PRIME RADIANT · THE COMPLETE RECORD · 2026-08-30 → 2026-09-01

The Prime Radiant Record

How a CDC FluSight forecaster went from an empty repository to a registered, publicly measurable system in three days — built agent-first with Claude Code, gated by a human at every outward step, and adversarially attacked at every phase boundary. This is the merged record of the eight phase walkthroughs, with every number carrying a commit, a run ID, or a refuter's verdict.

0.999989
relative WIS — our replica vs CDC's official baseline, full 2024-25 season

0.609
relative WIS — our LightGBM wins the 2025-26 league table

29 / 11
claims adversarially attacked / refuted and fixed with attack-derived tests

21
mutants run-and-killed in the final two phases alone — kills are facts only after the mutant runs

100%
test coverage — 963 statements, 0 missed, and the CI gate demands it stay there

≈$0
compute — pure statistics, CPU only, no LLM calls anywhere in the forecast path

CONTENTS

Repository
github.com/JeremyGracey-AI/prime-radiant MIT · CI, docs, release & publish automation

Live dashboard
huggingface.co/spaces/jeremygracey-ai/prime-radiant the frozen backtest record, interactive

Package
pypi.org/project/prime-radiant v0.1.0 via Trusted Publishing (OIDC)

Docs
jeremygracey-ai.github.io/prime-radiant model card, method, gates

Hub registration
cdcepi/FluSight-forecast-hub #3696 JGracey-prime_radiant, metadata PR open

Every claim gets a gate, every gate gets two proofs

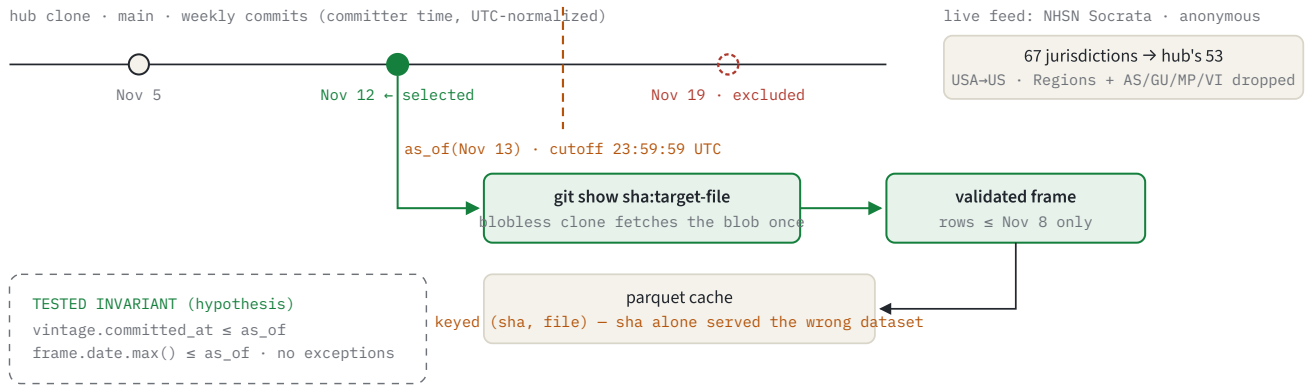
The design principle behind the whole build: an AI system is a governed, observable, evaluated composition of models, tools, data, and policies — and each of those words is a checklist item, not a slogan. In practice it meant four standing rules. **Enforce in structure, not prose:** anything outward-facing (a CDC submission, a deployment, the repository going public) sits behind gates that fail closed — event types, boolean inputs, repository variables, secrets that don't exist yet — and each gate is proven twice, once demonstrably shut, once demonstrably open. **Cite or it didn't happen:** results carry commits, CI run IDs, or refuter verdicts. **State the gap:** known weaknesses ship in the README, not in a drawer. **The human owns the go:** the agent proposes, builds, and verifies; every irreversible step waited for an explicit human word.

Each phase ran the same loop: a read-only fact-verification workflow before planning; a ten-line plan approved by the human; a test-driven build; the offline and integration gates run unpiped; then a panel of adversarial refuter agents paid to break the phase's claims, with every demonstrated defect fixed alongside a regression test derived from the attack itself. The project began as a Metaculus forecasting bot (that thread built its API scaffold and is parked with tests green), then turned to the harder, publicly-scored target: CDC FluSight quantile forecasts of weekly US influenza hospital admissions.

2 · DATA HONESTY

The hub's git history is the vintage store — and the invariant is a tested property

Honest backtesting dies on one mistake: scoring yesterday's forecast against today's revised data. The FluSight hub commits its truth file weekly, so its commit graph *is* a vintage store: `as_of(date)` resolves the last main-branch commit at or before end-of-day UTC and reads the file as it stood then. A hypothesis property test proves on synthetic repositories that nothing returned ever postdates the as-of date — the leakage rule is an executable invariant, not a comment.



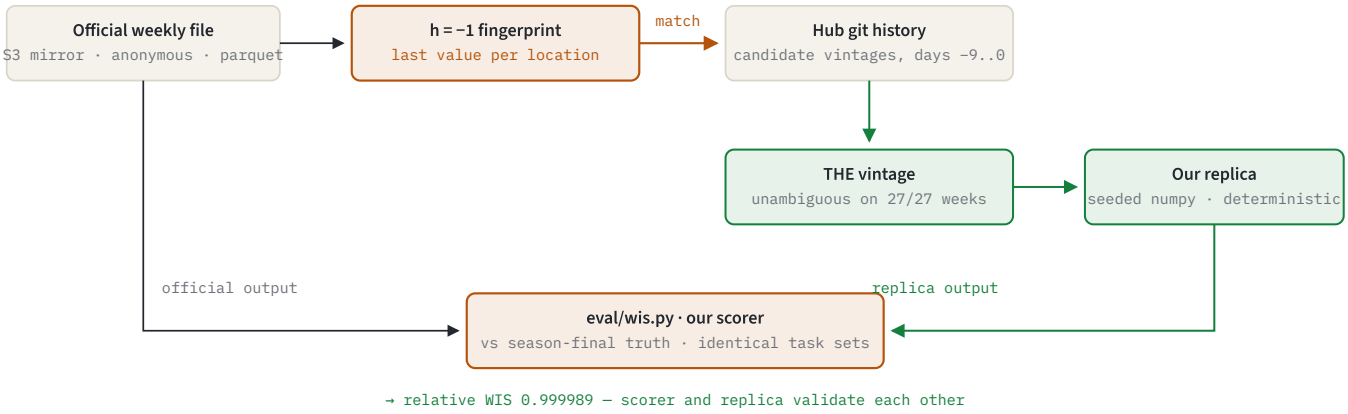
The vintage mechanism. Green: the selected path — the last main-visible commit at or before the cutoff, read at that sha, validated, cached by (sha, file). The amber annotation marks the bug the adversarial pass caught: real hub commits touch up to four target files at once, and a sha-only cache silently served hospital admissions when asked for ED visits. The NHSN inset is the live-season feed, filtered to the hub's exact location set so nothing double-counts.

What the refuters caught here: the vintage cache was keyed by commit sha alone — real hub commits touch several target files at once, and the cache silently served hospital admissions when asked for ED visits, with the wrong frame still passing the schema. Now keyed `(sha, file)`, proven by a poisoning test. The one-schema-for- five-targets contract and a vacuous cache round-trip test fell in the same pass: three of five claims refuted, all fixed with attack-derived tests.

Replicating CDC's baseline to five decimal places

Before any model of ours got scored, the measuring instruments had to be beyond doubt: a WIS scorer proven against the Bracher et al. algebra digit-for-digit, and a Python replica of the official FluSight-baseline validated against CDC's published output. The obstacle was knowing *which* data snapshot the official Wednesday-night run saw. The way out: every official file's horizon −1 rows are degenerate — all 23 quantiles equal the last observed value per location. That's a fingerprint, and matching it against candidate vintages from the hub's git history identifies the exact snapshot, unambiguously, on all 27 weeks of 2024-25.

0.999989 relative WIS, replica vs official, 5,724 forecast tasks across the full season — mean WIS 263.126 vs 263.129; 99.88% of horizon-0 cells exact, every residual a float-rounding ±1.

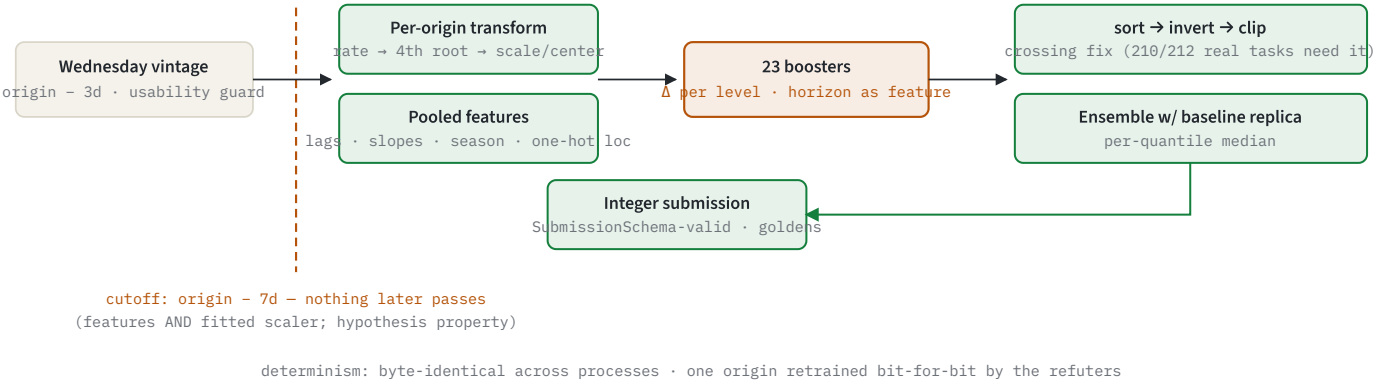


The cross-validation loop. Because the official run is deterministic at horizon 0, feeding the fingerprinted vintage through the replica should reproduce it — and does: the full-season sweep is 99.88% cell-exact, with every residual an R-vs-numpy float disagreement within 6×10^{-14} of an integer, amplified to ±1 by the hub's floor/ceil rounding rule.

What the refuters caught here: the WIS *validation gate* contained a mathematical tautology — one condition true for any input — so crossed quantiles could produce negative dispersion without complaint. The scoring math itself was correct on every valid input (independent agreement to 3.6×10^{-15}); the gate was rebuilt with nine rejection tests derived from the attacks.

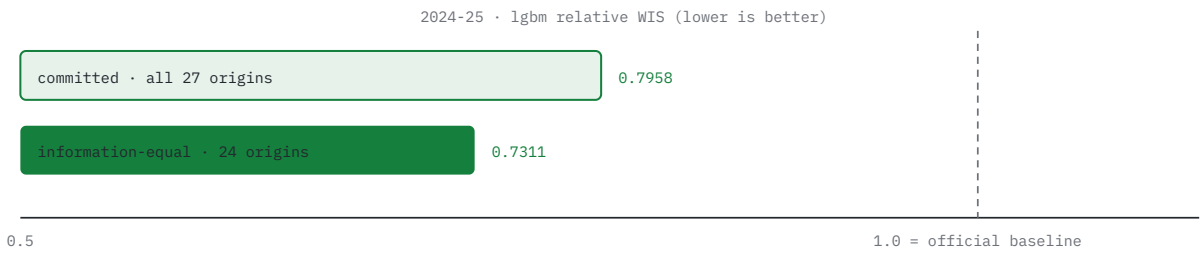
Flusion's winning shape, with the leak-proof cut drawn where it matters

The forecaster copies what actually won FluSight 2023/24: pooled across all locations, one LightGBM booster per quantile level, horizon as a feature, predicting the *change* in 4th-root transformed admission rates. The subtle discipline is the origin cut — not just the features but the fitted transform statistics come only from data at or before each origin's cutoff. A scaler fitted on final data would leak the future through normalization even with vintage-honest features; here that's a hypothesis property, not a convention. lightgbm 4.7.0 is exact-pinned because its determinism is binary-scoped, and the backtest parquets byte-regenerate across sessions.



The Phase C pipeline. The amber dashed line is the load-bearing discipline: the scaler statistics are refitted for every origin from data at or before its cutoff — a scaler fitted on final data would leak the future through normalization even with vintage-honest features. lightgbm 4.7.0 exact-pinned; deterministic + single-thread + fixed seed.

The phase gate demanded relative WIS < 1.0 against CDC's baseline on two seasons; it cleared at 0.59–0.90. And the adversarial pass proved the committed numbers *understate* the model: on three 2024-25 holiday weeks the official baseline's run saw data committed Thursday or later — data our live-Wednesday discipline refuses. On the 24 information-equal origins the model scores better than its own committed numbers.



The committed number includes three origins where the comparison is rigged against us. Proof chain: our replica pools to relative WIS 1.0000 against the official on the other 24 origins (and 0.999945 across all of 2025-26); the entire 1.0808 gap on 2024-25 decomposes to the three fingerprint-mismatched weeks. Documented in `tests/golden/wis_baseline.json`.

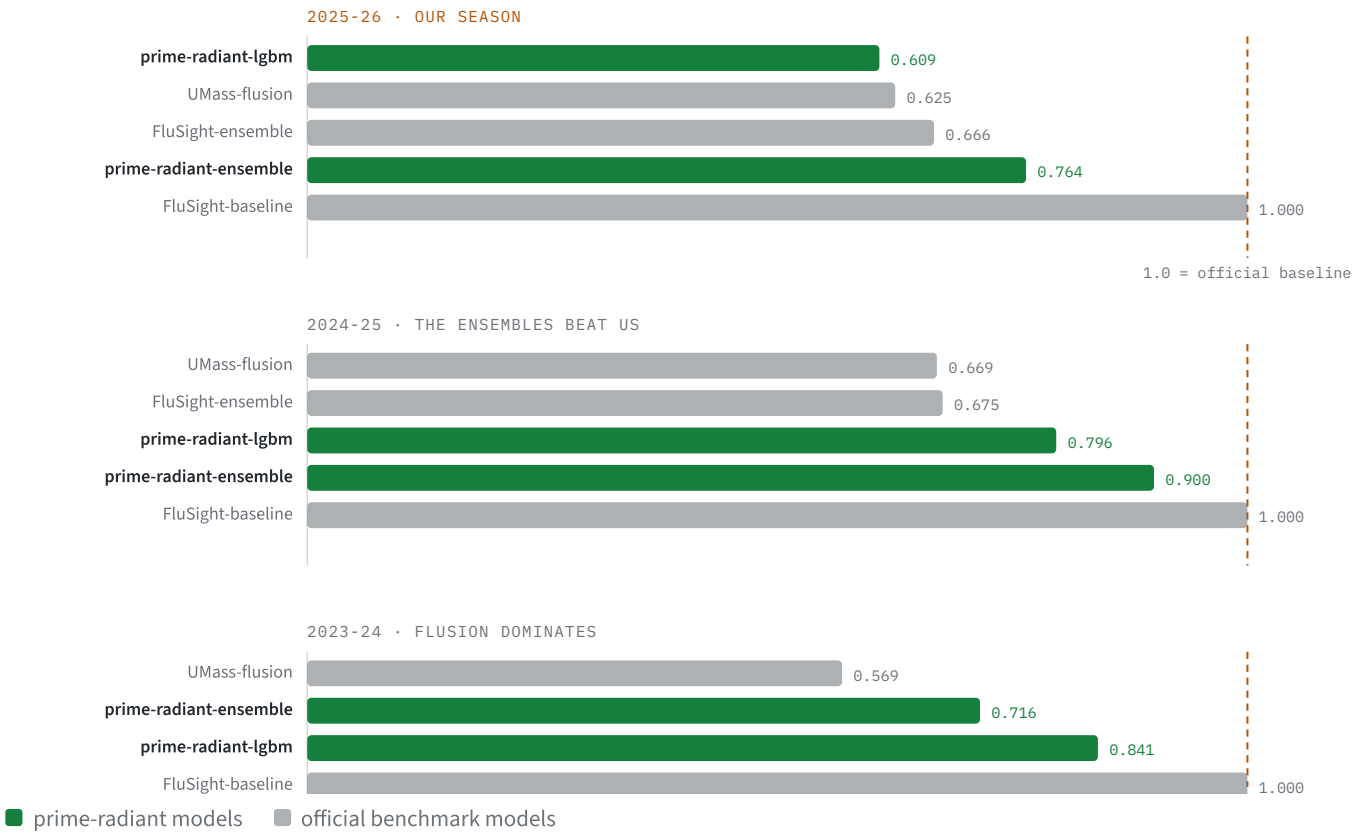
Kept anyway: the honest live-deadline discipline costs us in backtest and stays. The backtest is net biased against the model, and the decomposition proving it is committed in `tests/golden/wis_baseline.json`.

What the refuters caught here: the quantile-crossing test was vacuous — its smooth fixture never crossed, so deleting the sort passed every test while 210 of 212 real-vintage tasks crossed (worst inversion ≈4,570 admissions). Replaced with a self-validating fixture that first proves the raw boosters cross; the deleted-sort mutant now demonstrably fails.

5 · THE LEAGUE

Three seasons on the record — including the two we lose

Common-task relative WIS against the official FluSight models: 85 vintage-honest weekly origins, all 53 jurisdictions, horizons 0–3, truth pinned to a stated as-of date. In 2025-26 our LightGBM leads the entire table on the natural scale (on the log scale UMass-flusion edges it, 0.583 vs 0.585 — both columns are in the CSVs). In 2024-25 and 2023-24 UMass-flusion beats us and the tables say so. Every one of the 90 report rows was reproduced by an independent from-scratch scorer to 5×10^{-7}.



Overall relative WIS per season (FluSight-ensemble omitted from the 2023-24 panel at 0.730, between our two models — full tables incl. per-horizon rows, log scale, AE-median and coverage live in `reports/backtest_<season>.csv`). Hover any bar for its value. The 2024-25 number carries the Phase C conservatism finding: on information-equal origins our lgbm scores ~0.73.

The honest debt

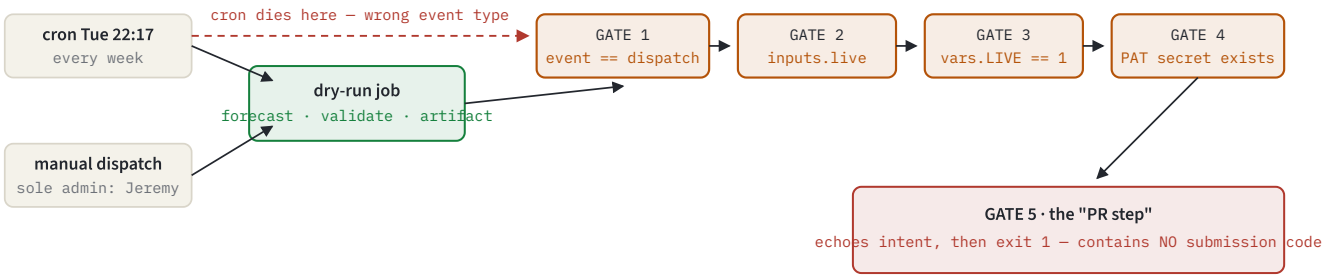
MODEL · 2025-26	50% INTERVAL COVERS	95% INTERVAL COVERS
prime-radiant-lgbm	39.8% UNDER	81.8% UNDER
FluSight-ensemble	52.6% GOOD	90.3% GOOD
FluSight-baseline	43.3%	86.4%

We win on WIS while being worse-calibrated than FluSight-ensemble — sharp but overconfident. Season-level bagging (flusion's stabilizer) is the known lever, deliberately parked and stated in the README rather than hidden.

What the refuters caught here: all in the presentation layer — the league CSV was ordered by WIS-as-string (lexicographic), the PNG regression checks passed on blank figures, and one sentence in the README's own honesty section overclaimed. The numbers survived; the packaging didn't, and that's exactly what the pass is for.

A submission pipeline that cannot submit

The weekly machinery — forecast, validate against the live hub config, package, prove it in CI — was built with the live path to CDC's hub made unreachable in structure. A refuter enumerated every trigger against live GitHub state and verified five independent gates between the weekly cron and a real submission, ending in a step that at the time contained no submission code at all. The CI proof: the dry-run job green on a cold runner with a 4,877-line hub-valid artifact, and the live job's conclusion literally `skipped`.

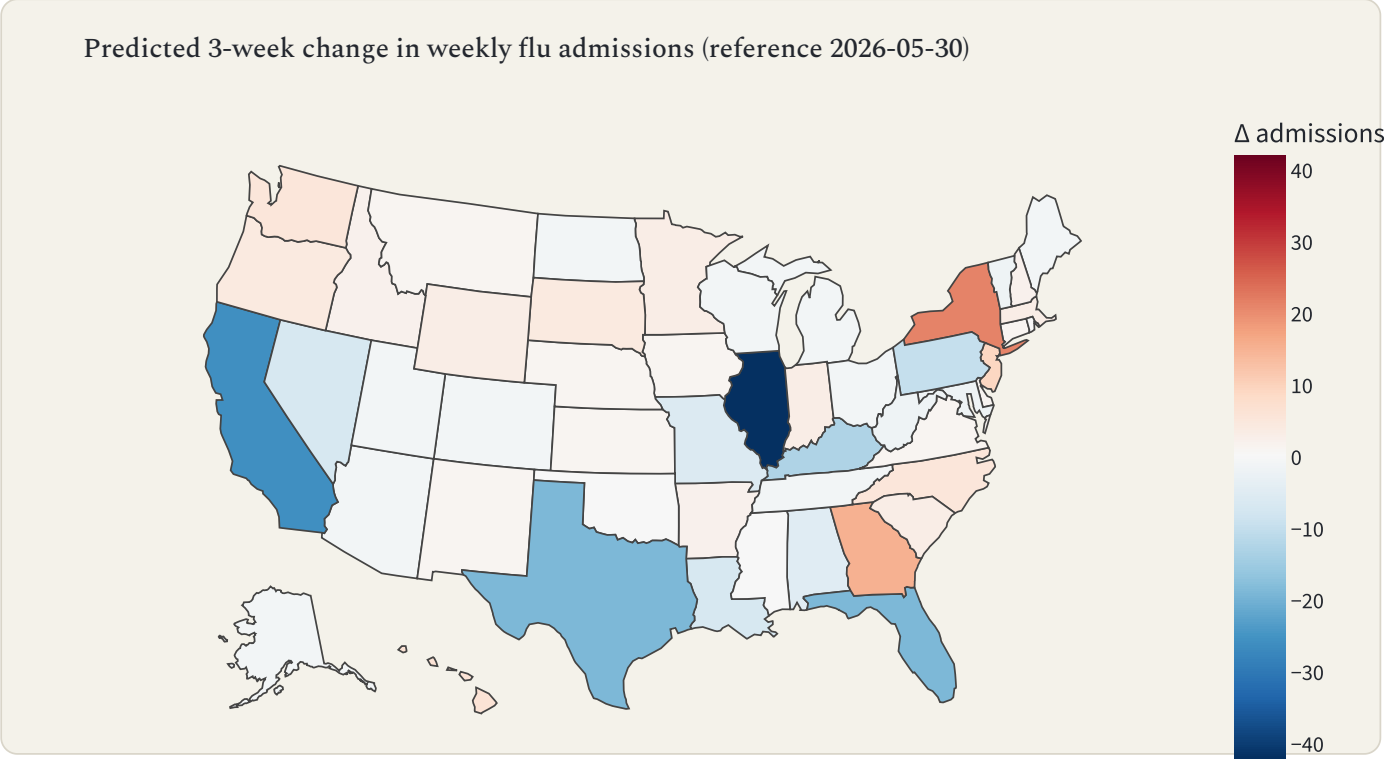


The refuter's verification against live GitHub state: no push/PR/fork event can reach the workflow; boolean-input semantics fail closed (doc-confirmed); the repo has zero secrets, zero variables, zero environments; the token is `contents: read`; and gate 5 is an `exit 1` until go-live. Go-live requires walking the runbook — fork, PAT, registration PR, 2026-27 rounds — and Jeremy's explicit go, none of it automated.

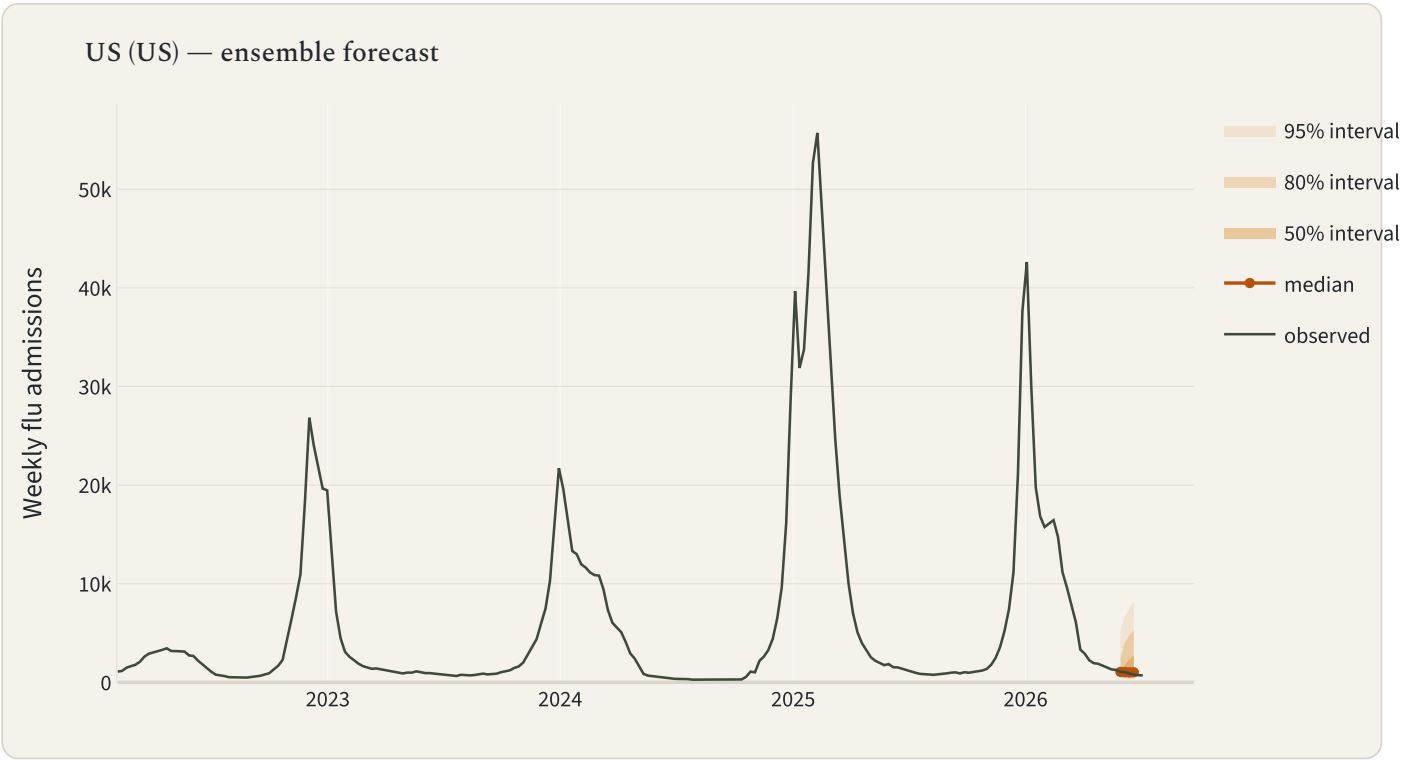
> **The instructive catch of the whole project:** a mutant appending `|| github.event_name == 'schedule'` to the live-job condition — making it cron-reachable — passed all seven workflow-honesty tests, because substring assertions can't see operator precedence. Every gate assertion in the repository is now an exact-conjunction equality, and that discipline caught real gaps twice more in later phases.

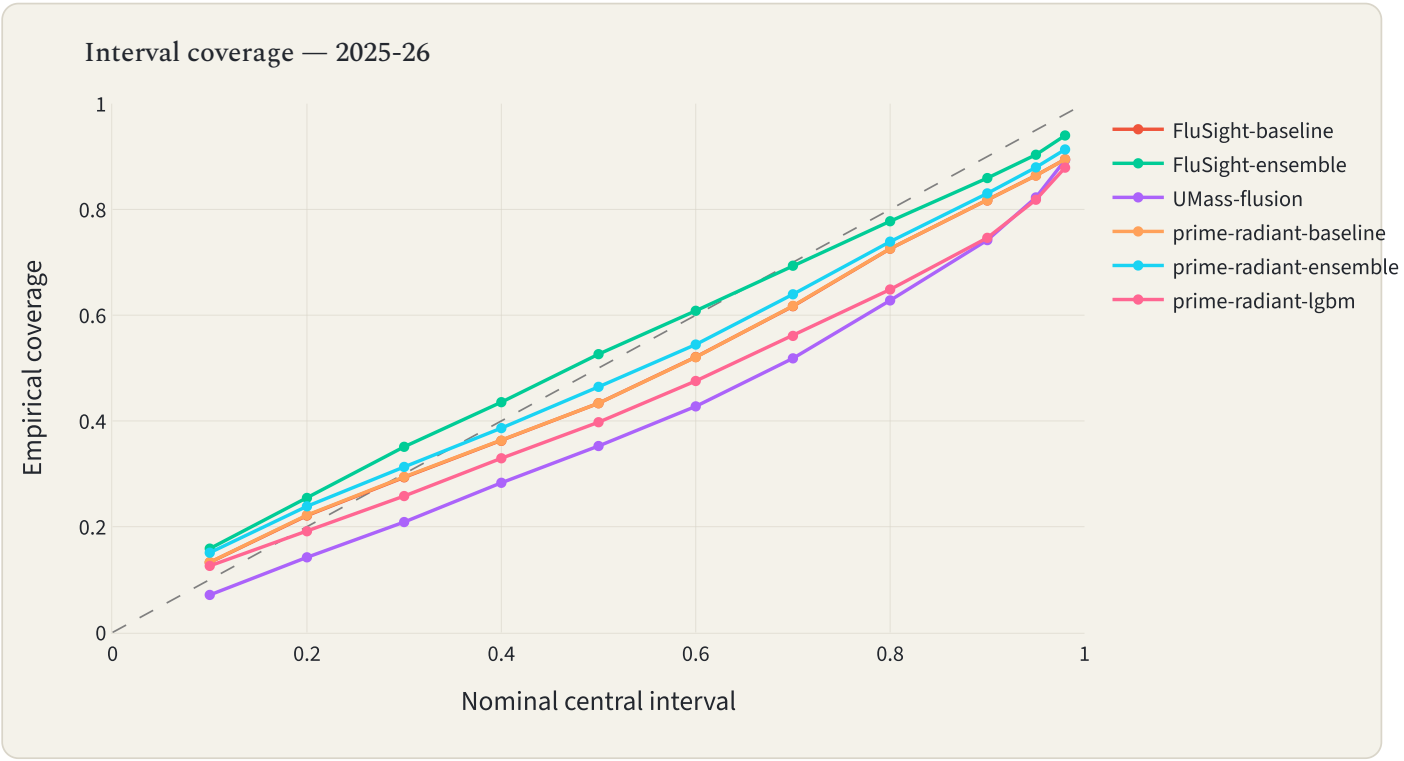
The backtest record, live — served from a 1.7 MB frozen bundle

The public face is a Gradio app on Hugging Face Spaces serving exactly four panels: a US choropleth of predicted 3-week change, per-state fan charts with observed history, reliability curves, and the league tables. The load-bearing decisions: the serve bundle is precomputed *offline* and committed (a CI rebuild could silently diverge — LightGBM's determinism is binary-scoped), and the Space never installs the project package, so no model runtime exists at serve time. The bundle byte-regenerates via an integration test, and its precomputed coverage curves match the league CSVs to six decimals on all 18 model-season pairs. The figures below are the dashboard's actual figures, embedded from the same bundle.



Predicted 3-week change, ensemble model, reference round 2026-05-30: horizon-3 median minus the last observation *at or before* the reference date — never later truth, though the frozen bundle's truth runs five weeks past the round. Red = predicted rise.





What the refuters caught here: the choropleth's change anchor originally read "latest observed" — which in a frozen bundle is truth from *after* the prediction target (an advisor catch, pre-build); Puerto Rico, undrawable in the map projection, silently set the color scale's maximum in two of three models; the fan chart's anchor point duplicated horizon-0's date in 53 of 53 locations; and a NaN truth value on the anchor date became the anchor. All four fixed with attack-derived regression tests; 9 of 9 mutants run and killed.

Every outward path proven shut, then proven open

The final phase made the repository public — but only after a full-history audit (secrets sweep clean, single git identity, no blob over 1 MB), and only behind release machinery whose every gate carries two proofs on the record:

GATE	PROVEN SHUT	PROVEN OPEN
docs → GitHub Pages `repository.private == false`	deploy job conclusion `skipped` while private (run 33480528961)	same gate, post-flip: deploy green, site serving (run 33531451092)
release (dispatch-only, noop default, the repo's only `contents: write`)	cron structurally absent; noop run changed nothing, printed `v0.1.0`	real dispatch: psr committed, tagged, released v0.1.0 (run 33540519945)
publish → PyPI (env `pypi`, OIDC only, public-gated)	untriggered until the tag existed; sdist scoped after a refuter found it shipping the repo's working notes	Trusted Publishing green on the tag ref (run 33540651371) — zero stored credentials
coverage upload (public ∧ ¬dependabot ∧ ready)	skipped while private; skipped for dependabot by clause	uploading via OIDC, badge at 96%
the visibility flip itself	held across two explicit asks	the human's word, 2026-09-01 — MIT detected, secret scanning + push protection on

One incompatibility deserved its own artifact: the repository's history is uniformly `[claude] type(scope): subject`, and python-semantic-release's stock parser rejects every such commit (regex anchored at string start — proven with a dry run: 26/26 unparseable). The fix is a 15-line parser subclass that strips exactly one actor prefix, proven end-to-end in a scratch clone: version stamped, changelog inserted above the preserved hand-written history, the regenerated lockfile riding inside the release commit, and the commit and tag authored under the human's git identity.

What the refuters caught here: the sharpest — post-flip, every Dependabot PR would have failed CI, because Dependabot runs get a read-only token (id-token silently dropped, proven from this repository's own run logs) and the coverage step's OIDC call would throw. Plus: the default sdist shipped the entire working tree to PyPI; the docker smoke never actually loaded lightgbm's native library; LICENSE deletion was invisible to every gate. Twelve findings, all guarded now; 12 of 12 mutants run and killed.

Seven adversarial panels, twenty-nine claims, eleven draws of blood

Every phase ended the same way: refuter agents briefed to *refute*, credited only for demonstrated defects with reproductions, followed by fixes carrying attack-derived regression tests. The pattern in the ledger is the argument for the method: the model and the math survived almost everything; what kept failing was tests that didn't test, presentation that overclaimed, and gates with unlocked corners — precisely the defects that make systems *look* verified.

PHASE	VERDICTS	SHARPEST DEMONSTRATED DEFECT
2A · data layer	2 survived · `3 refuted`	sha-only cache silently served the wrong dataset; schema accepted hub-invalid frames
2B · instruments	3 survived · `1 refuted`	a tautology in the WIS validation gate let crossed quantiles yield negative dispersion
2C · model	3 survived · `1 refuted`	vacuous crossing test: deleting the quantile sort passed everything; 210/212 real tasks crossed
2D · reports	3 survived · `1 refuted`	league rows sorted lexicographically by WIS-as-string; blank-figure PNGs passed the checks
2E · submission	3 survived · `1 refuted`	the `||`-mutant: cron-reachable live job passed all seven substring honesty tests
F · dashboard	3 survived · `1 partial` · 9/9 mutants killed	undrawable Puerto Rico set the map's color scale; fresh clones failed the typecheck gate
G · release	1 survived · `3 partial` · 12/12 mutants killed	Dependabot PRs would have broken CI post-flip; the sdist shipped the private working notes

Standing process rules that earned their keep, each paid for by a specific failure above: kill claims are facts only after the mutant runs; read refuter output in full before triage; never pipe the gate command; fixtures are recorded from real sources, never hand-built; and committed artifacts must byte-regenerate via an integration test.

Limitations on the record, calendar on the wall

Stated limitations. The LightGBM model's intervals are under-dispersed (50% intervals cover 34–40%), worsening with horizon — sharp but overconfident, visible in the committed calibration curves; season-bagging is the sanctioned fix, parked. Single data source, single target; 2022-23 is not backtestable (the hub's vintage history begins Oct 2023). The dashboard serves a frozen bundle, not a live feed. The 2023-24 baseline replica carries a documented, unexplained 2.4% edge over the official run in that season only. The backtest's known biases cut against the model and are quantified in the golden files.

The calendar. The hub's 2026-27 guidance lands around October — the task configuration gets re-verified then. The season opens around November: the weekly live-submission path gets implemented and the first real forecast goes to CDC only on an explicit human go, through the same runbook discipline that governed everything above. Until then the cron runs dry — forecast, validate, artifact, skip.

The record. Registered as `JGracey-prime_radiant` (hub PR #3696); v0.1.0 on PyPI; the dashboard live; the repository public with the full commit history, league tables, and stated gaps.

AFTERWORD · 2026-09-01, LATER THAT DAY

Coverage to 100 — and the gate raised to hold it

After this record was first assembled at 95.95% coverage, the remaining 39 uncovered lines were closed the same way everything above was built: each missed line read and classified honestly (logic guards → real unit tests; network and I/O boundaries → recorded real responses replayed through a patched boundary; nothing swept under an unexplained exclusion), every new test proven to bite by running a mutant and watching it fail, and the number landed at **100.00% — 963 statements, 0 missed, every file 100%**. Then the coverage gate itself was raised from 85 to 100 in the Makefile, the CI workflow, and the honesty tests: from that commit forward, an untested line fails the build structurally. Two confessions belong in the record too: one intermediate commit shipped a type error because the gate was run piped through `tail` — the exact failure mode this project had already documented once — and one test was first written to an assumed contract that the code's own schema rejected, which is how we learned the sorting responsibility lives in the model layer. Both were caught, fixed, and written down. The final suite: 335 offline tests plus the network integration suite, at a gate that can no longer drift.

Provenance. Built agent-assisted with Claude Code (Anthropic), 2026-08-30 to 2026-09-01, under human-gated go-live controls: the agent proposed, built, and verified; every outward action — hub registration, deployments, the public flip, the first release — waited for Jeremy Gracey's explicit word. Every phase was adversarially verified by independent refuter agents before being declared done. Review level: full. Errors land on the human, not the model. The machine-checkable declaration lives at AI-USE.md; the reproducible evidence behind every figure lives in the repository's `reports/`, `serve_data/`, and `tests/golden/`.

`prime-radiant v0.1.0 · assembled 2026-09-01 · MIT`