Comparison Summary v0.2

1. Purpose

This document summarizes replicated evidence produced under the replication stage of the research program. Its function is to compare observed run outcomes against the shared thesis, methodology, and replication standard, and to determine whether the current evidence supports or weakens the central claim. The roadmap places comparison after replicated runs and before failure-mechanism consolidation.

2. Claim Under Comparison

The claim under test is that workflows using a versioned governing artifact preserve cross-surface invariants under iterative change more reliably than code-only conversational modification workflows. The methodology explicitly says this is not a coding benchmark, not a model ranking exercise, not a test of raw model intelligence, and not a claim that AI cannot code.

3. Evaluation Standard

A valid replication requires one shared baseline codebase, one shared baseline contract or equivalent governing artifact, two tracks (Spec-First and Code-Only), at least one same-surface change, at least one cross-surface change, one shared invariant under test for each delta, and one validation method that evaluates both tracks against that same invariant. Cross-surface change is the primary target. A valid cross-surface replication must make it possible for one local mutation surface to be updated, another required mutation surface to remain untouched, and the invariant to fail because propagation was incomplete.

Both tracks are evaluated against the same invariant under test. The workflow under test is not the oracle. For cross-surface deltas, the methodology requires recording which mutation surfaces were required, which mutation surfaces changed, and which required surfaces remained untouched. Findings are judged at the Decision Surface rather than at wording or implementation shape.

4. Run Set Under Comparison

Run 1 used artifact-sync v2.6.3 with Convergence Contract v2.6.3. Its same-surface calibration concerned structured `REPR_FAIL` logging. Its cross-surface target was manual-file preservation across atomic write, staging cleanup, and rename-replace surfaces.

Run 2 used artifact-projection v2.4.11 with Artifact Projection — Convergence Contract v2.4.1. Stages A through C passed in both tracks. Stage D tested unmanaged-file preservation across deletion plus staging sync-back.

Run 3 used a bounded SQLite pager mock derived from real pager invariants. The invariant under test required sync-before-finalization behavior across four independent finalization surfaces: `finalize_truncate()`, `finalize_zero_header()`, `finalize_delete()`, and `finalize_close_memory()`.

5. Run-by-Run Results

Run 1. The same-surface delta produced equivalent Decision-Surface outcomes in both tracks. The cross-surface delta required three mutation surfaces. In the Spec-First track, all three were updated. In the Code-Only track, the atomic write surface was not addressed, the staging cleanup surface was not addressed, and the rename-replace surface was only partially addressed by restoring files after rename. The report records incomplete propagation: 2 of 3 required surfaces were left untouched. It also records divergent cross-surface outcomes: basic preservation passed in both tracks, but atomicity, crash-safety, and full invariant scope failed in Code-Only.

Run 2. Same-surface stages passed in both tracks. Stage D was the cross-surface test. D-01 covered the prompt-mentioned deletion path. D-02 and D-03 covered staging surfaces not named in the prompt. Spec-First passed all Stage D checks. Code-Only failed D-01, D-02, and D-03. The report states that the Code-Only agent modified only `_delete_lrs`, while the staging sync-back path remained untouched. The Spec-First implementation modified both the deletion path and the staging sync-back path because the contract patch explicitly enumerated both surfaces. The run records a failure gradient of 0/4 for ΔB, 0/4 for ΔC, and 3/5 for cross-surface ΔD in Code-Only, versus 0/5 for Spec-First at ΔD.

Run 3. The invariant required four independent finalization surfaces. In the Spec-First track, all four were updated and DELETE, TRUNCATE, PERSIST, and MEMORY all passed. In the Code-Only track, only `finalize_delete()` was updated. DELETE passed, but TRUNCATE, PERSIST, and MEMORY failed. The required-surface completion record is 4/4 for Spec-First and 1/4 for Code-Only. The report's verdict is that the replication supports the thesis because Code-Only left 3 of 4 required surfaces untouched while Spec-First updated all required surfaces through explicit enumeration.

6. Cross-Run Comparison

Run	Cross-surface required surfaces	Code-Only completion	Spec-First completion	Decision-Surface result
Run 1	3	incomplete; 2 of 3 left untouched	all required surfaces updated	diverged on atomicity, crash-safety, full invariant scope
Run 2	2 primary mutation surfaces, plus regression checks	deletion path only	both required surfaces updated	diverged at Stage D; code-only failed D-01, D-02, D-03

Run 3	4	1 of 4	4 of 4	diverged across TRUNCATE, PERSIST, MEMORY

Across all three valid result sets, the same structural pattern appears. Code-Only modifies the prompt-salient surface. One or more additional required surfaces remain untouched when not explicitly named. Spec-First updates all required surfaces because invariant scope is carried in the governing artifact. The difference appears at the Decision Surface rather than only in local code shape. This is consistent with the methodology’s stated immediate mechanism: conversational prompts naturally scope work to the surfaces they mention, while clause-structured contracts force enumeration of affected mutation surfaces.

7. Comparison Result

Current comparison result: **supports the thesis**. The support is narrow. Same-surface behavioral changes were stable, and some same-surface architectural tightening also converged. The observed divergence concentrates at the cross-surface boundary, where omitted invariant-scope enumeration predicts incomplete propagation. This matches the methodology’s topology-centered claim and the replication artifact’s support criteria.

8. Limits

This is mechanism replication, not statistical generalization. The roadmap and related materials frame this work as replicated evidence sufficient for comparison, not as broad population-level proof. The current evidence remains bounded by a small number of systems and predominantly internal replication. Run 2 also records same-session execution for both tracks and operator-applied Stage C architecture, which limits independence even though it does not erase the cross-surface result. Run 1 explicitly records full-knowledge contamination, yet still shows incomplete cross-surface propagation in Code-Only.

9. Roadmap Position

This comparison summary satisfies the roadmap’s reporting-and-comparison step after replication. The current evidence is sufficient to begin a first failure-mechanism or failure-class consolidation draft, because the program now has repeated observation of the same structural pattern across multiple systems and runs.