A Single LLM Is an Incomplete Code Reviewer

Evidence that Independent Review by Multiple LLM Families Recovers Code Defects Any One Model Misses

Tony Stone
Independent Researcher

June 2026

A single-organization, code-review-focused longitudinal case study (April–June 2026)

Status. Pre-print draft for practitioner dissemination. Single-organization case study; conclusions are directional, not a universal benchmark. Every quantitative claim is traceable to the measured corpus; claims the data cannot support are marked untested. Recall figures were recomputed by a pipeline whose self-test reproduces a frozen baseline exactly before any new data is added.

Abstract

Background. Teams increasingly route code review through a single large language model (LLM). Code review is the highest-stakes detection task an LLM review pipeline performs. We test whether one model's code review is complete relative to independent reviews by several different model families.

Primary hypothesis (H). A single code-review pass by one LLM detects materially fewer of the confirmed code defects than independent reviews by multiple different LLM families; i.e., single-model code review is incomplete relative to a multi-family panel.

Method. We analyzed every multi-model code review in a live software team's corpus that carried a human-reconciled answer key: 18 code/mixed artifacts, 154 confirmed issues, reviewed by eight model versions across five providers (April–June 2026). For each, we measured per-model recall against the reconciled confirmed-issue set (a deliberately generous denominator, conditional on the artifacts each model reviewed), with Wilson 95% confidence intervals, pairwise cross-family finding overlap (Jaccard), per-model unique-contribution counts, and a provider-coverage curve. Each model's report was produced by the team's standard two-pass self-adversarial protocol, so single-model figures reflect best-effort, self-corrected output. Coding and reviewing were done in separate, independent sessions, so a same-model review is a cold read, not an author reviewing its own pull request.

Results (what the data supports). On code, no single model exceeded ∼64% recall; among the non-Gemini models a single model missed roughly 35–53% of confirmed code defects (a typical one missed about half), and the retired Gemini 3.1 was far lower (15.4%). 56.5% of confirmed code defects (87/154) were found by exactly one model, cross-model overlap on code was low (Jaccard median ∼0.37). The provider-coverage curve shows the largest marginal gain from adding the second, different-provider model (33.6% → 57.1%), with diminishing returns thereafter; every provider owned code defects no other recovered. Severity-weighted recall tracked raw recall.

Results (what the data does NOT support). Three things are NOT established: that repeated passes of one model vary (untestable, no identical re-runs); that a newer family version detects more (weak and confounded); and any fine ranking among the seven non-Gemini versions (their intervals overlap and they were not scored on a common issue set). We report these as open questions, not findings.

Conclusion. Within this corpus, a single LLM pass is an incomplete code review, and independent, different-family review recovers the gap. Run a small panel of different-provider models independently on code, reconcile with a human who verifies findings against source, and set expectations at roughly half to two-thirds single-model code recall, not "near-complete."

Reading note: what "recall" means (please read first; the report hinges on it)

This report scores models by recall, a word that is unfamiliar to some readers, and the whole message depends on it. Recall is the fraction of the real defects actually present that the model caught. "GPT-5.5 had 59.5% recall on code" means that, of the confirmed code defects in the artifacts GPT-5.5 actually reviewed, it found about 60 of every 100 and missed about 40. (Recall is measured per model only over the artifacts that model reviewed; the denominator is the human-reconciled set of confirmed-real findings in those artifacts: the answer key.)

We use "recall," not the more familiar "accuracy," on purpose:

- Code review is a detection task, and recall is the standard, precise name for "share of real defects caught", the thing that actually hurts, because a missed defect ships.
- "Accuracy" rewards silence and hides misses. It also credits every correct "looks fine" judgment, and since most code is fine, a reviewer that flags almost nothing can post a high "accuracy" while missing real bugs. Recall ignores the clean code and asks only about the defects.
- "Accuracy" blends two opposite failures. Missing a bug (ships to users) and raising a false alarm (wastes triage time) cost very different things, and a single "accuracy" number hides which is happening. Recall isolates misses; false alarms are a separate axis (precision), reported separately and never blended in.
- Our denominator is deliberately generous (roughly the union of what the models collectively found, not every latent bug that could exist), so these recall figures are an upper bound on each single model's true recall. (Two consequences: the "a single model is incomplete" reading is conservative; and the denominator is not evidence the panel is complete, defects every model missed are invisible to it.)

In short, the entire thesis, "a single LLM is an incomplete code reviewer," is the statement that single-model recall is low. Incompleteness is low recall; that is why recall, not accuracy, is the load-bearing metric throughout this report.

1 Introduction

A widespread practice is to wire one LLM into the code-review pipeline and treat its output as "the review." This paper asks a narrow, testable question about the highest-stakes case: is one model's code review complete, or does it miss defects other models would catch? Code review is where a missed defect is most likely to reach production, so completeness matters most there.

The asset: for ∼two months a software team routed every code review through 2–4 different LLMs independently, then a human reconciled their outputs into a curated record of which findings were real and which model caught each, a longitudinal, multi-rater dataset with a per-artifact answer key.

What we claim, and what we do not. We claim, and support, that a single model's code review is incomplete relative to a multi-family panel, measured as defect recall. We do not claim a panel is "faster" (no latency data), that any one model is "best" beyond the coarse fact that one was clearly weakest (§4.1, the answer key was LLM-assisted and is conflicted, §3.6, and the versions were not measured on a common set), that repeated passes vary (untestable here), or that version upgrades help (weak/confounded here).

2 Hypotheses

Primary.

H. A single code-review pass by one LLM detects materially fewer confirmed code defects than independent reviews by multiple different LLM families. Confirmed if (a) the best single model's code recall is well below the panel's confirmed set, (b) a large share of confirmed code defects are found by only one model, and (c) cross-family finding overlap is low.

Secondary (examined, reported as open, not findings).

- **B1 (intra-model variance):** repeated passes of one model on identical input vary. Untestable here, no identical re-runs.
- **B2 (intra-family improvement):** a newer version detects more. Weak and confounded, version transitions are temporally disjoint and do not agree in direction.

3 Data and methods

3.1 Corpus

The team ran ∼450 independent LLM review files over ∼185 artifacts (a production codebase with client and server components) from early April to early June 2026. This report scores the code subset: every multi-model artifact whose review_type is code or mixed and that carries a human-reconciled synthesis (answer key): 18 artifacts, 154 confirmed issues. Eight are from an earlier analysis of the same corpus; ten are code-review records reconciled for this report, nine newly reconciled (late-May–June 2026) plus one pre-existing transcript. Each artifact was reviewed independently by 2–4 LLMs, then a human reconciled their outputs into the synthesis.

3.2 Model-version attribution

Anthropic-family versions were pinned from the reviewing model's logged API model ID in the team's agent session records (the gold standard this paper recommends in §7.1): Opus 4.6 ($\leq$ Apr 15), Opus 4.7 (Apr 16–May 27), Opus 4.8 ($\geq$ May 28). The recursive session-log extract was refreshed for this report (2,721 transcript files) and every Anthropic reviewer's filename slug was cross-checked against its logged ID, all matched. Other families were run through tooling that did not log model IDs and were attributed by file date $\times$ the team's "always use the latest model" policy: OpenAI GPT-5.4 (to Apr 22) and GPT-5.5 (from Apr 23, all new code reviews here are GPT-5.5); Google Gemini 3.1; Moonshot Kimi k2.6; Alibaba Qwen3.7-max. For replication: log exact model IDs and timestamps at review time; do not trust the artifact's self-reported version.

3.3 Ground truth and the primary metric

The per-artifact synthesis is the answer key; the metric is recall vs.\ synthesis (of the confirmed-real code issues in the artifacts a model reviewed, what fraction it caught). The denominator is deliberately generous (the synthesis-confirmed set, overwhelmingly the union of model findings), so single-model recall is an upper bound and the inferiority reported is conservative. The same generosity means the denominator cannot speak to whether the panel is complete: defects missed by every model are not in it (§6).

3.4 Eligibility and the answer keys

Of the multi-model code/mixed artifacts, 18 carry a synthesis and are scored (one is a zero-defect clean control (both reviewers correctly passed it) and contributes 0/0, inert). The ten new answer keys were Claude-drafted and maintainer-reviewed (the same Claude-assisted process as the earlier set, §3.6): reviewer findings were spot-checked against the actual server/client source (file:line + quoted snippet for the load-bearing claims, with database/schema claims checked live), each finding marked CONFIRMED or REJECTED and its severity recalibrated by user-harm impact; the remaining attributions rest on per-artifact agent extraction (§6). Two of the ten new artifacts are mixed code/design (Artifact M and the swing artifact, §4.4); their issue arrays include a minority of design-half findings, flagged in the data.

3.5 Extraction and statistics

Per-artifact capture records (which version caught which confirmed issue; which findings the synthesis rejected) were built by dedicated agents, then load-bearing claims were spot-verified against source. All statistics were computed in deterministic code: per-version recall, Wilson 95% CIs[1], severity-weighted recall (weights P0=3/P1=2/P2=1/P3=0.5), micro-averaged pairwise Jaccard[2], unique-find counts, and a provider-coverage curve, by a pipeline whose self-test reproduces a frozen baseline exactly (the same corpus's previously-fixed code-subset recall: Opus 4.6 22/36, Opus 4.7 13/30, GPT-5.5 14/23, GPT-5.4 25/43, Kimi 9/23, Gemini 6/39) before any new data is added.

3.6 Conflict-of-interest controls

The syntheses were drafted with assistance from a Claude/Opus-family model that was also one of the reviewers, which can bias that family's apparent recall. We therefore (a) assert no fine per-model ranking as a result (only that one model was clearly weakest, §4.1), (b) anchor on the single-vs-panel comparison and aggregate measures, (c) treat the direction of any same-model bias as unknown (§3.8), and (d) submitted an earlier analysis to two non-author families for adversarial checking (Appendix B). The Anthropic-family code recall here should be read with this conflict in mind, not as a leaderboard position.

3.7 Review protocol: each report is a two-pass, self-corrected product

Every reviewer's output was a two-stage product in one session: Pass 1 used a 176-line reviewer template (fixed scope, P0–P3 severity by user-harm impact, mandatory file:line citations, and adversarial checklists, production-path caller-graph tracing, multi-path parity, non-idempotent side-effect probes, schema-correctness checks, downstream-consumer assessment), and Pass 2 was a fixed prompt directing the model to "perform a more thorough, critical, and adversarial review ...

directly observe every claim ... guard strictly against hallucinations" and revise its own report. The first-pass output was overwritten, so within-model change across passes cannot be quantified; the surviving figures are each model's best, self-corrected effort, which makes the single-vs-panel shortfall conservative.

3.8 Same-model reviews, independence, and the COI policy

Several code artifacts were reviewed by the same model family that had earlier written the code under review (an Opus instance reviewing Opus-authored code). Coding and code-review were performed in separate, independent sessions, so the reviewing instance read the code cold, with no authoring context, the Co-Authored-By trailer marks the code commit, not the review. We therefore score these as independent single-model reviews (the inclusion policy). This is a weaker form of independence than cross-family review, because the reviewer shares model-level priors with the author. The direction of any resulting bias is unknown. In this corpus a same-model reviewer both missed defects that cross-family reviewers caught (R5-1, FCD-1) and uniquely caught the most consequential defects the panel missed (AF9-1, VG2-11); the data do not support a claim that inclusion inflates or depresses the Anthropic estimate in either direction. We adopt inclusion as a policy choice and, separately, make no fine per-model ranking (§4.1), which mitigates the conflict whichever way it runs. The one genuine same-session self-review (a model reviewing a fix it produced in the same session) is excluded.

Policy sensitivity. These figures use the inclusion policy. Under a stricter exclusion policy (counting same-model reviews as non-independent), the six confirmed defects only the same-model Opus reviewer caught would leave the scored set (148 issues), Opus 4.7 would have no code-subset data, Opus 4.8 would rest on one artifact (11/15), and the singleton rate would be 59.5% (88/148) rather than 56.5%. A middle "re-credit only" policy (credit shared findings, do not promote the six Opus-only ones) gives Opus 4.7 27/61 and Opus 4.8 15/25. The single-vs-panel conclusion holds under all three.

4 Results

4.1 Single-model code recall (issue-level, vs.\ the generous synthesis denominator)

Table 1: Single-model code recall (issue-level, vs. the generous synthesis denominator). The highest point estimates sit on the smallest samples with the widest intervals.

Model version	n	Recall	95% CI	sevW Δ
Opus 4.8	28	64.3%	46–79%	+6.8
Opus 4.6	36	61.1%	45–75%	+0.1
GPT-5.5	111	59.5%	50–68%	+2.8
GPT-5.4	43	58.1%	43–72%	+2.6
Qwen3.7-max	88	51.1%	41–61%	−1.4
Kimi k2.6	100	47.0%	38–57%	−3.4
Opus 4.7	64	46.9%	35–59%	+1.7
Gemini 3.1	39	15.4%	7–30%	−4.7

No single model exceeded ∼64% recall on code. The largest-sample estimates, GPT-5.5 (n=111),

Kimi (n=100), Qwen (n=88), Opus 4.7 (n=64), run 47–60%; the higher point estimates (Opus 4.8 n=28, Opus 4.6 n=36, GPT-5.4 n=43) rest on smaller samples with wide intervals. Among the non-Gemini models the best single model missed roughly a third to a half of confirmed code defects, and a typical one missed about half; the retired Gemini was far lower (15.4%).

What the data supports as a ranking, and what it does not. Only a coarse ordering is defensible: Gemini 3.1 was clearly weakest, its interval (7–30%) overlaps no other model's. The remaining seven versions are statistically indistinguishable (every pair of intervals overlaps) and, crucially, were not measured on a common issue set, recall is conditional on participation and no artifact was reviewed by all eight versions, so GPT-5.5's 59.5% (its 111 issues) and Opus 4.6's 61.1% (36 different issues) are not a like-for-like comparison. We therefore report the measured numbers and intervals but make no finer ranking, and specifically no claim that any Anthropic version is "best", the answer-key conflict (§3.6) would, if anything, fall on exactly the top rows. That two versions of the same family land at opposite ends (Opus 4.6 61.1%, Opus 4.7 46.9%) is itself a caution against over-reading the order. Severity-weighted recall was close to raw recall (and higher for several higher-recall versions); in this small sample there is no clear evidence that the higher-recall models preferentially missed severe defects.

Reading the Anthropic rows. Opus 4.6/4.7 carry their earlier code data plus, for 4.7, three newer cold-read reviews; Opus 4.8's 18/28 spans three artifacts (one borderline-mixed; see §4.4). Per §3.8 the direction of any same-model bias is unknown, and per §3.6 these are not ranking claims.

4.2 The anchor result: a single model misses what the panel catches

Four complementary lines of evidence, all derived from the same issue-by-catcher data, so corroborating rather than statistically independent, support H:

1. **The best single model missed a large fraction of confirmed code defects, and the panel caught them.** Best single-model code recall topped out at ∼60–64% (a typical model sat near 47–51%), so the best single model missed ∼35–53% of confirmed code defects (the retired Gemini far more); every confirmed issue was caught by some model in the panel. Each report was a two-pass, self-corrected best effort (§3.7), so this is not a one-hurried-pass artifact.

2. **More than half of all confirmed code defects were found by only one model.** 87 of 154 (56.5%) were singletons, cross-model overlap on code was low. The per-issue catcher-count distribution was {1: 87, 2: 43, 3: 10, 4: 14}. Dropping any single reviewer forfeits its singletons.

3. **Different families find substantially different code defects (low overlap), and each adds unique coverage.** Pairwise cross-family overlap (micro-averaged over co-reviewed artifacts; largest-sample pairs are the most reliable):

   The median pairwise overlap is ∼0.37, and the largest-sample pairs sit at 0.35–0.45, co-reviewing models agreed on only about a third of their combined code findings, and even the largest-sample pairs reached only ∼0.45. (Pairs with union < 10 and the Gemini-paired zeros rest on few co-reviewed artifacts and are indicative only.) Every provider produced confirmed code defects no other caught; per-version unique finds: GPT-5.5 = 21, GPT-5.4 = 15, Kimi = 14, Opus 4.6 = 12, Qwen3.7-max = 9, Opus 4.7 = 6, Opus 4.8 = 6, Gemini = 4 (sum 87).

4. **The coverage curve: the largest marginal gain is the second provider, and no provider is redundant.** Accumulating coverage of the 154-issue confirmed set, averaged over all provider subsets of each size (so no provider is privileged):

Table 2: Pairwise cross-family finding overlap (micro-averaged Jaccard over co-reviewed artifacts; largest-sample pairs are the most reliable).

Pair	Jaccard	shared / union
Opus 4.8 × Qwen3.7-max	0.50	11 / 22
GPT-5.5 × Qwen3.7-max	0.45	30 / 67
GPT-5.5 × Opus 4.8	0.43	9 / 21
Kimi k2.6 × Opus 4.7	0.39	15 / 38
Opus 4.7 × Qwen3.7-max	0.38	9 / 24
Kimi k2.6 × Opus 4.8	0.38	9 / 24
GPT-5.5 × Kimi k2.6	0.36	27 / 76
GPT-5.5 × Opus 4.7	0.35	17 / 48
Kimi k2.6 × Qwen3.7-max	0.32	18 / 56
GPT-5.4 × Opus 4.6	0.29	10 / 34
Gemini 3.1 × Opus 4.6	0.00	0 / 20
GPT-5.4 × Gemini 3.1	0.00	0 / 28

Table 3: Provider-coverage curve: expected coverage of the 154-issue confirmed set, averaged over all provider subsets of each size (so no provider is privileged).

Independent panel size	Expected coverage
1 provider	33.6%
2 providers	57.1%
3 providers	74.6%
4 providers	88.7%
5 providers	100.0%

Marginal gains ≈ 23.5 / 17.5 / 14.1 / 11.3, after an initial provider, the second provider gives the largest additional gain, with diminishing returns. A complementary drop-one view (order-independent): removing a provider forfeits OpenAI 23.4%, Anthropic 15.6%, Moonshot 9.1%, Alibaba 5.8%, Google 2.6% of confirmed code defects, every provider, including the lowest-recall one, accounted for code defects no other recovered. (The ∼100% ceiling at five providers is partly definitional (the denominator is the union of model finds), so the shape and marginal gains are the result, not the absolute ceiling, and the ceiling is not evidence the panel is complete.)

Together: a single model catches roughly half of confirmed code defects; the misses are largely disjoint across families; the second different provider is the biggest recovery; so independent multi-family code review recovers defects any one model misses. H is supported.

4.3 Secondary beliefs (reported honestly as not established)

- **B1 (repeated passes vary):** untestable here. No identical re-runs exist, repeated code reviews were iterative passes on changed code. (§7.4 proposes the experiment.) The protocol's guided second pass always changed the output, but that is guided re-prompting, not B1's unguided identical-input repetition, and its magnitude is unquantifiable.
- **B2 (newer version detects more):** weak, confounded, and direction-inconsistent. Versions are temporally disjoint, so no paired comparison exists. The unpaired signals do not agree in

direction: OpenAI GPT-5.4 → GPT-5.5 was roughly flat (58.1% → 59.5%), while Anthropic's Opus versions span 61.1% (4.6) → 46.9% (4.7) → 64.3% (4.8, small n), not monotone, and confounded by disjoint work, a changing OpenAI counterpart, and the answer-key conflict. We claim no version effect.

4.4 Convergent validity and the swing artifact

The team retired Gemini 3.1 mid-corpus for perceived poor quality, before this analysis; the code data agrees, Gemini had by far the lowest code recall (15.4%), though it still contributed 4 unique code defects, so "lowest" is not "no value." Because the synthesis was unblinded and LLM-assisted, this is convergent/face validity, not independent construct validity. One Opus 4.8 artifact (the swing artifact, Artifact S) is borderline between mixed and design (design-heavy document, but 13 of its 15 confirmed findings are code defects verified against deployed source); it is included as mixed. It is sensitive: dropping it removes 15 issues from the scored set, Opus 4.8 then falls to 7/13 (53.8%), and because three other versions co-reviewed it their figures also shift (GPT-5.5 57/96, Qwen 36/73, Kimi 36/85), the total drops to 139 issues, and the singleton rate rises to 59.0% (82/139). The qualitative pattern is unchanged, but the absolute numbers move, which is why both runs are reported (Appendix A) and the classification is flagged rather than buried.

5 Discussion

The pattern, moderate single-model recall, a high singleton rate, low pairwise overlap, steep early coverage gains, is what ensemble theory predicts[3] when base learners have uncorrelated errors: different providers appear to have different blind spots, so the union recovers proportionally more than any one model. A corroboration of the uncorrelated-errors reading: in this corpus a same-model reviewer (sharing the author's priors) both missed defects a cross-family reviewer caught (on two artifacts: R5-1, FCD-1) and uniquely caught the most consequential defects the panel missed (AF9-1, VG2-11), while the strongest unique catches overall were distributed across all five providers, consistent with differing blind spots rather than a one-directional same-model effect. We present this as interpretation; the corpus directly supports the overlap, singleton, and coverage measurements, not a causal claim about training.

What this does not say: that LLM code review is unreliable, that a panel is complete (it is not, see §6 and §7.1), that one model is best (only that one was clearly weakest), that panels are faster, or that more models are always better (returns diminish past the third provider, and false positives grow). It says one model's code review is incomplete, and that different-family review recovers the gap.

6 Threats to validity

Construct. "Recall vs.\ synthesis" measures agreement with a curated confirmed-issue set (overwhelmingly the union of model findings), not ground-truth recall; it overstates single-model recall (so the inferiority claim is conservative) and makes the coverage curve's absolute ceiling partly definitional. This is conservative for each single model's absolute recall, but it is not evidence the panel is complete, defects every model missed are invisible to this denominator. The answer keys were LLM-assisted and unblinded.

Internal. A Claude-family model co-authored the answer keys and was a reviewer (self-synthesis

bias), hence no fine model ranking is claimed (only "Gemini weakest" is supported), and the direction of any same-model bias is treated as unknown (§3.8). The ten new answer keys are Claude-drafted and maintainer-reviewed; capture records were built by agents with spot-checking of load-bearing claims against source, but the new-artifact non-Anthropic attributions rest primarily on the per-artifact agent extraction, and a correction during this work (an under-recording of one family's catches, found and fixed) shows attribution noise is real, mitigated rather than eliminated, and not quantified by a full audit. Each report is a two-pass self-corrected product; within-model cross-pass change is unquantifiable.

External. Single organization, single codebase, single domain; uneven per-model samples (Opus 4.8 n=28; Gemini retired mid-corpus → survivorship). Directional, not a benchmark.

Statistical. Several per-version code samples are modest (Opus 4.8 n=28, Opus 4.6 n=36, GPT-5.4 n=43, Gemini n=39) with correspondingly wide intervals; the temporal-disjointness of "always latest" precludes paired version (B2) and re-run (B1) comparisons, and the versions were not scored on a common issue set (precluding a fine ranking).

7 Recommendations for practice

7.1 How to configure code review (supported by §4)

1. **Use 2–3 different-provider models, run independently and blind to each other, then reconcile.** A single model missed roughly a third to a half of confirmed code defects here; the misses are recovered by other families; and the coverage curve shows the second provider is the largest marginal gain. Cost scales ∼linearly with panel size, so 2–3, not 5+.
2. **Prefer provider diversity over a single model or repeated passes.** Cross-family code overlap was ∼0.37 (much is complementary), so diversity recovers substantial additional defects. A same-model review adds less (shared priors), though it is not worthless.
3. **Keep a human reconciler as a first-class role**, to merge disjoint findings, remove false positives (including confident fabrications), and resolve disagreements, and have it verify every finding against source; a reviewer's "verified" label is not evidence.
4. **Treat the panel as a high-recall first pass, not a merge gate.** It is not complete. In the team's operational record, multi-round multi-model review still left severe defects for later layers, and reviewers sometimes verified logic and types but not the actual schema identifiers. Retain human review, tests, and staged rollout for high-consequence code.
5. **Log exact model IDs/dates and re-baseline periodically;** here the Anthropic versions were recoverable only because the agent harness logged the model ID per call.

7.2 How to set expectations

1. **Say "recall," not "accuracy," and quote a range.** A single model finds roughly half to two-thirds of confirmed code defects; none here exceeded ∼64%.
2. **Do not promise "consistency."** It is non-deterministic and was unmeasured; instrument it if required (§7.4).
3. **Track defect-escape rate over time** as the process KPI, even a multi-model panel let some code defects reach production.

4. **Budget for false-positive triage;** more models means more findings to sift, including confident fabrications under a "verified" label, the central reason an explicit source-verification step is non-negotiable.

7.3 Aspects practitioners commonly overlook

Severity stratification (here there was no clear evidence that models preferentially missed severe defects, but the sample is small, verify for your pipeline); independence/blinding (if models see each other's output they converge, erasing the diversity that creates the benefit); a reconciler that is human or a different vendor, never one that authored a review in the same session, and that verifies against source; model/version drift (pin and log); prompt sensitivity (hold the prompt fixed when comparing); redundancy $\neq$ diversity (the same model twice does not decorrelate errors); and human-in-the-loop for high-consequence code, because the panel is incomplete.

7.4 Proposed experiments to test what this corpus could not

- **B1 (consistency):** submit identical code-review requests (same model, input, prompt) $\geq$10$\times$ across a sample; report Krippendorff's α[4] and finding-set Jaccard.
- **B2 (version effect):** run predecessor and successor on the same held-out code artifacts (paired), removing the temporal confound.
- **Fine per-model ranking:** to rank models on code beyond "which is weakest," run all candidates on a common held-out set of code artifacts (so recall is measured on identical issues), with a blinded adjudicator.
- **External validity:** multi-organization replication with a blinded, non-LLM or cross-vendor adjudicator and pre-registered analysis.

8 Conclusion

Across two months of real code reviews, a single LLM (even given a two-pass, self-corrected protocol) was an incomplete code reviewer. No single model exceeded $\sim$64% recall on code (a typical one caught about half), so the best single model missed roughly a third to a half of confirmed code defects; 56.5% of confirmed code defects were found by only one model; co-reviewing families overlapped on only about a third of their findings; and the coverage curve shows the second, different-provider model is the single largest recovery of missed code defects, with no provider redundant. Independent review by multiple LLM families recovers code defects that any one model misses. We could not establish that repeated passes vary, that version upgrades help, or any fine ranking among the seven non-weakest models, and we decline to assert them. The actionable conclusion is narrow and well-supported: do not rely on one model's code review; run a small panel of different-provider models independently, reconcile with a human who verifies against source, and expect roughly half to two-thirds single-model code recall.

A Reproducibility

Metric: issue-level recall vs.\ per-artifact human-reconciled synthesis, conditional on the artifacts each model reviewed. **Scored set:** 18 code/mixed artifacts, 154 confirmed issues (inclusion policy, §3.8). **CIs:** Wilson score, 95%. **Overlap:** micro-averaged Jaccard over co-reviewed artifacts.

Coverage: expected coverage of the confirmed set averaged over all provider subsets of each size; drop-one over the full panel. **Version attribution:** Anthropic family from logged per-call model IDs (refreshed extract, 2,721 transcript files); other families by file date × "always latest" policy. **Aggregation:** deterministic code over structured per-artifact capture records; the pipeline's self-test reproduces the frozen baseline code-subset recall exactly (Opus 4.6 22/36, Opus 4.7 13/30, GPT-5.5 14/23, GPT-5.4 25/43, Kimi 9/23, Gemini 6/39) before any new data is added. **Policy:** same-model reviews are scored as independent because coding and review occurred in separate sessions (§3.8); the one same-session self-review is excluded; a real, source-verified defect caught by a single panelist is counted as a confirmed singleton regardless of which model caught it (the same rule applied to all models). The inclusion/exclusion/re-credit policy sensitivity is reported in §3.8, and the swing-artifact sensitivity in §4.4. A sanitized, PII-scrubbed dataset is required before any public data release.

B LLM-assisted adversarial sanity checks

An earlier analysis of this corpus (not this code-focused, inclusion-policy draft) was submitted to GPT-5.5 and Qwen3.7-max as adversarial reviewers instructed to refute its conclusion; both independently re-derived its headline recall figures from source and returned support for the core single-vs-panel conclusion. This is LLM-assisted checking, not independent human peer review, and it did not adjudicate the same-model inclusion policy used here. This code-focused draft was separately subjected to two independent adversarial LLM reviews (different families), whose corrections are incorporated.

C Per-model unique contributions and overlap (code subset)

Per-version unique (single-model) code finds: GPT-5.5 = 21, GPT-5.4 = 15, Kimi k2.6 = 14, Opus 4.6 = 12, Qwen3.7-max = 9, Opus 4.7 = 6, Opus 4.8 = 6, Gemini 3.1 = 4 (sum 87). Issues by number of distinct catchers: 1 → 87 (56.5%), 2 → 43, 3 → 10, 4 → 14. Provider-singletons: OpenAI 36, Anthropic 24, Moonshot 14, Alibaba 9, Google 4 (sum 87). Pairwise cross-family Jaccard: median ∼0.37, largest-sample pairs 0.35–0.45 (full table §4.2). Family-level recall on this subset (OpenAI 59.1%, Anthropic 45.5%, Moonshot 30.5%, Alibaba 29.2%, Google 3.9%, denom 154) is code-only and specific to this COI policy; it is a within-subset descriptive, not a general family benchmark.

D Observed code-review failure modes (qualitative)

Observations from the team's calibration record (not measured rates, per-model-per-version; no model was uniformly weak, each contributed confirmed code defects no other caught):

- **Google Gemini 3.1:** most frequently presented fabricated evidence under a "verified" label (non-existent file paths, invented identifiers), caught only by source-verification, yet still produced unique real code findings other models missed.
- **OpenAI GPT-5.x:** a tendency to over-classify severity, but mode-dependently, pronounced when reasoning abstractly with sparse evidence, largely absent when grounded in live data and file:line citations; in this code subset it caught two false-verify defects (a P0 and a P1) on an artifact two other models passed clean.

- **Anthropic Opus:** strong on local correctness; in this corpus its deepest unique catches were whole-source-omission and silent-false-verify chains, while a cold-read Opus reviewer also missed defects that cross-family reviewers caught (§5), consistent with differing blind spots.

Two cross-cutting observations on code: a guided second pass changed every reviewer's output and its visible thoroughness tracked its quality; and when two different-provider reviewers independently flagged the same source line, that convergence was a reliable high-confidence signal.

References

[1] Wilson, E. B. (1927). Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158), 209–212.

[2] Jaccard, P. (1912). The distribution of the flora in the alpine zone. *New Phytologist*, 11(2), 37–50.

[3] Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Multiple Classifier Systems (MCS 2000)*, LNCS 1857, pp. 1–15. Springer.

[4] Krippendorff, K. (2018). *Content Analysis: An Introduction to Its Methodology* (4th ed.). SAGE Publications.