GREEN BECAUSE THEY LEAKED: TEST-GATED POLICY UPDATES DON'T CERTIFY SAFETY

Anonymous authors
Paper under double-blind review

ABSTRACT

Automated judges—compiled rule systems, fine-tuned classifiers, or prompted language models—increasingly gate what AI agents are allowed to do. When the policy changes, the judge must be updated, and the standard safeguard is borrowed from continuous integration: admit an update only if it passes a suite of test cases. We show that this green checkmark means far less than it appears to. *First, the standard way of measuring the safeguard leaks.* When gating tests and evaluation cases share underlying evidence states, the gate is credited with catching errors on cases it has already screened: under such a protocol, our test-gated updater appears to buy monotone safety, reaching 100% error catch at large test budgets; under a corrected state-disjoint protocol, error catch saturates at 21% from budget one, and we explicitly retract the earlier result. *Second, with leakage removed, safety is bounded by coverage, not budget.* The effective budget is the number of unique evidence states that exist—about three per revision on our held-out split, zero for a third of revisions—and every admitted erroneous update lies on a criterion with no test coverage; neither test quality, gate strictness, nor prompted model scale repairs the gap. We then address both problems: the leakage, by the state-disjoint protocol itself, released with validators; the coverage, by a proven converse to our bound—checking the candidate against the archived old rule on the full typed state space outside the amendment's scope certifies zero protected-side errors, with no labels required—realizable by schema-based test generation, whose empirical validation is left open. Until then, inventory-drawn test suites make update gating an auditable *screen*, not a safety certificate. We release the benchmark, partitions, and audits.

1 INTRODUCTION

Suppose a customer-service agent may refund a purchase within thirty days, and an automated judge enforces this rule by approving or rejecting each refund action. The policy changes: fourteen days for one product class, while a defect exception remains. Updating the judge sounds like a prompt edit. In practice it carries a harder contract. A case at twenty days should now be denied; a case at ten days should not change; the defect exception and unrelated identity checks must remain intact; and if the amendment is withdrawn, the old judge should be recoverable exactly.

Such judges are becoming load-bearing infrastructure: they gate agent actions against policy, and the policies they encode change routinely, making the *update* the steady-state operating mode rather than a rare maintenance event. Yet the failure mode of a bad update is silent: a judge that breaks on protected cases does not error out—it confidently returns wrong verdicts—and aggregate accuracy cannot even detect the problem: a judge frozen at the *old* policy scores 90.28% on our post-amendment operator split while adopting none of the amendment (§2). The relevant object is the *transition* between judge versions—which verdicts move, which stay fixed, and whether the old version remains recoverable—not a snapshot of the new one. In model-editing terms, an update needs *edit success* on the truly affected cases and *locality* on everything else (Yao et al., 2023; Meng et al., 2023).

The industry-default safeguard for shipping such updates is the continuous-integration instinct: propose the update (here, with a language model that compiles the amended clause into a typed rule), then admit it only if it has valid structure and passes a suite of concrete test cases with known expected verdicts. This is regression-gated acceptance, in the spirit of behavioral test suites for NLP

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.

models (Ribeiro et al., 2020). This paper asks what that green checkmark is actually worth, and gives a two-part negative answer.

Finding 1: the standard way of measuring the safeguard leaks, and the leakage manufactures safety. An earlier version of this work evaluated the test gate at the level of rendered cases: removing a test case's ID from evaluation did not remove its underlying *evidence state*—the structured facts the rule actually reads (e.g., `booking_stage`)—which recurred under other surface renderings. Under that protocol, larger budgets appeared to buy monotone safety: 100% error catch and accepted precision at budget 32. Re-analyzed with gating tests and evaluation cases drawn from *disjoint* states—a grouped, leakage-free split at the state level—the same system shows no such trend: error catch saturates at 21.11% from budget one onward, and no metric improves above budget one (§4, Table 2); the earlier "catches" were largely the gate being re-examined on states it had already screened. We state this reversal as an explicit retraction. The mechanism is a close cousin of benchmark contamination—evaluation reusing what the system already saw—but it sits in the assurance pipeline rather than the training pipeline, a site current leakage audits do not examine.

Finding 2: with leakage removed, the binding constraint is coverage, and it is severe. The effective test budget is the number of unique evidence states available for a criterion, not the number requested. On our held-out operator split, requesting budgets up to 32 yields an effective 3.33 unique states per revision; 14 of 42 revisions have zero usable test states. Raising the budget from zero to one changes criterion-macro accuracy by $+0.28$ points (criterion-block bootstrap interval $[-1.07, +1.90]$) while lowering targeted-update recall by 4 points and sending 24.76% of proposals to review; budgets above one improve nothing (§5). The failure is not diffuse: on identical evaluation cases, newly incorrect protected flips are 38.89% and 26.67% on the two zero-coverage criteria and at most 1.27% everywhere else, and *every* admitted erroneous proposal on the locked split belongs to the single criterion with no test states. Errors live exactly where tests cannot reach. Ablations close the remaining escape routes—test quality, gate strictness, and prompted model scale (§6).

Together these findings support a simple thesis: *passing a finite test suite is screening, not certification*—a known lesson from test-guided program repair (Qi et al., 2015), transplanted here to policy-judge updating and made measurable. The gate yields auditable evidence about enumerated cases and no guarantee beyond them; and evaluation protocols that let test states reappear at evaluation time will systematically overstate what the gate provides.

Contributions. (i) An update contract and metric family for evaluating judge updates rather than judge snapshots: targeted-update recall (edit success), directionally decomposed collateral-flip rates (locality), no-op flips under meaning-preserving rewrites, deferral, and rollback replay, with explicit denominators and a structural comparability caveat (§2). (ii) A state-disjoint protocol that removes test–eval reuse by construction—each canonical evidence state is assigned exactly once to gating or evaluation, before budgets are chosen and independent of model predictions—under which all results here are computed, together with a leakage audit quantifying the inflation it eliminates (§4). (iii) The coverage-ceiling result and its localization, with matched-denominator baselines, criterion-block bootstrap uncertainty, and leave-one-criterion-out sensitivity (§5)—bracketed theoretically by a coverage bound and its certification converse (Propositions 1–2), which shows label-free locality certification is achievable from the archived old rule alone. (iv) Ablations on test quality, gate strictness, and model scale that eliminate the natural repairs (§6), plus a frozen, leakage-checked external candidate pool (CUAD) for future human-annotated evaluation (§3). All amendments, partitions, audits, and validation code are released; the claims are scoped to this constructed inventory, not to a policy population. Table 1 collects the paper's terminology in one place.

2 The Update Contract: Evaluating Transitions, Not Snapshots

Let J_{t-1} and J_t be judge versions before and after amendment Δ_t, producing binary verdicts. Typed semantics partition cases into the truly affected set $\mathcal{A}_t$ (whose correct verdict changes) and the protected set $\mathcal{U}_t$ (whose correct verdict does not). We report

$$\mathrm{TUR}_t = \Pr[J_t(x) = y_t(x) \,|\, x \in \mathcal{A}_t], \qquad \mathrm{CFR}_t = \Pr[J_t(x) \neq J_{t-1}(x) \,|\, x \in \mathcal{U}_t], \tag{1}$$

where y_t is the post-amendment label. TUR (targeted-update recall) is edit success; CFR (collateral-flip rate) is the locality-violation rate on protected inputs (Yao et al., 2023). The two are never combined: a frozen judge has perfect locality for the wrong reason—90.28% accuracy with zero

Table 1: **Terminology at a glance.** Formal definitions in §2 and Appendix A.

Term	Meaning
Criterion	One policy rule with a stable ID (e.g., `airline.insurance_timing`); the unit an amendment targets.
Amendment / revision	An explicit policy change to one criterion; *no-op* revisions are meaning-preserving rewrites (labels unchanged).
Evidence state	Canonical structured facts of a case (key-sorted JSON) that the rule reads; the semantic unit behind splits.
Rendering	One surface realization of an evidence state; many renderings share a state (median state: 126 renderings).
Gating test (*witness*)	An evidence state with known expected verdicts, used to screen a proposed rule before admission.
Budget b / effective b_{eff}	Requested vs. actually available number of unique gating states; $b_{\text{eff}} = \min(b, \lvert W_q \rvert)$.
Affected / protected	Cases whose correct verdict changes / must not change under the amendment; scope $\sigma(\Delta_t)$ is the affected region.
TUR	Targeted-update recall: accuracy on affected cases (edit success).
CFR^+ / CFR^-	Protected-case flips to correct / to incorrect (locality violations, split by direction).
No-op flip	Verdict change under a meaning-preserving rewrite (invariance failure).
Deferral	Proposal sent to an automatic review queue; the old rule is retained (abstention, not completed review).
Error catch / accepted precision	Share of empirically erroneous proposals rejected / of admitted proposals that are correct.
Operator / locked splits	Held-out criterion-ID splits with 3 / 6 IDs; "operator" is a legacy name (§3).

edit success on the operator split. Snapshot accuracy of J_t is structurally blind to update failure; the information lives in the pair (J_{t-1}, J_t) and the affected/protected partition.

Because a protected flip can repair an old mistake or create a new one, we decompose CFR on the same protected denominator:

$$\text{CFR}^+ = \Pr[J_{t-1}(x) \neq y_t(x),\ J_t(x) = y_t(x) \mid x \in \mathcal{U}_t]\,, \tag{2}$$

$$\text{CFR}^- = \Pr[J_{t-1}(x) = y_t(x),\ J_t(x) \neq y_t(x) \mid x \in \mathcal{U}_t]\,, \tag{3}$$

the *flip-to-correct* and *flip-to-incorrect* rates, with $\text{CFR} = \text{CFR}^+ + \text{CFR}^-$ for binary verdicts. One structural comparability caveat, flagged wherever mechanisms share a table: when the old-version prediction *is* the typed frozen old rule—as in our pipeline—J_{t-1} is correct on every protected case by construction, so $\text{CFR}^+ \equiv 0$ identically rather than empirically; learned judges instead predict both versions, so their CFR^+ largely records repair of their own old-version errors. CFR^+ and total CFR are thus not comparable across mechanism classes; CFR^- is.

Three further quantities complete the contract (Table 1): the *no-op flip rate*, a metamorphic invariance check (Ribeiro et al., 2020; Weng et al., 2026); *deferral*, abstention in selective-prediction terms (El-Yaniv & Wiener, 2010; Geifman & El-Yaniv, 2017); and *rollback*, checked by snapshot replay. Proposal-side rates use proposals P, accepted A, and empirically erroneous E (rule errs on some held-out evaluation state): accepted precision $(|A| - |A \cap E|)/|A|$, error catch $|E \setminus A|/|E|$, erroneous admission $|A \cap E|/|P|$ (conventions in Appendix A).

Why coverage bounds what a gate can certify. The following is elementary, but it organizes everything that follows. Let S_q be the set of canonical evidence states realizable for criterion q, and let $W_q \subseteq S_q$ be the states reserved for gating tests, disjoint from the evaluation states $S_q \setminus W_q$.

Proposition 1 (Coverage bound)**.** *A gate that accepts a candidate rule iff it is schema-valid and errs on none of b selected states from W_q constrains the candidate's behavior only on the selected states. Its effective budget is $b_{\text{eff}} = \min(b, |W_q|)$; requested budget beyond $|W_q|$ is vacuous. If $|W_q| = 0$, admission is independent of the candidate's correctness on every evaluation state, so among admitted proposals, errors face no screening on such criteria.*

The proposition predicts three signatures, all observed: metrics flat in requested budget beyond saturation (§5); admitted errors concentrated on zero-coverage criteria (Table 4); and no repair from test quality or strictness, which act only within W_q (§6). It also identifies the sense in which a passing gate is a *screen*: it certifies behavior on W_q and is silent about the rest—the plausibility/correctness gap of test-guided repair (Qi et al., 2015). The bound has an exact converse, which tells us what *would* close the gap:

Proposition 2 (Closed-world certification)**.** *Let Σ_q be the typed state space induced by the schema's finite field universes, r_{t-1} the archived old rule, and $\sigma(\Delta_t) \subseteq \Sigma_q$ the amendment's declared scope. A gate that admits candidate $\hat{r}_t$ only if (a) $\hat{r}_t(s) = r_{t-1}(s)$ for all $s \in \Sigma_q \setminus \sigma(\Delta_t)$ admits no candidate with a protected-side error:* $\text{CFR}^- = 0$ *on every typed evaluation state, with* no labeled tests required*, since on protected states the old rule's verdict is the expected verdict. If additionally (b) $\hat{r}_t(s) = y_t(s)$ for all $s \in \sigma(\Delta_t)$, then* $\text{TUR} = 1$ *as well.*

The proof is immediate (evaluation states lie in Σ_q; outside a correctly declared scope, $y_t = y_{t-1}$, which is the archived rule's verdict)—and that is the point: certification is blocked by no deep obstacle, only by drawing tests from an inventory instead of the typed space. Condition (a) is checkable by generation or enumeration and buys locality for free; (b) needs in-scope labels; both are conditional on a faithful schema and correct $\sigma(\Delta_t)$—the abstraction gap where residual risk concentrates. §5 shows the inventory-drawn gap bites hard; Appendix M gives algorithms for (a)–(b), whose empirical validation is future work.

3 System and Benchmark

3.1 PolicyPatch: an LLM-proposed, test-gated update pipeline

The evaluated pipeline receives an old/current clause pair together with a benchmark-supplied target criterion ID q. A small language model (the *compiler*) proposes a typed JSON rule $\hat{r}_t = C_\theta(c_{t-1}, c_t)$. Evaluation code then checks the proposal's schema and executes the gating tests (Table 1). A proposal is admitted only if it is schema-valid and makes zero errors on the selected test states; otherwise it is deferred to review and the typed old rule is retained, so a rejected update is a no-op. Verdicts themselves come from a deterministic typed-rule *executor*, not from a language model: the LLM proposes updates; the gate decides; compiled rules judge. Both versions are recorded, and rollback is checked by replay.

The path is deliberately closed-world, with its assists stated plainly: the benchmark supplies the target criterion ID (no jurisdiction discovery); the trained compiler receives a revision-specific schema derived from the reference rule (kind, keys, field/unit, finite value universes; target values undisclosed); the evidence extractor receives the exact per-case field-key inventory and predicts only values. A revision-independent contract with a global schema and explicit `ABSTAIN_UNREPRESENTABLE` option is exercised only in the prompted stress test of §6. Appendix L walks through one audited amendment.

3.2 PolicyDeltaBench: typed policy amendments with executable ground truth

We construct amendments over the public policy documents of τ^2-bench's airline, retail, and telecom domains (Yao et al., 2025; Barres et al., 2025), pinned to a fixed commit; we neither modify nor claim the official benchmark score. We type 15 criterion IDs per domain (45 total, drawn from 38 exact-normalized passages; five duplicate groups). Four semantic revisions per ID yield 180 threshold, exception, obligation, scope, priority, or disclosure changes; ten meaning-preserving rewrites per ID yield 450 no-op revisions. Each revision has 72 rendered cases spanning affected boundary examples and protected controls (4,394 affected / 40,966 protected renderings; 45,360 total). Labels come from typed execution, not an LLM judge.

The apparent scale must be qualified—it is the crux of the paper. Renderings are repeated measurements: a criterion ID has only 3–16 unique typed evidence states (median 8), each recurring across many renderings (Table 1). Criterion IDs split 27/6/6/3/3 into training, calibration, a locked test frozen before any analysis here, a three-ID held-out split (legacy-named *operator*), and a synthetic linguistic-shift split; IDs never cross splits, though one exact-normalized passage is shared between a training and a calibration ID. The operator split mixes rule kinds and does *not* test generalization to unseen edit operators; a claim to that effect in the earlier version of this work is withdrawn (§4). We aggregate by criterion ID and treat block resampling as descriptive uncertainty for this fixed inventory, not population inference.

For future work beyond constructed data, we freeze a leakage-checked pool of 475 CUAD contract clauses (Hendrycks et al., 2021) with a 75/400 split having *zero* overlap across text families (0.50 Jaccard), filers, documents, contexts, and clauses (438 review families; largest 5); an initial leaky split was replaced before any outcome existed. Amendments, tests, and labels remain open; we report no CUAD outcomes.

3.3 MODELS, BASELINES, AND COMPARATORS

All benchmark-trained systems use Qwen3-1.7B (Yang et al., 2025) with LoRA (Hu et al., 2022) (rank 16, lr 10^{-4}, BF16, seeds $\{17, 29, 43, 71, 101\}$; Appendix E). Two fine-tuned judge baselines predict verdicts directly from text and structured evidence: *Current-clause LoRA* (current clause only) and *Delta LoRA* (plus old clause and natural-language delta), trained on 5,512 balanced case labels. The patch compiler trains on 216 balanced revision targets. Deterministic endpoints bracket the range: the *frozen old rule* (no update) and a *typed oracle patch* (perfect update). Two prompted open-weight comparators are evaluated on the identical evaluation cases: Qwen3.6-35B-A3B-FP8 (sparse MoE, ≈3B active parameters) and gpt-oss-120b (OpenAI, 2025) (≈5.1B active, MXFP4), both pinned to hash-verified revisions with fully recorded serving contracts (Appendix F). Trained systems use greedy decoding or first-token logits; prompted comparators are sampled (five serial repeats). Systems are matched on evaluation denominators, not on supervision or compute. No frontier or API-served judge is evaluated: a provider-served judge cannot be pinned to an immutable revision for reproduction.

4 FINDING 1: TEST–EVAL STATE REUSE MANUFACTURES SAFETY

4.1 WHERE THE LEAK WAS

The earlier analysis operated on rendered cases: gating cases (*witnesses*, in the earlier version's terminology) and evaluation cases had distinct case IDs. But distinct renderings frequently share the same canonical evidence state—the serialization the executor actually reads—so removing a case ID from evaluation did not remove its state. Quantitatively, in the legacy fixed-budget frame, gating states covered 41.19% of evaluation states at budget 4, 85.22% at budget 16, and 98.43% at budget 32; the legacy locked diagnostic at budget 4 was fully state-overlapping (all 268 selected states and 336 gating renderings recur in evaluation). "Catching" an erroneous proposal then often meant re-testing it on states it had already been screened on—the assurance-pipeline analogue of evaluating a model on data it trained on.

4.2 THE STATE-DISJOINT PROTOCOL

The corrected protocol splits at the level of underlying states rather than surface items—the update-gating analogue of the grouped, deduplicated train–test splits used to prevent near-duplicate leakage. Each canonical evidence state within a criterion is assigned exactly *once* to the gating pool W_q or the evaluation pool—zero overlap, fixed across revisions, budgets, and seeds—conditioning only on state identity, revision grouping, and typed transition labels, never on model predictions, proposal correctness, or admission outcomes. Singleton states are forced into evaluation; the at-most-20 freely assignable states per criterion are placed by exhaustive search (maximize covered strata, avoid an empty gating pool, balance the split, stable hash tie-break). Within a revision, a requested budget b caps unique gating states selected by stratum-balanced round robin; evaluation uses one minimum-hash representative rendering per held-out revision–state pair. Appendix B gives the code-matched procedure; the released validator reports zero partition and selected-state overlap. This analysis was constructed post hoc, after the legacy results, and is descriptive rather than preregistered.

4.3 THE REVERSAL

Table 2 is the paper in miniature. Under the leaky protocol, the earlier manuscript's headline result—a monotone budget–safety tradeoff reaching 100% error catch and 100% accepted precision at budget 32—appears real. Under state disjointness, the identical system's error catch saturates at 21.11% from budget one, accepted precision *declines* with budget, and nothing improves above budget one. We retract the earlier claim explicitly. A second claim—that the three-ID split tested generalization "under held-out edit operators" at 97.67% accuracy—is also withdrawn: the split mixes rule kinds; its legacy name is kept for release continuity only.

We suspect the lesson generalizes beyond our own prior mistake: any test-gated update evaluation split at the level of surface items (renderings, paraphrases, templated cases) rather than underlying states risks the same inflation—largest at large budgets, where reuse approaches totality. We offer the

Table 2: **The leakage reversal.** Left: the earlier rendering-level sweep (state reuse permitted); right: the state-disjoint replacement, same system. The monotone budget→safety improvement does not survive removing reuse. Aggregation conventions differ as in their sources; the point is the direction reversal, not a cell-by-cell match.

	Rendering-level (leaky)		State-disjoint	
Quantity (%)	$b = 4$	$b = 32$	$b = 1$–4	$b = 8$–32
Current accuracy	97.83	99.40	87.66	87.66
Targeted-update recall	98.56	89.69	90.67	90.67
Deferral	33.81	53.33	24.76	25.24
Accepted precision	70.50	100.00	66.25	65.92
Error catch	62.39	**100.00**	**21.11**	21.11

Table 3: **State-disjoint budget sweep.** Outcome metrics use held-out revision–state representatives; proposal metrics use revision proposals; rates are criterion-ID macros averaged over five seeds. Effective budget: ID-macro mean of unique gating states selected per revision. Operator: 3 IDs / 42 revisions; Locked: 6 / 84; 14 revisions per split have zero capacity. $\mathrm{CFR}^{+} \equiv 0$ by construction (§2). Values describe this fixed inventory only.

		Test capacity	Held-out representatives (%)				Revision proposals (%)		
Split	Req. b	Eff. $\bar{b}$	Acc.	TUR	CFR^{-}	No-op	Defer	Acc. prec.	Err. catch
Operator	0	0.000	87.38	94.67	14.06	16.44	0.00	68.10	0.00
Operator	1–4	0.667–2.667	87.66	90.67	13.38	15.56	24.76	66.25	21.11
Operator	8–32	3.333	87.66	90.67	13.38	15.56	25.24	65.92	21.11
Locked	0–32	0–2.560	96.19	100.00	4.44	5.33	0.00	94.29	0.00

state-disjoint protocol as a leakage audit for such pipelines; the broader claim is a hypothesis inviting audits, not a demonstrated finding.

5 Finding 2: The Coverage Ceiling

5.1 Budgets buy nothing beyond one

Table 3 is the central empirical result, and it is negative. On the three-ID operator split, moving from b=0 to b=1 changes criterion-macro accuracy by $+0.28$ points (criterion-block bootstrap interval $[-1.07, +1.90]$; Table 6) while targeted-update recall falls 4.00 points and 24.76% of proposals are deferred. That is the entire safety purchase: no budget above one improves accuracy, TUR, CFR^{-}, or error catch, while accepted precision declines (68.10% at b=0 to 65.92% at $b\geq 8$) and deferral rises to 25.24%. The mechanism is exactly Proposition 1: at $b\geq 8$ all 42 revisions saturate at their unique-state capacity (effective mean 3.33), 14 of 42 have zero capacity, and only 2 of 12 semantic revisions have any affected-state capacity. Larger budgets are not more protective; they are infeasible.

5.2 Errors localize exactly where coverage is zero

Table 4 shows the failure is pinned, not diffuse. On identical evaluation representatives, PolicyPatch's flip-to-incorrect rate is 38.89% on the zero-coverage `telecom.roaming_scope` criterion and 26.67% on the zero-coverage `airline.insurance_timing` criterion, and at most

Table 4: **Localization: per-criterion flip-to-incorrect on identical representatives (%, seed means).** PolicyPatch's CFR^{-} concentrates entirely on the two zero-coverage criteria; fine-tuned judges spread smaller errors. Full table in Appendix G.

Split	Criterion (gating pool size)	PolicyPatch b=1	CC-LoRA	Delta-LoRA
Operator	airline.cancel_exceptions (6)	1.27	9.11	7.59
Operator	retail.gift_balance (4)	0.00	2.75	0.78
Operator	telecom.roaming_scope (**0**)	**38.89**	0.00	2.78
Locked	airline.insurance_timing (**0**)	**26.67**	7.22	10.56
Locked	five remaining criteria (max)	0.00	1.60	1.15

Table 5: **Strict same-representative comparison (%, criterion-ID macro).** All methods evaluated on the identical 182 operator (16 affected, 166 protected) or 337 locked (35/302) representatives; trained rows average five seeds, prompted rows five sampling repeats. $^{\dagger}\mathrm{CFR}^{+}$ identically zero by construction; CFR^{+}/total CFR are not comparable across mechanism classes, CFR^{-} is (§2). Supervision, targets, and compute are unmatched: a matched-denominator comparison, not a superiority test. Dispersions and prompted 1.7B controls in Appendix H.

Split	Method	Acc.	TUR	CFR^{+}	CFR^{-}	No-op	Defer
Operator	Typed frozen old rule	90.28	0.00	$0.00^{\dagger}$	0.00	0.00	N/A
	Current-clause LoRA	67.70	92.22	6.65	3.95	2.83	N/A
	Delta LoRA	69.84	98.67	6.70	3.72	6.94	N/A
	Qwen3.6-35B-A3B-FP8, prompted	75.79	79.11	9.12	2.60	5.28	N/A
	gpt-oss-120b, prompted	72.54	74.89	13.38	11.33	22.50	N/A
	POLICYPATCH, b=1	87.66	90.67	$0.00^{\dagger}$	13.38	15.56	24.76
	Typed oracle patch	100.00	100.00	$0.00^{\dagger}$	0.00	0.00	N/A
Locked	Typed frozen old rule	89.43	0.00	$0.00^{\dagger}$	0.00	0.00	N/A
	Current-clause LoRA	93.49	99.44	2.01	1.73	1.42	N/A
	Delta LoRA	93.38	99.44	1.84	2.19	3.03	N/A
	Qwen3.6-35B-A3B-FP8, prompted	79.17	90.56	9.07	3.14	8.34	N/A
	gpt-oss-120b, prompted	80.42	89.52	12.76	6.13	17.52	N/A
	POLICYPATCH, b=1	96.19	100.00	$0.00^{\dagger}$	4.44	5.33	0.00
	Typed oracle patch	100.00	100.00	$0.00^{\dagger}$	0.00	0.00	N/A

Table 6: **Criterion-block bootstrap (percentage points).** 10,000 criterion-ID resamples within split (3 operator / 6 locked IDs); contrasts paired on identical resampled multisets. Intervals are wide by construction—the width is itself a finding—descriptive for this fixed inventory, without multiplicity adjustment; not population confidence intervals.

Split	Contrast	Point	95% interval	Excl. 0
Operator	b=1 − b=0: accuracy	$+0.28$	$[-1.07, +1.90]$	no
Operator	b=1 − b=0: TUR	-4.00	$[-12.00, 0.00]$	no
Operator	POLICYPATCH − CC-LoRA: accuracy	$+19.96$	$[+14.76, +25.36]$	yes
Operator	POLICYPATCH − Qwen3.6-35B: accuracy	$+11.87$	$[+8.93, +13.81]$	yes
Operator	POLICYPATCH − Qwen3.6-35B: CFR^{-}	$+10.78$	$[-3.80, +38.89]$	no
Operator	POLICYPATCH − gpt-oss-120b: accuracy	$+15.12$	$[-0.48, +30.48]$	no
Locked	POLICYPATCH − CC-LoRA: accuracy	$+2.70$	$[+0.24, +7.14]$	yes
Locked	POLICYPATCH − Qwen3.6-35B: accuracy	$+17.02$	$[+8.81, +25.03]$	yes
Locked	POLICYPATCH − gpt-oss-120b: accuracy	$+15.77$	$[+6.13, +26.37]$	yes
Locked	POLICYPATCH − gpt-oss-120b: CFR^{-}	-1.69	$[-6.73, +6.17]$	no

1.27% on every criterion where any gating state exists. On the locked split, *every* empirically erroneous admitted proposal belongs to `airline.insurance_timing`—the one criterion where the gate cannot act—and all of them are admitted (erroneous admission 5.71% of proposals, split-level micro; conditional error catch 0%). Leave-one-out analysis confirms it: excluding that criterion leaves the remaining five locked IDs at 100% accuracy, 100% TUR, zero CFR^{-}, and zero deferral at every budget (Appendix G). Zero coverage explains the mechanism, not the outcome: on the same cases, the fine-tuned baselines score 0–9% CFR^{-}.

5.3 MATCHED-DENOMINATOR COMPARISON

Table 5 evaluates every method on identical representatives. Three readings matter. First, the frozen-judge row makes snapshot blindness numerical: 90.28% / 89.43% accuracy with zero edit success. Second, POLICYPATCH buys the highest accuracy and exact old-side behavior—at a cost stated in the unfavorable direction: on the comparable CFR^{-} column it is the *worst* operator-split method (13.38%; $1.2\times$ gpt-oss-120b, $3.6\times$ Delta LoRA, $5.1\times$ Qwen), with no offsetting CFR^{+}, fully concentrated on the zero-coverage criteria (Table 4): gating protects exactly where tests exist and is blind elsewhere. Third, scale does not rescue prompted judges: the larger gpt-oss-120b is weaker than the smaller Qwen comparator on five of six headline cells (§6).

Uncertainty uses a criterion-ID block bootstrap (Table 6): POLICYPATCH's accuracy advantages persist under resampling; the budget-one tradeoff and every adverse CFR^{-} gap do not exclude zero. With three operator and six locked IDs these intervals are wide by construction; we report them

because the width is itself what this inventory can and cannot certify. Operator seed SDs at $b{=}1$ are 1.99, 3.65, 2.36, and 2.13 points for accuracy, TUR, CFR^{-}, and deferral (Appendix H).

6 No Escape: Quality, Strictness, and Scale Do Not Repair Coverage

Noisier tests reject correct updates. If coverage is the constraint, test *quality* should not fix it. We corrupt gating tests deterministically (label flips, evidence-key omission, evidence-value mutation) at nominal rates up to 25% under a strict zero-mismatch gate at requested $b{=}16$. On the locked split the sweep is structurally non-discriminative: every erroneous proposal sits on the zero-coverage criterion, so there is nothing to corrupt where screening could act—erroneous admission stays at 5.71% and error catch at zero under *every* condition, a null entailed before any noise is applied. On the operator split, where clean error catch is nonzero, 25% label corruption raises catch from 11.60% to at most 21.87% (split micro)—but deferral more than doubles (25.24% $\rightarrow$ 55.14%), TUR falls from 94.25% to 72.07%, and accepted precision from 62.44% to 44.04%, while erroneous admission barely moves (28.10% $\rightarrow$ 24.81%). Corrupted tests reject more of everything; the marginal rejections are overwhelmingly *correct* proposals. Quality is no substitute for coverage on either split (full tables in Appendix I).

Maximal strictness over-rejects. Could a stricter gate at least reject what the schema cannot express? On an outcome-blind locked set of 15 paired clause clusters spanning five constructed out-of-schema families, the schema check alone rejects only $30.67 \pm 8.94\%$ of unrepresentable amendments (all explicit abstentions; the validator catches none), ranging from 100% (Boolean composition) to 0% (three of five families). Adding gating-test execution rejects *all* schema-admitted unrepresentable compilations at every tested budget—but the mechanism is type coverage, not semantic detection: every rejection is a fail-closed execution failure on missing fields, never a label disagreement. And the cost is severe: final false rejection on matched *representable* controls reaches $76.00 \pm 5.96\%$ (paired-cluster interval $[61.33, 92.00]$), invariant across budgets (the small pools saturate). An abstention-removed ablation shows roughly half of schema-stage rejection is hedging, not detection: forced compilation drops it from 30.67% to 14.67%, while end-to-end constructed-OOD admission stays 0% under both variants (Appendix J). Strictness trades one failure for another; it does not create coverage.

Bigger prompted models do not change the picture. On the identical representatives (Table 5), the $\approx$3B-active-parameter Qwen comparator reaches 79.17% / 75.79% accuracy (locked/operator); the larger $\approx$5.1B-active gpt-oss-120b is *weaker* on five of six headline cells (80.42% / 72.54%; 6.13% / 11.33% CFR^{-}), and the Qwen comparator admits 69.33% of constructed unrepresentable amendments in the schema stress test. These fixed-inventory prompted results under unmatched settings show only that neither comparator replaces explicit update and coverage checks; nothing here speaks to frontier systems.

7 Related Work

Judge reliability and rubrics. Evidence-grounded rubrics improve execution of fixed criteria (Hong et al., 2026); invariance tests expose sensitivity to equivalent wording (Weng et al., 2026); rubrics serve as measurement specifications (Roy et al., 2026) or co-evolve with the generator (Wang et al., 2026); NS-DPO addresses preference drift (Son et al., 2025). We study a complementary object: an *intentional, explicit* amendment, evaluated as a transition with protected non-change.

Model editing and continual learning. Model editing seeks local parameter changes with edit-success and locality metrics (Yao et al., 2023; Meng et al., 2023); continual fine-tuning forgets prior behavior (Luo et al., 2023) and can weaken alignment even under benign customization (Qi et al., 2024). Our update object is a typed, versioned rule rather than weights, making exact old-side behavior and rollback checkable; the edit/locality framework transfers (CFR^{+} caveat, §2).

Testing and selective prediction. Behavioral test suites brought software-testing discipline to NLP evaluation (Ribeiro et al., 2020); test-guided program repair showed that passing a finite suite does not imply correctness (Qi et al., 2015)—the lesson our gate inherits. Deferral is abstention in the

selective-prediction sense (El-Yaniv & Wiener, 2010; Geifman & El-Yaniv, 2017; Angelopoulos et al., 2024); Catala-style rules-as-code supplies the typed-rule substrate we assume (Merigoux et al., 2021); our reporting follows BetterBench (Reuel et al., 2024). An evaluator gating agent behavior is an institution (North, 1990), motivating separable rules, scope, adjudication, and versioning; we make no legal claim (Hart, 1994; Fuller, 1964).

8 LIMITATIONS

The evidence inventory is small and nested: 45 criterion IDs over 38 passages in three source documents; the locked and operator splits contain six and three IDs. Renderings, corruption seeds, and repeated passages are not new policy samples; all uncertainty is descriptive, and the state-disjoint analysis was constructed post hoc. The evaluated compiler is closed-world (revision-specific gold-shaped schemas; supplied target criterion ID and field-key inventory); the pipeline does not test jurisdiction discovery, field discovery, document differencing, or a complete version log, and the global-schema abstention contract is exercised only via a prompted stress test. Amendments, cases, and tests are synthetic, not professionally authored; deterministic corruption is not a model of human disagreement; the CUAD pool awaits human review. Comparators are two pinned open-weight sparse-MoE models under unmatched settings; no conclusion extends to frontier-class systems. Locality is conditional on a correct target criterion; tests can miss errors or be wrong; replay can reproduce a bad judge. Propositions 1–2 are elementary, inherited in principle from the testing literature (Qi et al., 2015); our contribution is showing the gap binds hard and that standard measurement hides it. These limits preclude claims about open-world policy understanding, legal use, or deployment safety.

9 CONCLUSION

When gating tests and evaluation cases share underlying states, a test-gated pipeline looks like it certifies safety; ours reached 100% error catch that way, and the number was an artifact. With states disjoint, the same pipeline hits a coverage ceiling—about three usable states per revision, a third of revisions with none—beyond which budget, quality, strictness, and scale buy nothing, and admitted errors concentrate where no test can reach. The checkmark was green because the evaluation leaked; what remains after fixing the leak is a screen, not a certificate.

Two practices follow: builders should treat a passing gate as auditable evidence about enumerated cases, report coverage (unique gating states; the zero-coverage set) alongside pass rates, and route zero-coverage amendments to human review; evaluators should split at semantic states, not surface items, and report the overlap removed.

Buying coverage. The ceiling binds because tests are *inventory-drawn*; Proposition 2 shows what closes it, and typed rules over finite universes make its conditions realizable by *generation* (Appendix M): *scope-differential testing* implements condition (a) with zero labels; *coverage-directed generation* targets predicate branches (Claessen & Hughes, 2000); *exhaustive checking* of small universes meets both. All three shift residual risk to the abstraction gap—schema fidelity, evidence extraction, trusted scope—which generation cannot address; validating them, with preregistered evaluation and the human-annotated CUAD track, is the next step.

ETHICS STATEMENT

This work evaluates update mechanisms for automated policy judges on constructed benchmarks; policy authorship, approval, appeal, privacy, and accountability remain human responsibilities. We make no deployment or legal-use claim, and caution against reading a passing gate as a safety certificate—the misreading documented here.

REPRODUCIBILITY STATEMENT

The supplement contains the benchmark, hashes, fixed partitions, code, tests, notebooks, and all summary files; raw predictions and CUAD text remain internal. All results use five named seeds and the released zero-overlap partitions (validation and caveats: Appendix N).

AI USE STATEMENT

In this work, we used LLM-based tools for code search and for copy editing of the manuscript, both tasks in ICLR's recommended-disclosure category. We did not use generative AI tools for any task in the required-disclosure category: all annotations, gating tests, labels, results, and statistics come from versioned scripts and frozen files, and no tool output supplied an annotation, test, or CUAD outcome (Appendix N). The language models inside the evaluated POLICYPATCH pipeline and the prompted comparators are objects of study, specified in §3.3, not research assistants. We reviewed all AI-assisted code and text, and we take responsibility for the final content of this work.

REFERENCES

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *International Conference on Learning Representations*, 2024.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*, 2025.

Koen Claessen and John Hughes. QuickCheck: A lightweight tool for random testing of Haskell programs. In *Proceedings of the Fifth ACM SIGPLAN International Conference on Functional Programming*, pp. 268–279, 2000.

Alejandro Cuadron, Pengfei Yu, Yang Liu, and Arpit Gupta. SABER: Small actions, big errors—safeguarding mutating steps in LLM agents. *arXiv preprint arXiv:2512.07850*, 2025.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11:1605–1641, 2010.

Lon L. Fuller. *The Morality of Law*. Yale University Press, 1964.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

H. L. A. Hart. *The Concept of Law*. Clarendon Press, 2 edition, 1994.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. CUAD: An expert-annotated NLP dataset for legal contract review. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Yihan Hong, Huaiyuan Yao, Bolin Shen, Wanpeng Xu, Hua Wei, and Yushun Dong. From rubrics to reliable scores: Evidence-grounded text evaluation with LLM judges. *arXiv preprint arXiv:2601.08654*, 2026.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*, 2023.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *International Conference on Learning Representations*, 2023.

Denis Merigoux, Nicolas Chataing, and Jonathan Protzenko. Catala: A programming language for the law. *Proceedings of the ACM on Programming Languages*, 5(ICFP):77:1–77:29, 2021.

Douglass C. North. *Institutions, Institutional Change and Economic Performance*. Cambridge University Press, 1990.

OpenAI. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *International Conference on Learning Representations*, 2024.

Zichao Qi, Fan Long, Sara Achour, and Martin Rinard. An analysis of patch plausibility and correctness for generate-and-validate patch generation systems. In *Proceedings of the 2015 International Symposium on Software Testing and Analysis*, pp. 24–36, 2015.

Anka Reuel, Amelia Hardy, Chandler Smith, Max Lamparth, Malcolm Hardy, and Mykel J. Kochenderfer. BetterBench: Assessing AI benchmarks, uncovering issues, and establishing best practices. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 37, 2024.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

Swastik Roy, Rajkumar Pujari, Tharindu Kumarage, Charith Peris, Rahul Gupta, Anna Rumshisky, Pradeep Natarajan, and Venkatesh Saligrama. PReMISE: Policy rubrics as measurement specifications for LLM judges. *arXiv preprint arXiv:2605.30803*, 2026.

Seongho Son, William Bankes, Sayak Ray Chowdhury, Brooks Paige, and Ilija Bogunovic. Right now, wrong then: Non-stationary direct preference optimization under preference drift. In *International Conference on Machine Learning*, 2025.

Beining Wang, Weihang Su, Hongtao Tian, Hao Kong, Tao Yang, Ting Yao, Qingyi Pan, Yueyue Wu, Qingyao Ai, Min Zhang, and Yiqun Liu. Co-evolving LLM evaluators and policies via DynamicRubric. *arXiv preprint arXiv:2607.20083*, 2026.

Shihao Weng, Yang Feng, and Xiaofei Xie. Beyond accuracy: Policy invariance as a reliability test for LLM safety judges. *arXiv preprint arXiv:2605.06161*, 2026.

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations*, 2025.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. Editing large language models: Problems, methods, and opportunities. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

A Metric Definitions and Denominators

For old target y_{t-1}, new target y_t, old prediction $\hat{y}_{t-1}$, and new prediction $\hat{y}_t$:

$$\mathrm{TUR} = \frac{\sum_i \mathbb{1}[y_{t-1,i} \neq y_{t,i}]\mathbb{1}[\hat{y}_{t,i} = y_{t,i}]}{\sum_i \mathbb{1}[y_{t-1,i} \neq y_{t,i}]}, \qquad \mathrm{CFR} = \frac{\sum_i \mathbb{1}[y_{t-1,i} = y_{t,i}]\mathbb{1}[\hat{y}_{t-1,i} \neq \hat{y}_{t,i}]}{\sum_i \mathbb{1}[y_{t-1,i} = y_{t,i}]}. \tag{4}$$

The no-op flip rate applies the same prediction-change expression to revisions whose labels are unchanged under the typed executor. CFR^+ additionally requires $\hat{y}_{t-1} \neq y_t$ and $\hat{y}_t = y_t$; CFR^- requires $\hat{y}_{t-1} = y_t$ and $\hat{y}_t \neq y_t$; each is divided by the protected count. Deferral is one minus proposal acceptance; an invalid, abstained, or rejected proposal retains the archived old rule. For proposal set P, accepted $A \subseteq P$, and empirically erroneous $E \subseteq P$ (candidate rule errs on at least one held-out revision–state representative): accepted precision $= (|A| - |A \cap E|)/|A|$; error catch $= |E \setminus A|/|E|$; erroneous admission $= |A \cap E|/|P|$.

Aggregation. For the clean state-disjoint tables, each rate is computed per criterion ID within a seed (an ID is omitted from a metric only when its denominator is zero; error catch has 2–3 eligible operator IDs per seed, mean 2.6 of 3, and 1 of 6 locked IDs), macro-averaged over eligible IDs, then averaged over five seeds. The noisy-test analysis instead pools all revisions within each model-seed/corruption-seed condition (84 locked or 42 operator revisions; state metrics use one representative per evidence state within each revision; gate rates count revisions) and averages repeated conditions; its 5.71% erroneous admission is a split-level micro rate, not a criterion macro. The implementation field named `scope_recall` defines "in scope" by $J_t(x) \neq J_{t-1}(x)$; it is realized-change recall on affected cases, not recall of a learned jurisdiction.

B Fixed State-Partition Algorithm

The following deterministic procedure matches the released implementation.

1. For every case x, serialize its evidence as compact sorted-key JSON s_x, and derive $z_x = (\texttt{truly_affected}, y_{t-1}, y_t)$ from typed labels.
2. For each revision–transition stratum, collect its distinct states; let S_q be the union over all revisions of criterion q. Force a state into evaluation if it is the sole member of any stratum. Hash-order the remaining states by SHA-256 of `criterion-id|canonical-state`. The implementation rejects a criterion with more than 20 freely assignable states rather than silently approximating the exhaustive search.
3. Enumerate every assignment of the free states. Lexicographically minimize (i) uncovered splittable strata, (ii) an empty gating pool when free states exist, (iii) distance of the evaluation-pool size from $\lceil |S_q|/2 \rceil$, (iv) the assignment mask. Fix the winning partition for every revision, requested budget, and seed of the criterion.
4. Within a revision, intersect present states with the fixed gating pool. Hash-order states inside each typed transition stratum, order strata by $(-\texttt{truly_affected}, y_{t-1}, y_t)$, round-robin across strata, and take the first b states. If fewer exist, the effective budget saturates.
5. Represent each selected or evaluated state by the rendering with minimum SHA-256 of `case_id`. Evaluate on one representative per held-out revision–state pair, retaining all evaluation renderings for the overlap audit.
6. Assert that selected and evaluation states are disjoint and that the fixed partition covers every state present in the revision.

The partition objective uses typed benchmark semantics and state identities only—never proposal correctness, prediction error, or admission outcomes.

C Conditional Design Properties

Two conditional properties are design conditions, not learned guarantees. **Criterion noninterference.** For a supplied target key q, the executor represents criterion state as a map and writes only q;

functional map update leaves each $q' \neq q$ unchanged. Verdict noninterference additionally requires that the final decision not change when only q varies. The derivation assumes the supplied key is correct; it is not a jurisdiction-discovery guarantee. **Prefix replay.** If a complete system stored an immutable base and an append-only ordered patch sequence, deterministic execution on identical evidence would reproduce a prior prefix. The experiment implements a narrower state-dictionary replay check: twenty exactly compiled semantic replacements spanning 18 supplied focal criterion IDs and 151 unique typed states give, across five seeds, 100% final state-dictionary accuracy, 0% maximum off-target-key flips, and 0% replayed-prefix state error. This $b{=}0$ run is a deterministic regression test; it does not establish a general differ, jurisdiction discovery, a complete version log, or robustness to long noisy histories.

D BENCHMARK CONSTRUCTION DETAILS

Table 7: POLICYDELTABENCH v0 counts. Renderings are repeated measurements; criterion IDs are split labels, not independent policy samples.

Object	Observed count
Domains / source policy documents	3 / 3
Criterion IDs / unique passages	45 / 38
Duplicate passage groups	5
Semantic / no-op revisions	180 / 450
Total revisions	630
Affected / protected renderings	4,394 / 40,966
Total renderings	45,360

The public τ^2-bench repository is pinned to commit `363133ada1936491fb5bcec33cd62c3518a99f65` (MIT licensed; source pin and file hashes in the release); use of the policies does not imply endorsement by the benchmark authors. Airline contributes 13 unique passages, retail 14, telecom 15. Integrity checks: all revision and case IDs unique; every case joins to one revision; domains, criterion IDs, families, split assignments, and no-op flags agree across files; every semantic revision contains affected and protected cases; no no-op changes its typed label; `truly_affected` equals label change for every case. The three operator IDs mix rule kinds: airline and retail test an exception kind absent from training; telecom tests a scope kind seen in training. The telecom criterion's 65.75% affected-rendering share would dominate a rendering micro-average, motivating equal criterion weights.

Controlled expressivity stress set. We pair one representable control with one amendment outside the six-kind schema for each of the 45 criterion IDs (90 revisions; 6,480 renderings), across five OOD families—Boolean disjunction/negation, relational comparison, temporal state, aggregation, multi-effect conflict—with nine paired clusters each. A richer reference AST supplies labels; a static capability checker (a benchmark-construction check, not a learned detector) confirms whether each amendment is inside or outside the smaller schema. Before any review analysis, an outcome-blind split using only cluster IDs and family names fixed 15 locked and 30 calibration clusters; control and OOD partners never cross the boundary.

CUAD candidate pool. From 510 raw CUAD records, 475 are retained (474 unique exact full contexts), one candidate span each, passing fixed checks on length, content, clause boundaries, balanced delimiters, predicates/modals, endings, and redaction; no exact-normalized duplicates; largest token-5-gram Jaccard 0.8125. A review family is the union of the 0.50-Jaccard text family, filer provenance, source-document identity, exact context hash, and exact clause hash; whole families are assigned to one side. The frozen split has 75 calibration / 400 locked records, 438 families, largest family 5, zero crossing groups at every identity level. CUAD's pre-existing annotation categories were used only to preserve coverage and approximate balance; all 31 categories with at least two records appear in both splits; three singleton categories remain locked. No model outcome or new human judgment was used, and no CUAD outcomes are reported.

Table 8: Shared training configuration.

Setting	Value
Backbone	Qwen3-1.7B (commit `70d244cc86cc...`)
Precision	BF16; TF32 enabled
LoRA rank / alpha / dropout	16 / 32 / 0.05
Target modules	attention and MLP projections
Learning rate / scheduler	10^{-4} / cosine
Weight decay / warmup	0.01 / 0.03
Classifier / compiler / extractor epochs	1 / 10 / 3
Per-device batch / accumulation	4 / 4
Seeds	17, 29, 43, 71, 101

E Training and Serving Details

Classification uses the first-token ALLOW/DENY logit; compilation uses greedy JSON generation. Both fine-tuned judges receive the same 5,512 case-label examples and optimizer steps; the compiler's 216 structured-rule targets are a different unit, so systems are not supervision- or compute-matched. A semantic-only compiler ablation (108 revisions, unmatched size) improves locked exact rule match (98.57% vs. 94.29%) and locked no-op flips (1.24% vs. 5.33%) but worsens no-op flips on calibration (28.92% vs. 24.42%) and operator (24.78% vs. 18.78%): removing no-op training is not uniformly beneficial. The auxiliary extractor trains on 3,578 unique rendered cases; held-out sets contain 597 / 602 / 216 / 216 unique texts (calibration / locked / operator / linguistic). Its prompt supplies the exact field-key inventory and the model predicts only values; a legacy rendering-level end-to-end audit sends predicted values through admission and held-out execution while retaining supplied keys—a template check of the text–value–gate path, not field discovery or held-out-state safety evidence.

F Comparator Serving Contracts

Qwen3.6-35B-A3B-FP8 is pinned to revision `95a723d08a9490559dae23d0cff1d9466213d989`: an open-weight sparse mixture-of-experts checkpoint of the Qwen3 family with roughly 3B active parameters per token, served in FP8, decoded by sampling (five serial repeats at concurrency one), on a single RTX 4090 against SHA-256-verified weights (56/56 files). Its strict-comparison runs contain 1,820/1,820 (operator) and 3,370/3,370 (locked, merged with 1,660 broad-run tasks) parse-valid tasks with zero missing, duplicate, or unexpected rows, and per-request prompt-hash verification against a frozen manifest. gpt-oss-120b is pinned to revision `ed282e66b414f4e05b8999ef7a9a24e05478b2fd`: roughly 5.1B active parameters per token in the released MXFP4 quantization, low reasoning-effort instruction, sixteen-way request concurrency, two-way tensor parallelism on two RTX 5090s, weights hash-verified file-by-file (27/27). Runner identities bind model revision, prompt, inference settings, data and split hashes, budget, seeds, and inventory, and reject a resume whose identity differs. A separate coverage run of the Qwen comparator on the frozen legacy-criterion inventory (57,120 rendered-case tasks; same-state prompts; rendering-micro statistics) gives $80.98 \pm 3.60\%$ current accuracy, $87.03 \pm 1.51\%$ TUR, $7.95 \pm 0.79\%$ CFR^{+}, $2.96 \pm 0.24\%$ CFR^{-}, and $7.64 \pm 0.45\%$ no-op flips; it is a behavior check, not held-out-state evidence.

G Per-Criterion Results and Sensitivity

Leave-one-criterion-out sensitivity. From $b{=}0$ to $b{=}16$ on the operator split, the four-metric change vector depends on the omitted ID: accuracy changes sign across deletions (Δacc -0.54 / $+0.95$ / $+0.42$ omitting airline / retail / telecom), TUR and CFR^{-} each flatten under one deletion, and deferral rises under all three ($+2.14$ / $+35.71$ / $+37.86$ points). Excluding the structural zero-capacity telecom ID preserves the full-table directions: accuracy 97.74% $\rightarrow$ 98.15%, CFR^{-} 1.65% $\rightarrow$ 0.63%, TUR 92.00% $\rightarrow$ 86.00%, deferral 0% $\rightarrow$ 37.86%. On the locked split, excluding `airline.insurance_timing` yields 100% accuracy and TUR, zero CFR^{-}, and zero deferral at both budgets: all observed locked errors localize to that ID. The sensitivity audit reconstructs 315

Table 9: Per-ID operator diagnosis. Slash-separated metrics are b=0 / b=1; percentages average five seeds. Affected share counts held-out affected renderings and does not weight the criterion macro. Pool size is maximum disjoint gating-state capacity.

Criterion ID	Aff. share	Max pool	Acc.	TUR	CFR^-	Defer
airline.cancel_exceptions	13.70	6	96.90 / 98.81	100 / 100	3.29 / 1.27	0 / 70.00
retail.gift_balance	20.55	4	98.57 / 97.50	84 / 72	0 / 0	0 / 4.29
telecom.roaming_scope	65.75	0	66.67 / 66.67	100 / 100	38.89 / 38.89	0 / 0

Table 10: Per-criterion directional CFR on identical representatives (%, seed means), all methods. POLICYPATCH's CFR^+ is identically zero by construction. The prompted Qwen comparator ranges 1.15–17.60% CFR^+ and 0.94–5.60% CFR^- across the six locked IDs, and 5.57–12.78% / 0.00–5.06% across the three operator IDs; gpt-oss-120b ranges 12.91–13.89% / 2.75–15.70% (operator) and 3.60–27.20% / 1.60–10.00% (locked).

		POLICYPATCH b=1		CC-LoRA		Delta-LoRA	
Split	Criterion ID	CFR^+	CFR^-	CFR^+	CFR^-	CFR^+	CFR^-
Operator	airline.cancel_exceptions	0.0	1.27	5.32	9.11	6.58	7.59
Operator	retail.gift_balance	0.0	0.0	3.53	2.75	10.2	0.78
Operator	telecom.roaming_scope (0 states)	0.0	38.89	11.11	0.0	3.33	2.78
Locked	airline.cancel_reason	0.0	0.0	0.0	0.0	1.6	0.4
Locked	airline.insurance_timing (0 states)	0.0	26.67	11.67	7.22	8.33	10.56
Locked	retail.modify_fields	0.0	0.0	0.38	1.54	0.38	1.15
Locked	retail.single_payment	0.0	0.0	0.0	0.0	0.31	0.62
Locked	telecom.plan_price	0.0	0.0	0.0	0.0	0.4	0.0
Locked	telecom.refuel_steps	0.0	0.0	0.0	1.6	0.0	0.4

seed–criterion–budget units from frozen predictions, cross-checks 4,410 clean revision decisions and 9,135 metric cells, with zero directional-CFR residual. These are fixed-inventory exclusion calculations, not population or causal estimates.

H SEED DISPERSIONS AND PROMPTED 1.7B CONTROLS

Table 11: Sample standard deviations across five repetitions (percentage points). Trained rows use saved seeds; prompted rows use serial sampling repeats. N/A denotes a nongated method. Dispersions describe only the fixed three-ID and six-ID inventories.

Split	Method	Acc. SD	TUR SD	CFR^+ SD	CFR^- SD	Defer SD
Operator	Current-clause LoRA	4.37	10.48	1.03	2.03	N/A
Operator	Delta LoRA	7.34	2.98	1.58	1.80	N/A
Operator	Qwen3.6-35B-A3B-FP8	1.90	11.37	2.69	0.50	N/A
Operator	gpt-oss-120b	3.58	7.27	1.81	3.49	N/A
Operator	POLICYPATCH b=0	2.27	5.58	0.00	2.82	0.00
Operator	POLICYPATCH b=1	1.99	3.65	0.00	2.36	2.13
Operator	POLICYPATCH b=16	1.99	3.65	0.00	2.36	1.30
Locked	Current-clause LoRA	1.58	1.24	0.47	0.28	N/A
Locked	Delta LoRA	2.07	1.24	0.62	2.81	N/A
Locked	Qwen3.6-35B-A3B-FP8	4.84	2.48	1.25	0.70	N/A
Locked	gpt-oss-120b	0.81	2.18	0.80	1.00	N/A
Locked	POLICYPATCH $b \in \{0, 1, 16\}$	2.05	0.00	0.00	2.40	0.00

The same-representative reconstruction found zero missing, duplicate, or wrong-split predictions, zero representative drift across budgets, and maximum directional-CFR identity residual 2.78×10^{-17}.

I NOISY-TEST DETAILS

Corruption modes: *label flip* negates a selected test's stored expected verdict; *evidence omission* deletes one deterministically chosen evidence key; *evidence value* mutates one deterministically chosen evidence value. Selection uses a stable hash of (corruption seed, revision, case, mode) at

Table 12: Single-run 1.7B prompted-base controls (%, criterion-ID macro) on the same representatives; no across-seed dispersion is available. The prompted base compiler emits parseable JSON for every locked revision but has 0% v0 schema validity and 0% exact rule match, so the schema gate defers all its proposals.

Split	Control	Acc.	TUR	CFR^+	CFR^-
Operator	Current clause	55.56	42.22	0.00	1.31
Operator	Delta+current clause	56.55	42.22	0.65	1.08
Locked	Current clause	50.03	58.73	2.34	0.00
Locked	Delta+current clause	48.62	58.73	0.96	0.32

nominal rates {6.25, 12.5, 25}%. The strict gate accepts only with zero mismatches; a tolerate-1 variant permits one (tolerate-1 rows in the release). Realized corruption rates match nominal within sampling error (e.g., label flip: 6.30% / 12.91% / 25.58% over 4,300 selected slots $\times$ 20 replicates).

Table 13: Noisy-test results at requested b=16, strict gate (%, split-level micro; mean±SD over 5 model seeds $\times$ 20 corruption seeds). Locked error catch is structurally zero at every condition—the only erroneous proposals belong to a zero-capacity criterion—so its sweep is non-discriminative; operator error catch moves with corruption, but the deferral and precision columns show the added rejections are dominated by false positives.

Split	Condition	TUR	Defer	Acc. prec.	Err. adm.	Err. catch
Locked (6 IDs)	clean	100.00±0.00	0.00±0.00	94.29±3.08	5.71±3.08	0.00±0.00
	evid. omission 25%	58.85±6.64	49.76±4.67	88.53±5.68	5.71±2.77	0.00±0.00
	evid. value 25%	87.24±5.48	17.50±4.99	93.05±3.41	5.71±2.77	0.00±0.00
	label flip 25%	59.76±7.61	48.15±4.90	88.88±5.52	5.71±2.77	0.00±0.00
Operator (3 IDs)	clean	94.25±2.25	25.24±1.30	62.44±2.20	28.10±1.99	11.60±7.70
	evid. omission 25%	76.70±5.57	48.00±5.63	51.20±5.09	25.14±1.63	20.80±7.21
	evid. value 25%	82.64±4.96	39.62±6.10	55.25±4.94	26.81±2.24	15.66±7.78
	label flip 25%	72.07±4.74	55.14±5.10	44.04±6.53	24.81±1.40	21.87±6.45

Intermediate rates (6.25%, 12.5%) interpolate monotonically between the clean and 25% rows for every column and mode (full tables in the release: `witness_state_disjoint_v1`, `witness_state_disjoint_operator_v1`).

J Gate Strictness and Constructed-OOD Details

Table 14: Outcome-blind locked schema-OOD split (%, mean±SD over five sampling seeds; 15 paired clusters, three per family). The schema gate misses most unrepresentable amendments; adding test execution rejects every schema-admitted OOD compilation but raises false rejection on matched representable controls to 76%.

Gate	OOD rejection	Control false rejection
Schema only	30.67±8.94	49.33±5.96
Schema + test execution (b=4)	100.00±0.00	76.00±5.96

By constructed family, schema-stage OOD rejection is 100% for Boolean composition, $53.33 \pm 38.01\%$ for temporal state, and 0% for relational comparison, aggregation, and multi-effect conflict; every rejection is an explicit `ABSTAIN_UNREPRESENTABLE` and the schema validator itself rejects none. On the full 45-cluster pool, pooled schema-stage rejection is $30.67 \pm 7.60\%$ and schema-stage control false rejection $54.67 \pm 3.37\%$ (locked-split values in the main text). **Abstention-removed ablation** (same items, seeds, generation settings, model pin): parse coverage stays 100%; schema-stage rejection of unrepresentable amendments falls to $14.67 \pm 8.69\%$ (all fail-closed schema invalidity), control false rejection falls to $40.00 \pm 0.00\%$, exact-envelope accuracy stays 6.67%. Roughly half of abstain-mode rejection behaves like hedging; forcing compilation turns it into schema-valid but wrong outputs. Replaying forced compilations through state-disjoint test execution again

Table 15: Test-gate replay on all 45 paired expressivity clusters (%, criterion-cluster macro, mean±SD over five sampling repeats). Results are identical at $b \in \{1, 4, 16\}$ because every rejection fires on the first test state. *Mechanism disclosure*: all test rejections, in both classes, are fail-closed execution failures (compiled rules reference evidence fields absent from the richer typed states); label-level disagreement never fires—type-coverage screening, not semantic unrepresentability detection. Three control revisions have zero test capacity (schema-only admission; none schema-admitted). The replay uses the prompted compiler; the trained v0 adapters were not retained.

Amendment class	Schema adm.	Test rej. of adm.	Final adm.	Err. catch
Unrepresentable (45)	69.33±7.60	100.00±0.00	0.00±0.00	100.00±0.00
Representable controls (45)	45.33±3.37	56.07±5.45	20.00±3.51	100.00±0.00

rejects every schema-admitted unrepresentable compilation at every budget—end-to-end constructed-OOD admission 0% under both prompt variants—while end-to-end control admission rises from $20.00 \pm 3.51\%$ to $35.56 \pm 0.00\%$, every admitted control empirically correct on the listed states. The independent validator passes on the ablation run (450/450 tasks). These are constructed families and serial model repeats, not a natural-distribution OOD estimate.

K Legacy Rendering-Level Analyses (Reuse Record)

Retained only as the reuse record behind §4. In the legacy fixed-rendering locked analysis (5,712 renderings after excluding four selected case positions, five-seed means), the frozen old rule scores 89.79% accuracy; Current-clause LoRA 93.47% (total CFR 3.59%); Delta LoRA 93.58% (3.73%); and PolicyPatch at b=4 reaches 100% accuracy, 100% TUR, 0% CFR with 5.71% deferral—a result now attributable to full state overlap: all 84 revisions overlap, all 268 selected unique states recur in evaluation, and all 336 selected gating renderings have an evaluation rendering of the same state. A related audit of the pinned τ-Bench Verified release (Cuadron et al., 2025) compares 164 retail/airline task objects against the source snapshot (53 fix groups; 42 interface, 51 adjudication, 16 both; 24 policy references), motivating the separation of task interface from adjudication; it does not run PolicyPatch and contains no policy amendment.

L Audited Amendment Example

Table 16: One amendment audit: the first stored semantic revision in the locked split, first reported seed (not hand-selected). The release records full case identifiers and raw generated JSON.

Object	Recorded value
Old clause	The user cannot add insurance after initial booking.
Amendment	Insurance may be added at initial booking or after booking.
Compiled rule	`field = booking_stage`; `allowed = {initial_booking, after_booking}`; declared universe also contains `after_departure`.
Affected tests	Two surface forms with `booking_stage = after_booking`: DENY → ALLOW.
Protected tests	`initial_booking`: ALLOW → ALLOW; `after_departure`: DENY → DENY.
Admission record	Schema valid; four of four tests correct; candidate rule selected for this experiment.

M From Screening Toward Certification: Generating Coverage

The coverage ceiling of §5 binds because gating tests are drawn from an existing case inventory: $|W_q|$ is capped by which evidence states happen to occur. Nothing in the gate requires this. A typed rule over fields with finite declared universes induces an enumerable state space Σ_q (the product of

field universes, intersected with validity constraints), and tests can be generated from Σ_q directly. We sketch three mechanisms realizing the conditions of Proposition 2, ordered by assumption strength, as *proposals*: none is empirically validated in this paper, and each carries stated assumptions.

(1) Scope-differential testing. Let r_{t-1} be the archived typed old rule, $\hat{r}_t$ the candidate, and $\sigma(\Delta_t) \subseteq \Sigma_q$ the amendment's declared scope (the states whose verdict it is intended to change). Generate states $s \in \Sigma_q$; reject $\hat{r}_t$ if $\hat{r}_t(s) \neq r_{t-1}(s)$ for any $s \notin \sigma(\Delta_t)$. No labeled tests are needed: on protected states the old verdict *is* the expected verdict, so locality testing scales with generation budget rather than inventory. Within scope, generated states still require labels from the amendment's intended semantics (or human review), so this mechanism buys locality coverage, not edit-success coverage. *Assumption:* a trusted $\sigma(\Delta_t)$—exactly the jurisdiction input our evaluated pipeline receives from the benchmark; an incorrect scope specification converts locality violations into accepted behavior. Directly targeting our worst observed failure, this mechanism is applicable to zero-inventory criteria, where the current gate cannot act at all.

(2) Coverage-directed generation. Rather than sampling Σ_q, select states to cover the structure of r_{t-1} and $\hat{r}_t$: each branch of each typed predicate, each boundary of threshold and temporal comparisons, and each pairwise (more generally t-way) combination of field values, in the combinatorial-testing sense. For rules in the six-kind schema, predicates are shallow and solver support is unnecessary—boundary states are computable in closed form; richer schemas (Boolean composition, relational and temporal predicates; §6) admit solver-produced boundary states per region. This is the property-based-testing discipline (Claessen & Hughes, 2000) applied to rule diffs, and it subsumes the stratified round-robin selection of Appendix B when the inventory is replaced by generation.

(3) Exhaustive checking. When $|\Sigma_q|$ is small—plausible for many of our criteria, whose declared universes are finite and shallow—enumerate it. Then admission is equivalence checking of $\hat{r}_t$ against r_{t-1} outside $\sigma(\Delta_t)$ and against the intended semantics inside it, and the gate *is* a certificate within the closed world, in the model-checking rather than testing regime. Cost grows with the product of universe sizes; hybrid policies (exhaustive below a size threshold, mechanisms 1–2 above it) are the practical shape.

What generation cannot buy. All three mechanisms certify behavior on the *typed* space Σ_q. The residual risk moves to the abstraction gap: whether the schema represents the policy (§6 shows the six-kind schema silently admits most out-of-schema amendments absent abstention), whether the evidence extractor maps real cases to correct typed states, and whether $\sigma(\Delta_t)$ is right. Generated states must also respect validity constraints (Σ_q may over-approximate realizable evidence, producing vacuous tests), and none of this addresses open-world evidence. Coverage of the typed space is therefore necessary for certification but sufficient only relative to the abstraction—which is where we expect the hard residual problems to live, and why the human-annotated external track matters.

N Audit Summary

The data-quality notebook checks source and derived hashes, key coverage, split assignments, duplicate passage groups, label consistency, state-disjoint partitions, evidence-state reuse, and exact reconstruction of reported metrics. Recomputation found no numeric mismatch in 92 classifier/compiler run-splits or 20 extractor run-splits; a closed-vocabulary downstream audit covered 20 run-splits comprising 85,680 case decisions, also with no mismatch. The state-disjoint validator reports zero partition and selected-state overlap; 228 tests and 202 parameterized subtests pass under Python 3.11. Language-model tools assisted code search and copy editing; all labels, results, and statistics come from versioned scripts and frozen files, and no tool output supplied an annotation, test, or CUAD outcome. We claim numeric reconstruction where case-level outputs exist, not an exact end-to-end source snapshot. Known reproducibility caveats: one legacy cases-file hash is stale; saved compiler rows exhibit writer/code drift; three evaluation-only metadata files omit benchmark data hashes; and LoRA adapter bytes were not archived, so adapter identity is bounded by the pinned base revision, training metadata, seeds, and prediction-file hashes rather than byte reproducibility. The authors checked the text, citations, and computations and retain responsibility for the manuscript.