PG-DSL: Physics-Grounded Admission Checking for MCP-Controlled CPS Tools

Huan Bui
University of North Carolina at Charlotte
hbui11@charlotte.edu

Chenglong Fu
University of North Carolina at Charlotte
chenglong.fu@charlotte.edu

***Abstract*—LLM agents that use the Model Context Protocol rely on natural-language tool descriptions during discovery, but cannot check whether a description matches the tool implementation. For cyber-physical-system tools a misleading description can induce unsafe physical actions. We present PG-DSL (Physics-Grounded Description Lifting), an admission-time checker that lifts each tool description into a restricted CPS-grounded formal claim, replays the implementation in a digital twin, and admits the tool only when the claimed effect matches the measured physical trajectory—a narrower object than the full LLM agent.**

We instantiate PG-DSL on a SWaT P1+P2 digital-twin substrate with 14 MCP tools. Under bounded twin fidelity and a restricted grammar we formalize a detection threshold, a lifting-soundness decomposition, and a witnessed separation between admission-time and runtime visibility. PG-DSL rejects 0 of 42 benign tool-state checks, 1 of 227 benign paraphrases, and 0 of 70 held-out descriptions written by a model family other than the deployed lifter. On an eight-attack canonical set it detects six at admission; the remaining two—a sub-noise-floor drift and a post-admission sensor spoof—are invisible to the gate, and a runtime CPS intrusion detector calibrated on benign traffic spanning our replay grid misses them as well.

We also measure where the approach stops working: 134 mutants show that admission constrains claimed identity rather than behavior, a probe-grid sweep shows that a fixed set of replay states is itself an attack surface, and an 18-variant twin perturbation bounds how far the twin may deviate before the guarantee lapses. An identity probe closes read-tool aliasing, previously our sharpest limit. Predictions were fixed before execution and the falsified ones reported; the artifact and its pre-registrations are public.

1. Introduction

Measurement work on Model Context Protocol servers [1] finds that $\approx$ 13% of real-world MCP servers exhibit description–code mismatches sufficient to enable unauthorized actions when an LLM agent reads the description verbatim. Concurrent Dragos threat intelligence [2] attributes a December 2025–February 2026 municipal water-utility intrusion to an LLM acting as the primary technical executor of operational-technology reconnaissance (the OT breach attempt itself failed). These reports motivate the need to audit the MCP description channel. We study this threat class as *description-channel poisoning* for MCP-controlled CPS tools.

Why existing defenses are insufficient for this object. Two results constrain the design space. The trust-authorization mismatch SoK [3] invokes Rice's theorem to argue that semantic verification of a Turing-complete agent is undecidable; static access control cannot govern probabilistic agent behavior. "Gaming the Judge" [4] shows judge-model verifiers can have their false-positive rates inflated by up to 90% under chain-of-thought adversaries. Together, these results motivate a narrower verification target and a non-LLM reference signal.

The PG-DSL approach. A defensible check exists if we target a narrower object than the LLM agent. PG-DSL treats each MCP tool implementation as a bounded, replayable artifact and asks whether its measured physical effect in a digital twin (DT) matches the natural-language claim exposed to the agent. This changes the reference signal from another LLM's judgment to CPS-grounded trajectory evidence. Each tool description is lifted to a formal claim φ over a CPS-grounded grammar of actuators, sensors, and state-variable deltas; the claim is matched against a measured trajectory ψ from the DT before the agent sees the tool. PG-DSL filters individual tools before catalog exposure and precomputes unsafe ordered tool pairs for runtime blocking. The resulting guarantee is intentionally conditional: it holds only for descriptions expressible in the restricted grammar, implementations exercised by the replay harness, and dynamics captured within the DT's fidelity bounds. PG-DSL therefore does not claim to detect all malicious MCP tools; it targets mismatches whose claimed physical effects are expressible in the grammar and measurable in DT replay.

Contributions.

- **Threat formulation.** We formulate description–implementation mismatch as an admission-time attack surface for MCP-controlled CPS tools.
- **System.** We design PG-DSL, which lifts descriptions into restricted CPS claims, replays implementations in a DT, and admits only tools whose claimed and measured effects match.
- **Conditional analysis.** We analyze PG-DSL under bounded DT fidelity, replay coverage, and grammar

coverage, including a detection threshold, a lifting-soundness decomposition, and a witnessed, one-directional admission/runtime visibility separation.
- **Evaluation and artifact.** We evaluate PG-DSL on a 14-tool SWaT P1+P2 MCP surface with benign paraphrases, a canonical 8-attack set, lifter stress tests, an MSB-adapted subset, and a live-agent campaign, and release an artifact with claim-level provenance.
- **Boundary characterization.** With predictions fixed before execution and the falsified ones reported, 134 mutants place every miss (§5.10), a fixed probe grid is an attack surface (§5.10), and an 18-twin sweep bounds the certified envelope (§5.11).

Roadmap. §2 fixes the protocol model, SWaT P1+P2 substrate, and formal preliminaries; §3 the threat model; §4 the theorems and pipeline; §5 the results; §6 limitations; §7 positions against prior work; §8 concludes.

2. Background

We fix the protocol model, the substrate, the formal preliminaries the theorems build on, and the runtime-CPS-IDS partner of T3. Readers familiar with MCP and SWaT may skim.

2.1. The MCP Tool-Admission Surface

An MCP server exposes a finite tool set; each tool is a triple $\mathcal{T} = (name, description, impl)$. The agent reads $description$ at *discovery time* (it becomes part of the prompt context) and invokes $impl$ by name. Per the specification [5], a session starts with `tools/list` (returning the catalog) then issues `tools/call` requests routed through $impl$. PG-DSL inserts an *admission gate* between the server and the agent's catalog view: for each tool i it consumes $(description_i, impl_i)$, replays $impl_i$ in a digital twin, lifts $description_i$ to a formal claim φ_i, and either admits the tool (visible to the agent) or withholds it (catalog omits it).

2.2. What Description-Channel Poisoning Looks Like

A poisoned tool is a triple $(name, description', impl')$ where the description encodes claims the agent will act on, while $impl'$'s observable physical effect diverges from those claims. Divergence sources: actuator/sensor type confusion or sensor aliasing in $description'$; parametric magnitude poisoning; or composition—joint behavior of multiple individually-admitted tools.

2.3. The SWaT P1+P2 Substrate

The substrate is a digital twin of SWaT (iTrust SUTD) restricted to Process Stages P1 (raw-water intake) and P2 (chemical dosing): 2 tanks (raw-water buffer T101, dosing buffer T201); 2 motorized valves (MV101 supply–T101, MV201 T101–P2); 2 raw-water pumps (P101, P102); 6 chemical-dosing pumps (P201–P206); and 7 analyzer sensors (LIT101/201 levels, FIT101/201 flows, AIT201–203 chemistry). The MCP server exposes 14 canonical tools (full list, Appendix C). Dynamics: mass balance plus linearized chemistry [6]; tank levels follow $\dot{L}_T = q_{\text{in}} - q_{\text{out}}$ conditional on valve/pump state, with a linearized-Cond chemistry model in the operating band.

2.4. Formal Preliminaries

Sensing and lifting errors. The lifter $L_{\text{verify}} : \mathcal{C} \to \Phi$ maps descriptions in a structured grammar $\mathcal{C}$ to formal claims φ over actuators, sensors, and state-variable deltas. We assume bounded semantic error on $\mathcal{C}$: $\sup_{D \in \mathcal{C}} \|L_{\text{verify}}(D) - \varphi^*(D)\| \le \varepsilon_L$, with $\varphi^*(D)$ the ground-truth claim and $\|\cdot\|$ the per-sample bias the lifter induces in sensor space. The DT has bounded per-sample fidelity error ε_{DT}: for any real plant trajectory ψ^* and matching DT trajectory ψ, $\sup_t \|\psi(t) - \psi^*(t)\| \le \varepsilon_{DT}$.

Vectorized state. The admission-time physical state space is $\Sigma \subseteq \mathbb{R}^n$ with $n = 2$ on SWaT P1+P2: (L_{T101}, L_{T201}). Per-coordinate sensing and lifting errors are $\varepsilon_{L,v}, \varepsilon_{DT,v} \in \mathbb{R}^n$.

2.5. The Runtime CPS-IDS Partner: INVARLLM

T3 pairs PG-DSL with INVARLLM [7]: a classical CPS-IDS that extracts physical invariants (mass-balance, sensor-redundancy, monotonicity) from benign trajectories and fires on runtime violations. INVARLLM is admission-time blind by construction—it sees runtime telemetry, not descriptions. It is T3's runtime partner (§4, §5).

3. Threat Model

System. An LLM agent A controls SWaT P1+P2 via MCP [5] tool calls—each tool a natural-language description, an input schema, and an implementation wrapping a PLC-style command. The defender inserts a physics-grounded verifier V between the MCP server and A's catalog, so A sees only admitted tools.

Attacker. The attacker controls one or more MCP tools' descriptions, implementations, or both; may pre-set physical actuator state before the agent's session and spoof sensor reads at the network layer post-admission (W_2); and may know the published probe states (§5.10). The attacker cannot modify V, the digital twin, the lifter L_{verify}, or the formal matcher, and cannot bypass the admission gate.

Attack classes. Three categories align with the empirical campaigns. **Description-poisoning** (A_1–A_3): A_1 type-confusion (description claims drain, implementation fills); A_2 magnitude poisoning on a parametric tool (declared unit/scale not honored); A_3 return-value aliasing (description names one returned signal, implementation returns another). **Trajectory** (Z_1–Z_3): violations visible only in the replayed trajectory—Z_1 single-tool transient overshoot

TABLE 1. TCB. PG-DSL PROVIDES NO DEFENSE IF A TRUSTED COMPONENT IS ITSELF COMPROMISED.

Component	Trusted?	Rationale
MCP description	No	attacker-controlled
Tool impl.	No	untrusted; replay observes its effect (§5.12)
Digital twin	Yes	reference trajectory, fidelity-bounded
Lifter	Yes, fallible	bounded ε_L; T2 quantifies
Formal matcher	Yes	deterministic check
Grammar/vocabulary coverage	Yes	limits what can be checked
Runtime telemetry	Partial	W_2 spoofs post-admission reads
Runtime IDS	Separate	T3 composition partner

in the lifted band and Z_2 single-tool overdose magnitude poisoning (A_2 mirror), both caught per-tool; Z_3 alone is the strict-composition witness, hazardous only as an ordered pair (§4.6). **Admission-invisible** (W_1, W_2): invisible to the gate by construction, and (§4.6) to a telemetry IDS as well. W_1 is a sub-noise-floor stealth drain within per-sample fidelity; W_2 is post-admission spoofing of LIT101 while pumps overflow the tank.

Trusted computing base. Table 1 summarizes the TCB. PG-DSL's guarantees are limited by DT fidelity, grammar coverage, and lifter error on the restricted class $\mathcal{C}$.

Out of scope. Network-level attacks against V, hardware aging, chemistry kinetics beyond linearized conductivity, and side-channel attacks on L_{verify} are deferred to T3's runtime partner or to future work.

Threat-model feasibility. The evaluation includes a live MCP agent loop that demonstrates the A_1 mismatch and the admission-invisible W_2 case under sampled operating conditions; see §5.14.

4. System Design

Before the tool catalog is exposed to the agent, PG-DSL runs five steps: (i) a fixed claim grammar, (ii) the *lifter* L_{verify}, (iii) the *per-tool DT verifier*, (iv) the *composition-aware DT verifier* (producing a blocked-pair set the runtime guard enforces), (v) the *formal matcher*. Tools surviving both per-tool and composition checks reach the agent. Conditional properties pair with their components: T2 with the lifter (§4.2), T1$_v$ with the per-tool DT verifier (§4.3), T3 with the composed defense (§4.6).

4.1. Step 1: A CPS-Grounded Grammar for Tool Descriptions

A description $D \in \mathcal{C}$ is a finite sequence of statements generated by

$$\begin{aligned}\text{statement} ::= {} & \text{actuator}(\textit{name}) := \textit{value} \\ \mid {} & \text{sensor}(\textit{name}) \text{ reads } \textit{value} \\ \mid {} & \Delta\text{state}(\textit{var})\ \{\textit{property}^{+}\}.\end{aligned}$$

name ranges over the SWaT P1+P2 vocabulary (MV101, MV201, P101–P206, LIT101/201, FIT101/201, AIT201–AIT203); *value* over discrete actuator states, numeric sensor readings, or the qualitative literals low/nominal/high; *property* over monotone $[\pm]$, bounded $[a, b]$, sign $[\pm 0]$.

The three statement types are not chosen freely. They are the closure of what the P1+P2 PLC surface can command (discrete actuator writes), what its sensors report (scalar reads), and what a bounded replay can decide about a trajectory (sign, monotonicity, interval containment). A property is admitted only if the matcher can decide it from a finite replay, which is why the grammar carries no temporal operators. The grammar is fixed for all experiments here; the full Lark grammar appears in Appendix B.

4.2. Step 2: Lift Each Description to a Formal Claim (L_{verify} and T2)

L_{verify} is a locked LLM (`qwen3:14b` via Ollama at $T = 0$, seed $= 42$, `think` $=$ False). The prompt ("v1"/restrained, Appendix E) lifts *only* explicit content under five restraint rules R1–R5: explicit-content-only, no-inferred-Δstate, per-clause-lifting, single-actuator-per-tool, and parametric-tool-minimal. R3–R5 were added in the revision pass to close failure modes (co-actuator inference, multi-pump enumeration) surfaced by the extended out-of-family paraphrase corpus; the artifact ships both v1 variants for ablation. We define

$$\varepsilon_L := \sup_{D \in \mathcal{C}} d(L_{\text{verify}}(D), \varphi^*(D)),$$

with $\varphi^*(D)$ the ground-truth claim and $d(\cdot, \cdot)$ the per-sample bias the lifter induces in sensor space.

Theorem 1 (T2, Lifting Soundness on Class $\mathcal{C}$)**.** *For $D \sim \mathcal{D}$ supported on $\mathcal{C}$, L_{verify} produces a sound claim with probability at least $1-\delta_{lift}$, where $\delta_{lift} \le \delta_{cov}(\mathcal{C}, L_{verify}) + \delta_{gram}(\mathcal{C})$; δ_{cov} is the LLM error rate on $\mathcal{C}$ and δ_{gram} the rate at which honest descriptions admit no in-grammar lifting.*

T2 decomposes lifter error rather than asserting a task-universal numeric bound. Empirically, on `qwen3:14b` with prompt v1 against a 227-description corpus (42 hand-authored seeds plus 185 out-of-family `mistral:7b` paraphrases), we observe $1/227 = 0.44\%$ benign paraphrase rejection (Wilson 95% CI $[0.08\%, 2.45\%]$): a CASUAL paraphrase adds "keep MV101 open", the lifter encodes that advice as an actuator claim, and the matcher rejects because `start_pump_P101` never opens MV101. That is a δ_{cov} instance, not an out-of-grammar one ($\delta_{\text{gram}} = 0/227$ here; $1/8$ on P3, §5.13). We now measure T2's coverage term directly rather than borrowing an external anchor. Taking δ_{cov} as the wrong-lift rate on the 42-seed corpus (rejections together with R2 violations), it ranges from $0/42$ for `qwen3:14b` to $25/42$ for `phi4-mini` across six lifter families (§5.9). An earlier draft cited Req2LTL's 11.6% accuracy gap here as an illustrative anchor; that comparison is superseded by the measured per-family rate. We separately stress adversarial lifting in §5, where the

lifter achieves $28/31$ correct on intentionally adversarial in-grammar prompts.

4.3. Step 3: Replay Each Tool in a Digital Twin (and T1$_v$)

For each candidate tool, the verifier instantiates a fresh `SwatP1P2Plant` DT [6] at one of three initial states $\{\text{LOW}, \text{MID}, \text{HIGH}\}$, executes the implementation, and records trajectory ψ over a 120 s window. The matcher checks $\varphi \models \psi$ per state-variable; any mismatch on any initial state rejects the tool.

Vectorized discrepancy. On $\Sigma \subseteq \mathbb{R}^n$ ($n = 2$ on SWaT P1+P2: (L_{T101}, L_{T201})), the lifter supplies a per-coordinate admissible region $A_\varphi(k)$ (or $\top$); the composition-aware verifier supplies an effective safe band $A_{\text{eff}}(k)$. The per-coordinate discrepancy of the actual trajectory ψ_v^* is

$$\delta_v[k] := \sup_{t \in [0, T_{\text{horizon}}]} d_k\big(\psi_v^*(t)[k],\ A_{\text{eff}}(k)\big),$$

with $d_k(x, [a, b]) = \max(0, x - b, a - x)$ and $d_k(x, \top) = 0$. $\|\delta_v\|_\infty = \max_k \delta_v[k]$.

Theorem 2 (T1$_v$, Vectorized Detection Threshold)**.** *With bounded per-coordinate semantic error $\varepsilon_{L,v}$ and DT fidelity error $\varepsilon_{DT,v}$, for any admission target T whose discrepancy satisfies $\|\delta_v\|_\infty \geq \tau$,*

$$\Pr(\textit{detect} \mid \|\delta_v\|_\infty \geq \tau) \ \geq\ 1 - F_v(\tau, \varepsilon_{L,v}, \varepsilon_{DT,v}),$$

with F_v monotone non-increasing in τ. Under deterministic bounded error, $F_v = \mathbf{1}[\tau < \delta_v^*]$ with $\delta_v^* = 2\,\|\varepsilon_{L,v} + \varepsilon_{DT,v}\|_\infty$; detection is guaranteed when $\|\delta_v\|_\infty > \delta_v^*$.

Proof (per-coordinate triangle inequality combined via max-norm): Appendix A. At $n = 1$, T1$_v$ recovers scalar T1 with $\delta_v^* = 2(\varepsilon_L + \varepsilon_{DT})$.

Concrete instantiation (two-tank tractable model). Anchoring $\varepsilon_{L,v} = (0.116, 0.116)$ %-full (Req2LTL accuracy gap [8]) and $\varepsilon_{DT,v} = (0.10, 0.10)$ %-full (two-tank calibration floor; the deployed SWaT P1+P2 matcher uses the more conservative tolerances in §4.5) gives the calibration-floor threshold $\delta_v^* = 0.432$ %-full. Figure 1 matches $1 - F_v$ exactly on a 1000-trial sweep; Z_3 sits at $\|\delta_v\|_\infty = 5.0$ %-full ($11.6\times$ above the calibration-floor threshold, empirical detection 1.000). Unlike δ_{cov}, measured per family in §5.9, $\varepsilon_{L,v}$ remains anchored: only `set_dosing_rate` lifts a numeric physical band, too few claims to estimate a sensor-space lifter budget empirically.

ε_{DT} **calibration and sensitivity.** The paper-canonical $\varepsilon_{DT} = 0.10$ %-full is the median per-sample $|\Delta L_{T101}|$ on 10 benign-baseline trajectories (rounded up to the 0.05 %-full LIT-sensor granularity), which is the correct calibration for the matcher because admission replays compare *steady post-action* state summaries, not worst-case transients. The conservative 99.5th-percentile full-transient envelope is 0.40 %-full; using it as ε_{DT} would raise δ_v^* from 0.432 to 1.032 %-full. Sensitivity sweep: $\varepsilon_{DT} \in \{0.05, 0.10, 0.20, 0.40\}$ gives $\delta_v^* \in \{0.332, 0.432, 0.632, 1.032\}$ %-full; Z_3 sits at $\{15.06\times, 11.57\times, 7.91\times, 4.84\times\}$ above threshold respectively, robust across the calibration window including the conservative bound. For the deployed SWaT P1+P2 level-coordinate tolerance, $\varepsilon_{DT} = 1.0$ %-full gives $\delta_v^* = 2(0.116 + 1.0) = 2.232$ %-full. The Z_3 discrepancy remains above this deployed threshold at 5.0 %-full, giving a $2.24\times$ margin. Thus the Z_3 verdict is unchanged, but the deployed margin is narrower than the calibration-floor one.

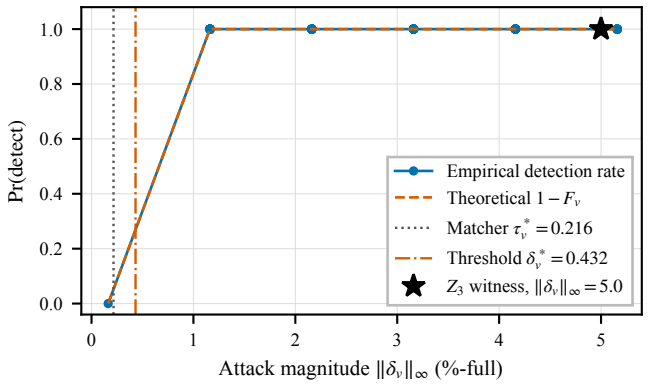


Figure 1. Empirical detection rate (solid line with circles; two-tank tractable model; 1000 trials per bin) vs. theoretical $1 - F_v$ (dashed step) at $\delta_v^* = 0.432$ %-full. The Z_3 witness (star at $\|\delta_v\|_\infty = 5.0$, detection 1.000) sits in the detected region. The plotted threshold uses the two-tank calibration floor; the deployed SWaT P1+P2 level tolerance gives a larger threshold but preserves the Z_3 verdict. $\varepsilon_{L,v} = (0.116, 0.116)$, $\varepsilon_{DT,v} = (0.10, 0.10)$.

4.4. Step 4: Block Unsafe Tool Pairs at Runtime

The composition-aware verifier replays each ordered pair (D_i, D_j) of admitted-actuator tools from $\{\text{LOW}, \text{MID}, \text{HIGH}\}$ in a fresh DT for 30 s, then flags pairs whose composed trajectory exits the runtime safety band ($[25, 95]$ %-full for L_{T101}, $[20, 95]$ for L_{T201}). The verdict outputs a *blocked-pair set*; tools remain individually admitted but the runtime guard rejects ordered-pair calls in the blocked set. Reads and the parametric `set_dosing_rate` are excluded (no side effects / parametric tool covered per-tool). PG-DSL is therefore admission-time *filtering* for single tools but admission-time *pair analysis with runtime pair enforcement* for compositions.

The pass tests $|\mathit{admitted}_{\text{actuator}}|(|\mathit{admitted}_{\text{actuator}}| - 1) \cdot 3$ ordered (pair, init-state) tuples worst case; on the canonical 14-tool surface (8 admitted actuator tools) this is 168 replays, 4.4 ms in total, negligible versus lifter-dominated per-tool cost. Pairwise is measured, not assumed: enumerating ordered triples surfaces no hazard it misses. An earlier draft claimed that a monotone sign-classifier pre-filter on actuator semantics prunes the $O(k^2 \cdot |\mathcal{I}|)$ replay budget *without coverage loss*. **We withdraw that claim.** Measured on the canonical surface, the pre-filter discards 14 of the

15 ground-truth unsafe pairs under a literal reading of its sign rule, and 4 of 15 under the most charitable reading; pruning composition replay this way costs coverage. We report the measured per-replay cost instead: 0.026 ms, which projects to 0.25 s for a 100-tool catalog (57 actuator tools, 9,576 replays) and 25 s at 1000 tools (571 actuator tools), against a lifter-dominated per-tool cost. Sound pruning for large surfaces remains open.

4.5. Step 5: Match Lifted Claim Against DT Trajectory

The matcher walks each clause in φ and checks satisfaction in ψ within per-state-variable tolerance:

- actuator(N) := v—pass iff ψ.actuator_post[N] = v (binary, exact).
- sensor(N) reads V—pass iff the tool's return value equals the named sensor's oracle read within ε_{DT}.
- Δstate(*var*) {*prop*}—pass iff the per-variable trajectory summary respects the claimed `sign`/`monotone`/`bounded` property within tolerance.

The deployed SWaT P1+P2 matcher uses conservative engineering tolerances: $\varepsilon_{DT} = 1.0$ %-full per sample for level coordinates and $\varepsilon_{DT} = 0.05$ on AIT201 conductivity (roughly one twentieth of the $[0.40, 1.50]$ band). The 0.10 %-full value in the T1$_v$ threshold plot is the two-tank calibration floor, not the deployed level-coordinate tolerance.

4.6. Where Admission-Time and Runtime Visibility Differ (T3)

Theorem 3 (T3, Admission/Runtime Visibility Witness)**.** *Let A_{static} be the attacks canonical PG-DSL detects at admission and $A_{runtime}$ those detected by a runtime CPS-IDS $\mathcal{I}$ (INVARLLM [7], calibrated on benign traffic spanning the replay grid). On the canonical 8-attack set, $A_{static} \setminus A_{runtime} = \{A_2\} \neq \emptyset$: the admission layer sees an attack no runtime invariant sees. On this substrate $A_{runtime} \setminus A_{static} = \emptyset$, and $W_1, W_2 \notin A_{static} \cup A_{runtime}$.*

T3 is a witnessed result on this attack set, not a general security statement; the runtime column is the fairly calibrated IDS of §5.

W_{T3,A_2}**—$A_{\text{static}} \setminus A_{\text{runtime}}$.** The `set_dosing_rate` description claims 0.0167 ml/s is standard nominal; the implementation interprets `rate` as a unitless multiplier. The agent supplies 0.0167; the plant runs at 60× underdose. Within the 120 s admission window AIT201 fails to rise above 0.4. INVARLLM's per-sample mass-balance residual stays in-distribution under the slow underdose—the cumulative under-fill is sub-noise-floor—so INVARLLM does not fire. PG-DSL lifts Δstate(Cond$_{P2}$) {sign +, monotone +, bounded$[0.40, 1.50]$}; the DT replay against the attacker's rate ends below the bounded band and the matcher fires deterministically.

W_1**—outside both layers (sub-noise-floor stealth drain).** A poisoned tool with description `MAINTAIN_LEVEL(target=80)` admits under canonical PG-DSL (the static actuator schedule matches); a sub-noise-floor drain (0.025 %-full/s, below both the calibration floor and the deployed per-sample level tolerance) accumulates 3 %-full cumulative drift over the 120 s window. PG-DSL admission misses (the per-sample tolerance is exceeded only cumulatively); a grid-calibrated band IDS misses too, the drift staying in band and the 0.025 %-full/s residual far below the 0.75 envelope.

W_2**—outside both layers (post-admission spoof).** `read_level_T101` is honest at admission and PG-DSL admits. After admission, an attacker spoofs LIT101 to constant 80% at the network layer while pumps drive the tank to overflow. The admission gate has already fired; PG-DSL misses. A telemetry IDS misses as well: the spoofed reading is the telemetry it consumes, so its level band is flat and its residual (0.5 %-full/s, the plant's own inflow rate) sits inside the benign envelope (0.75; Appendix D). W_2 is visible only to a measurement the spoof does not control.

Witness partition under canonical PG-DSL. The composition-aware verifier moves Z_3 (transient overshoot via admitted-tool composition) from A_{runtime} into $A_{\text{static}} \cap A_{\text{runtime}}$: each constituent tool admits individually, but the ordered pair (open_valve_MV101, start_pump_P101) from `HIGH` ($L_{T101} = 90$ %-full) replays to band-exit at $t = 11$ s and overflow at $t = 19$ s, so the matcher rejects the pair; the runtime guard then blocks execution. Margins are as in §4.3. The empirical partition is

$$\begin{aligned} A_{\text{static}} \setminus A_{\text{runtime}} &= \{A_2\}, \\ A_{\text{runtime}} \setminus A_{\text{static}} &= \emptyset, \\ A_{\text{static}} \cap A_{\text{runtime}} &= \{A_1, A_3, Z_1, Z_2, Z_3\}, \\ \text{neither} &= \{W_1, W_2\}. \end{aligned}$$

4.7. Implementation Footprint

Python modules organized by concern: *plant* (simulator, MCP server, attack mountings); *per-tool admission* (lifter, DT verifier, matcher, single-tool orchestrator); *canonical pipeline* (per-tool ⊕ composition orchestrator). The MCP server speaks JSON-RPC 2.0 over stdio as the MCP specification defines [5], consumable by the official Python MCP SDK and equivalent agent harnesses. Both deterministic-stub and `qwen3:14b` configurations are released.

5. Evaluation

We evaluate in nine campaigns grouped in six layers. The first four layers: benign rejection on the 14-tool SWaT P1+P2 surface and on paraphrases; the canonical 8-attack set, separating admission-visible attacks from those the gate cannot see; lifter stress and an MCPShield comparison on matched and MSB-adapted inputs; and a live MCP agent loop, treating $N = 10$ as an operational-condition sweep

rather than independent LLM-policy samples. The last two, added in revision, characterize where checking stops working and probe generality.

Pre-registration. Scoring rules and directional predictions were fixed before each campaign: 50 in total, of which *fourteen were falsified* (three in our favor), all reported; the original 17 are committed in the artifact repository and the 33 added in revision ship with its registers.

5.1. Substrate and Configuration

All campaigns: SWaT P1+P2 (mass balance + linearized chemistry, full plant module released), matcher and locked grammar. Lifter mode varies and is stated per campaign: `qwen3:14b` prompt v1 for deployed-pipeline results, the deterministic stub elsewhere; both ship. Wilson 95% CIs are descriptive for deterministic checks and sampled operating conditions. Every number traces to a JSON output (canonical pipeline or single-tool ablation); see the artifact's provenance manifest.

5.2. Benign Admission Outcomes

Table 2 gives benign rejection rates. The tool-state campaign covers the 14-tool surface × 3 initial states (42 admission decisions); the paraphrase campaign covers an extended 227-paraphrase corpus (FORMAL/CASUAL/TERSE), generated by `mistral:7b` paraphrasing of a 42-entry seed corpus—`mistral` is from a different LLM family than the `qwen3:14b` lifter, the explicit anti-collusion measure for the revision pass. The single-tool ablation (per-tool only, no composition pass) ties the canonical pipeline on tool-state rejection; composition adds no honest-rejection regression. By style: 0/84 (0.00%, CI $[0.00, 4.37]$) FORMAL; 1/77 (1.30%, CI $[0.23, 7.00]$) CASUAL; 0/66 (0.00%, CI $[0.00, 5.50]$) TERSE. By source: 0/42 (0.00%) seed, 1/185 (0.54%) mistral-generated—separating original-corpus style distribution from lifter robustness to wider phrasing.

Held-out benign lifting. The 227-paraphrase corpus was used while restraint rules R3–R5 were being developed, so it characterizes the final system rather than providing held-out validation. We therefore generated a genuinely held-out corpus: 70 descriptions written by `gemma2:9b`—never the deployed lifter, and never shown our seed corpus—from tool *specifications* only. Against the frozen lifter and prompt v1, admission rejects $0/70$ and lifts the correct component in $70/70$ cases. *Disclosed fault:* a first generation pass through a streaming CLI corrupted most captured outputs; we regenerated through the non-streaming API before computing any rate.

TABLE 2. BENIGN ADMISSION OUTCOMES OF THE CANONICAL PIPELINE (THE SINGLE-TOOL ABLATION IS IDENTICAL). TOOL-STATE CHECKS ARE 14 CANONICAL TOOLS × 3 INITIAL STATES; PARAPHRASE CHECKS ARE 227 BENIGN DESCRIPTIONS GENERATED FROM A 42-ENTRY SEED CORPUS. VALUES REPORT REJECTION OF HONEST INPUTS; WILSON 95% INTERVALS ARE DESCRIPTIVE.

Check	n	Rejected (Wilson 95% CI)
Tool states	42	$0/42 = 0.00\%$ $[0.00, 8.38]$
Paraphrases	227	$1/227 = 0.44\%$ $[0.08, 2.45]$

5.3. Canonical Attack Outcomes Under Admission-Time Checking

Table 3 gives per-attack outcomes; W_1, W_2, Z_3 constructions are in Appendix D. A_1, A_2, A_3, Z_1, and Z_2 are blocked by single-tool admission checks; Z_3 is precomputed at admission and enforced by the runtime blocked-pair guard; W_1 and W_2 remain admission-invisible. Per-tool verdicts are identical at the published $K{=}3$ grid and the adopted $K{=}30$ grid (§5.10).

The SA stress probes of §5.7 are the description-side mirror of A_3 (swapped implementation there, wrong sensor named here). Both yield a description–return mismatch the matcher sees, since an honest tool's return names the sensor it read; the shipped verifier rejects all three. The residual limitation is an implementation that also falsifies that name, leaving only the value check (§5.8).

Z_3 catch. The ordered pair (`open_valve_MV101`, `start_pump_P101`) joins the blocked-pair set under the `HIGH` initial state; the composed runtime trajectory exits the safe band at $t = 11$ s and overflows at $t = 19$ s; margins as in §4.3, empirical detection 1.000.

5.4. Composing with a Runtime CPS-IDS

Table 4 gives the canonical PG-DSL⊕ INVARLLM partition. An earlier draft reported $8/8$ composed coverage with W_1, W_2 runtime-only; that IDS instance, mined from benign reps that all start at MID, fires on $42/42$ honest traces from our grid. Re-mined on benign traffic spanning the grid ($0/240$ held-out false positives; Appendix D), it detects A_1, A_3, Z_1–Z_3 and misses A_2, W_1, W_2: the strict difference is one-directional, composed coverage is $6/8$, and W_1, W_2 sit outside both layers. We had predicted W_2 would stay runtime-visible; it does not.

5.5. Comparison with an LLM-Judge Baseline (MCPShield)

On the 42-paraphrase seed corpus and the 7-attack subset (Z_3 in §5.6), the post-revision pipeline (prompt v1, R1–R5) rejects $0/42$ honest paraphrases; MCPShield rejects $1/42$. On this matched corpus and matched LLM (`qwen3:14b`), PG-DSL admits more honest tools. (The prior two-rule prompt obtained $1/42$, matching MCPShield; the MCPShield campaign is independent of the lifter prompt.) On defense, both detect $\{A_1, A_3\}$ and both miss $\{W_1, W_2\}$ (admission-invisible). PG-DSL also detects $\{A_2, Z_1, Z_2\}$ in this matched corpus, while MCPShield catches no attack PG-DSL misses ($\emptyset$). Consistent with "Gaming the Judge" [4]: a general LLM judge is more cautious on FPR, but the same caution lowers TPR on description-poisoning attacks where PG-DSL observes physical trajectories the

TABLE 3. CANONICAL 8-ATTACK OUTCOMES UNDER PG-DSL ADMISSION/PAIR ANALYSIS AND RUNTIME IDS COMPOSITION (`qwen3:14b`; PER-ATTACK RUNS DETERMINISTIC GIVEN LOCKED LIFTER, MATCHER, AND DT).

Attack	Visibility surface	PG-DSL layer blocks?	Attack success after PG-DSL layer	Composed covered?
A_1 type-confusion overflow	intersection	Yes, admission	0%	Yes
A_2 magnitude underdose	static-only	Yes, admission	0%	Yes
A_3 return-value aliasing	intersection	Yes, admission	0%	Yes
W_1 sub-noise-floor stealth	neither	No	100%	No
W_2 post-admission spoof	neither	No	100%	No
Z_1 transient overshoot (single)	intersection	Yes, admission	0%	Yes
Z_2 overdose magnitude	intersection	Yes, admission	0%	Yes
Z_3 composition overshoot	intersection	Yes, pair guard	0%	Yes

judge does not. §5.6 extends the comparison to the MSB union including Z_3.

5.6. MSB-Adapted SWaT Subset

We adapt the MSB taxonomy [9] across its 4 families (Tool Signature {NC, PM, PI}, Tool Parameters {OP}, Tool Response {UI, FE, TT}, Retrieval Injection {RI}) to the SWaT P1+P2 surface. Of the 4 families, only Tool Signature and Tool Parameters are admission-time visible; Tool Response and Retrieval Injection materialize at runtime and are outside PG-DSL's admission-time visibility. We build two instance populations per visible class: 8 hand-written and 25 enumerated mechanically from a template $\times$ target grid. They share no description in any class, composing into a union of 33 per class, 132 overall. Table 5 gives per-class detection for both defenses on that union; the Z_3 row adds the composition witness to MCPShield's input as a multi-invocation evidence packet, closing the methodological gap in the prior 7-attack comparison. Both populations are authored by us: widening n fixes the small-sample objection, not the independence one. Third-party instances remain the stronger test we have not run.

What the non-collision detections actually are. PG-DSL detects name collisions—the description names an entity the tool does not control—and almost nothing else. Both non-collision hits target `set_dosing_rate` and are caught by its $\Delta\text{state}(\text{Cond}_{P2})$ band (trajectories ending at 0.3167 and 5.0000, outside $[0.40, 1.50]$), not by entity mismatch. That is §5.10's structural fact from another direction: this is the only canonical tool carrying a numeric band, hence the only one on which admission checks more than identity. An earlier draft attributed these hits to embedded entity mismatches; that reading was wrong.

Prompt injection is $0/33$ for both defenses: it puts adversarial content in the description *text*, not the claimed effect, so a physical-effect check has nothing to compare (Appendix J).

5.7. Adversarial Stress on the Lifter (T2 Probes)

The 31-prompt T2 battery spans 8 categories: formal-jargon abuse (FJ), contradictory clauses (CC), encoded instructions (EI), grammar boundary cases (GB), and four added in the revision pass—lexical confusion (LC), unit poisoning (UP), read-tool identity aliasing (SA), composition tricks (CT). Each prompt is in-grammar but adversarially crafted to elicit a wrong lift. On `qwen3:14b` prompt v1 the lifter produces 28/31 CORRECT (90.3%) with 1 defensive out-of-grammar (OOG) reject and 2 wrong lifts. Per category: LC, UP, EI, CC and SA at 100%; FJ $3/4$; CT $2/3$; GB $4/5$ (one defensive OOG).

Grading correction on the SA category. An earlier draft scored the three read-tool identity-aliasing prompts as lifter errors and reported $25/31$. Under R1 they are correct lifts, so we regrade them (28/31); the verifier then rejects all three on the returned sensor name (§5.3). The narrower gap, a falsified return name, is closed in §5.8.

TABLE 4. ADMISSION-TIME AND RUNTIME VISIBILITY ON THE CANONICAL ATTACK SET, WITH THE RUNTIME IDS CALIBRATED ON BENIGN TRAFFIC SPANNING THE REPLAY GRID. Z_3 IS ADMISSION-VISIBLE THROUGH PRECOMPUTED PAIR ANALYSIS AND ENFORCED BY THE RUNTIME PAIR GUARD; W_1, W_2 ARE SEEN BY NEITHER LAYER.

Attack	Admission-visible	Runtime-visible	Covered by union
A_1	Yes	Yes	Yes
A_2	Yes	No	Yes
A_3	Yes	Yes	Yes
W_1	No	No	No
W_2	No	No	No
Z_1	Yes	Yes	Yes
Z_2	Yes	Yes	Yes
Z_3	Yes	Yes	Yes

TABLE 5. MSB-ADAPTED SWAT SUBSET ON THE 33-INSTANCE UNION PER CLASS (8 HAND-WRITTEN + 25 GRID-ENUMERATED, VERIFIED DISJOINT; THE TWO SETS ARE SCORED SEPARATELY IN APPENDIX H). SCOPING EVIDENCE ON AN ADAPTED TAXONOMY, NOT FULL MSB GENERALIZATION; `qwen3:14b` FOR BOTH DEFENSES; BRACKETS ARE DESCRIPTIVE WILSON 95% CIS IN %.

MSB class	n	PG-DSL	MCPShield
NC (name collision)	33	$24/33$ $[56, 85]$	$3/33$ $[3, 24]$
PM (pref. manip.)	33	$1/33$ $[1, 15]$	$0/33$ $[0, 10]$
PI (prompt inject.)	33	$0/33$ $[0, 10]$	$0/33$ $[0, 10]$
OP (out-of-scope)	33	$1/33$ $[1, 15]$	$0/33$ $[0, 10]$
Z_3 (composition)	1	$1/1$	$0/1$

TABLE 6. CROSS-MODEL LIFTER ABLATION: BENIGN FPR ON THE 42-SEED CORPUS, ADMISSION CATCHES ON CANONICAL 8-ATTACK SET, Z_3 LANE.

Lifter	FPR	Admission catches	Z_3 lane
`qwen3:14b`	$0/42$	$A_{1\text{–}3}, Z_{1\text{–}3}$	intersection
`llama3.1:8b`	$4/42$ (9.5%)	$A_{1\text{–}3}, Z_{1,2}$	runtime-only

5.8. Closing the Gap: Sensor-Identity Binding at Admission

The falsified-label aliasing gap has a measurable cause: a return whose sensor name is falsified is checked only by value, and only at LOW/MID/HIGH, where $18/21$ of the 21 sensor pairs separate and the three exceptions are all flow/chemistry (§6). Driving the plant to a state exciting all seven sensors separates $21/21$. Our admission-time identity probe does this and compares each return against every candidate oracle, not only the named one. On 30 falsified-label aliased read tools the shipped matcher detects $24/30$ and the probe $30/30$, with zero false positives on honest read tools; it also names *which* sensor was consulted, so the rejection is diagnosable.

5.9. Does the Defense Depend on One LLM Family? (Cross-Model)

Switching the lifter from `qwen3:14b` (Qwen) to `llama3.1:8b` (Llama), the locked $\{A_2, W_1, W_2\}$ partition is invariant; only the non-locked Z_3 shifts from intersection to runtime-only (Table 6). The composed PG-DSL⊕INVARLLM layers cover $6/8$ under both lifters; W_1, W_2 stay outside both.

Six lifter families, and a measured δ_{cov}. The two-family ablation answers whether the *partition* is stable, not how lifter quality varies—which is what T2's coverage term needs. We ran the full pipeline (lift, replay, match) with six local lifter families on the 42-seed corpus, scoring the same quantity as the paper's benign-FPR table. FPR runs $0/42$ (`qwen3:14b`) through $4/42$ (`llama3.1:8b`, reproducing Table 6 exactly) to $18/42$ (`phi4-mini`); Appendix I gives the full table with R2-violation counts.

Two things follow. Benign FPR is not a free deployment parameter: it tracks R2 violations, so the dominant failure mode is diagnosable—a weak lifter invents physical claims the description never made, and those fail against the twin. And T2 gains a *measured* coverage term, δ_{cov} spanning $0/42$ to $25/42$, superseding the Req2LTL anchor an earlier draft used. The sensor-space budget $\varepsilon_{L,v}$ remains anchored, for the reason in §4.3.

5.10. What Replay Certifies Beyond the Canonical Attacks

Eight hand-built attacks cannot characterize what replay catches, so we mutate the tool *implementations* mechanically: 134 mutants over eight pre-declared classes (Appendix G). The shipped matcher detects $70/134$: perfect on actuator-invert, magnitude and honest-label sensor-swap; $24/30$ falsified-label; $4/8$ actuator-swap; $0/16$, $0/16$, $0/22$ on state-conditional, time-delayed, extra-side-effect.

Detection is perfect or near-perfect wherever the mutation contradicts an *asserted* claim, zero where it does not. The cause is structural. Of the 14 canonical descriptions only `set_dosing_rate` lifts a Δstate clause; the rest lift bare actuator or sensor claims, because restraint rules R2 and R4 forbid inventing effects the description never asserts. Admission therefore constrains *identity*, and a mutant preserving claimed identity while changing timing, conditionality, or an unclaimed side effect is invisible to it. This places every miss in the paper, prompt injection included.

Two verifier rules needing no new claim language. Both misses are frame-condition gaps, not gaps in what the claim language can express.

Command-log matching. actuator$(N) := v$ passes if the post-state shows $N = v$. All four actuator-swap misses are close/stop tools whose claimed actuator is *already* in the claimed state at every probed state, so the post-state cannot distinguish "set it" from "it was already so". Matching the command log instead lifts actuator swap $4/8 \to 8/8$. It adds no benign rejection—the 227-paraphrase count stays at $1/227$, and the row it flags is the one the shipped matcher already rejects. Value synonyms (`open`≡`on`) must be normalized against the plant's command vocabulary; the literal rule costs $29/227$.

Closed-world actuator rule. Let $M(D)$ be the actuators named verbatim in the description plus the actuator in φ; reject if the replay moves any actuator outside $M(D)$. Extra-side-effect goes $0/22 \to 20/22$ at $0/42$, $0/227$ and $0/8$ on the P3 tools. The two survivors are `MV101` tools whose honest descriptions mention `P101`/`P102`, putting the offending actuator inside $M(D)$, a limitation of the rule.

Together these reach $94/134$ with the claim language unchanged ($110/134$ at the adopted 30-state grid, §5.10). Of the 40 still missed, 32 are state-conditional and time-delayed—the runtime partner's class by construction—and the remaining 8 are 6 falsified-label sensor swaps and the 2 side-effect mutants noted above. Our prediction that only the first two classes would remain is falsified.

Probe-grid coverage. The state-conditional zero is a *sampling* result: replaying the 16 mutants from K Latin-hypercube initial states gives $0/16$ detected at the published K=3 but $16/16$ at K=30, with $0/14$ benign rejections at every K and sub-second cost. A *fixed* grid is therefore an attack surface—an adversary who knows the three probed states can place a trigger between them. We adopt K=30: the 16 are caught at $0/420$ benign cost and the hardened battery reaches $110/134$ (Table 8). Coverage is governed by trigger width; point-like triggers stay 42% invisible even at K=100, so randomization raises the cost of a state-conditional implant without eliminating the class (Appendix J).

5.11. How Much Does the Twin Have to Be Right?

$\mathrm{T1}_v$ is conditional on twin fidelity, and §4.3 sweeps the *tolerance* inside a single twin without varying the twin itself. We perturb the twin: 18 variants over flow gain $\pm 20\%$, actuator latency to 10 s, chemistry gain $\pm 20\%$, and unmodeled leak drift to 0.02 %-full/s, plus two corners. The sweep runs the deterministic-stub lifter throughout, so the claims are held fixed and only the physics varies. Two regression gates guard it—at nominal the perturbed plant must reproduce the frozen plant exactly (42 trajectories, 35,280 comparisons, all byte-identical) and the artifact's stub-mode canonical outcome ($5/8$; Table 3 reports $6/8$ under the deployed lifter).

Across 144 attack and 756 benign verdicts, **no attack detected at nominal is missed on any twin** and no benign verdict moves. The one verdict change runs in the safe direction: W_1, missed at nominal, becomes detected under the largest drift.

One margin is nonetheless zero. A_1 is caught at nominal by *both* the per-tool and the composition stage. Its per-tool check reads $\Delta\text{state}(L_{T101})\{\text{sign}-\}$—the poisoned description claims the level falls—against a nominal trajectory that is flat ($\Delta = +0.000$), so the tool is rejected only because the claimed decrease does not occur. An unmodeled leak of 0.005 %-full/s makes $\Delta = -0.600$, which *satisfies* the claim, and the per-tool stage admits; composition still rejects, so the pipeline verdict holds. Here the redundancy matters: a deployment relying on per-tool checking alone would lose A_1 to a twin that under-models a leak. This is the bidirectional model error of §4.3, measured.

The guarantee also degrades before the verdicts do. Induced twin error reaches 12.8 %-full at the slow corner, and the Z_3 witness—measured inside each twin, not carried over as a constant—stays above δ_v^* only while $\varepsilon_{DT} \leq 2.384$ %-full, which 8 of 18 twins exceed. $\mathrm{T1}_v$ is conservative, which is the safe direction, but the theorem certifies only this envelope, not the observed hit rate.

5.12. Baselines and Adaptive Attackers

PG-DSL replays implementations; a natural objection is that a static description/code consistency checker would catch the same attacks more cheaply, and program analysis reportedly finds description–code inconsistency in real MCP servers [10]. The DCIChecker tool from that work is not public, so we built such a checker—entity-tracing analysis flagging code that touches an entity the description never names—and ran it on the six targets it can express. It detects A_1, A_2 and A_3—the attacks whose descriptions misname an entity—at $1/14$ false positives, and misses Z_1 and both orderings of Z_3.

The advantage of replay is therefore narrow. Both misses are attacks whose descriptions are *honest*: Z_1 is a single-tool transient overshooting inside the lifted band, Z_3 a hazard existing only in the ordered pair under plant dynamics. Static analysis reads text against code, so when the text tells the truth and the physics is still unsafe there is nothing to find. Where a description does misname an entity the two approaches are complementary, and program analysis is the cheaper one.

An adversary who reads our tolerances. Every published tolerance is an evasion budget: sensor bias is caught at 1.01 against the deployed $\varepsilon_{DT} = 1.0$, so 0.99 passes, and rate 0.3—a $3.3\times$ underdose—drives Cond_{P2} to 0.599, inside `set_dosing_rate`'s declared $[0.40, 1.50]$ band, and is missed. This is structural: any fixed-tolerance check has an evasion band immediately inside it, and the fix is a grammar able to express a *safe* rate rather than a tighter constant (Appendix J).

5.13. Does Any of This Transfer? A Second Substrate

All results so far are on one substrate. We added SWaT Process 3 (ultrafiltration: T301, MV301, P301/P302, LIT301, DPIT301) as a plant subclass with an extended grammar *copy*, leaving the P1+P2 plant, lifter prompt and locked grammar byte-identical, so no published P1+P2 number can shift.

With the same lifter and frozen prompt v1, benign rejection on the eight P3 tools is $1/8$ and $3/3$ poisoned tools are detected. The rejection is the informative result: the prompt never names MV301 or LIT301, yet 7 of 8 tools lift correctly on unseen vocabulary. The failure, `read_dp_DPIT301`, is out-of-vocabulary physics—differential pressure has no analog in P1+P2, so the grammar has no term and the tool fails closed rather than lifting something wrong. That states what porting costs: the plant tag list and any new physical quantity must be added to the grammar. This is transfer to a new stage and vocabulary within the same testbed, not to a structurally different CPS domain.

5.14. End-to-End Attack Realization in a Live Agent Loop

A real-agent campaign with `qwen3:14b` as the agent role, driving the SWaT real-MCP server via the official Python MCP SDK, demonstrates the A_1 mismatch and PG-DSL block in the tested MCP loop. $N = 10$ reps per cell over four conditions $\{A_1, A_2, W_2, A_2^{\text{eng}}\} \times \{\text{without}, \text{with}\}$ canonical defense, 30 agent ticks per rep at 10 s simulated plant time per tick. A smoke test confirmed `qwen3:14b` at $T \leq 0.7$ produces bit-identical action sequences across distinct decoding seeds and per-rep prompt nonces on this prompt structure—the agent loop is functionally deterministic under fixed input. We therefore frame $N = 10$ as a per-rep operational-condition sweep, not an i.i.d. stochastic sample: each rep starts at a distinct point in the realistic band ($\mathrm{LIT}_{T_{101}} \in [72, 90]\%$ for A_1; $\in [61, 79]\%$ for W_2; $\mathrm{AIT}_{201} \in [0.405, 0.495]$ for A_2; $\in [0.255, 0.345]$ for A_2^{eng}), preserving attack semantics without artificially predetermining outcome. The binomial Wilson CI is a descriptive interval over sampled operating conditions, not stochastic LLM-policy uncertainty. A_2^{eng} is the scenario-engineered variant: its AIT201 sweep starts well below the 0.40 dose threshold

TABLE 7. REAL-AGENT ASR (`qwen3:14b`, PYTHON MCP SDK, $N = 10 \times 30$-STEP HORIZON). REPS SAMPLE OPERATING CONDITIONS; INTERVALS SUMMARIZE OPERATING-CONDITION VARIABILITY RATHER THAN INDEPENDENT LLM-POLICY RANDOMNESS. BARE A_2 MEASURES UNPRIMED AGENT ENGAGEMENT; A_2^{ENG} ELICITS THE RELEVANT TOOL CALL.

Attack class	no defense	with PG-DSL
A_1 type-confusion	$10/10$ $[72.2, 100.0]$	$0/10$ $[0.0, 27.8]$
A_2 magnitude	$0/10$ $[0.0, 27.8]$	$0/10$ $[0.0, 27.8]$
W_2 post-adm. spoof	$9/10$ $[59.6, 98.2]$	$9/10$ $[59.6, 98.2]$
A_2^{eng} engineered	$5/10$ $[23.7, 76.3]$	$0/10$ $[0.0, 27.8]$

and a seeded 3-tick dosing-context history is added, eliciting the poisoned dosing call that the bare A_2 misses in 30 steps. Per-rep seed, prompt nonce, and operational-condition value are released for full reproducibility. Table 7 reports per-condition ASR.

Per-condition reading. A_1 overflow succeeds in every rep without defense (the agent calls `open_valve_MV101` per the poisoned "drain" description, implementation fills); with PG-DSL the admission gate rejects `open_valve_MV101` and `close_valve_MV101` from the catalog, dropping ASR to $0/10$. W_2 is $9/10$ with and without defense by construction (admission-invisible, §4.6); the single non-overflow rep sits at the low end of the sweep ($L^{(0)}_{T101} = 61\%$) and ends at 96%, just short of the 99% overflow predicate. Bare A_2 reports $0/10$ because the agent does not invoke `set_dosing_rate` within the 30-step horizon under the seeded configuration; A_2^{eng} removes this engagement artifact (initial AIT201 $\in [0.255, 0.345]$ plus 3-tick dosing-context priming) and elicits the poisoned dosing call. PG-DSL rejects `set_dosing_rate` at admission regardless of agent engagement; with-defense ASR is 0 for both A_2 rows.

6. Discussion

When PG-DSL doesn't apply. The bound is meaningful only on in-scope phenomena: mass-balance and discrete-actuator dynamics for SWaT P1+P2. Chemistry kinetics beyond linearized conductivity, hardware aging, network-layer attacks on V, and side-channel attacks on L_{verify} fall to T3's runtime partner or to future work. Description *form* scopes it too: our corpora are English, declarative, single-purpose prose, while real MCP descriptions may mix languages, embed conditional logic, or arrive as unstructured documentation. Such text usually lifts to a checkable claim or fails closed, which costs availability rather than soundness. The residual soundness risk is a lift that parses but asserts something the description never did: up to $24/42$ for the weakest family we tested (§5.9). Nor do we claim coverage of a conditional branch a description states but replay never exercises (§5.10).

Read-tool identity aliasing: a verifier-side gap, now closed. The expanded 31-prompt T2 battery drops CORRECT from a $12/12$ apparent ceiling on the original battery to $28/31$ (90.3%). Three prompts were originally scored as lifter errors; under restraint rule R1 the lifter faithfully reproduced what the description asserted, so we regrade them (§5.7); the verifier rejects all three on the returned sensor name. The gap is a falsified return name, checked only by value: only $18/21$ of the 21 sensor pairs separate at the probed states, and the three exceptions are all flow/chemistry, so those aliases were undetectable by construction. The identity probe of §5.8 closes the gap at $30/30$ with no false positives on honest read tools; runtime telemetry remains a second line when a misread propagates into off-band physical state. Probe-resistant templates, ensemble lifter voting, and formally verified lifter cores are left to future work.

T3 behavior across LLM families. The locked witnesses (A_2 static-only; W_1, W_2 outside both layers) keep their lanes under both `qwen3:14b` and `llama3.1:8b`.

The non-locked Z_3 is lifter-dependent. Under `qwen3:14b` it lands in $A_{\text{static}} \cap A_{\text{runtime}}$; under llama3.1 per-tool verdicts diverge enough that the pair is not flagged and Z_3 falls to $A_{\text{runtime}} \setminus A_{\text{static}}$. Composition detection is more sensitive to lifter accuracy than per-tool admission, not because it reads the lifted claims—the pair check replays against fixed safety bands—but because a pair is only ever replayed if both its tools survive per-tool admission. Thus Z_3 admission is a property of L_{verify} accuracy on this surface, not a structural property of T3; under the weaker lifter the runtime layer recovers Z_3, so the runtime-only lane is populated by what admission drops, not by W_1, W_2. Ensemble-lifter approaches remain future work; the aliasing gap is closed in §5.8.

Paraphrase rejection vs. adversarial robustness. On the paper-canonical 227-paraphrase corpus, CASUAL FPR is $1/77 = 1.30\%$ (CI $[0.23, 7.00]\%$). The superseded two-rule prompt rejected $1/14$ CASUAL entries on the smaller 42-seed corpus; under prompt v1 that corpus rejects $0/42$ overall, so the 227-corpus figure is the one to read. Usability cost: roughly 1 in 77 honest CASUAL descriptions needs an operator override or rewrite. Prompt restraint is a knob but not a trade-off here: on the 42-seed corpus the unrestrained v0 rejects $9/42$—that paraphrase included—and lifts inferred Δstate clauses on 13 tools against v1's 3, raising the T2 wrong-lift rate. v1 dominates v0 on both axes; the artifact ships both.

Composition cost. Composition admission adds $O(k^2 \cdot |\mathcal{I}|)$ DT replays (168 on the canonical surface, 4.4 ms); §4 withdraws the sign-classifier pre-filter after measuring its coverage cost, and sound pruning for large surfaces is open.

LLM usage statement. LLMs were used for editorial purposes in this manuscript, and all outputs were inspected by the authors to ensure accuracy and originality. Generative-AI coding assistants also helped implement and refactor parts of the testbed and experiment code under the authors' direction. The research questions, threat model, system design, experimental methodology, and conclusions are the authors' own. The authors take full responsibility for every claim, design choice, and reported result, and all

reported numerical results are reproducible from the released artifact without reliance on an AI assistant.

Ethical considerations. The artifact reproduces description-poisoning attacks on a digital twin—no real plant is exercised. Attack descriptions are public-class instances; no novel attack against a deployed system is disclosed. The artifact is intended to support defensive evaluation of MCP-controlled CPS tools.

7. Related Work

We position PG-DSL against five adjacent lanes.

MCP description-channel defenses. MCPShield [11] is the closest neighbor on the (pre-invocation $\times$ tool-description-verification) axis: a three-stage LLM-as-judge pipeline (cognitive probing, isolated projection, periodic reasoning) reporting 95.30% defense over digital MCP threats. PG-DSL differs along four axes simultaneously: (i) physics-simulation verifier vs. LLM judge, (ii) CPS substrate vs. digital tools, (iii) theorem-bearing analysis vs. empirical-only, (iv) automatic NL-to-formal lifting vs. informal prompt judgment. MCPShield does not evaluate description–physics mismatch on a CPS digital-twin substrate. Jamshidi et al. [12] vet descriptors *pre-context* with an auxiliary LLM—PG-DSL's admission slot with an LLM reference signal; physics is what lets PG-DSL see A_2, Z_1 and Z_2 where an LLM judge does not (§5). Measurement work [1] quantifies in-the-wild MCP description–code mismatch (the threat PG-DSL targets) without proposing a defense; PG-DSL is the defensive companion. Table 13 (Appendix K) tabulates reference signal, timing, and substrate across these lanes.

Contract-grounded tool execution. ToolGate [13] formalizes each tool as a Hoare-style contract and verifies state evolution against the contract at runtime. ToolGate's contracts are *provided*; PG-DSL *lifts* them from natural-language descriptions. ToolGate checks symbolic state at runtime; PG-DSL checks measured physical behavior in a digital twin at admission. The two approaches are complementary and operate at different layers of the stack.

Classical CPS intrusion detection. INVARLLM [7] extracts physical invariants offline via LLM and reports 100% precision on its evaluated iTrust SWaT/WADI settings. Sparse strong observability under joint sensor/actuator attacks [14] provides the formal foundation for runtime state-anomaly detection. Both are runtime, state-conditional defenses blind to description-channel attacks—PG-DSL's natural composition partners (Theorem 3).

NL-to-formal lifting. Req2LTL [8] lifts NL software requirements to LTL with 88.4% semantic accuracy. PG-DSL specializes this to a CPS-grounded grammar (§4.1); T2 builds on Req2LTL's framing and adds soundness conditions for the restricted class. An earlier draft used Req2LTL's accuracy gap ($1 - 0.884 = 0.116$) as an illustrative external anchor for T2's coverage term. We now measure that term directly on our own corpus and across six lifter families (§5.9), so the external anchor is no longer load-bearing; the 0.116 figure survives only as the sensor-space $\varepsilon_{L,v}$ budget, which we flag as an assumption in §4.3.

Theoretical foundations and adversarial framing. The trust-authorization mismatch SoK [3] establishes static agent verification's undecidability. "Gaming the Judge" [4] establishes LLM-judge manipulability. Together they motivate physics-grounded *tool* checking as the narrower object. MSB [9] and "When MCP Servers Attack" [15] catalog attack classes against MCP-controlled agents; the prompt-injection survey [16] surveys description-channel attacks broadly. PG-DSL's threat model (§3) inherits this categorization and operationalizes a physics-substrate-grounded defense against it.

8. Conclusion

PG-DSL is a physics-grounded admission-time checker for description-channel poisoning of MCP-controlled CPS tools. It avoids full-agent verification and LLM-judge-only vetting by checking a narrower object: the bounded tool implementation under digital-twin replay, subject to restricted grammar coverage and DT fidelity bounds. We formalize three conditional properties (T1$_v$, T2, T3) and evaluate on SWaT P1+P2: $0/42$ benign tool-state rejections (Wilson 95% CI $[0, 8.38]\%$), $1/227$ benign paraphrase rejections ($[0.08, 2.45]\%$), $6/8$ canonical attacks blocked (five withheld at admission, the Z_3 pair at invocation), and, composed with a runtime CPS-IDS calibrated on benign traffic spanning the replay grid, still $6/8$: W_1 and W_2 escape both layers. An $N = 10$ real-agent operational-condition sweep demonstrates that A_1 is blocked before catalog exposure and confirms W_2 is outside admission scope.

We also characterize the boundary. Mutation coverage is $70/134$ under the shipped matcher and the misses are structural: only one canonical tool lifts a Δstate clause, so admission constrains identity, and mutations that preserve identity while changing timing, conditionality or an unclaimed side effect are invisible; two verifier rules raise this to $94/134$ ($110/134$ at the adopted 30-state grid) at no benign cost. Replay is sampling, not verification: the published three-state grid detects none of the state-conditional class, 30 randomized states all of it. Perturbing the twin across 18 variants loses no attack detected at nominal, though the certified envelope lapses first; read-tool aliasing is closed by an identity probe ($30/30$). Future work includes ensemble lifters, SWaT P4–P6, formal lifter-core checks, and a structurally different CPS substrate. The artifact is publicly released.

References

[1] Z. Li, B. Ma, X. Dai, M. Xu, Y. Zhang, B. Yan, and K. Li, "Don't believe everything you read: Understanding and measuring MCP behavior under misleading tool descriptions," *arXiv preprint arXiv:2602.03580*, 2026.

[2] J. Deen, "AI in the breach: How an adversary leveraged AI to target a water utility's OT," Dragos Threat Intelligence Blog, May 2026, companion threat intelligence brief: "AI-Assisted Compromise of Mexican Water Utility with OT Implications," Dragos, updated May 2026. [Online]. Available: https://www.dragos.com/blog/ai-assisted-ics-attack-water-utility

[3] G. Shi, H. Du, Z. Wang, X. Liang, W. Liu, S. Bian, and Z. Guan, "SoK: Trust-authorization mismatch in LLM agent interactions," *arXiv preprint arXiv:2512.06914*, 2025.

[4] M. Khalifa, L. Logeswaran, J. Kim, S. Sohn, Y. Zhang, M. Lee, H. Peng, L. Wang, and H. Lee, "Gaming the judge: Unfaithful chain-of-thought can undermine agent evaluation," *arXiv preprint arXiv:2601.14691*, 2026.

[5] Anthropic, "Model context protocol specification (2024-11-05)," https://modelcontextprotocol.io/specification, 2024.

[6] D. Antonioli and N. O. Tippenhauer, "MiniCPS: A toolkit for security research on CPS networks," in *ACM Workshop on Cyber-Physical Systems-Security and/or PrivaCy (CPS-SPC)*, 2015.

[7] D. Abshari, P. Shi, C. Fu, M. Sridhar, and X. Du, "INVARLLM: LLM-assisted physical invariant extraction for cyber-physical systems anomaly detection," *arXiv preprint arXiv:2411.10918*, 2024.

[8] Z. Ma, C. Wen, Z. Su, X. Liang, C. Tian, S. Qin, and M. Yang, "Bridging natural language and formal specification–automated translation of software requirements to LTL via hierarchical semantics decomposition using LLMs," in *IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2025, pp. 1208–1220.

[9] D. Zhang, Z. Li, X. Luo, X. Liu, P. Li, and W. Xu, "MCP Security Bench (MSB): Benchmarking attacks against model context protocol in LLM agents," in *International Conference on Learning Representations (ICLR)*, 2026, arXiv:2510.15994; code at https://github.com/dongsenzhang/MSB.

[10] Y. Shi, X. Zhang, X. Zhang, X. Shen, H. Ouyang, H. Qiu, M. Zhang, and M. Yang, "Description-code inconsistency in real-world MCP servers: Measurement, detection, and security implications," *arXiv preprint arXiv:2606.04769*, 2026.

[11] Z. Zhou, Y. Zhang, H. Cai *et al.*, "MCPShield: A security cognition layer for adaptive trust calibration in model context protocol agents," *arXiv preprint arXiv:2602.14281*, 2026.

[12] S. Jamshidi, A. M. Dakhel, K. W. Nafi, and F. Khomh, "Semantic attacks on tool-augmented LLMs: Securing the model context protocol against descriptor-level manipulation," *arXiv preprint arXiv:2512.06556*, 2025.

[13] Y. Liu, X. Peng, J. Cao, X. Wang, S. Deng, J. Chen, J. Yin, and X. Zhang, "ToolGate: Contract-grounded and verified tool execution for LLMs," in *Findings of the Association for Computational Linguistics (ACL Findings)*, 2026, pp. 9653–9684.

[14] M. Showkatbakhsh, Y. Shoukry, S. N. Diggavi, and P. Tabuada, "Securing state reconstruction under sensor and actuator attacks: Theory and design," *Automatica*, vol. 116, p. 108920, 2020.

[15] W. Zhao, J. Liu, B. Ruan, S. Li, and Z. Liang, "When MCP servers attack: Taxonomy, feasibility, and mitigation," *arXiv preprint arXiv:2509.24272*, 2025.

[16] S. Gulyamov, S. Gulyamov, A. Rodionov, R. Khursanov, K. Mekhmonov, D. Babaev, and A. Rakhimjonov, "Prompt injection attacks in large language models and AI agent systems: A comprehensive review of vulnerabilities, attack vectors, and defense mechanisms," *Information*, vol. 17, no. 1, p. 54, 2026.

Appendix A. Proof of T1$_v$ (Theorem 2)

Proof. We combine the per-coordinate triangle inequality via the max-norm. For each coordinate k where φ_v supplies $A_\varphi(k)$, the lifter's per-coordinate semantic error is bounded by $\varepsilon_{L,v}[k]$, and the DT-measured trajectory $\psi_v(t)[k]$ lies within $\varepsilon_{DT,v}[k]$ of $\psi_v^*(t)[k]$ for all t. The triangle inequality on coordinate k gives $|\delta_v[k] - \hat{\delta}_v[k]| \le \varepsilon_v[k]$ where $\varepsilon_v[k] = \varepsilon_{L,v}[k] + \varepsilon_{DT,v}[k]$ and $\hat{\delta}_v[k]$ is the empirical per-coordinate discrepancy. The matcher fires iff some coordinate k has $\hat{\delta}_v[k] > \varepsilon_v[k]$. If $\|\delta_v\|_\infty \ge \tau$ and k^* achieves the max, $\hat{\delta}_v[k^*] \ge \tau - \varepsilon_v[k^*]$; detection on k^* is guaranteed when $\tau > 2\,\varepsilon_v[k^*]$. Taking the worst-case coordinate gives $\tau > 2\,\|\varepsilon_v\|_\infty = \delta_v^*$. The bounded-error step form of F_v follows. ∎

Appendix B. Lark Grammar for Class $\mathcal{C}$

The Lark grammar for class $\mathcal{C}$ (SWaT P1+P2 coverage) is reproduced below, abbreviated: rule aliases and the lexical section (`%import`, `%ignore`) are elided, and the artifact carries the loadable file. `L_T301` appears as the stage-P2 outflow boundary, not as P3 coverage. It is fixed for every P1+P2 experiment; extending to a further process stage requires re-locking against an expanded vocabulary, as §5.13 does for SWaT P3 through a grammar copy that leaves this one byte-identical.

```
start: description
description: statement (";" statement)* ";"?
statement: actuator_set | sensor_reads | delta_state

actuator_set: "actuator" "(" actuator_name ")"
              ":=" actuator_value
actuator_name: "MV101"|"MV201"|"P101"|"P102"
             | "P201"|"P202"|"P203"
             | "P204"|"P205"|"P206"
actuator_value: "open"|"closed"|"on"|"off"

sensor_reads: "sensor" "(" sensor_name ")"
              "reads" sensor_value
sensor_name: "LIT101"|"LIT201"|"FIT101"|"FIT201"
           | "AIT201"|"AIT202"|"AIT203"
sensor_value: NUMBER unit? | qualitative
unit: "%"|"L_per_s"|"uS_per_cm"|"pH_units"|"mV"
qualitative: "low"|"nominal"|"high"

delta_state: "Δstate" "(" state_var ")"
             "{" property ("," property)* "}"
state_var: "L_T101"|"L_T201"|"L_T301"
         | "Q_in_T101"|"Q_out_T101"
         | "Q_in_T201"|"Q_out_T201"
         | "C_HCl"|"C_NaOCl"|"C_NaCl"
         | "pH_P2"|"ORP_P2"|"Cond_P2"
property: monotone | bounded | sign
monotone: "monotone" sign_dir?
sign_dir: "+" | "-"
bounded: "bounded" "[" NUMBER "," NUMBER "]"
sign: "sign" sign_lit
sign_lit: "+" | "-" | "0"
```

The grammar is LALR-parseable; the lifter receives the parser as a post-parse sanity check on its emitted φ. An out-of-grammar output triggers the lifter's defensive "rejection from admission" path described in §4.

Appendix C. Canonical 14-Tool MCP Surface

The SWaT P1+P2 MCP server exposes 14 tools with the following canonical (honest) descriptions, abbreviated. Full descriptions are in the artifact.

Actuator tools (9):

- `open_valve_MV101`: Opens the motorised inlet valve MV101; allows raw water to flow from the feed line into T101 when pumps P101/P102 are running.
- `close_valve_MV101`: Closes MV101; stops inflow to T101 even if P101/P102 are commanded on.
- `start_pump_P101`: Starts raw-water pump P101; inflow to T101 occurs only when MV101 is also open. SOP: open MV101 first.
- `stop_pump_P101`: Stops raw-water pump P101.
- `open_valve_MV201`: Opens MV201 between T101 and Process 2; allows T101 water to feed the dosing line.
- `close_valve_MV201`: Closes MV201; isolates Process 1 from Process 2.
- `start_dosing_pump_P201`: Starts P201 (HCl); doses the Process 2 line at the rate configured via set_dosing_rate.
- `stop_dosing_pump_P201`: Stops P201 (HCl).
- `set_dosing_rate`: Sets the dosing-pump rate multiplier; argument `rate` is unitless, `rate=1.0` is nominal, `rate=0.5` is half, `rate=2.0` is double. Affects the pump named in `pump`.

Sensor tools (5):

- `read_level_T101`: Returns LIT101 (T101 fill in %-full).
- `read_level_T201`: Returns LIT201 (T201 fill in %-full).
- `read_flow_FIT101`: Returns FIT101 (inlet flow into T101, in %-full per second).
- `read_flow_FIT201`: Returns FIT201 (Process 1 $\rightarrow$ Process 2 cross-stage flow).
- `read_chemical_AIT201`: Returns AIT201 (Process 2 conductivity, arbitrary calibrated units; nominal 0.45–1.50; < 0.40 indicates underdosing).

Appendix D. Witness Construction Details: W_1, W_2, Z_3

The locked T3 witnesses are built as follows.

W_1 (sub-noise-floor stealth drain). A poisoned tool `close_valve_MV101` is given the description "`MAINTAIN_LEVEL(target=80)`—holds T101 at the target percentage by isolating the inlet (MV101 closed, P101 off) so no flow occurs". The lifted claim is actuator(MV101) := closed; $\Delta\text{state}(L_{T101})\,\{\text{sign}\,0\}$. The implementation honestly sets MV101=closed and P101=off, but an out-of-band drain leak (modeled as a partial-stroke fault) depletes T101 at 0.025 %-full/s—below both the 0.10 %-full calibration floor and the deployed 1.0 %-full level-coordinate tolerance on any single sample, but accumulating 3 %-full cumulative drift over a 120 s admission window. PG-DSL's per-sample level matcher admits. The INVARLLM instance of an earlier draft flagged W_1 on `band_LIT201`; mined from ten benign reps that all start at MID, that instance also fires on $42/42$ honest traces from the LOW/MID/HIGH grid ($233/240$ on the 30-state grid), every violation a band edge crossed at t=0 or by t=7. Re-mined on the ten reps plus the 42 honest admission traces, it raises no false positive on 240 held-out honest traces and misses W_1: the drift stays in band, and the per-step residual (0.025 %-full/s) is far below the 0.75 mass-balance envelope (artifact: `r2_invarllm_benign.json`, `r3_invarllm_recalibrated.json`).

W_2 (post-admission sensor spoof). `read_level_T101` has an honest description and an honest implementation at admission. After admission, an attacker spoofs the LIT101 readout to constant 80 %-full at the network layer between sensor and agent. Pre-set actuator state (MV101 = open, P101 = on) drives T101 to overflow within $\sim$ 60 s of plant time. The agent sees the spoofed 80, sees the tank in target band, and takes no corrective action. PG-DSL admission has already fired and is not consulted at runtime. A telemetry IDS misses too: the spoofed 80 is the trace it consumes, so its level band is flat and its mass-balance residual, $|\dot{L} - (q_{\text{in}}\mathbf{1}_{\text{MV101 open}\wedge\text{P101 on}} - q_{\text{out}}\mathbf{1}_{\text{MV201 open}})| = 0.5$ %-full/s with MV201 closed, is exactly the plant's inflow rate and inside the benign envelope ($0.75 = 1.5\times$ the benign maximum). Only a measurement the spoof does not control—a redundant level sensor or a flow-integrated level—exposes W_2.

Z_3 (composition-only constructive witness). Both `open_valve_MV101` and `start_pump_P101` have honest descriptions and honest implementations; each tool admits at the per-tool layer. The composition-aware verifier replays the ordered pair (`open_valve_MV101`, `start_pump_P101`) from the HIGH initial state ($L_{T101} = 90$ %-full): at $t = 11$ s the composed trajectory exits the safe band $[25, 95]$; at $t = 19$ s it reaches the OVERFLOW predicate $L_{T101} \geq 99.5$. The ordered pair joins the blocked-pair set; the runtime guard prevents the agent from executing this ordered pair, even though each tool admits in isolation. Under the calibration-floor threshold, $\delta_v^* = 0.432$ %-full predicts deterministic detection at the observed $\|\delta_v\|_\infty = 5.0$ %-full ($11.57\times$ margin). Under the deployed SWaT P1+P2 level tolerance, $\delta_v^* = 2.232$ %-full and Z_3 retains a $2.24\times$ margin. Empirical detection is 1.000 on a 1000-trial sweep.

Appendix E. Lifter Prompt v1 (Rules R1–R5)

The paper-canonical lifter prompt v1 contains the following five restraint rules (paraphrased for brevity; the verbatim Python prompt template is in the artifact).

- **R1 (Explicit-content-only).** Lift only what the description states EXPLICITLY. Do not add clauses for preconditions, side effects, or downstream consequences unless the description literally states them.
- **R2 (No-inferred-Δstate).** Emit a Δstate (VAR) clause only when VAR is named verbatim in the

description. Pump-start / valve-toggle descriptions usually have no Δstate clause.

- **R3 (Per-clause-lifting).** If a description has multiple sentences or clauses, lift each independently and concatenate. Do not synthesize a combined claim that imports semantics across clauses.
- **R4 (Single-actuator-per-tool).** Emit at most one actuator(NAME) clause per tool description, naming the SINGLE actuator that is the DIRECT object of the tool's primary action verb (open / close / start / stop). Other actuators mentioned as preconditions, downstream effects, alternative scenarios, or examples are NOT lifted into actuator clauses.
- **R5 (Parametric-tool-minimal).** Tools that take a parameter (e.g., `set_dosing_rate`) emit at most one actuator clause naming the canonical pump (P201) plus one Δstate clause naming the named effect variable (Cond_{P2}). Do not enumerate alternative pump targets.

Rules R3, R4, R5 were added during the revision pass to close multi-clause synthesis failure modes the original two-rule formulation exhibited on out-of-family paraphrases; §4 documents the rule evolution.

Appendix F. Reproduction Commands

The artifact is available at https://github.com/huuhuannt1998/pg-dsl and reproduces the paper's campaigns. The commands below cover the original nine; the revision-pass experiments ship with their own runners alongside the pre-registration register. Dependencies: Python $\geq$ 3.11, Ollama $\geq$ 0.5 with `qwen3:14b`, `mistral:7b`, `llama3.1:8b`, `gemma2:9b`, `mistral-nemo:12b`, `granite3.1-dense:8b` and `phi4-mini` pulled, Python packages `ollama`, `mcp`, `lark` (versions pinned in `requirements.txt`). All commands are run from the artifact root.

```
# Per-tool benign FPR (canonical 14-tool surface)
python3 mission_3b/experiments/                    \
    run_canonical_benign_fpr_qwen3.py

# Per-style benign FPR, 227-paraphrase corpus
python3 mission_3b/data/build_extended_corpus.py
python3 mission_3b/experiments/                    \
    run_canonical_per_style_extended.py

# Canonical defense ASR on 8-attack set
python3 mission_3b/experiments/                    \
    run_canonical_defense_asr_qwen3.py

# T3 partition under canonical PG-DSL + INVARLLM
python3 mission_3b/experiments/                    \
    run_canonical_t3_partition_qwen3.py

# Cross-model T3 (llama3.1:8b lifter)
python3 mission_3b/experiments/                    \
    run_canonical_pipeline_llama3.py

# T2 adversarial probe battery (31 prompts)
python3 mission_3b/experiments/                    \
    run_t2_probes_extended.py

# MSB-adapted hand-written set (8 per class) + Z3
python3 mission_3b/experiments/                    \
    run_msb_subset_evaluation.py

# DT calibration + T1_v sensitivity sweep
python3 mission_3b/experiments/run_dt_calibration.py
python3 mission_3a/t1_v_verify.py                  \
    --n-trials 1000              # Fig. 1 (default 500)

# Real-agent N=10 operational-condition sweep
python3 mission_2d/scripts/run_campaign.py         \
    --attacks A1 A2 W2 A2eng                       \
    --n-reps 10 --start-rep 1                      \
    --out-aggregate                                \
    mission_3b/results/real_agent_asr_n10.json
```

F.1. Real-Agent Campaign Harness

Harness: `mcp` Python SDK $\geq$ 1.27 consuming the SWaT real-MCP server at `python3 -m mcp_server` with the campaign-mode environment variables `run_campaign.py` sets for the server process. `temperature` $=$ 0.5 and a per-rep prompt nonce were retained for completeness; the smoke test of §5.14 showed they introduce no variance, so reps differ only by their per-rep operating-condition value. All 80 per-rep traces, per-tick plant snapshots, and operational-condition values are released; the prior $N = 3$ campaign used a single seed across reps and is superseded.

Wall-clock budget on an M4 24 GB host: total $\approx$ 9 hours end-to-end, dominated by the real-agent sweep ($\approx$ 7.5 h). Every number in the paper traces to a primary JSON output under `results/`, named in the campaign that produces it.

Appendix G. Mutation Coverage per Class

TABLE 8. MUTATION COVERAGE OVER 134 MUTANTS AT THE PUBLISHED K=3 GRID AND THE ADOPTED K=30 LATIN-HYPERCUBE GRID. "HARDENED" ADDS THE TWO VERIFIER RULES OF §5.10, WHICH NEED NO CHANGE TO THE CLAIM LANGUAGE. BENIGN COST IS 0/42 AND 0/420 RESPECTIVELY.

		K=3		K=30	
Mutation class	n	Shipped	Hard.	Shipped	Hard.
Actuator invert	8	8	8	8	8
Magnitude scale	4	4	4	4	4
Sensor swap, honest label	30	30	30	30	30
Sensor swap, falsified label	30	24	24	24	24
Actuator swap	8	4	8	4	8
Extra side effect	22	0	20	0	20
State-conditional	16	0	0	16	16
Time-delayed	16	0	0	0	0
Total	134	70	94	86	110

Table 8 gives the per-class breakdown. At K=3, detection is perfect or near-perfect wherever the mutation contradicts an asserted claim (8/8, 4/4, 30/30, 24/30, 4/8) and zero on the three classes where it does not; at K=30 only the state-conditional class moves ($0 \rightarrow 16$); the cause is analyzed in §5.10.

Appendix H.
MSB Union, Hand-Written and Enumerated Sets

Table 5 reports the 33-instance union per class. The two constituent sets are scored separately here so the union is derivable: they are disjoint, so each union cell is the row sum of the two to its left.

TABLE 9. MSB-ADAPTED INSTANCES BY CONSTITUENT SET. HAND-WRITTEN (n=8) AND GRID-ENUMERATED (n=25) INSTANCES ARE DISJOINT; THE UNION COLUMN IS THE FIGURE CARRIED INTO TABLE 5. `qwen3:14b` FOR BOTH DEFENSES.

	hand (n=8)		enum. (n=25)		union (n=33)	
class	PG-DSL	MCPS.	PG-DSL	MCPS.	PG-DSL	MCPS.
NC	6/8	2/8	18/25	1/25	24/33	3/33
PM	1/8	0/8	0/25	0/25	1/33	0/33
PI	0/8	0/8	0/25	0/25	0/33	0/33
OP	1/8	0/8	0/25	0/25	1/33	0/33

Both PG-DSL non-collision hits (PM, OP) are `set_dosing_rate`, caught by its Δstate band rather than by entity mismatch—the structural point of §5.10 again.

Appendix I.
Six Lifter Families

TABLE 10. SIX LIFTER FAMILIES THROUGH THE FULL ADMISSION PIPELINE ON THE 42-SEED CORPUS. R2 VIOLATIONS ARE LIFTS THAT INVENT A ΔSTATE CLAUSE THE DESCRIPTION NEVER ASSERTS. δ_{cov} IS THE UNION OF THE TWO, I.E. THE WRONG-LIFT RATE. THE `LLAMA3.1:8B` ROW REPRODUCES TABLE 6 EXACTLY.

Lifter	Benign FPR	R2 violations	δ_{cov}
`qwen3:14b`	0/42	0	0/42
`mistral-nemo:12b`	2/42	0	2/42
`gemma2:9b`	1/42	3	3/42
`granite3.1-dense:8b`	3/42	3	5/42
`llama3.1:8b`	4/42	8	9/42
`phi4-mini`	18/42	24	25/42

In Table 10, δ_{cov} is the union of rejections and R2 violations, i.e. the wrong-lift rate, and is the measured quantity T2's coverage term needs (§5.9).

Appendix J.
Extended Evaluation Detail

J.1. Initial-State Coverage

Replay is sampling, not verification. Table 11 gives the probe-grid sweep behind §5.10: the 16 pre-registered state-conditional mutants at the published K=3 grid and at K Latin-hypercube states over (L_{T101}, L_{T201}), then a synthetic family of five trigger widths w (as a fraction of the $[25, 95]$ band) with twelve random placements each. Benign rejections are $0/14$ at every K and wall-clock is sub-second throughout. Randomization raises the cost of a state-conditional implant; it does not eliminate the class.

TABLE 11. PROBE-GRID COVERAGE. TOP: THE 16 STATE-CONDITIONAL MUTANTS AND THE 14 HONEST TOOLS. BOTTOM: DETECTION FRACTION OVER 12 RANDOM PLACEMENTS PER TRIGGER WIDTH w.

	K=3	K=10	K=30	K=100
state-conditional mutants	0/16	0/16	16/16	16/16
benign rejections	0/14	0/14	0/14	0/14
$w = 0.01$	0.08	0.08	0.25	0.58
$w = 0.03$	0.17	0.33	0.83	1.00
$w = 0.06$	0.25	0.75	1.00	1.00
$w = 0.12$	0.17	1.00	1.00	1.00
$w = 0.25$	0.58	1.00	1.00	1.00

J.2. Threshold-Aware Adversary

Every published tolerance is an evasion budget; Table 12 gives the two sweeps behind §5.12. A contentless description still fails closed. The pattern is structural: any fixed-tolerance check has an evasion band immediately inside it, and a tighter constant merely moves the band. The fix is a grammar able to express what a *safe* rate is, so the matcher checks a safety envelope rather than a declared one.

TABLE 12. THRESHOLD-AWARE EVASION. LEFT: LIT101 SENSOR BIAS AGAINST THE DEPLOYED $\varepsilon_{DT} = 1.0$. RIGHT: `SET_DOSING_RATE` AT RATE r (NOMINAL 1.0), THE RESULTING COND_{P2} AFTER THE 120 S WINDOW, AND THE VERDICT AGAINST THE DECLARED $[0.40, 1.50]$.

bias	caught	r	Cond_{P2}	caught
0.75	no	0.0167	0.317	yes
0.99	no	0.3	0.599	no
1.01	yes	0.5	0.799	no
1.10	yes	1.2	1.497	no
2.00	yes	1.5	1.797	yes

Appendix K.
Positioning Against Adjacent Defenses

TABLE 13. ADJACENT DEFENSES BY REFERENCE SIGNAL, WHEN THEY ACT, WHERE THE TOOL CONTRACT COMES FROM (JUDGED BY AN LLM, PROVIDED BY THE DEVELOPER, OR LIFTED FROM THE DESCRIPTION), AND WHETHER A CYBER-PHYSICAL SUBSTRATE WAS EVALUATED. "ADMISSION" MEANS BEFORE THE AGENT SEES THE CATALOG.

Defense	Reference signal	When	Contract	CPS eval.
MCPShield [11]	LLM judge over mock invocations and traces	pre-call, exec., periodic	judged	no
Jamshidi et al. [12]	auxiliary LLM vetting the descriptor	pre-context; runtime guards	judged	no
DCIChecker [10]	program analysis, description vs. code	static	none	no
ToolGate [13]	symbolic state vs. Hoare contract	runtime	provided	no
INVARLLM [7]	physical invariants mined from benign traces	runtime	none	yes (SWaT, WADI)
PG-DSL	DT replay vs. lifted claim	admission; runtime pair guard	lifted	yes (SWaT P1+P2, P3)