PUBLIC BENCHMARK METHODOLOGY / 2026

Measuring change. Informing decisions.

AI Stupid Level public methodology for continuous model evaluation

From executable evidence to longitudinal model intelligence

Controlled benchmarking and explicit data provenance.
Version-aware drift analysis and failure attribution.
Production routing informed by measured quality and delivery outcomes.

METHODS / MEASUREMENTS / PUBLIC REVIEW

Public methodology edition 2026 - Revision 3, 15 September 2026

Prepared for public technical review, research discussion and community critique

Live task identities, hidden tests and selected production calibration details are intentionally withheld.

PUBLIC METHODOLOGY

1. Executive technical overview

AI Stupid Level (ASL) connects repeated model evaluation, longitudinal change detection and request-level model selection in one operating platform. The core question is not only which model scores highest once, but whether a model continues to perform a known workload over time, how repeatable its outcomes are, and how that evidence should influence model selection.

What this public edition is for

This document explains the measurement principles, benchmark surfaces, statistical interpretation and evidence-to-routing logic needed for external review.

It intentionally does not publish the complete live task bank, hidden test cases, full prompt-variation catalogue, exact detector calibration constants, internal source paths, operational schedules or production routing constants. The objective is to make the methodology inspectable without turning the live benchmark into a training target or a replication cookbook.

An integrated measurement cycle. Controlled requests produce task-level execution evidence. Repeated trials expose variability. Configuration identities define comparable time series. Longitudinal analysis asks whether behavior changed within the same instrument. Attributed production outcomes can then refine routing decisions.

Recorded evidence in 2026

Evidence type	Count	Unit
Canary score records	59,805	Lightweight early-warning results
Logged tool executions	102,195	Individual tool calls with result and latency
Captured raw outputs	89,710	Returned model text and extraction outcomes
Per-test-case results	52,535	Individual pass/fail records
Coding task runs	178,101	Individual coding task executions across trials
Reasoning sessions	4,598	Multi-turn reasoning sessions
Tool sessions	62,837	Sandboxed tool-use sessions

These are distinct evidence types and are not added together as a single benchmark total. Every empirical claim in this document should be read with its own cohort, sampling unit and observation window. Execution evidence - runs, outputs, tool executions and test-case results - is real regardless of date. Only score rows before about 10-11 September 2026 can be synthetic placeholders; those rows are marked.

REVISION 3 / CORRECTIONS OF RECORD

0. What changed since the 8 September edition

13 September 2026 - Coding instrument. The coding suite is now predominantly repository bug-fix tasks graded by visible and hidden tests. Its corpus was revised again on 14 September, and the coding score curve became linear on 12 September 2026.

13 September 2026 - Reasoning coverage. Every model now runs all four reasoning tasks every day. The previous implementation ran one task per day by rotation, which made between-day comparison unsound because successive days did not measure the same workload mix.

13 September 2026 - Canary redesign. The canary now runs two trials per task and compares a model's recent probes with that model's own prior week using a two-sample test that requires both statistical significance and a minimum effect. The previous threshold rule had no significance test. Its 446 historical incidents were retracted.

13 September 2026 - Drift series separation. Drift detection now runs separately per suite on each suite's own series. The previous implementation interleaved three suites that sit on different scales. Forty-six provider-level incidents derived from that implementation were retracted. A further 1,031 historical change points from the same blended series remain in the record and should be treated as unreliable.

11 to 13 September 2026 - Missing-suite handling. A missing or expired suite now drops out of the combined score and the remaining suite weights renormalise. A composite from fewer than three suites is shown but not ranked. Coverage is stated on the row.

13 September 2026 - Provider refusals. Provider refusals are recorded as events. The declined task drops out of the score, the row is marked partial with its coverage, and nothing is imputed.

13 September 2026 - Ranking uncertainty. Leaderboard ranks now use a two-sample comparison based on each model's run-to-run standard error. Models inside each other's noise share a rank. A row graded on fewer tasks than its neighbours is never called tied. The rule that a row graded on fewer tasks than its neighbours is never called tied was added on 14 September 2026.

Standing disclosure - Historical score placeholders. Score history before roughly 10-11 September 2026 contains synthetic placeholder rows generated during a provider-credit outage. They are marked, excluded from period views and from drift baselines, and a decision on their removal is scheduled for 12 October 2026. Measured data resumed about 11 September 2026.

14 September 2026 - Operational resilience. A benchmark sweep interrupted by maintenance now resumes for the models it had not reached. A missed daily run is caught up within hours. A provider that is not answering is skipped and caught up later. None of these paths publishes a synthetic score.

FROM A CONTROLLED TASK TO AN EXPLAINABLE MODEL CHOICE

2. Measurement framework

From a controlled task to an explainable model choice

Layer	Core question
Request contract	Were models evaluated under declared and comparable conditions?
Capability	Did the returned work satisfy the task and evaluator?
Repeatability	Were outcomes, tool choices and resource use consistent?
Scoring and uncertainty	What is aggregated, and what does the uncertainty represent?
Longitudinal analysis	Did behavior change within a comparable instrument?
Reporting and routing	How are measured signals interpreted and used for selection?

Four principal benchmark surfaces

Surface	Configured cadence	Unit of work
Canary	Hourly	A small fixed probe, two trials per task, compared against the model's own prior week
Coding	Every four hours, six sweeps a day	7 tasks x 7 trials per model per sweep; every model runs every task in every sweep
Reasoning	Daily	Four multi-turn sessions per model per day, one per task; all four every day
Tooling	Daily	9 tasks, one session each, on a real tool surface in an isolated sandbox

Interpret each layer separately

Execution-based correctness, deterministic heuristics, repeatability metrics, statistical detector alerts and routing scores answer different questions. A provider fingerprint supports attribution; an uncertainty interval describes uncertainty under assumptions; a routing score expresses a configured decision rule.

A scientific result needs a declared unit of observation, a comparable evaluation contract and a traceable output. ASL preserves model identity, task identity, request conditions, execution evidence, measurement outputs, methodology identity and time so that a score can be connected back to the evidence that produced it.

THE BENCHMARK IS AN INSTRUMENT

3. Fairness, integrity and contamination controls

The benchmark is an instrument

Cross-model comparison requires a declared request contract and explicit treatment of provider exceptions. ASL applies a common request policy where provider support allows it, while recording cases where a provider or reasoning mode rejects or overrides a requested setting.

Control	Public description	Why it matters
Common request contract	Temperature, output budget and allowed request fields are controlled where supported.	Reduces avoidable cross-model configuration differences.
Provider exceptions	Unsupported or provider-imposed behaviors are identified rather than silently normalized.	Prevents false comparability.
Batch-level variation	Surface identifiers and prompt envelopes are deterministically varied between batches.	Reduces exact-response reuse and dependence on one phrasing.
Prompt equality within a batch	Every model receives the same task variant for the same batch.	Keeps the comparison fair within that run.
Execution-based evaluation	Generated code is executed against test cases where direct execution is available.	Avoids making a changing LLM judge the primary correctness authority.
Sandboxing	Tool and code execution are network-isolated and resource-bounded.	Limits side effects and keeps evaluation conditions controlled.
Methodology identity	Score-moving benchmark changes create a new configuration identity.	Prevents methodology edits from being misread as model drift.
Hidden tests	Repository tasks are graded by executing tests the model was shown plus tests it never saw; the hidden set carries three quarters of the grade.	A fix that passes only the tests it was shown has silenced the symptom and left the defect; the suite exists to catch that.
One output floor for all families	Reasoning models draw thinking tokens from the same output budget as the answer; one shared floor applies to every model family.	A floor set per family is a handicap set per family.
Refusals are not failures	A provider decline is detected from the provider's own signal, recorded as an event, and the task drops out of the score.	A refused task is one the model could not be asked, not one it could not do.
Whitelist	Only the models listed on the board are evaluated.	Nothing off the board consumes evaluation or appears in a ranking.

On 14 September 2026 one hidden assertion in one coding task was found to grade an implementation choice rather than the reported bug, and was corrected. The correction and its evidence are part of the record.

What these controls do not prove

No single control establishes immunity to benchmark contamination, provider-side recognition, hidden serving changes or task-family memorization. The goal is to reduce these risks, make them observable where possible and avoid overstating what the instrument can prove.

ASL also records served-model metadata or provider fingerprints when available. A changed identifier is useful evidence for investigating a serving transition, but it does not by itself prove that weights changed or that a measured quality shift was caused by that transition.

DIFFERENT TASKS ANSWER DIFFERENT QUESTIONS

4. Capability surfaces

Different tasks answer different questions

Coding

Since 13 September 2026 the primary coding suite is predominantly a repository bug-fix instrument. The model receives a small multi-file Python project, a complaint written the way a customer would write it - naming no file, function or cause - and the real failing test output. It must localise the defect itself and return one corrected file. Grading executes the visible tests plus a hidden set. Six of the seven current tasks use this shape; one single-function task remains.

The per-task outcome is the median over seven trials. The score curve is linear: it uses no exponents, gates or variance terms. Correctness carries the majority of the weight. Exact live task identities, hidden tests and weighting constants are not published in this edition.

On 14 September three tasks that every model passed in every sweep were retired. One task was added after a validation campaign that ran seventeen candidate tasks against all 24 models on the board and shipped only the one that at least a fifth of the fleet failed across three independent runs.

The 2026 fleet solves small-repository repairs whose requirement is stated in the complaint at or near ceiling. The suite's discriminating power therefore comes from a small number of tasks that each turn on one inference the codebase supports and the complaint withholds. Coding scores are correspondingly coarse, and the leaderboard's tie rule exists because of it.

Reasoning

The reasoning suite uses multi-turn sessions to test sustained behavior rather than a single isolated answer. The sessions examine specification adherence, use of information introduced earlier in the exchange, coherent code modification and long-context task completion. Where possible, code turns are validated by execution; other turn types use declared deterministic checks or heuristics.

Every model runs all four sessions every day. The published reasoning figure is the mean of four per-task composites over nine scoring axes, not a pass rate. A model can pass every turn of every task and still not publish 100.

On 15 September 2026 four axes were retracted from this composite and the reasoning benchmark configuration was restarted: two of them had been identical for every model in the fleet, and the other two had measured the length of a model's replies rather than their content. The same revision changed how a model's code is applied to the task repository, after a replay of stored sessions showed that thorough multi-part answers were being mishandled by the harness and scored as failures. Reasoning scores published before and after that date are not comparable, and the reasoning suite's change detector restarts its baseline from it.

Tooling

The tooling suite gives a model a real, sandboxed tool surface spanning file inspection, modification, search and command execution. It measures whether the overall task worked, whether appropriate tools were selected, whether parameters were correct, how efficiently the model acted, whether it recovered from errors, and whether it respected context and safety constraints.

The suite contains nine tasks, one session each per day, scored over seven axes. Until the sandbox image was fixed in September 2026, two of the nine tasks could not be passed by any model because the image lacked the interpreters those tasks required. Tooling scores from before that fix are not measurements of models.

Canary

The canary is a lightweight hourly probe designed for early warning, not for a confident degradation conclusion by itself. It is intentionally small and statistically weak.

It now runs two trials per task. The detector compares the model's recent probes with its own prior seven days using a two-sample test and requires both statistical significance and a minimum effect before opening an incident. It does not open a second incident while one is already open.

The earlier design used a fixed percentage threshold with no significance test on a probe whose answers were being truncated for some vendors. It produced 446 incidents over eleven months. All 446 were retracted on 13 September 2026 and are excluded from every published count.

Capability is not availability

A provider delivery failure, caller authentication problem, quota issue or local transport problem is not automatically treated as evidence that the model lacks capability.

ASL separates valid task outcomes from unavailable measurements and carries that distinction into reliability analysis and routing. A provider that is not answering is probed and then skipped for the whole sweep rather than scored. Skipped models are caught up when the provider answers again. A sweep interrupted by maintenance resumes for the models it had not reached. A missed daily run is caught up within hours. None of these recovery paths publishes a synthetic score.

CONSISTENCY IS ITS OWN DIMENSION

5. Reliability and repeatability

Consistency is its own dimension

Capability asks whether the work was completed. Repeatability asks whether comparable runs produce stable outcomes, similar action patterns and similar resource requirements. ASL keeps these dimensions separate rather than compressing them into a single reliability number.

Dimension	Method family	Question
Outcome consistency	Agreement across repeated success/failure outcomes	Does the model reach the same result on comparable attempts?
Trajectory distribution	Pairwise Jensen-Shannon similarity over tool-use distributions	Does it use similar mixtures of tools?
Trajectory sequence	Normalized edit-distance similarity over ordered actions	Does it use tools in a similar order?
Resource consistency	Variation across latency, token use and action count	Is the resource cost of completing the work stable?

Failure direction matters. A model can be highly consistent because it succeeds repeatedly or because it fails repeatedly. ASL therefore interprets outcome agreement alongside pass rate, task coverage and the direction of the result.

Coverage guards matter. Model-level consistency should not be inferred from a tiny surviving subset of tasks. Results are interpreted with task coverage and nullable values where the available comparison set is insufficient.

Disagreement between dimensions is useful. A model can choose similar tools but in a changing order, or maintain similar outcomes while using highly variable resources. Those differences are information, not noise to be hidden by an all-purpose reliability score.

No extra model calls are required for retrospective repeatability analysis

The reliability layer can be computed from already-recorded trial and tool-execution evidence. This keeps the repeatability view tied to the same observed work rather than introducing a separate evaluation surface.

A SCORE IS A CONFIGURED SUMMARY

6. Score composition and uncertainty

A score is a configured summary

ASL publishes suite-specific views and a combined quality view. Coding, reasoning and tooling remain distinct measurements; the combined view gives coding the greatest influence and uses reasoning and tooling as additional capability surfaces. Exact production weighting constants are intentionally omitted from the public edition.

View	Interpretation
Coding	Executable coding performance under the current coding instrument
Reasoning	Performance over defined multi-turn reasoning sessions
Tooling	Task completion and tool-use behavior in the sandboxed tool environment
Combined	Configured summary used for broad quality ranking; not a universal measure of intelligence

Uncertainty metadata

Every published score carries a standard error derived from run-to-run repeatability on the current configuration, using the last five measured runs. The combined score's standard error is the weight-combined error of its contributing suites, and the reported interval is centred on the published score.

Leaderboard ranking uses that uncertainty directly. A model's rank is one plus the number of models leading it by more than the two-sample threshold at the 95% level. Models inside each other's noise share a rank and are marked as tied on the board. A row graded on fewer tasks than its neighbours because a provider declined some tasks keeps its position, but is never called tied and shows its coverage.

A missing or expired suite contributes no default value. It drops out of the composite and the remaining suite weights renormalise; the missing coverage is stated on the row. A composite from fewer than three suites is still shown with its score but is excluded from ranking, because dropping the hardest suite would otherwise reward a model for being measured less.

Interpretation rule

A composite score is a configured decision summary. A repeatability score is not a pass rate. A detector statistic is not causal attribution. Each quantity should be used for the question it was designed to answer.

DRIFT ONLY MAKES SENSE WITHIN A STABLE INSTRUMENT

7. Versioning and baseline continuity

Drift only makes sense within a stable instrument

A change detector asks whether a model moved relative to its own past. That comparison is meaningful only while the measurement instrument is sufficiently stable. If a task changes, a test becomes stricter, a prompt contract changes or a scoring rule is retuned, every model can move even when the models themselves did not.

Configuration identity

ASL assigns a content-based benchmark configuration identity to score-moving inputs. The hashed inputs include task definitions, repository fixtures including their hidden tests, the bug reports, the trial count, the tasks per sweep, scoring weights and the score-curve shape. When a covered input changes, the affected time series begins a new baseline. Non-score-moving edits such as comments, logging changes or unrelated refactors should not reset historical continuity.

Twenty-two configurations have been minted to date. The coding series restarted on 14 September 2026 after a deliberate change that bundled a grading correction and a corpus revision into a single reset while the standing baseline was four sweeps old. Every reset restarts the detector's cold start. September 2026 saw several resets because the instruments were being corrected.

Scenario	Expected interpretation
Material score drop with the same configuration	Eligible for longitudinal change detection
Identical score drop at a configuration change	Baseline transition; do not call it model drift without separate evidence
Historical row with unknown configuration identity	May remain visible, but comparability is weaker and must be stated

Why this matters publicly

A leaderboard can hide instrument changes inside a single number. Version-aware analysis makes the comparison boundary explicit, so a methodology revision does not silently rewrite the model history.

Within-model drift analysis therefore requires more than a timestamp. It requires the model, suite, effective request policy, benchmark configuration and relevant provider metadata needed to establish that the observations belong to the same measurement regime.

CHANGE DETECTION, NOT CAUSAL STORYTELLING

8. Drift detection

Change detection, not causal storytelling

ASL uses a downward Page-Hinkley style detector over a sequence of daily medians. The detector is designed to accumulate evidence when a model remains below its learned historical level and to reset after a confirmed change so that it can learn the new regime. The detector runs separately for each suite on that suite's own series, never blended, because the suites sit on different scales.

Calibration and regimes

Detector sensitivity is calibrated against historical behavior. Model-specific historical variability is used when interpreting stable, volatile, degraded and recovering regimes. Exact production thresholds, cold-start lengths, rearm settings and secondary-scan constants are withheld from the public edition so that the live detector remains less gameable.

A second screening layer

The second layer is the canary described in Section 4. It runs hourly and uses two trials per task. Its detector compares a model's recent probe results against its own prior seven days using a two-sample test, and requires both statistical significance and a minimum effect before opening an incident.

Measured operating characteristics

The following measurements describe the detector as an instrument, not its withheld production threshold. A sustained step decrease was injected into a stationary series, with 400 repetitions per cell. Detection was counted if it occurred within 30 days; delay is the median number of days to first detection.

Noise regime	Sustained decrease	Detection within 30 days	Median delay	False-alarm rate
Typical of 20 of 24 models; run-to-run standard deviation 1.5 to 3 points	3 points	88-91%	15-18 days	0 to 0.02 per 100 days
Typical of 20 of 24 models; run-to-run standard deviation 1.5 to 3 points	5 points	100%	7-8 days	0 to 0.02 per 100 days
Typical of 20 of 24 models; run-to-run standard deviation 1.5 to 3 points	8 points	100%	4 days	0 to 0.02 per 100 days
Noisiest models; standard deviation about 5 points	3 points	84%	11 days	About 0.55 per 100 days
Noisiest models; standard deviation about 5 points	5 points	98%	5 days	About 0.55 per 100 days
Block-bootstrapped real histories with genuine changes removed	-	-	-	Fleet rate 1.16 per 100 model-days, about one spurious alert per model every 86 days, dominated by the noisiest models

Current state

As of 15 September 2026 the Page-Hinkley detectors for all three suites are in cold start after configuration changes and cannot fire before about 24 September 2026. The canary's baseline on its current configuration completed on 14 September 2026, and it is the one layer able to report until the Page-Hinkley detectors arm. The platform labels the cold start warming up rather than showing an empty incident list as if it were a clean bill of health.

Corrections of record

On 13 September 2026 ASL retracted 446 incidents produced by the previous canary rule and 46 provider-level incidents derived from the previous blended-suite drift implementation. A further 1,031 historical change points from that same blended series remain in the record and should be treated as unreliable.

WHAT REPEATED MEASUREMENT REVEALS

9. Longitudinal observations

What repeated measurement reveals

The within-day versus between-day analysis published in the 8 September edition is retired. It used 31,702 rows across 51 models and reported medians of 3.21 and 8.27, pooled values of 10.10 and 9.75, and excess-over-sampling figures of 1.9x and 1.1x. That analysis was computed on the previous amplified score curve and on series that include synthetic placeholder score rows, so it should not be used as evidence about the current instrument.

Measured within a single benchmark configuration on the linear curve, using measured rows only, the fleet is quieter than the detector's calibration range. On configuration 19 the median model's run-to-run standard deviation is 0.25 points, with 16 of 24 models below 1.5. On configuration 22, the current coding instrument, the median is 0.95 points with 14 of 24 below 1.5. Four models on configuration 19 and three on configuration 22 exceed 3 points, the noisiest near 8.3 to 8.7. Two consecutive sweeps on an unchanged configuration agreed within one point for 17 of 24 models on configuration 19 and 15 of 24 on configuration 22.

These low figures should not be read as instrument precision. A run-to-run standard deviation near zero on this suite partly reflects how coarse the coding score is: many models return an identical score sweep after sweep because the corpus resolves into few distinct values. The detector benefits from the quiet, but the quiet is a property of a quantised score as much as of a stable model.

Measuring this quantity across a configuration boundary is meaningless and produces standard deviations of 8 to 18 points for every model, because a configuration change moves the whole fleet. All figures here are within one configuration.

Study	Observation	Interpretation boundary
Current coding repeatability, September 2026	Median run-to-run standard deviation 0.25 points on configuration 19 and 0.95 on configuration 22; 16 of 24 and 14 of 24 models respectively below 1.5 points; three to four models above 3, the noisiest near 8.3 to 8.7. Two consecutive unchanged sweeps agreed within one point for 17 of 24 models on configuration 19, 15 of 24 on configuration 22.	Within-configuration, measured-rows-only evidence; not a cross-suite variance claim. Low values partly reflect a coarse, quantised score rather than instrument precision.
Cache-collapse check, historical observation retained from the 8 September edition	805 batches examined; output length varied across trials in 78.1% of batches.	Rules out universal identical-completion reuse in those batches; does not rule out all caching or benchmark recognition.
Correctness agreement, historical observation retained from the 8 September edition	16 models with both per-trial records and axis scores agreed within roughly 0.6-0.8 points.	Supports approximate agreement in that observed subset, not universal interchangeability.

Why continuous observation matters

Historical model series can spend most days inside a narrow band while still producing short-lived low-score episodes. Weekly or monthly snapshots can miss those events entirely. Continuous measurement is therefore valuable even when a model returns to its prior level before the next conventional benchmark refresh.

Do not over-read a single dip

A low task result can be localized to one workload, one provider condition or one temporary boundary effect. Aggregate change should be traced back to task composition, configuration identity and delivery metadata before being generalized.

SEPARATE MODEL BEHAVIOR FROM DELIVERY CONDITIONS

10. Attribution and reporting

Separate model behavior from delivery conditions

ASL combines per-request failure classification with cross-model correlation. The purpose is to distinguish a valid task failure from a provider delivery problem, a caller configuration error or a shared infrastructure condition.

Class	Representative cause	Treatment
Model/provider delivery	Provider error, overload, timeout or throttling	Eligible for delivery reliability analysis when attribution is sufficient
Provider not answering	Health probe and live confirmation fail	Whole provider skipped for the sweep; nothing scored; caught up when it answers
Caller request	Malformed payload or unsupported request construction	Do not treat as model quality
Caller authentication	Missing or invalid caller credentials	Do not penalize the model
Caller quota	Caller account quota or entitlement	Do not penalize the model
Transport	Local network or transport fault	Do not penalize the model
Unknown	Insufficient evidence	Leave unassigned rather than forcing attribution

Provider-wide interpretation requires correlation across multiple tracked models rather than a single failing endpoint. Simultaneous movement across providers motivates a broader platform investigation. Correlation helps localize a problem, but it does not establish cause.

Reporting rules

Rule	Meaning
Compare with the model's own history	A baseline represents that model's measured level under the comparable instrument
Match the suite	Coding, reasoning and tooling distributions answer different questions
Match the configuration	Methodology changes define comparison boundaries
Declare the basis	State whether the result is a threshold, trend, variance or availability statement
Preserve coverage and time	Report the sampling unit, sample count, observation window and chronology

MEASUREMENT BECOMES A DECISION INPUT

11. The Smart Router

Measurement becomes a decision input

ASL exposes an OpenAI-compatible routing layer that can select a model per request. The public methodology describes the decision structure while withholding exact production calibration constants and ranking formulas.

Decision input	Role in selection
Benchmark quality	Ranks eligible models on the capability surface relevant to the caller's strategy
Caller constraints	Filters by provider availability, cost, latency, tool support, streaming and exclusions
Fallback policy	Allows an eligible alternative when the primary request fails
Observed delivery outcomes	Can adjust rankings when sufficient model-attributable production evidence exists
Request-level logging	Preserves selected model, rationale, candidate pool, latency, tokens, estimated cost and outcome metadata

Evidence earns influence

Sparse production feedback is shrunk strongly toward the benchmark prior, so one or two requests cannot overturn a mature benchmark ordering. As model-attributable observations accumulate, delivery reliability and observed latency can receive more influence. Caller ratings are bounded because they are sparse, self-selected and potentially gameable.

Why the benchmark and router remain conceptually separate

The benchmark measures model behavior under controlled conditions. The router makes a production decision under a caller's actual constraints. A model can have the highest benchmark quality and still be the wrong choice for a specific request because of latency, cost, unavailable credentials, missing tool support or recent delivery reliability.

The routing product is a heuristic, not a probability

Router performance should be evaluated on delivered outcomes for the target workload. The decision logic is inspectable, but the final ranking is not presented as a calibrated probability that a request will succeed.

WHAT ASL CLAIMS - AND WHAT IT DOES NOT

12. Scientific boundaries and public reproducibility

What ASL claims - and what it does not

Surface	Current interpretation boundary
Coding	Python repository bug-fix tasks and one function task, graded by executing visible and hidden tests; no cross-language generalization claim
Reasoning	Defined multi-turn tasks with execution-based or deterministic heuristic evaluators
Tooling	The exposed sandboxed tool surface and tiered task families
Repeatability	Outcome, trajectory and resource consistency inside comparable windows
Safety	Limited benchmark axes plus separate adversarial protocols; not a comprehensive safety guarantee
Cost	Maintained price estimates; not a direct measurement of every provider invoice or discount
Creative quality	No dedicated creative-writing benchmark in this edition
Historical data	Score rows before about 10-11 September 2026 may be synthetic placeholders and are marked; only measured rows support research claims

Public reproducibility contract

A reproducible public result should identify the workload family, model and provider identity, request contract, evaluator type, analysis cohort, benchmark configuration, estimator or detector family, observation window and sampling unit. Historical claims should retain enough evidence to distinguish a definition, a scheduled attempt and an observed empirical result.

Intentionally withheld from the public edition

Complete live task identities, hidden tests, full prompt variants, exact detector calibration constants, internal source maps, table names, cron schedules and production routing constants. These details can be shared selectively for technical diligence or research review when appropriate, while keeping the public benchmark less vulnerable to benchmark-specific optimization.

References

[R1] Rabanser et al. (2026), Towards a Science of AI Agent Reliability. Multidimensional reliability ideas informing ASL's consistency and robustness adaptations.

[R2] NIST/SEMATECH e-Handbook, CUSUM Average Run Length. Statistical process-monitoring reference for interpreting observations before a signal.

[R3] SciPy documentation, **scipy.stats.sem**. Reference for standard error and the distinction between mean-based uncertainty and uncertainty for other estimators.

Public methodology principle

ASL aims to publish enough methodology for serious external criticism while preserving the integrity of the live measurement instrument. The benchmark should be inspectable, but it should not become easier to optimize against than the real workloads it is intended to measure.