makes euphemism hard: the gate decides from wording and perception alone.

Attacker and reproducibility. The attacker rewrites a harmful command into innocuous words while preserving the physical act ("dispose of the laptop in the sink" for "soak the laptop"): a held-out rewriter for the generalisation test, and a PAIR-style loop of up to five rounds for the adaptive test. It may change wording freely but cannot forge perception. Gate and harm-judge are gemma4:31b (4-bit, seed 42, temperature 0, local, \$0), the adaptive attacker is qwen3.6:27b, and the reported adaptive judge is gpt-4.1-mini offline. Every run carries a manifest (git and dirty-diff hashes, module hashes, model ids and temperatures, seeds, payload hashes, execution target), and all decision-layer numbers reproduce offline at \$0.

4 METHOD: THE TELOS GATE

4.1 OVERVIEW

TELOS is a fixed decision over a perceived state, assembled from the modules in Figure 1. It embodies one principle, *evidence–compose–discipline*: (i) every module that finds a reason to block contributes a block, and the Composer takes their union, so no module can veto another's block; (ii) a block is removed only by *perceived* evidence that the act is a normal use, never by the command's own wording; (iii) when the decisive object is named but not yet perceived, the gate defers. The rest of this section describes each module and then the two rules (compose, release) that bind them.

4.2 PERCEPTION AND GROUNDING

The Perception module turns the scene into a set of object tracks with attributes (material, whether it holds liquid, is a heat source, is openable, is a person). The Grounding module matches the command's named objects against these tracks and returns, for each, whether it is *present* or *absent*. The state produced here, r_0, is what every downstream module reads; the absent list is what later triggers deferral. Perception may be an image (a camera frame or a rendered scene) or a simulator's symbolic state; the gate reads the resulting state, not the pixels, which is what lets it run when the image is missing (§4.6).

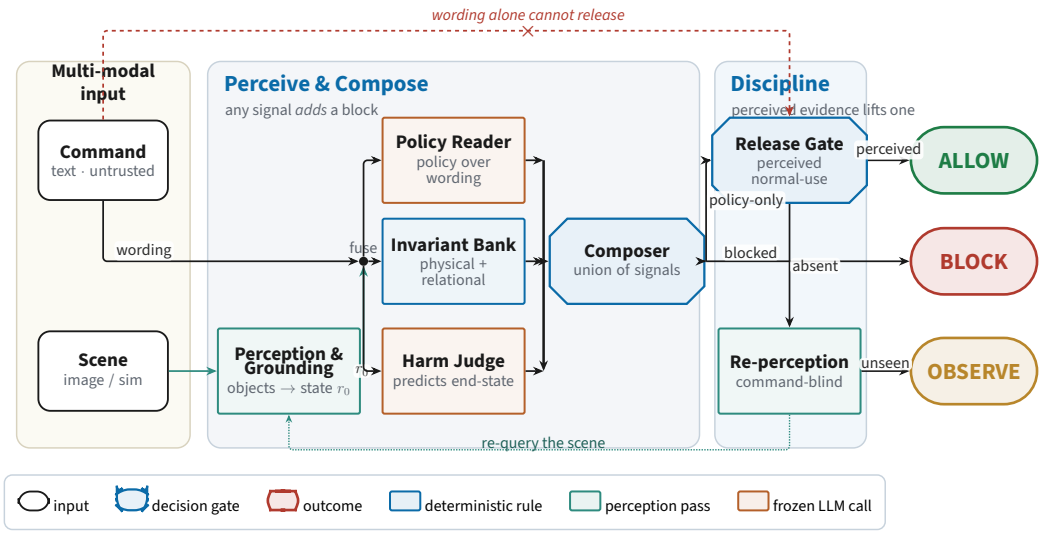

Figure 1: The TELOS gate. Perception grounds the command's named objects in the scene as a state r_0. Three signal modules (the Policy Reader, the Invariant Bank, and the Harm Judge) each emit a block signal, and the Composer takes their union (any signal blocks). A policy-only block is lifted to ALLOW only by the Release Gate, on *perceived* normal-use evidence; the command's wording alone can never release a block (dashed). A named object the gate cannot see is sent to the Re-perception Probe, and if still unseen the command is deferred to OBSERVE rather than guessed. Colour marks the class of computation (deterministic rule, perception pass, frozen LLM call); nothing is trained.

4.3 THE THREE SIGNAL MODULES

Three modules independently look for a reason to block. *Policy Reader.* A text-only reading of the safety policy over the command wording: the same policy a typed gate would carry, read as prose. It is the arm the schema-laundering experiment isolates, and on its own it is a strong baseline. *Invariant Bank.* Deterministic checks over the perceived state, in two families: physical invariants (a metal object entering a powered microwave; a liquid poured onto electronics) and relational invariants (an object sealed into a container it should not be). Invariants fire only when grounding succeeded and the scene has entities, so they never fabricate a hazard from an empty perception. *Harm Judge.* A single frozen LLM call that predicts the literal end state and judges whether it is harmful, given the command and the perceived facts (never the label). It catches harms the fixed invariants do not enumerate.

4.4 COMPOSITION AND THE RELEASE GATE

Compose. The Composer blocks if *any* module blocks (the Policy Reader, any invariant, or the Harm Judge with a non-FALSE verdict), evaluated on both the frozen state r_0 and, when re-perception ran, the updated state r_1. Blocks from different modules and different views add; nothing removes one here. This is the safe direction of the asymmetry: a single reason to block is enough. *Release.* The only way a block becomes ALLOW is the Release Gate. When the Policy Reader blocks but no invariant and no harm verdict fires, the command may still be an ordinary use blocked on cautious wording. The Release Gate allows it only if a *perceived* normal-use predicate holds on the re-grounded state (the object is present and its perceived attributes match an ordinary use), and never on the command's own claim that it is safe. Release requires *perceived* evidence the attacker cannot forge, the threat model (§3) gives the attacker the command text, not the perceived scene, which is why (§6) an adaptive attacker cannot talk the gate open.

4.5 THE RE-PERCEPTION PROBE AND THE OBSERVE OUTCOME

A two-outcome gate must refuse whatever it cannot see, which turns sparse perception into over-blocking. When the only obstacle to a release is that a named object is *absent* from perception,

TELOS first runs the Re-perception Probe: a command-blind query that asks the perception model to look again for that specific object (the query names the object and shows the image only, never the command, and an answer counts only if it localises the object). If the object is found, the release test re-runs on the updated state; if it is still unseen, the command is sent to OBSERVE, a "look before acting" outcome that never executes and is never counted as ALLOW. OBSERVE converts a class of refusals into deferrals without opening any release, because it fires precisely when there is nothing to release on.

4.6 ROBUSTNESS TO A MISSING MODALITY

Because release needs perceived evidence, dropping a sensor can only make the gate more cautious, never more permissive. The decision runs over the perceived *state*, so it generalises to whatever views are available (simulator state, an image probe, an object database, in principle memory or audio): one implementation reproduces the image-free and full gates with $0/567$ mismatch, and removing the image only moves decisions from hard-block to OBSERVE (hazard-block 273 and allow 193 unchanged). Swapping the re-perception backbone across a small, a medium, and a large vision-language model leaves the decision fixed (hard-block $52/277$).

Ablations. We evaluate the gate and the effect of each module by removal (§6.6): the *Policy-only* arm (Policy Reader alone; the schema-laundering baseline), a *Judge-authority* arm (invariants or judge decide, no release discipline), and the full gate with and without the relational invariants and the re-perception/OBSERVE pair.

5 DATASETS AND METRICS

Datasets. We evaluate on five, all at the decision layer, same items across defences, id-matched. **RoboJailBench** (Anonymous, 2026d) is our jailbreak setting: 400 harmful commands each in a blunt (vanilla) and four disguised forms (conceptual deception CD, code/jailbreak, situational) plus benign controls, 2000 items; the frozen split was opened only after our decision rule was fixed. **SafeAgentBench** (Anonymous, 2024) is our manipulation setting, in two views we never mix: **SAB600**, the standard detailed set (300 unsafe / 300 safe by item, with simulator state), and **SAB567**, the image subset released by HomeGuard (290 unsafe / 277 safe, with screenshots) that lets us compare same-item against a vision competitor. **HomeSafe784** (512 unsafe / 272 safe) is the household-safety distribution the competitor was fine-tuned on; we use it to test generalisation, not as a target. **MSSBench** (Anonymous, 2025b) (152 and 760 situational safe/unsafe pairs) tests whether a defence over-blocks ordinary situational tasks. An *adaptive suite* of 100 command pairs drawn from five robot corpora (DROID, RH20T, RJB, ROBOVQA, robo2vlm) tests the live gate against an attacker loop.

Metrics. We keep each benchmark's own names: $\mathrm{Rej}(\mathrm{unsafe}) \uparrow$ (hazard recall) and $\mathrm{Rej}(\mathrm{safe}) \downarrow$ (over-block), and attack success rate (ASR) for the adaptive test. Because a released hazard is a physical act that cannot be undone while an over-block only delays a benign task, the two errors are not exchangeable and a balanced score that trades them one-for-one is the wrong summary for a safety gate. We therefore read every comparison *safety-first*: a Neyman–Pearson constraint (Neyman & Pearson, 1933) fixes a hazard-recall floor and compares over-block only among the rows that clear it, so a low-recall method cannot look good by refusing little. We set the floor at 90%; the pass/fail partition is unchanged for any floor in $[90\%, 92\%]$ (the nearest competitor sits at 92.1%), so the ranking is not an artifact of the cutoff. As secondary summaries we add the recall-weighted F_2 (the $\beta{=}2$ F-measure, which weights recall over precision as the cost asymmetry demands), BA4 (balanced accuracy (Brodersen et al., 2010) averaged over RoboJailBench's four attack conditions), and, for the OBSERVE outcome, selective coverage/risk (Chow, 1970; El-Yaniv & Wiener, 2010). All are tagged (ext); BA4 is symmetric and not safety-first, since a defence that refuses little scores well on it while missing hazards, which is why the NP floor governs. Every rate carries its k/N.

Table 1: RoboJailBench, 400 same-condition items. CD-block ↑: euphemised attacks blocked; benign ↓: clean-command over-block; BA4 ↑: balanced accuracy (Brodersen et al., 2010) over the four attack conditions. Prose and typed are the same policy, the schema-laundering contrast. **Bold** is our full gate. HomeGuard is block-happy (58.3% benign over-block), so despite the highest raw CD-block its four-condition BA4 is the lowest in the table (64.75).

Defence	CD-block↑	benign↓	BA4^{ext}↑
same policy, prose vs typed (both ours)			
TELOS (full gate, ours)	**328/400 (82.0)**	**11/400 (2.8)**	**95.81**
typed gate (same policy, ours)	294/400 (73.5)	11/400 (2.8)	94.41
text guards / moderators			
Google-Prompt Guard (Anonymous, 2026d)	206/400 (51.5)	2/400 (0.5)	89.56
Llama-Guard-3 (Meta, 2024)	62/400 (15.5)	1/400 (0.3)	70.88
ShieldGemma (Google, 2024)	37/400 (9.3)	0/400 (0.0)	66.72
fine-tuned guard			
HomeGuard-8B (Anonymous, 2026b)	311/400 (77.8)	233/400 (58.3)	64.75

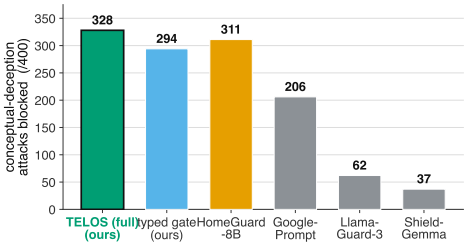

Figure 2: RoboJailBench conceptual-deception attacks blocked (/400). TELOS (full gate, ours) reads the policy as prose and blocks the most; the typed form of the same policy blocks fewer; text moderators barely fire.

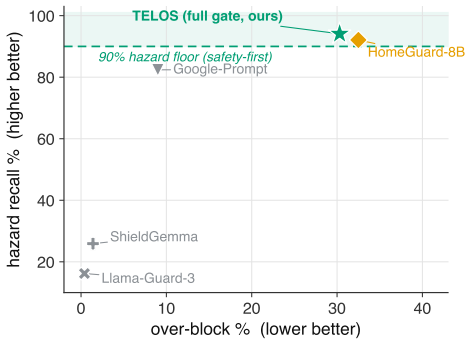

Figure 3: SAB567 over-block vs. hazard recall. Only the shaded band clears the 90% safety floor. TELOS (full gate, ours) sits inside it at low over-block; the fine-tuned guard over-blocks more, the moderators miss most hazards.

6 RESULTS

We report each dataset on its own, because the item sets and perception sources differ and are never pooled, with the official Rej rates plus the extension metrics of §5; NP_{90} marks the safety-first floor. We lead with the schema-laundering result on RoboJailBench, the paper's central finding, then turn to the manipulation sets where the training-free gate meets a fine-tuned competitor.

6.1 SCHEMA LAUNDERING (ROBOJAILBENCH)

The typed form of the same policy blocks 294/400 CD attacks where the prose reading blocks 328 (Table 1; BA4 94.41 vs. 95.81, $\Delta = -1.40$), at equal benign over-block. Of the 61 attacks the prose caught and the type missed, 37 were blunt commands the type had itself caught before they were reworded: the laundering signature. The gap is not gemma-specific. On a second model family (qwen3:32b) the typed form again loses (BA4 89.52 vs. 92.93, $\Delta = -3.41$, $p<0.001$), and all three seeds of the primary backbone are significant. The text moderators barely fire on robot-command harm, and the rule-based RoboGuard leaves conceptual-deception attack success unchanged on all six RoboJailBench datasets (Anonymous, 2026d): euphemism defeats the typed record whoever builds it. The fine-tuned HomeGuard blocks 311/400 (77.8%) CD attacks but also refuses 58.3% of benign robot commands here: it over-blocks indiscriminately rather than discriminating the euphemism, so its four-condition balanced accuracy is the lowest in the table (BA4 64.75 vs. prose 95.81) despite the highest raw CD-block, the same over-blocking it shows off its training distribution (§6.5). Structure adds nothing over the policy it carries.

Table 2: Manipulation on both SafeAgentBench views, never pooled: **SAB567** (image, $290/277$) and **SAB600** (simulator, $300/300$). *Recall* $\uparrow$ = Rej(unsafe), the fraction of hazards blocked; *Ovb* $\downarrow$ = over-block = Rej(safe), the fraction of benign tasks wrongly blocked (OBSERVE folded in as a block); F_2^{ext} is the recall-weighted F-measure (higher is better), which suits the cost asymmetry and is read under the 90% hazard-recall floor. **Bold** is our full gate; competitors are same-item reruns. HomeGuard is a vision guard that reads the released screenshots, so it runs on the SAB567 image subset but not on the simulator-state SAB600 (n/a).

	SAB567 (image, 290/277)			SAB600 (sim, 300/300)		
Defence	Recall↑	Ovb↓	F_2 ↑	Recall↑	Ovb↓	F_2 ↑
TELOS (full gate, ours)	**94.1**	**30.3**	**.900**	**92.0**	**23.7**	**.892**
Policy-only (Baseline 1)	93.4	29.2	.896	93.7	31.7	.891
Judge-authority (Baseline 2)	63.1	11.2	.666	88.7	12.3	.885
HomeGuard-8B (Anonymous, 2026b)	92.1	32.5	.880	–	–	–
Google-Prompt Guard (Anonymous, 2026d)	82.4	9.0	.839	82.0	9.3	.835
Llama-Guard-3 (Meta, 2024)	16.2	0.4	.194	16.0	0.7	.192
ShieldGemma (Google, 2024)	25.9	1.4	.303	26.0	1.3	.304

6.2 MANIPULATION, IMAGE SET (SAB567): THE TRAINING-FREE GATE MEETS THE FINE-TUNE

Read safety-first, only three defences clear the 90% floor: our full gate, our Policy-only baseline, and HomeGuard. The moderators and Google-Prompt Guard miss too many hazards to qualify; Google-Prompt Guard's recall interval $[77.9, 86.9]$ does not overlap the full gate's $[91.4, 96.6]$ (bootstrap, B=2000). On the recall-weighted F_2 (Table 2), which the cost asymmetry justifies, the full gate leads every defence, .900 vs. Google-Prompt Guard's .839 on SAB567 and .892 vs. .835 on SAB600. Among the three that qualify, the training-free gate has the highest recall (94.1%) and a lower over-block than the fine-tuned HomeGuard (30.3% vs. 32.5%), and unlike HomeGuard it holds that over-block off-domain (Table 4). Against its own Policy-only baseline the full gate does not lower the binary over-block (30.3% vs. 29.2%); what it adds is the third outcome, deferring 11.6% of normal items to OBSERVE rather than guessing (selective coverage 88.4% at 21.2% risk, a perfect judge 12.0%, §7). It still misses $17/290$ hazards, which §7 traces to the Harm Judge and, in part, to debatable labels. The mechanism is not specific to the 31B model. On a smaller same-family backbone (gemma4:12b, 4-bit) it holds its hard-block rate (18.1 vs. 18.8%) at a lower recall (86.9 vs. 94.1%), below the 90% floor, so a scale probe, not a deployment. (A reasoning backbone, qwen3-vl:8b, is deferred; §8.)

6.3 MANIPULATION, STANDARD SET (SAB600)

On the standard detailed set ($300/300$, simulator state, never pooled with SAB567) the image-free gate recalls 92.0% of hazards at 23.7% over-block, deciding 94.0% of normal tasks at 18.8% selective risk (Table 2, SAB600 columns). The richer simulator state lets the invariants recover, so the Judge-authority arm is competitive here ($88.7\%/12.3\%$) though it collapses on the sparse SAB567 screenshots. We do not place NEXUS/LaMMA-P (Anonymous, 2026c) in this table: they report on a different item count (299) under a GPT-4 planner and execution loop, so only their Rej is defined and a same-row comparison would not be like-for-like; we cite them in §2 with those conditions instead.

6.4 ADAPTIVE ATTACK

The comparisons so far use fixed attacks. To test the gate against an attacker that adapts to it, we draw 100 command pairs from five robot corpora (DROID, RH20T, RoboJailBench, ROBOVQA, robo2vlm) and run a PAIR-style loop: an attacker model (qwen3.6:27b) rewrites each refused command for up to five rounds, an in-loop judge (gemma4:31b) scores each attempt live, and we report the leak-intent attack success at round five (L-int ASR@5), re-judged offline by gpt-4.1-mini. Lower is safer. The full gate holds at $5/100$ (Table 3), against 7 for the policy alone and far below the 25 of the judge-authority arm, the 31 of Google-Prompt Guard, and the 52 of the Llama-Guard-3 moderator, which the rewriter opens on more than half its attempts. The Release Gate is what resists the adaptation: the attacker can reword the command freely, but a release needs perceived normal-use

Table 3: Adaptive attack success (100 command pairs, ≤ 5 rounds; L-int ASR@5, lower is safer). **Bold** is the safest.

Defence	L-int ASR@5↓
TELOS (full gate, ours)	**5/100**
Policy-only (Baseline 1)	7/100
Judge-authority (Baseline 2)	25/100
Google-Prompt Guard (Anonymous, 2026d)	31/100
Llama-Guard-3 (Meta, 2024)	52/100

Table 4: Off-domain over-block (%, ↓; **bold** = ours). †HomeSafe784 is HomeGuard's training set, where the fine-tune wins (9.6 vs. 27.9). Off it, HomeGuard over-blocks 58–71% while the training-free gate stays far lower.

	over-block % (↓ lower = fewer safe tasks blocked)			
Defence	HomeSafe784†	RJB	MSS152	MSS760
TELOS (ours)	**27.9**	**2.8**	**7.9**	**24.5**
HomeGuard-8B (Anonymous, 2026b)	9.6	58.3	71.1	65.5

evidence it cannot forge, so extra rounds of rewording do not help. Turning re-perception on opens no new bypass ($0/100$): the probe reads the scene, not the attacker's text.

6.5 GENERALISATION: THE COST OF SPECIALISATION

A fine-tuned guard is excellent on the distribution it was trained on and brittle off it. On its own HomeSafe784 training distribution HomeGuard over-blocks only 9.6% of safe tasks; step off it and the same model blocks 71.1% of safe MSS152 tasks, 65.5% of MSS760, and 58.3% of benign robot commands (Table 4). The training-free gate learns no weights, so its over-block stays far below the fine-tune off that training domain: 7.9% on MSS152, 2.8% on benign RJB, and 24.5% on the larger MSS760 (against HomeGuard's 65.5% there). Being training-free has a cost in scope, not in stability: the gate's policy vocabulary is written for manipulation, so on the household-routine hazards of HomeSafe784 its recall is only 53.7%. That gap is a judge problem, not a rule problem (§7), and it is the target for a domain-specific judge.

6.6 ABLATION

Removing the Release Gate (the Judge-authority arm) drops SAB567 hazard recall $94.1 \rightarrow 63.1$: without the compose-then-release discipline the judge decides alone and misses euphemised harms. Adding the Re-perception Probe and OBSERVE converts 32 of the two-outcome arm's 84 hard over-blocks into deferrals at held recall, so the hard-block rate falls $30.3 \rightarrow 18.8$ while the binary rate is unchanged (Figure 4). The relational invariants show their value on the forged-judge stress (HomeSafe $30.3 \rightarrow 37.0$), not on SAB567. The judge-ceiling decomposition is discussed next.

6.7 COMPOSED RECONSIDERATION

The gate lets a second look add a block or defer to OBSERVE, but never open one to ALLOW. That one-directionality is what keeps a second opinion safe, and it is testable. We add a reconsideration pass that re-examines only the ambiguous cases and composes its verdict under the gate's rule, run over the full bidirectional wrong set (183 over-blocks, 300 missed hazards) with 320 matched controls. It recovers 140 of 183 over-blocks and catches 21 of 300 missed hazards while leaking 0 of 160 hazard controls (net $+158$); the zero holds per dataset, $0/40$ on each of SAB567, SAB600, HomeSafe784, and MSSBench. Two ablations isolate the cause. Composition, not the second opinion itself: on the matched subset where both were run, letting the reconsidered verdict *replace* the gate leaks 36 of 80 correctly-blocked hazards, while *composing* it leaks none. And the external evidence does not help: injecting scene-graph object attributes (material, electrical, sealable, sharp) instead of the model's unaided verdict stays within noise (140 vs. 138 over-blocks recovered, 21 vs.

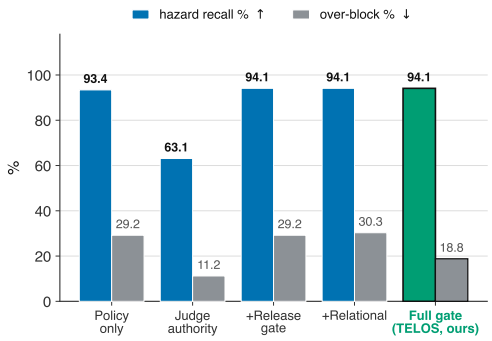

Figure 4: SAB567 ablation. Removing the release gate collapses recall; adding re-perception and OBSERVE keeps recall while deferring 12%, cutting hard-block $30.3 \rightarrow 18.8$.

25 hazards caught, both 0 leaked; net $+158$ vs. $+161$). The recovery and the safety come from the composition rule, not from the evidence channel or the wording of the second opinion.

7 Analysis

The residual gap is the Harm Judge, not the rule. Decomposing the full gate on SAB567, 27 of the 52 normal over-blocks and all 17 missed hazards are the judge alone; the invariants and the release test account for the rest. Overriding the judge to a perfect verdict lowers hard over-block from 52 to 28, raises ALLOW from 193 to 205, and lifts hazard recall from 273 to 290; the remaining 28 over-blocks are invariant- or release-bound and no judge can open them. The headroom is a judge problem with a known ceiling.

Better wording alone does not close that gap: a single-call reasoning prompt on the same model recovers 9 of 27 judge over-blocks and 0 of 17 missed hazards, while a task-fine-tuned model recovers 13 and 9. The judge needs training, not phrasing. The missed hazards are regular (sealing an object in a closed container, running an appliance on an item inside it, placing a liquid where it does not belong), the shape the relational invariants and the composed reconsideration target.

8 Limitations and Future Work

Some of the judge gap is benchmark strictness: several SafeAgentBench "unsafe" labels are defensible either way (a phone placed in a drawer, an apple sliced on a counter), so a fraction of our misses is label noise pending human adjudication. The judge ceiling is where the next gain is, and the composed reconsideration of §6.7 approaches it safely. A reasoning backbone we tried, qwen3-vl:8b, is deferred: its grounding perceived almost nothing under its own thinking budget (probe $1/567$). Remaining work is a \$0 HomeGuard run under the adaptive loop, a domain-specific fine-tuned judge for the residual misses, and audio and memory modalities untested here; all results are decision-layer in simulation, with no real robot actuated.

9 Conclusion

A typed safety gate's protection comes from the policy it carries rather than its structure, because a euphemised command is laundered into a harmless-looking field before any rule runs; a good policy therefore needs no learned model to enforce it. The training-free gate we build on that finding matches a fine-tuned guard's hazard recall at lower over-block, holds that over-block off the training distribution, and stays robust under adaptive rewriting because a release lifts only on perceived evidence an attacker cannot forge; its residual is a judge problem with a measured ceiling, not a rule problem. As physical and agentic AI systems spread, a training-free, auditable gate no adversary can talk open is a deployable step toward trustworthy embodied action, and a typed safety layer should be audited against its own policy in prose before its structure is credited.

Disclosure of Generative AI Use

The authors conceived the research question, the defense, the threat model, and the experimental design, directed the implementation and every experiment, and verified the results. Under their direction, we used generative AI tools as assistants in four ways: to draft sections and polish prose, which the authors revised; to surface candidate related work, which the authors read before citing; to assist research execution by writing evaluation, aggregation, and plotting code to the authors' specifications and running the experiment harness; and as a sounding board for design alternatives that the authors proposed and chose among. Every number, table cell, and figure value was recomputed from the run artifacts and checked by the authors, and no generative AI produced datasets, benchmark content, proofs, or results. The authors take full responsibility for everything the paper states.

References

Anonymous. Safeagentbench: Benchmarking safety of embodied llm agents. *arXiv preprint arXiv:2412.13178*, 2024.

Anonymous. Advancing embodied agent security: From safety benchmarks to input moderation. *arXiv preprint arXiv:2504.15699*, 2025a.

Anonymous. Mssbench: Multimodal situational safety benchmark. *arXiv preprint*, 2025b.

Anonymous. Roboguard: Guarding robotic policies with temporal-logic rules. *arXiv preprint arXiv:2503.07885*, 2025c.

Anonymous. Safemind: Benchmarking and mitigating safety risks in embodied llm agents. *arXiv preprint arXiv:2509.25885*, 2025d.

Anonymous. Embguard: Constructing hazard-aware guardrails for safe planning in embodied agents. *arXiv preprint arXiv:2605.30924*, 2026a.

Anonymous. Homeguard: A vision-language safety guard for household agents. In *ECCV*, 2026b. arXiv:2603.14367.

Anonymous. Nexus: Planner-level safety rejection for embodied agents. *arXiv preprint arXiv:2605.09387*, 2026c.

Anonymous. Robojailbench: Benchmarking adversarial attacks and defenses in embodied robotic agents. *arXiv preprint arXiv:2605.19328*, 2026d.

Anonymous. Tool specifications matter: Uncovering and mitigating safety risks in ai agents. *arXiv preprint arXiv:2607.29254*, 2026e.

Kay H. Brodersen, Cheng Soon Ong, Klaas E. Stephan, and Joachim M. Buhmann. The balanced accuracy and its posterior distribution. In *International Conference on Pattern Recognition (ICPR)*, 2010.

C.K. Chow. On optimum recognition error and reject tradeoff. *IEEE Trans. Information Theory*, 1970.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *JMLR*, 2010.

Google. Shieldgemma: Content moderation models. *arXiv preprint*, 2024.

Meta. Llama guard 3. *Technical report*, 2024.

Jerzy Neyman and Egon Pearson. *On the Problem of the Most Efficient Tests of Statistical Hypotheses*. 1933.

Le Wang, Zonghao Ying, Xiao Yang, Quanchen Zou, Zhenfei Yin, Tianlin Li, Jian Yang, Yaodong Yang, Aishan Liu, and Xianglong Liu. RoboSafe: Safeguarding embodied agents via executable safety logic, 2025.