KNOWING THE PARTS IS NOT ENOUGH: MEASURING COMPOSITION AFTER MATCHED PRECURSOR SUCCESS

Anonymous authors
Paper under double-blind review

ABSTRACT

Aggregate code-generation benchmarks report how often a model solves a problem. They cannot separate two failures: the model never demonstrated the required capability, or it succeeded on a matched precursor task for each constituent yet could not combine them. ConceptBench targets the second case by conditioning composed-task failure on matched precursor success. Each of 74 families over five concept pairs is instantiated as three tasks sharing one semantic core — capability A alone, B alone, and their composition AB — where A and B are projections of the AB rule onto input domains in which the other capability is structurally absent. The outcome is the **raw PPF** (both precursors pass, the composition fails), reported as $P(\text{AB fails} \mid A \wedge B)$ with its eligible-family denominator: a protocol-conditional diagnostic, not an estimate of how often composition fails.

Under one pinned inference protocol, models differ sharply even on the same jointly eligible families. On the same ten families of one concept pair, for which both models passed both matched precursors, Qwen2.5-Coder 14B produces nine raw PPFs and Qwen2.5-Coder 32B none; two of the nine are mechanically verified precursor-coverage gaps, and setting those observations aside the contrast remains 7–0. Eight further matched common-support cells differ by at least three raw PPFs, in both directions. Eligibility selection alone therefore cannot explain this heterogeneity, though the contrast is not shown to be composition-specific. Pair-level rates also vary, but their rank ordering is identical to that of unconditional AB failure (P3 $>$ P2 $>$ P5 $>$ P1 $>$ P4), so pair ordering alone does not identify composition-specific structure. Mechanical screens expose specific confounds; beyond them, mechanistic attribution was not human-adjudicated, and no global count of composition-specific failures is reported. As descriptive external validation, two frontier models with reasoning disabled each produce one raw PPF; with reasoning enabled and a larger output budget, both failures become passes and each produces one new raw PPF.

1 INTRODUCTION

A model that scores well on HumanEval (Chen et al., 2021) has demonstrated that it can solve a distribution of independent programming problems, not that it can combine two capabilities it has separately demonstrated. When a model fails a problem requiring recursion and branching, the failure is equally consistent with "it did not solve the recursion task", "it did not solve the branching task", and "it solved both and cannot put them together." Aggregate pass rates cannot tell these apart.

The third case is where constituent benchmarks actively mislead: passing the matched precursor tasks separately does not certify their composition. ConceptBench isolates that case by construction. Every family is a triple of tasks built from one semantic core. Two tasks project that core onto input domains where one constituent capability is *structurally absent*; the third requires both. A family speaks about composition only where the model has already succeeded on both matched precursors, so eligibility is reported in every cell. *Matched precursor success* means one passing response on each matched precursor under the declared inference protocol, not general mastery of the named capability (§3).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Weak models are eligible on few families, so their rates rest on small, heavily selected subsets: Granite 20B is eligible on 15 of 74 families and none at all for two pairs. Strong models are eligible almost everywhere, so the frontier panels serve only as descriptive external validation; the controlled open-weight panel, whose inference protocol we control, is the main measurement panel.

Contributions. **(1) A matched semantic-projection instrument.** ConceptBench's contribution is not conditioning on precursor success itself: Press et al. (2023) named the compositionality gap, and EvoEval applies a related conditional in code (Xia et al., 2024) (§2). It is a matched semantic-projection instrument: A and B are restrictions of one AB semantic function onto domains where the complementary capability is structurally absent, enabling eligibility-controlled comparisons on matched families, with projection-equivalence audits and mechanical forensic screens (§3, §6). **(2) Model heterogeneity on fixed common support.** Under one pinned inference protocol, models differ sharply even on exactly the same families for which both show matched precursor success: 9 raw PPFs against 0 on the same ten P5 families, and 7 against 0 after setting aside two mechanically verified coverage gaps (§5.1). **(3) Measurement boundaries.** Pair rank ordering is identical with and without conditioning, so pair-level raw PPF does not isolate composition-specific difficulty (§5.2); mechanical screens expose confounds inside raw counts; and program-level case studies show failures consistent with integration errors, as individual cases rather than a ledger (§6).

Raw PPF is therefore a protocol-conditional diagnostic, not a prevalence estimate of composition failure. We report no global count of composition-specific failures, do not attribute the frontier results to scale or to reasoning, and make no claim about decomposition prompting or contamination likelihood (§8).

2 Related work

Execution-based code-generation benchmarks. HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021) established the field's default framing: independent problems scored by unit tests, summarised as pass@k. That answers "how capable is this model" well, but cannot ask whether a *specific pair* of capabilities composes, because nothing relates one problem to another. BigCodeBench (Zhuo et al., 2025) scales the setting to 1,140 tasks over 139 libraries and uses the word *compositional* for orchestrating many library calls under complex instructions — a different quantity from ours.

Rigour, test adequacy, and contamination. Two results shape our protocol. EvalPlus (Liu et al., 2023) extends HumanEval's tests 80-fold, drops pass@k by up to 28.9 points, and shows that test insufficiency causes outright *mis-ranking* of models. LiveCodeBench (Jain et al., 2025) time-stamps problems so contamination can be bounded by date. A survey of 672 code benchmarks (Cao et al., 2026) finds that around 70% took no data-quality assurance measures and that many highly cited benchmarks contain incorrect reference code or tests.

We take three things from this literature and are explicit about the one we do not. We hand-author hidden tests and validate every reference under the shipped evaluator (§4); we treat prompt wording as an intervention, following evidence that meaning-preserving format changes move accuracy by up to 76 points (Sclar et al., 2024); and we publish the full development chronology. What we **lack** is LiveCodeBench's temporal control: our tasks are unpublished but untested for contamination.

Compositional generalization. SCAN (Lake & Baroni, 2018) and COGS (Kim & Linzen, 2020) established that sequence models can master primitives and fail their combinations. SCAN's sharpest split "requires compositional application of words learned in isolation" — the nearest conceptual ancestor of our design. The difference is where isolation lives: in SCAN it is a property of the *training split*, whereas we require each model to *succeed on* both matched precursors at test time, per family. Keysers et al. (2020) generalise the split-based control, constructing train and test sets that maximise compound divergence at small atom divergence; that is a distributional control over training exposure, complementary to conditioning on per-model test-time performance, and it presumes a training set we do not have. Hupkes et al. (2020) translate accounts of compositionality into five tests — systematicity, productivity, substitutivity, localism and overgeneralisation — on a synthetic grammar; ConceptBench tests none directly and is closest in spirit to systematicity. Dziri et al. (2023) formulate compositional tasks as computation graphs and argue transformers reduce multi-step compositional

reasoning to linearised subgraph matching; there constituent competence is a property of graph nodes inside one task, not of separately administered tasks the model must pass first. Skill-Mix (Yu et al., 2023) combines a random k-subset of N rhetorical skills; grading is LLM-based and constituent performance is not separately established.

The conditional metric is not ours. Press et al. (2023) define the *compositionality gap* as how often a model correctly answers all sub-questions but not the composed question, on multi-hop factual questions whose facts are unlikely to co-occur in pretraining. That is the same conditional we use, named and published first, in a different domain and with sub-questions rather than projections as constituents. We therefore claim neither conditional composition evaluation nor conditioning on constituent success as novel. Composition can also be *elicited* rather than measured: least-to-most prompting (Zhou et al., 2023) supplies the decomposition, and self-ask has the model pose its own follow-up questions. Both are interventions that narrow composition gaps; ConceptBench measures unaided composition under direct prompting and bounds its claims accordingly (§8).

Closest work: composed code tasks. EvoEval (Xia et al., 2024) evolves HumanEval through seven transformations, **two** of which compose problems. *Combine* merges two problems "by integrating the concepts from both", pairing them by a heuristic on input-argument type. *Combine-naive* is stricter and manually constructed: "Let A and B be two separate problems with x as input(s) for A, we aim to create a new problem C with same inputs where the solution can be written as B(A(x))" — explicit sequential functional composition, filtered so the output type of A matches the input type of B.

EvoEval already conditions on constituent success, for both variants. It partitions parent outcomes into *pass both* / *pass one* / *pass none* and reports a *Composition Percentage*: the share of *pass both* problems the model still solves once combined. Removing extraneous logic helps but does not close the gap — GPT-4 moves from 53.8% on Combine to 75.2% on Combine-naive, still failing a quarter of problems whose parents it had already solved. Together with Press et al. (2023), this settles the point: we do **not** claim the conditional metric, or filtering on constituent success, as novel.

The remaining distinction is precise, and it is the whole of our novelty claim. Prior work composes separate parent problems (EvoEval Combine), separate sub-questions (Press), or sequential functions $B(A(x))$ (Combine-naive); ConceptBench constructs A and B as controlled semantic projections of one AB semantic function onto domains where the complementary capability is structurally absent. In EvoEval a failure is consistent with "cannot chain two functions" or "cannot satisfy two requirements at once"; in ConceptBench the two capabilities interact inside a single function.

HumanEval Pro / MBPP Pro (Yu et al., 2024) builds self-invoking problems that must call the model's own base solution, reporting large drops (o1-mini: $96.2\% \rightarrow 76.2\%$); it reports overlap as a confusion matrix rather than a conditional rate, and its constituent is one function reused rather than two distinct capabilities. ClassEval (Du et al., 2024) and ClassEval-Pro (Chen et al., 2026) find that method-level ability does not reflect class-level ability and identify *cross-method coordination* as the bottleneck; both are compositional code *creation*, neither conditions on matched constituent success per family, and neither isolates a named concept pair. A full comparison is in Appendix F.

3 CONCEPTBENCH METHODOLOGY

3.1 THE CONDITIONAL MEASUREMENT

Let a family be a triple of tasks (A, B, AB) sharing one semantic core. For a model m, write $a, b, ab \in \{\text{pass}, \text{fail}\}$. The family is **eligible** if $a = b = \text{pass}$. Among eligible families, a **raw PPF** (pass–pass–fail) is one where $ab = \text{fail}$. The reported quantity is

$$P(\text{AB fails} \mid A \wedge B) = \frac{|\text{raw PPF}|}{|\text{eligible}|},$$

computed **within a concept pair**, never pooled across pairs as if families were independent draws. Families within a pair share prompt scaffolding, structure and difficulty; a pooled proportion would understate its own uncertainty. Where we report an aggregate we label it a descriptive aggregate and show the stratified breakdown alongside.

Matched precursor success, defined. A model has succeeded on precursor A of family f when one response to f's role-A task, generated under *that panel's declared inference protocol*, passes *all* of that task's hidden tests in the sandboxed evaluator; B likewise, and eligibility is the conjunction. It is panel-relative because the protocols differ: the controlled open-weight panel uses the pinned decoding manifest at temperature 0, the frontier panels the provider-native protocols of §4. It is one response, one task per capability per family, and pass/fail against a finite suite, so it certifies nothing beyond that family's domain and inherits that suite's coverage.

What the rate estimates. It is a **descriptive conditional proportion** over a model-specific, endogenously selected family set: the conditioning event is the model's own success, so the denominator is by construction the subset that model found easiest. Models are scored on different family sets, per-model rates are not estimates of a common population parameter, and two models are directly comparable only on their common support (§5.1). The 74 families were hand-authored, not sampled. Wilson intervals (Wilson, 1927) assume within-pair independence that does not hold and are lower bounds on uncertainty. Without a difficulty-matched non-compositional control the rate is not a causal estimate of excess difficulty attributable to composition alone.

Evidence tiers. **Tier 1, raw PPF**, is the fully automatic outcome, and every quantitative rate in the paper is a raw-PPF rate. **Tier 2, mechanically verified confound screens**, are programmatic checks for specific known failure modes — instrumentation or extraction failure, an orthogonal sentinel convention, a precursor-coverage gap with a concrete valid-domain witness — that yield specific exclusions or flags, not a proof that the remaining cases are composition-specific. **Tier 3, mechanistic case studies**, are selected cases read at the program level. Neither Tier 2 nor Tier 3 yields a global count of "true" composition failures, and none is reported (§6).

We report all eight $A/B/AB$ outcome patterns (Appendix A). PFP, FPP and FFP — a constituent fails and the composition passes — are an **instrument non-monotonicity diagnostic** (§7).

3.2 Matched families and the semantic-projection rule

Each family fixes one semantic function and instantiates it three times. The requirement is not that the three prompts be verbatim identical, but that A and B be **faithful projections** of the AB semantics onto input domains where the other capability is structurally absent. A precursor must not mention an operation that is semantically vacuous on its own input domain.

Worked example (`c5_010`, state $\times$ rollback). All three tasks run a list of `(kind, n)` operations against a running total starting at 0 and return how many of the values it held, the initial 0 included, were ≥ 1. **AB**: operations are `add` (total $+= n{+}1$), `mul` (total $\times= n$) and `undo` (restore the value before the most recent operation still in effect). **A** is the same rule on inputs with no `undo`, so rollback is structurally absent and its clause is omitted. **B** is the same rule on inputs with at most one `add` or `mul`, the rest `undo`, so a history of several state changes cannot arise. On each restricted domain the precursor's reference agrees with the AB reference; only AB inputs contain several state changes for `undo` to roll back through.

This is stronger than a verbal restriction such as "assume depth ≤ 1" on a general representation: models do not reliably honour such promises, and a violated promise is indistinguishable from a composition failure. That design was tried and rejected (Appendix B).

3.3 The projection-equivalence audit, and its limits

For every family in P2–P5 we check that the projected precursor rule and the full AB rule agree on the precursor's input domain. The check is **empirically validated over an enumerated in-domain battery**, not proved: each battery sweeps a *bounded* sub-domain exhaustively and samples beyond it. This audit is distinct from the **behavioural-distinctness audit** (Appendix E), which counts how many distinct functions a pair's families realise. **P1 is a legacy pair**: it predates this protocol, uses role-specific representations, and has *no* projection-equivalence audit, so its construct validity rests on the representation design alone. Its results are reported unchanged (Table 4).

Table 1: Controlled open-weight panel. Cells show **raw PPF /** eligible families and the raw-PPF rate $P(\text{AB fails} \mid A \wedge B)$, whose **denominator is the eligible-family count shown in the same cell**. Raw PPF is the automatic benchmark outcome and carries every rate in the paper. Cells with fewer than 3 eligible families are marked FLOOR : no rate exists there, which is not the same as 0%.

Pair	Qwen 14B	Qwen 32B	DeepSeek 33B	Phi-4 14B	Granite 20B
P1 recursion × branching	4/9 44%	3/10 30%	2/4 50%	2/8 25%	FLOOR (0 elig.)
P2 nesting × delimiters	7/10 70%	6/9 67%	2/3 67%	1/5 20%	3/6 50%
P3 aggregation × partial failure	8/9 89%	4/10 40%	7/7 100%	4/5 80%	6/6 100%
P4 nesting × type dispatch	0/4 0%	4/11 36%	0/4 0%	1/6 17%	FLOOR (0 elig.)
P5 state × rollback	9/10 90%	1/12 8%	4/6 67%	1/9 11%	3/3 100%

3.4 Eligibility and screening

Cells with fewer than three eligible families are rendered FLOOR rather than as a rate: a proportion over one or two families carries no information. Before authoring, each candidate pair passes a screen: both projections must be non-constant, families must not collapse onto one another, and the composition must not admit a trivial decomposition. One candidate pair was withdrawn at this stage, with no tasks ever written for it.

4 Experimental protocol

Frozen benchmark. Five pairs, 74 families, 222 tasks; per-pair composition and behavioural function counts are in Table 3, Appendix E. Each dataset carries a content fingerprint and a freeze commit. No task was modified after freezing, and none in response to a model outcome. Superseded versions and failed runs are archived, not discarded.

Controlled open-weight panel. Five models — Qwen2.5-Coder 14B and 32B (Hui et al., 2024), DeepSeek-Coder 33B (Guo et al., 2024), Phi-4 14B (Abdin et al., 2024) and Granite 20B (Mishra et al., 2024) — on one execution stack: vLLM (Kwon et al., 2023), Singularity-sandboxed evaluation (Kurtzer et al., 2017), and a decoding manifest pinning temperature 0, top-p 1, top-k -1 and repetition penalty 1.0. The manifest fails before generating a token on any requested-vs-effective mismatch (§7).

Frontier external validation. Two proprietary APIs, run with reasoning explicitly disabled, one response per task, no tools, no web access, no code execution, no retries, no few-shot examples, and identical prompts, extraction and evaluator to the open panel. **Neither API exposes sampling control**: reasoning models do not honour temperature or top-p, and the installed provider SDK removes them from the request. We record for every row what was requested, what was sent, and that sampling was provider-default and not caller-pinnable. The panels are therefore never pooled: they differ in protocol as well as in model, so any comparison between them is confounded by construction.

A third panel: reasoning enabled. An intervention, pre-specified in a written protocol and frozen before execution (Appendix G), reruns the same 222 frozen tasks on both frontier models. Tasks, prompts, extraction, evaluator, model identity, and sampling-field policy were held fixed; provider-native reasoning configuration and the output-token ceiling changed. Reported separately in §5.4, never pooled.

What was and was not varied. Provider-native reasoning *was* varied, together with the output ceiling, as the third panel above. Explicit chain-of-thought and decomposition prompting were *not* run, and neither was multi-sample estimation: every panel is one response per task. We did not test whether the failures we report survive being told to decompose.

Table 2: Open-weight panel by pair, over all five models. Precursor pass rates and joint eligibility are shares of all model × family cells. *Unconditional* AB failure ignores the precursors; the *conditional* raw-PPF rate is $P(\text{AB fails} \mid A \wedge B)$. Both orderings are P3 > P2 > P5 > P1 > P4.

Pair	A pass	B pass	Joint	Unconditional AB fail	Conditional raw PPF
P1 recursion × branching	60.0%	41.0%	31.0%	54.0% (54/100)	35.5% (11/31)
P2 nesting × delimiters	100.0%	55.0%	55.0%	75.0% (45/60)	57.6% (19/33)
P3 aggregation × partial failure	91.7%	66.7%	61.7%	83.3% (50/60)	78.4% (29/37)
P4 nesting × type dispatch	86.7%	34.7%	33.3%	52.0% (39/75)	20.0% (5/25)
P5 state × rollback	90.7%	53.3%	53.3%	68.0% (51/75)	45.0% (18/40)

5 RESULTS

5.1 PRIMARY RESULT: MODEL HETEROGENEITY ON FIXED COMMON SUPPORT

Each model's raw-PPF rate in Table 1 is computed over the families that model was eligible for, so per-model rates compare different family sets. We therefore restrict each pair of open-weight models to the families on which *both* are jointly eligible under the declared protocol. Holding the evaluated family set fixed removes model-specific eligibility differences within this comparison.

On the same ten jointly eligible P5 families, Qwen 14B produces nine raw PPFs and Qwen 32B none — same tasks, same pinned stack, same prompts. Under the pinned protocol, models can thus differ sharply in conditional AB failure even on exactly the same families for which both demonstrate matched precursor success. Two of the nine, `c5_002` and `c5_007`, are mechanically verified precursor-coverage gaps (§6); setting aside only those two observations, the contrast remains **7 vs 0**. This sensitivity removes only the two mechanically demonstrated precursor-coverage confounds; it does not certify the remaining seven as composition-specific. Eight further cells show a gap of at least three raw PPFs on an identical family set, in both directions, across P3 and P5 (Appendix A).

Eligibility selection alone cannot explain all observed model heterogeneity. This does not identify the contrast as composition-specific excess difficulty: unconditional AB difficulty and other model differences remain possible contributors. Common support is itself selected, and its cells overlap; the matrix is descriptive bookkeeping, with no test statistic, effective sample size or p-value computed from it.

5.2 A NEGATIVE RESULT: PAIR ORDERING MIRRORS UNCONDITIONAL DIFFICULTY

Pair-stratified raw-PPF rates span 58 percentage points: P4 20.0% (5/25), P1 35.5% (11/31), P5 45.0% (18/40), P2 57.6% (19/33), P3 78.4% (29/37). Table 2 sets them beside unconditional AB failure over all model × family cells. Conditioning on matched precursor success changes the levels substantially — P4 falls from 52.0% to 20.0% — and changes the model ordering: Qwen 14B overtakes DeepSeek 33B, and Qwen 32B overtakes Phi-4 14B. **But the pair rank ordering is unchanged**, P3 > P2 > P5 > P1 > P4. Pair-level ranking therefore cannot by itself distinguish composition-specific structure from ordinary AB difficulty.

Eligibility is part of the selection structure. In every pair the role-B pass rate is below the role-A rate, most sharply in P2, where every model passes every role-A task and joint eligibility equals the role-B pass rate (55.0%). This is not a flaw in itself, but the conditional rate is computed over the families whose role-B task each model passed.

5.3 DESCRIPTIVE PANEL RATES

Across the controlled open-weight panel, 82 of 166 eligible model–family observations are raw PPFs (49.4%), a descriptive aggregate over this finite panel rather than an estimate of composition-failure prevalence. Model-stratified rates range from 27.3% (Phi-4 14B, 33 eligible) to 80.0% (Granite 20B, 15 eligible) over different, endogenously selected family sets (Table 1; Figure 4, Appendix I). Some failures recur: for `rt_008`, two Qwen models produce near-identical incorrect solutions with the same leaf/merge boundary error, and three P3 families are raw PPF for four of five open models.

5.4 FRONTIER PANELS: DESCRIPTIVE EXTERNAL VALIDATION

With reasoning disabled, GPT-5.6 Sol is jointly eligible on 71/74 families and produces 1 raw PPF, and Claude Sonnet 5 is eligible on 65/74 with 1 raw PPF, with no refusal, truncation, extraction or API failure across 444 generations; each case is a single generation from an unpinned sampler.

In the **pre-specified third panel**, tasks, prompts, extraction, evaluator, model identity, and sampling-field policy were held fixed; provider-native reasoning configuration and the output-token ceiling changed. For OpenAI, reasoning moved from `none` to `medium` and `max_completion_tokens` from 2,048 to 16,384; for Anthropic, thinking from disabled to `adaptive`, `effort` from absent to `high`, and `max_tokens` from 2,048 to 16,384. GPT-5.6 Sol is then jointly eligible on 73/74 with 1 raw PPF, and Claude Sonnet 5 on 70/74 with 1 raw PPF. Both previously observed AB failures became passes under the reasoning-enabled, higher-output-budget configuration while precursor eligibility was preserved (`rt_003` 16/17 $\rightarrow$ 17/17 tests; `c5_010` 4/16 $\rightarrow$ 16/16). Each provider also produced one new raw PPF. Because generation is provider-controlled, the output ceiling also changed, and the new raw PPFs are not assigned a definitive mechanistic label, this panel is descriptive rather than causal (Appendix G). These four events are not independent samples; per-cell raw-PPF percentages are reported descriptively, but no inferential comparison or pooled rate is computed across panels.

6 FORENSIC SCREENING AND MECHANISTIC ANALYSIS

Raw PPF is the only automatic quantitative endpoint. Some raw PPFs can be shown mechanically to arise from known confounds; attributing the rest to the integration of A and B requires reading programs, and that judgement is not fully automatable. A forensic audit found that the verdict of such an attribution criterion changes materially under reasonable operationalizations. The forensic analysis was AI-assisted, and no independent human adjudication or inter-rater reliability study was performed. **We therefore report no global count of composition-specific failures after screening**, and none of the screens below is subtracted from a raw count or divided into a rate.

6.1 MECHANICALLY VERIFIED SCREENS (TIER 2)

Every reasoning-off raw PPF was re-executed against every AB hidden test, and each precursor's valid domain was searched for inputs on which the AB program fails; wherever one was found, the model's matched precursor program was run on it (Appendix D).

Instrumentation. Re-execution reproduces every recorded failure, with zero load or extraction failures across the 84 cases.

Orthogonal sentinel (P4). All five P4 raw PPFs reproduce *only* on the AB specification's -1 convention for inputs with no scalar values; nesting and type dispatch are otherwise correct. P4 thus contributes raw signal but no mechanistic evidence for the intended interaction in those five cases; its 20.0% is a raw-PPF endpoint only, and the pair is retained rather than pruned.

The P3 sentinel is not an artifact. In P3 a -1 return expresses the partial-failure requirement itself. That 14 of the 29 P3 raw PPFs involve -1 in every failure is descriptive, not a confound classification: sentinel involvement is not automatically an artifact, and none of those cases is excluded on that ground.

Precursor-coverage gaps. In six raw PPFs the matched precursor program is itself wrong on a concrete valid input of its own domain that its hidden tests missed — three in P2, three in P5 (Appendix D lists each witness). These are specific exclusions. Finding none elsewhere is not proof of absence, and we draw no rate or bound from the search.

6.2 MECHANISTIC CASE STUDIES (TIER 3)

The two cases below were selected for depth, not sampled, and are qualitative program-level interpretations rather than validated classifications.

`c5_010`, Claude Sonnet 5 — shared-state conflation. The only family that is raw PPF in both the open panel (Qwen 14B) and the reasoning-disabled frontier panel. In role B (§3), Sonnet's program keeps rollback state and the visitation record in **two separate structures**. In role AB it collapses both duties into a single `history` list, so `undo` pops values out of the count. The program diff localizes the defect to the integration step, and the error is **consistent with an integration-induced shared-state conflation**. The defect also manifests on a role-B-domain input on which the role-B program is correct; that locates where it can manifest, not why it was introduced. Programs: Appendix I; reasoning-enabled outcome: Appendix G.

`rt_003`, GPT-5.6 Sol — lossy subtree summary. Both precursor strategies work on their restricted domains: role A sorts all values, valid because every pair in a chain is related, and role B compares the root with each child. Neither survives a branching, deep tree. In AB the program compares each node only with the minimum and maximum of each child's subtree, a summary that is exact only when that subtree holds a single value. The mechanism is consistent with a failure introduced while combining the requirements. Appendix D adds `c8_006`, an aggregation-level collapse recurring across four open models, and `c5_007`, a coverage gap.

7 Benchmark and protocol validation

Five lessons bear on the results; the full chronology and our own analysis errors are in Appendix B.

Representations must not vary with role unless the design requires it. Early versions gave precursors bespoke representations, letting models fail for representation reasons that scored as composition failures. P2–P5 use one representation across all three roles; P1 retains role-specific types and is a legacy pair (Appendix E).

Decoding defaults are a live confound. Part of our pilot ran with an inherited repetition penalty of 1.05 present in no configuration file. Every run now builds a decoding manifest and refuses to generate on a requested-versus-effective mismatch.

Clarifying prompts are interventions. One sentence added to role B of P2 collapsed one model's role-B score from 12/12 to 1/12 between dataset versions, taking eligibility with it; a controlled probe with that sentence removed scored 9/12 against 1/12, isolating it as the cause, and the wording was reverted — matching evidence that meaning-preserving format changes move accuracy substantially (Sclar et al., 2024).

Entry-point names leak structure, unrepaired. A family's three roles share one function name, so eight of twelve P2 entry points contain *segment*, including role B, whose prompt never mentions segmentation. Renaming per role would break matched-family identity, so the leak is recorded rather than patched (Appendix B).

Instrument non-monotonicity. In P1 and P4 the role-B pass rate sits *below* the AB pass rate (41% vs 46%; 35% vs 48%) and PFP/FPP/FFP concentrate there. This difficulty inversion does *not* falsify the semantic-projection rule; it is a diagnostic that weakens interpretation of those two pairs, and Appendix A gives the per-pair counts and the argument.

8 Limitations

1. **Mechanistic attribution is not validated.** Whether a raw PPF arises from integrating A and B is a program-level judgement whose verdict changed materially under reasonable operationalizations. The forensic analysis was AI-assisted, with no independent human adjudication and no inter-rater agreement statistic, so case studies are interpretations and no global count of composition-specific failures is reported.
2. **Pair ordering mirrors unconditional difficulty.** Pair ranks are identical with and without conditioning, so they cannot by themselves separate composition from AB difficulty.
3. **Frontier sampling is uncontrolled**: single draws from a provider-default sampler, so any open-to-frontier comparison is confounded by protocol; one raw PPF per model is existence evidence, not characterisation.

4. **P1 is a legacy pair, and no projection check is a proof.** P1 uses role-specific representations and has no projection-equivalence audit (Appendix E); for P2–P5 the equivalence is empirically validated over bounded batteries of unbounded domains.
5. **P4 and the screens.** All five P4 raw PPFs are orthogonal-sentinel cases; its 20.0% is a raw-PPF endpoint only. P1 and P4 are non-monotonic (§7). The coverage-gap search can confirm a gap but not rule one out, so undetected gaps are unbounded.
6. **Entry-point naming leak**, unrepaired (§7).
7. **Contamination is untested.** The tasks are unpublished but built from constructs resembling common exercises; no contamination test was performed and we make *no* claim about how likely it is.
8. **Eligibility-selection bias.** Conditioning on $A \wedge B$ selects a different family subset per model, by construction the one it found easiest, so per-model rates are not estimates over a common population; this is the single largest threat to cross-model comparison. Granite 20B sits at the instrument floor with 15/74 eligible and zero on two pairs.
9. **Residual semantic dependence.** Under the behavioural-distinctness audit P2 roles A and B have 8 distinct functions across 12 families, and P5 role B 12 across 15. No AB role contains an equivalence class, so the dependence acts on eligibility, not on composed outcomes; we do not recompute the metric over deduplicated families after the fact.
10. **Condition and sampling scope; the reasoning intervention is not a causal test.** Every panel uses direct prompting and one response per task per condition; nothing bounds these models under decomposition prompting (Zhou et al., 2023; Press et al., 2023). Reasoning configuration and output ceiling changed together, the two providers' controls are not comparable, and Anthropic exposed no thinking block on any of its 222 generations, so internal reasoning is not observable.
11. **Selection, twice over; five pair groups.** Common support removes the selection *difference* between two models, not selection from the study; the pairwise cells overlap and yield no test statistic, and five pair groups cannot support a random-effects variance estimate. We examined 35 model $\times$ pair cells and highlight extremes with no multiplicity correction.

9 Conclusion

ConceptBench shows that under a single pinned inference protocol, models can differ substantially in conditional AB failure even when evaluated on the same families for which both passed the matched precursors. The sharpest observed contrast is 9 vs 0 on the same ten jointly eligible P5 families, and it remains 7 vs 0 after setting aside the two mechanically verified precursor-coverage-gap observations. This does not make the remaining failures a composition-specific rate. Pair ordering mirrors unconditional AB difficulty, showing that pair-level raw PPF alone does not isolate composition-specific difficulty. Matched semantic projection provides a controlled way to study what happens after observed precursor success, while exposing where interpretation requires additional controls, such as difficulty-matched non-compositional tasks.

REPRODUCIBILITY STATEMENT

Every number, table and figure in this paper regenerates from the frozen raw results with no model or API access. The anonymous supplementary artifact contains the complete evaluator and all hidden tests, the frozen task sets with per-file SHA256 fingerprints, the raw generation for every task and model, and the analysis scripts, together with exact commands and a checksum manifest. The experimental protocol, including the pinned decoding manifest and the API-native protocol resolved for each provider, is in Appendix C; per-family outcomes are in Appendix A; the mechanical confound screens, with every re-execution and witness input, are in Appendix D; and the per-pair construct-validity status, including P1's legacy status, is in Appendix E. A consistency audit shipped with the artifact re-derives every reported number and exits non-zero on any disagreement.

AI USE STATEMENT

In this work we used generative AI tools throughout the research process, and we disclose that use in full.

AI assistance was used for: feedback on methodology and experimental design, including critiques that led us to withdraw one candidate concept pair and to revise the projection protocol; implementation and debugging of the benchmark harness, evaluation pipeline and analysis code; interpretation of results; **mechanistic forensic analysis of individual failures and candidate failure classification**; adversarial review of our own claims, which produced several of the weakenings recorded in the limitations; literature discovery and summarisation, including identification of the EvoEval and Press et al. (2023) precedents, which together narrowed our novelty claim to the matched projection instrument; drafting and editing of the manuscript; production of figures and artifacts; and reproducibility auditing, including the consistency and identity-leak checks shipped with the supplementary material.

No independent human adjudication of failure mechanisms was performed, and no inter-rater agreement statistic exists. An earlier draft reported a global count of adjudicated composition-specific failures produced by AI-assisted classification; we withdrew it because the classification changed materially under reasonable operationalizations of its attribution criterion and had not been independently validated. Mechanistic interpretations in this paper are therefore presented as case-level analyses, not as a complete, independently validated ledger. The confound screens we do report are programmatic re-executions whose outputs anyone can regenerate.

Other AI-assisted work was checked as follows. Every empirical claim is regenerated from frozen raw results by an auditing script that fails on any disagreement with the manuscript; AI-assisted code is covered by the project's test suite; every cited work was verified against the paper or its official record, and no reference was accepted on the basis of a generated summary alone; and the case studies were checked by reading the models' actual generations and the failing tests. Where AI-assisted analysis was found to be wrong, the errors were corrected and documented rather than quietly repaired (Appendix B).

We take responsibility for the final content of this work, including all text, claims and artifacts produced with the aid of generative AI.

REFERENCES

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024. URL https://arxiv.org/abs/2412.08905.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021. URL https://arxiv.org/abs/2108.07732.

Jialun Cao, Yuk-Kit Chan, Zixuan Ling, Wenxuan Wang, Shuqing Li, Mingwei Liu, Ruixi Qiao, Yuting Han, Chaozheng Wang, Boxi Yu, Pinjia He, Shuai Wang, Zibin Zheng, Michael R. Lyu, and Shing-Chi Cheung. Position: Code benchmarks should prioritize rigor, reliability, and reproducibility. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2501.10711.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Yeheng Chen, Chaoxiang Xie, Yuling Shi, Wenhao Zeng, Yongpan Wang, Hongyu Zhang, and Xiaodong Gu. ClassEval-Pro: A cross-domain benchmark for class-level code generation. In *Proceedings of the 3rd ACM International Conference on AI-Powered Software (AIware)*, 2026. doi: 10.1145/3805760.3814925. URL https://arxiv.org/abs/2604.26923.

Xueying Du, Mingwei Liu, Kaixin Wang, Hanlin Wang, Junwei Liu, Yixuan Chen, Jiayi Feng, Chaofeng Sha, Xin Peng, and Yiling Lou. Evaluating large language models in class-level code generation. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering (ICSE)*, 2024. doi: 10.1145/3597503.3639219. URL https://arxiv.org/abs/2308.01861.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. Faith and fate: Limits of transformers on compositionality. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Y. Wu, Y. K. Li, Fuli Luo, Yingfei Xiong, and Wenfeng Liang. DeepSeek-Coder: When the large language model meets programming — the rise of code intelligence. *arXiv preprint arXiv:2401.14196*, 2024. URL https://arxiv.org/abs/2401.14196.

Binyuan Hui, Jian Yang, et al. Qwen2.5-Coder technical report. *arXiv preprint arXiv:2409.12186*, 2024. URL https://arxiv.org/abs/2409.12186.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795, 2020.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2403.07974.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations (ICLR)*, 2020.

Najoung Kim and Tal Linzen. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9087–9105, 2020. URL https://aclanthology.org/2020.emnlp-main.731/.

Gregory M. Kurtzer, Vanessa Sochat, and Michael W. Bauer. Singularity: Scientific containers for mobility of compute. *PLOS ONE*, 12(5):e0177459, 2017. doi: 10.1371/journal.pone.0177459.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, 2023.

Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning (ICML)*, pp. 2879–2888, 2018. URL https://arxiv.org/abs/1711.00350.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023. URL https://arxiv.org/abs/2305.01210.

Mayank Mishra, Matt Stallone, Gaoyuan Zhang, Yikang Shen, Aditya Prasad, Adriana Meza Soria, Michele Merler, Parameswaran Selvam, Saptha Surendran, Shivdeep Singh, Manish Sethi, Xuan-Hong Dang, Pengyuan Li, Kun-Lung Wu, Syed Zawad, Andrew Coleman, Matthew White, Mark Lewis, Raju Pavuluri, Yan Koyfman, Boris Lublinsky, Maximilien de Bayser, Ibrahim Abdelaziz, Kinjal Basu, Mayank Agarwal, Yi Zhou, Chris Johnson, Aanchal Goyal, Hima Patel, Yousaf Shah, Petros Zerfos, Heiko Ludwig, Asim Munawar, Maxwell Crouse, Pavan Kapanipathi, Shweta Salaria, Bob Calio, Sophia Wen, Seetharami Seelam, Brian Belgodere, Carlos Fonseca, Amith Singhee, Nirmit Desai, David D. Cox, Ruchir Puri, and Rameswar Panda. Granite code models: A family of open foundation models for code intelligence. *arXiv preprint arXiv:2405.04324*, 2024. URL https://arxiv.org/abs/2405.04324.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=RIu5lyNXjT.

Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927. doi: 10.1080/01621459.1927.10502953.

Chunqiu Steven Xia, Yinlin Deng, and Lingming Zhang. Top leaderboard ranking = top coding proficiency, always? EvoEval: Evolving coding benchmarks via LLM. *arXiv preprint arXiv:2403.19114*, 2024. URL https://arxiv.org/abs/2403.19114.

Dingli Yu, Simran Kaur, Arushi Gupta, Jonah Brown-Cohen, Anirudh Goyal, and Sanjeev Arora. Skill-mix: a flexible and expandable family of evaluations for AI models. *arXiv preprint arXiv:2310.17567*, 2023. URL https://arxiv.org/abs/2310.17567.

Zhaojian Yu, Yilun Zhao, Arman Cohan, and Xiao-Ping Zhang. HumanEval pro and MBPP pro: Evaluating large language models on self-invoking code generation. *arXiv preprint arXiv:2412.21199*, 2024. URL https://arxiv.org/abs/2412.21199.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed H. Chi. Least-to-most prompting enables complex reasoning in large language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, Simon Brunner, Chen Gong, Thong Hoang, Armel Randy Zebaze, Xiaoheng Hong, Wen-Ding Li, Jean Kaddour, Ming Xu, Zhihan Zhang, Prateek Yadav, Naman Jain, Alex Gu, Zhoujun Cheng, Jiawei Liu, Qian Liu, Zijian Wang, Binyuan Hui, Niklas Muennighoff, David Lo, Daniel Fried, Xiaoning Du, Harm de Vries, and Leandro Von Werra. Bigcodebench: Benchmarking code generation with diverse function calls and complex instructions. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2406.15877.

A Complete family-level outcomes

Every family, every model. PPP–FFF is the A/B/AB outcome triple. **PPF** marks a raw PPF. Two mechanically verified confound screens are annotated (Appendix D): † marks an orthogonal -1 sentinel case (all five in P4), and ‡ marks a precursor-coverage gap, where the matched precursor program is itself wrong on a valid input of its own domain. Unmarked raw PPFs carry no verdict either way. A blank cell means no complete A/B/AB triple was produced for that model.

P1 — RECURSION × BRANCHING

Family	Qwen 14B	Qwen 32B	DeepSeek 33B	Phi-4 14B	Granite 20B	GPT-5.6 Sol	Claude Sonnet 5
rt_001	PPP	PPP	PFP	FPP	PFP	PPP	PPP
rt_002	PFP	FPP	PPP	PPP	FFF	PPP	PPP
rt_003	FPF	FPF	FFF	FPF	FFF	**PPF**	PPP
rt_004	**PPF**	PPP	PFF	PFP	FFF	PPP	PPP
rt_005	FFF	FPP	FFF	FFF	PFF	PPP	PPP
rt_006	FFF	PFF	PFF	**PPF**	FFF	PPP	PPP
rt_007	FFP	FFP	FFF	FFF	FFF	PPP	PFP
rt_008	**PPF**	**PPF**	FPF	FPP	FFF	PPP	PPP
rt_009	**PPF**	PFF	PFF	PFP	FFF	PPP	PPP
rt_010	FFP	FPP	FFP	PPP	FFP	PPP	PPP
rt_011	PFF	PFP	PFF	PPP	PFF	PPP	PPP
rt_012	**PPF**	PPP	PFP	PFF	FFF	PPP	PPP
rt_013	PFF	PPP	PFF	**PPF**	PFF	PPP	PPP
rt_014	PPP	PPP	PPP	PPP	PFP	PPP	PPP
rt_015	FFF	FPF	FFF	FFF	FFF	PPP	FPP
rt_016	PFP	PPP	PFP	PPP	PFP	PPP	PPP
rt_017	PPP	**PPF**	**PPF**	PPP	FFF	PPP	PPP
rt_018	PFP	PFP	PFF	FFP	FFF	PPP	PPP
rt_019	PPP	PPP	**PPF**	PFF	PFF	PPP	PPP
rt_020	PPP	**PPF**	FFP	FFP	FFF	PPP	PPP

P2 — NESTING × DELIMITERS

Family	Qwen 14B	Qwen 32B	DeepSeek 33B	Phi-4 14B	Granite 20B	GPT-5.6 Sol	Claude Sonnet 5
c9_001	PFF	PFF	PFF	PFF	PFF	PPP	PPP
c9_002	PPP	PPP	PFF	PPP	PPP	PPP	PPP
c9_003	**PPF**	**PPF**	PFF	PFP	PFF	PPP	PPP
c9_004	**PPF**	**PPF**‡	PFF	PPP	PFF	PPP	PPP
c9_005	PPP	PPP	PPP	PPP	PPP	PPP	PPP
c9_006	PFF	**PPF**	**PPF**	PFF	PFF	PFF	PFF
c9_007	**PPF**‡	PFF	PFF	PFF	PFF	PPP	PPP
c9_008	**PPF**‡	PFF	PFF	**PPF**	**PPF**	PPP	PPP
c9_009	**PPF**	**PPF**	PFF	PFF	**PPF**	PFP	PPP
c9_010	PPP	PPP	**PPF**	PPP	PPP	PPP	PPP
c9_011	**PPF**	**PPF**	PFF	PFF	**PPF**	PFF	PFF
c9_012	**PPF**	**PPF**	PFF	PFF	PFF	PPP	PPP

P3 — AGGREGATION × PARTIAL FAILURE

Family	Qwen 14B	Qwen 32B	DeepSeek 33B	Phi-4 14B	Granite 20B	GPT-5.6 Sol	Claude Sonnet 5
c8_001	**PPF**	PPP	PFP	PPP	PFF	PPP	PPP
c8_002	**PPF**	**PPF**	**PPF**	PFF	**PPF**	PPP	PPP
c8_003	PPP	PPP	PFF	PFF	PFF	PPP	PPP
c8_004	**PPF**	PPP	**PPF**	PFF	**PPF**	PPP	PPP
c8_005	**PPF**	**PPF**	**PPF**	PFF	**PPF**	PPP	PPP
c8_006	**PPF**	**PPF**	**PPF**	**PPF**	PFF	PPP	PPP
c8_007	PFP	**PPF**	PFF	**PPF**	**PPF**	PPP	PPP
c8_008	PFF	FPF	FFF	**PPF**	FFF	PPP	PPP
c8_009	PFF	FPF	FPF	**PPF**	PFF	PPP	PPP
c8_010	**PPF**	PPP	**PPF**	PFF	PFF	PPP	PPP
c8_011	**PPF**	PPP	**PPF**	PFF	**PPF**	PPP	PPP
c8_012	**PPF**	PPP	**PPF**	PFF	**PPF**	PPP	PPP

P4 — NESTING × TYPE DISPATCH

Family	Qwen 14B	Qwen 32B	DeepSeek 33B	Phi-4 14B	Granite 20B	GPT-5.6 Sol	Claude Sonnet 5
c1_001	PPP	PPP	PPP	PPP	PFP	PPP	PPP
c1_002	PFF	**PPF**†	PFP	PFP	PFF	PPP	PPP
c1_003	FFF	**PPF**†	PFF	PFF	FFF	PPP	PPP
c1_004	PPP	PPP	PFP	PPP	PFP	PPP	PPP
c1_005	PFF	PFF	PFF	PFF	PFF	PPP	PPP
c1_006	FFF	**PPF**†	FFF	PFF	FFF	PPP	PPP
c1_007	PPP	PPP	PPP	PPP	PFF	PPP	PPP
c1_008	PFP	PPP	PFP	PFP	PFP	PPP	PPP
c1_009	PFF	PFF	FFF	PFF	FFF	PPP	PPP
c1_010	PFP	PPP	PPP	PPP	PFF	PPP	PPP
c1_011	PFF	PFF	PFF	PFF	PFP	PPP	PPP
c1_012	PFF	PFF	FPF	**PPF**†	PFF	PPP	PPP
c1_013	PPP	PPP	PPP	PPP	PFF	PPP	PPP
c1_014	PFP	PPP	PFP	PFP	PFP	PPP	PPP
c1_015	PFF	**PPF**†	FFF	PFP	FFF	PPP	PPP

P5 — STATE × ROLLBACK

Family	Qwen 14B	Qwen 32B	DeepSeek 33B	Phi-4 14B	Granite 20B	GPT-5.6 Sol	Claude Sonnet 5
c5_001	PPP	PPP	PFF	PPP	PFF	PPP	PPP
c5_002	**PPF**‡	PPP	PPP	PPP	**PPF**	PPP	PPP
c5_003	**PPF**	PPP	**PPF**	PPP	PFF	PPP	PPP
c5_004	PFF	PFF	PFF	PFF	PFF	PPP	PFF
c5_005	**PPF**	PPP	PFF	FFP	FFF	PPP	PFP
c5_006	PFF	**PPF**	PPP	PPP	PFF	PPP	PPP
c5_007	**PPF**‡	PPP	**PPF**	**PPF**‡	FFF	PPP	PFP
c5_008	**PPF**	PPP	PFP	PPP	**PPF**	PPP	PPP
c5_009	PFF	PFF	PFF	PFF	PFF	PPP	PFF
c5_010	**PPF**	PPP	PFF	FFF	FFF	PPP	**PPF**
c5_011	PFF	PPP	**PPF**	PPP	**PPF**	PPP	PPP
c5_012	**PPF**	PPP	**PPF**	PPP	PFF	PPP	PPP
c5_013	**PPF**	PPP	PFF	PPP	PFF	PPP	PPP
c5_014	PFF	PFF	PFF	PFF	PFF	PPP	PFF
c5_015	**PPF**	PPP	PFF	FFF	FFF	PPP	PPP

MODEL-STRATIFIED TOTALS, OPEN-WEIGHT PANEL

Model	Eligible	Raw PPF	Raw-PPF rate
Qwen 14B	42	28	66.7%
Qwen 32B	52	18	34.6%
DeepSeek 33B	24	15	62.5%
Phi-4 14B	33	9	27.3%
Granite 20B	15	12	80.0%
Panel total	**166**	**82**	**49.4%**

ALL EIGHT OUTCOME PATTERNS

PFP, FPP and FFP — a constituent fails and the composition passes — are the **instrument non-monotonicity** diagnostic. They are not logically impossible: semantic projection requires agreement on the restricted domain, not that a model's pass probability be monotonic in domain restriction. They concentrate in P1 and P4, where role B is empirically harder than AB, and they weaken interpretation of those two pairs (§7).

Controlled open-weight panel — by pair

Pair	PPP	PPF	PFP	PFF	FPP	FPF	FFP	FFF	cells	non-mono
P1	20	11	13	16	5	5	8	22	100	26
P2	14	19	1	26	0	0	0	0	60	1
P3	8	29	2	16	0	3	0	2	60	2
P4	20	5	16	24	0	1	0	9	75	16
P5	22	18	1	27	0	0	1	6	75	2

Controlled open-weight panel — by model

Model	PPP	PPF	PFP	PFF	FPP	FPF	FFP	FFF	cells	non-mono
DeepSeek 33B	9	15	9	28	0	3	2	8	74	11
Granite 20B	3	12	8	29	0	0	1	21	74	9
Phi-4 14B	24	9	7	23	2	1	3	5	74	12
Qwen 14B	14	28	7	17	0	1	2	5	74	9
Qwen 32B	34	18	2	12	3	4	1	0	74	6

Frontier panel — by pair

Pair	PPP	PPF	PFP	PFF	FPP	FPF	FFP	FFF	cells	non-mono
P1	37	1	1	0	1	0	0	0	40	2
P2	19	0	1	4	0	0	0	0	24	1
P3	24	0	0	0	0	0	0	0	24	0
P4	30	0	0	0	0	0	0	0	30	0
P5	24	1	2	3	0	0	0	0	30	2

Frontier panel — by model

Model	PPP	PPF	PFP	PFF	FPP	FPF	FFP	FFF	cells	non-mono
Claude Sonnet 5	64	1	3	5	1	0	0	0	74	4
GPT-5.6 Sol	70	1	1	2	0	0	0	0	74	1

PAIRWISE COMMON SUPPORT (OPEN-WEIGHT PANEL)

For every model pair and concept pair, the family set is restricted to the families on which *both* models are jointly eligible, and raw AB failures are compared on that fixed set. Cells with fewer than three jointly eligible families are omitted.

These comparisons overlap and are not independent. The same family and the same model recur across many rows, and the models are not independent of one another. The table is descriptive bookkeeping. No test statistic, effective sample size or p-value is computed from it, and it does not show that eligibility-selection bias has been eliminated — only that eligibility selection alone cannot explain all observed model heterogeneity.

Pair	Models A vs B	Support	A raw PPF	B raw PPF	Both	Discordant
P1	Qwen 14B vs Qwen 32B	8	3	3	1	4
P1	Qwen 14B vs DeepSeek 33B	3	0	2	0	2
P1	Qwen 32B vs DeepSeek 33B	3	1	2	1	1
P1	Qwen 32B vs Phi-4 14B	4	1	1	0	2
P1	DeepSeek 33B vs Phi-4 14B	3	1	0	0	1
P2	Qwen 14B vs Qwen 32B	8	5	5	5	0
P2	Qwen 14B vs Phi-4 14B	5	2	1	1	1
P2	Qwen 14B vs Granite 20B	6	3	3	3	0
P2	Qwen 32B vs DeepSeek 33B	3	1	2	1	1
P2	Qwen 32B vs Phi-4 14B	4	1	0	0	1
P2	Qwen 32B vs Granite 20B	5	2	2	2	0
P2	Phi-4 14B vs Granite 20B	4	1	1	1	0
P3	Qwen 14B vs Qwen 32B	9	8	3	3	5
P3	Qwen 14B vs DeepSeek 33B	7	7	7	7	0
P3	Qwen 14B vs Granite 20B	5	5	5	5	0
P3	Qwen 32B vs DeepSeek 33B	7	3	7	3	4
P3	Qwen 32B vs Phi-4 14B	3	2	2	2	0
P3	Qwen 32B vs Granite 20B	6	3	6	3	3
P3	DeepSeek 33B vs Granite 20B	5	5	5	5	0
P4	Qwen 14B vs Qwen 32B	4	0	0	0	0
P4	Qwen 14B vs DeepSeek 33B	3	0	0	0	0
P4	Qwen 14B vs Phi-4 14B	4	0	0	0	0
P4	Qwen 32B vs DeepSeek 33B	4	0	0	0	0
P4	Qwen 32B vs Phi-4 14B	5	0	0	0	0
P4	DeepSeek 33B vs Phi-4 14B	4	0	0	0	0
P5	Qwen 14B vs Qwen 32B	**10**	**9**	**0**	**0**	**9**
P5	Qwen 14B vs DeepSeek 33B	4	4	3	3	1
P5	Qwen 14B vs Phi-4 14B	7	6	1	1	5
P5	Qwen 32B vs DeepSeek 33B	6	1	4	0	5
P5	Qwen 32B vs Phi-4 14B	9	1	1	0	2
P5	Qwen 32B vs Granite 20B	3	0	3	0	3
P5	DeepSeek 33B vs Phi-4 14B	6	4	1	1	3
P5	Phi-4 14B vs Granite 20B	3	0	3	0	3

Descriptive bookkeeping over the 33 cells above: 56 discordant raw-PPF observations. These counts are reported for completeness and are *not* an effective sample size.

B BENCHMARK DEVELOPMENT CHRONOLOGY

Every superseded dataset version and every superseded run is retained in the research archive; nothing was discarded. This appendix is kept out of the main narrative except where a lesson justifies a methodological choice (§7).

Task-set lineage. v2.0 restricted precursors by a *verbal promise* ("assume depth ≤ 1") on a general representation, and was rejected on construct-validity grounds: models do not reliably honour such promises, and a violated promise is indistinguishable from a composition failure. v2.1 gave precursors role-specific representations so the excluded structure is absent by construction; it is retained as P1, and those role-specific types are its principal residual limitation. One candidate pair (sequential $\times$ per-key state) was **withdrawn before authoring** at the projection screen because its projections collapsed; no tasks were written.

P2 across four versions. P2 went through four iterations. In v1.1 a single sentence added to role B to clarify the rule collapsed one model's role-B score from 12/12 to 1/12, taking eligibility with it; a controlled probe isolated the sentence and v1.2 restored the original wording. v1.2 still showed role contamination through a splitter carried over from role A's framing. v1.3 applied role-scoped semantic projection mechanically to all twelve families from the semantic decomposition, not family-by-family from observed failures, and contamination went to zero.

What v1.3 could not fix: on role A's domain several distinct-sounding quantities are all equal to segment length, so twelve families reduce to eight behaviourally distinct functions under the audit battery. This is a property of the pair, not of the wording — no phrasing separates quantities that are equal on a domain — and it is recorded as residual semantic dependence (§8).

Two design lessons kept out of the main narrative. *Sentinel return values.* P4's AB specification returns -1 when an input contains no scalar values. That convention proved to be a failure mode of its own: every P4 raw PPF turned out to be a model mishandling the sentinel rather than the concept

interaction, which the orthogonal-sentinel screen records (Appendix D). A specification-level sentinel unrelated to either requirement competes with the interaction under test and should be avoided in future pairs; P3's -1, which expresses its partial-failure requirement, is a different case.

Entry-point name leakage, unrepaired. A family's three roles share one function name, because the name is part of the family's identity. In P2, eight of twelve entry points contain the word *segment* (`widest_segment`, `sparsest_segment`, ...), so role B's function is named for segmentation although its prompt never mentions it. Renaming per role would break the matched-family identity the design rests on. The leak is a name rather than a clause, and we record it rather than patch it.

Errors found in our own analysis. Recorded because they bear on how much weight the results can carry.

1. An interim checkpoint gave a wrong exclusion reason for P1; P1 was re-run canonically with identical results and the document corrected.
2. The claim "composition failure is a property of the interaction, not the model" was made and withdrawn twice. It is wrong and is now out of bounds.
3. The coverage-asymmetry criterion was initially too aggressive: program identity was treated as sufficient for exclusion. It is not; exclusion now requires the matched precursor program to fail a concrete valid-domain witness (Appendix D).
4. An undersized audit battery (16–19 inputs per role, drawn only from the families' own tests) produced three equivalence classes that do not exist. All three break within a few thousand generated inputs. The audit now uses a 15,000-input in-domain battery and the counts are converged.
5. Infrastructure errors were briefly mis-attributed to a model; they were node contention from a concurrent job of our own.
6. An AI-assisted adjudication first admitted all five P4 raw PPFs despite the sentinel criterion already being stated; per-test re-execution showed every failure to be the orthogonal sentinel.
7. **The global adjudicated count was withdrawn.** Earlier drafts reported a single count of composition-specific failures surviving adjudication. A later forensic audit applied a precursor-domain probe to every raw PPF rather than only to flagged cases, found five precursor-coverage gaps beyond the one already identified, and showed that the attribution criterion's verdicts changed materially under reasonable operationalizations. With no independent human adjudication, the count is not defensible as ground truth; the paper now reports raw PPF, mechanical screens and case studies only (Appendix D). This is the largest correction listed here. No raw count or reported rate was affected, and the historical sensitivity set is preserved rather than rewritten.

C Reproducibility and protocol

Open-weight panel. vLLM (Kwon et al., 2023) with eager execution and prefix caching disabled; decoding pinned and validated before generation at temperature 0, top-p 1, top-k -1, repetition penalty 1.0, maximum 2048 new tokens, $n = 1$, fixed seed. The runner builds a decoding manifest, compares requested against effective values, and aborts on mismatch. Evaluation runs in a Singularity container (Kurtzer et al., 2017). One response per task.

Frontier panel (API-native). Both providers were run with reasoning or extended thinking explicitly disabled, a fixed maximum output of 2048 tokens, one response per task, and no tools, web access or code execution. **Neither request carried temperature or top-p**: reasoning models do not honour them and the installed provider SDK removes them from the request entirely. Sampling was therefore provider-default and **not caller-controlled**. Every result row records the requested configuration, the configuration actually sent (null for anything omitted), and the resolved protocol including the reasoning/thinking setting.

Instrumentation outcomes, frontier panel. Across 444 generations: zero refusals, zero truncations, zero extraction failures, zero API or infrastructure errors, zero retries. Every non-pass is a genuine wrong answer.

Regenerating every number. All tables and figures regenerate from the frozen raw results without model or API access; see the reproduction README shipped with the artifact. A consistency audit re-derives every manuscript number from the canonical artifacts and exits non-zero on any disagreement.

D FORENSIC SCREENS AND CASE STUDIES

Evidence tiers. **Raw PPF** (Tier 1) is the automatic benchmark outcome: A passes, B passes, AB fails. Every rate in the paper is a raw-PPF rate. **Mechanically verified confound screens** (Tier 2) test for specific known failure modes by re-executing frozen programs; each produces a specific exclusion or flag, anyone can regenerate it, and none certifies that an unscreened case is composition-specific. **Mechanistic case studies** (Tier 3) are selected cases read at the program level. No tier yields, and the paper does not report, a global count of composition-specific failures; no screen result is subtracted from a raw count or divided into a rate.

Why there is no global count. Attributing a raw PPF to the *integration* of A and B, rather than to an unrelated slip, requires a program-level judgement. A forensic audit of an earlier AI-assisted classification showed that its verdicts on that attribution changed materially under reasonable operationalizations: whether an AB defect that also misbehaves on a precursor-domain input counts against integration, and whether an AB program identical to a precursor program does. No independent human adjudication and no inter-rater reliability study was performed, so the classification is withdrawn and only the mechanical screens below are retained.

THE SCREENS

All 84 reasoning-off raw PPFs — 82 open-weight, 2 frontier — were re-executed against every AB hidden test. For each, the valid input domain of each precursor was searched for an input on which the AB program disagrees with that role's reference; an input counts only if the AB reference and the role reference agree on it, so that it lies inside the projection. Wherever such an input was found, the model's matched precursor program was run on it.

Screen	Mechanical test	Result
Instrumentation, extraction, evaluator	frozen AB program re-executed on every AB hidden test; references re-validated under the same evaluator	every recorded failure reproduces; 0 load and 0 empty-extraction failures
Orthogonal sentinel	every failing test has -1 on exactly one side, and the specification's -1 convention concerns neither requirement	5 cases, all P4; not applied to P3 (below)
Precursor-coverage gap	matched precursor program wrong on a valid input of its own domain	6 cases, three P2 and three P5
Representation mismatch	role representations identical by construction in P2–P5	not mechanically separable in P1, which is flagged at pair level only
Byte-identity	AB program textually identical to a precursor program	screening flag only; 2 flagged

P4: orthogonal sentinel. The P4 AB specification returns -1 when the input contains no scalar values, a convention unrelated to nesting or type dispatch. All five P4 raw PPFs reproduce *only* on that convention:

Family	Model	Failures	Sentinel	Other	Example
c1_002	Qwen 32B	1/16	1	0	got -1, expected -2
c1_003	Qwen 32B	4/16	4	0	got -1, expected 0
c1_006	Qwen 32B	3/16	3	0	got -1, expected 0
c1_015	Qwen 32B	4/16	4	0	got -1, expected 0
c1_012	Phi-4 14B	5/16	5	0	got -1, expected 0

In every failing case the model conflated "no scalar values" with "the computed value is ≤ 0" and returned -1 where a real value was due, or the converse; nesting was traversed and types dispatched correctly throughout. P4 therefore contributes raw signal but no mechanistic evidence for the intended interaction in these five cases.

P3: the sentinel is the capability. In P3 the AB specification returns -1 when too few items are usable, which is how the partial-failure requirement itself is expressed. In 14 of the 29 P3 raw PPFs every failure has -1 on one side, but mishandling that value may be the very failure being measured, so **sentinel involvement is not automatically an artifact** and no P3 case is excluded on that ground.

Precursor-coverage gaps. In six raw PPFs the witness input lies in the role-B domain and the model's own role-B program, which passed all of role B's hidden tests, produces the same wrong output as its AB program:

Pair, family	Model	Domain	Witness input	Reference	Precursor	AB
P2 `c9_004`	Qwen 32B	B	`'((2)b)'`	`0`	`1`	`1`
P2 `c9_007`	Qwen 14B	B	`'((2)b)'`	`0`	`1`	`1`
P2 `c9_008`	Qwen 14B	B	`'((2)b)'`	`0`	`1`	`1`
P5 `c5_002`	Qwen 14B	B	`[('undo', 0), ('mul', 0), ('undo', 0), ('undo', 0), ('undo', 0)]`	`0`	`ZeroDivision`	`ZeroDivision`
P5 `c5_007`	Phi-4 14B	B	`[('undo', 0), ('mul', 0), ('undo', 0), ('undo', 0), ('undo', 0)]`	`0`	`ZeroDivision`	`ZeroDivision`
P5 `c5_007`	Qwen 14B	B	`[('undo', 0), ('mul', 0), ('undo', 0), ('undo', 0), ('undo', 0)]`	`0`	`ZeroDivision`	`ZeroDivision`

`ZeroDivision` abbreviates a raised `ZeroDivisionError`. Each is a specific coverage-gap exclusion: the defect was reachable in the precursor and its tests missed it. In all three P5 cases the programs undo `mul n` by integer division by `n`, which fails at $n = 0$; an earlier targeted search for `c5_007` (Qwen 14B) found the shorter witness `[('mul',0),('undo',0)]`. The search is bounded, so finding no gap in the other cases is not proof of absence, and **no rate or bound** is drawn from it.

Byte-identity is a flag, not a verdict. Two Qwen 14B P5 programs were flagged as byte-identical to the role-B program: `c5_007`, a confirmed coverage gap above, and `c5_008`, whose failing AB input needs at least two state-changing operations, which role B's domain forbids; no witness was found for it, so it is not excluded. Submitting a precursor strategy unchanged is itself a failure to integrate, so identity alone supports no conclusion.

A historical sensitivity analysis. An earlier draft reported that excluding pre-documented limitation families moved the open rate from 49.4% (82/166) to 48.1% (78/162). That set is exactly five families, `{c1_003, c1_009, c1_012, c8_008, c8_009}`: a P4 subset `{c1_003, c1_009, c1_012}` annotated as -1 sentinel readings, and a P3 subset `{c8_008, c8_009}` annotated as a correlated two-level phrasing difficulty. The analysis code records both as limitations noted before the frontier runs; version history does not independently establish that ordering. The P4 subset is preserved here exactly as recorded, and it is **not** the five-case sentinel screen above: `c1_009` is not a raw PPF, and `c1_002`, `c1_006` and `c1_015` are sentinel cases it omits. No claim in the paper depends on this analysis.

Additional case studies

Selected for depth, not sampled; interpretations, not validated classifications.

`c5_010` (Claude Sonnet 5). Beyond the program diff in §6, the search finds a role-B-domain input on which the AB program fails while the role-B program is correct, `[('undo',0),('add',4),('undo',0),('undo',0),('undo',0),('undo',0)]`. That input locates where the frozen defect can manifest, not why it was introduced, and is neither positive nor negative evidence for integration attribution.

`rt_003` (GPT-5.6 Sol). The search found no input on which the AB program fails among 584 valid role-A and 640 valid role-B inputs, an absence that is evidence rather than proof. Its summary is

Table 3: Benchmark composition. *Distinct functions* are behavioural equivalence classes under the **behavioural-distinctness audit**, a finite battery of 15,000 + in-domain inputs per role, not proved mathematical equivalences; each count is therefore a *lower* bound on distinctness, since a larger battery can only split classes further, which raises the count. This audit is separate from the projection-equivalence audit and has different sizes. Construct-validity status per pair is in Appendix E.

Pair	Capabilities $A \times B$	Families	Tasks	Distinct fns (A/B/AB)
P1	recursion × branching	20	60	20 / 20 / 20
P2	nesting × delimiters	12	36	8 / 8 / 12
P3	aggregation × partial failure	12	36	12 / 12 / 12
P4	nesting × type dispatch	15	45	15 / 15 / 15
P5	state × rollback	15	45	15 / 12 / 15
Total		**74**	**222**	

Table 4: Construct-validity status by pair. P1 is a **legacy** pair: it predates the projection-audit protocol, uses role-specific representations, and has *no* projection-equivalence audit (it does have the separate behavioural-distinctness audit of Table 3). P2–P5 follow the later projection-based methodology and carry empirical projection-equivalence validation over the documented in-domain batteries. No battery is exhaustive over its (unbounded) domain, so none of these is a mathematical proof. P1's results are reported unchanged.

Pair	Shared representation	Projection audit	Projection-audit size (A/B)	Construct-validity status
P1	role-specific (`ChainNode` / `Star` / `Node`)	none	—	**legacy**: predates the projection-audit protocol; weaker construct-validity guarantees
P2	shared `str`	yes	64,380 / 50,352	projection protocol, empirically validated
P3	shared `list` of items	yes	36,480 / 40,092	projection protocol, empirically validated
P4	shared nested `list`/`dict`	yes	39,840 / 39,840	projection protocol, empirically validated
P5	shared `list` of `(kind, n)`	yes	31,275 / 780	projection protocol, empirically validated

exact only when a child's subtree holds a single value, the situation role B guarantees, and branching removes the total ancestor order that makes role A's global sort valid.

`c8_006` — aggregation-level collapse. Raw PPF for four of five open models. Role A counts items whose maximum reading is ≥ 2; role B applies a single-level usability filter. In AB the models implement the filter and then replace the per-item `max` reduction with a flat scan over all present readings — indistinguishable at one level, which is what role B guarantees. `c8_006` is behaviourally distinct from every other P3 family in all three roles (Appendix E). The four failures are **four model-specific observations on the same task**, not four independent observations.

`c5_007` — a coverage gap. Qwen 14B emits a byte-identical program for role B and role AB; B passes, and AB raises `ZeroDivisionError` on `mul 0` followed by `undo`. A witness inside role B's domain triggers the same defect, so the bug was reachable in the precursor and B's hidden tests did not cover it. It is a raw PPF by the metric and an excluded case under the coverage screen, and it identifies a real limitation of matched-family designs.

E Construct validity and projection audits

Reading the table. P1 is a **legacy** pair. It was authored before the projection-audit protocol existed, uses role-specific representations rather than a shared one, and has no projection-equivalence audit at all; its construct validity rests on the representation design alone. It is reported here as satisfying a

weaker standard than P2–P5, not as satisfying the later protocol. Its results are neither recomputed nor removed.

P2–P5 follow the projection methodology and carry empirical projection-equivalence validation over the documented in-domain batteries. Each projection battery combines an exhaustive sweep of a *bounded* sub-domain with a large random sample beyond it. The underlying domains are unbounded — arbitrary string length, list length and integer range — so **no battery is a mathematical proof over the domain**, and none is described as one.

Two audits, not one. Table 4 reports the **projection-equivalence audit**, which asks whether a precursor's rule equals the AB rule restricted to that precursor's domain. Its size differs per pair and per role because each enumerates a bounded parameterisation of that role's own domain: P5 role B's 780 inputs are small because that domain permits at most one state-changing operation, so the enumeration is nearly complete, not because the battery is undersized.

Table 3 reports the **behavioural-distinctness audit**, a different question with a different battery: how many behaviourally distinct functions a pair's families realise in a given role, measured by executing the frozen reference solutions on 15,000 + in-domain inputs per role, converged (identical at 3,000 and 15,000). P1 has *no* projection-equivalence audit but does have this one (15,060 / 15,057 / 15,101 inputs, 20/20/20 distinct), so "P1 has no audit" would be wrong as stated.

The distinct-function counts are *lower* bounds on distinctness: a larger battery can only split classes further, never merge them, which raises the count. The error direction is the safe one, since it understates distinctness and so overstates the dependence concern. An earlier 16-input battery produced three equivalence classes that do not survive, which is why the battery size is reported alongside every count.

F CLOSEST-WORK COMPARISON

Novelty, stated precisely. ConceptBench contributes a matched semantic-projection instrument for studying composition after demonstrated constituent performance: unlike prior composition benchmarks, which combine separate parent problems, separate sub-questions, sequential functions $B(A(x))$, or larger structural units, its constituents are controlled projections of one composed semantic function onto domains where the complementary capability is structurally absent. We claim neither conditional composition evaluation nor conditioning on constituent success as novel. Press et al. (2023) state the conditional first; EvoEval's *Combine* is prior conditional composition evaluation in code and *Combine-naive* is prior sequential functional composition $B(A(x))$.

G THE PRE-SPECIFIED REASONING-ENABLED INTERVENTION

Tasks, prompts, extraction, evaluator, model identity, and sampling-field policy were held fixed; provider-native reasoning configuration and the output-token ceiling changed. For GPT-5.6 Sol, `reasoning_effort` moved from `none` to `medium` and `max_completion_tokens` from 2,048 to 16,384. For Claude Sonnet 5, `thinking` moved from disabled to `adaptive`, `output_config effort` from absent to `high`, and `max_tokens` from 2,048 to 16,384. Sampling parameters were sent to neither. The two controls are provider-native and **no equivalence between them is claimed**; because the reasoning setting and the output ceiling changed together, no difference between conditions is attributed to either alone. This panel is never pooled with the reasoning-off panel or with the controlled open-weight panel, and it is descriptive rather than causal.

Pre-specification, not public registration. The design was written down and committed to version control before any reasoning-enabled request: the protocol on 11 September 2026 (commit `0a5c8d6`), and one pre-execution amendment at 01:47 UTC on 12 September (commit `7c6c718`) replacing a manual thinking budget, which Claude Sonnet 5 rejects, with adaptive thinking at effort `high`. Models, settings, output ceilings, analysis tables and interpretation rules were unchanged from then until execution; later pre-execution commits recorded the data-policy gate, credential checks and non-benchmark smoke tests, with the tooling that ran them, the last at 10:12 UTC (`7fa5209`). All 444 generations ran between 10:15 and 10:23 UTC that day; the raw layer was later fingerprinted and

Table 5: Closest work. *Conditions on constituent success* is the column that matters for our novelty claim, and we claim no priority for that move: Press et al. (2023) named and measured the conditional first, and EvoEval answers yes for both of its composing variants. Our claim rests on the *matched constituents* column instead.

Work	Unit of evaluation	Notion of composition	Matched constituents?	Conditions on constituent success?	Main distinction from ConceptBench
Compositionality gap (Press et al., 2023)	one multi-hop question	composed question over two independently stated facts	no	**yes**	**Earliest statement of the conditional we use.** Constituents are separate sub-questions in factual QA, not projections of one function; no concept-pair stratification, no program-level case analysis
EvoEval *Combine* (Xia et al., 2024)	one evolved problem	two *unrelated* seed problems merged, paired by input-argument type	no	**yes**	Our A, B are projections of one AB semantic function, not two separate tasks; we stratify per concept pair rather than reporting one aggregate
EvoEval *Combine-naive* (Xia et al., 2024)	one composed problem	sequential functional composition $B(A(x))$, type-matched, manually built	no	**yes**	Composition is *chaining two functions*; our precursors are one function restricted to domains where the other capability cannot arise
HumanEval Pro / MBPP Pro (Yu et al., 2024)	base + self-invoking pair	harder problem must call the base solution	partly	no (confusion matrix only)	One function reused, not two distinct capabilities
ClassEval (Du et al., 2024)	a class (410 methods)	methods composed into a class	no	no	Structural composition (methods → class)
ClassEval-Pro (Chen et al., 2026)	a class (300 tasks, 11 domains)	cross-domain class assembly; cross-method coordination	no	no	Unit is a whole class; composition architectural, not concept interaction
BigCodeBench (Zhuo et al., 2025)	one task	many library calls under complex instructions	no	no	"Compositional" means multi-API orchestration
SCAN (Lake & Baroni, 2018)	one command	primitives + modifiers	yes, by construction	split-level only	Isolation is a training-split property, not demonstrated test-time success
COGS (Kim & Linzen, 2020)	one sentence	familiar words in new structures	yes, by construction	no	Synthetic language; logical-form match
ConceptBench	matched family (A, B, AB)	two concepts interacting inside *one* semantic function	**yes**	**yes**, per family per model	—

tagged `phase4-raw-frozen` (provenance below). The protocol was **not deposited in a public registry**. It also pre-specified per-case adjudication of every raw PPF under the earlier criteria; that layer is not reported (§6).

GPT-5.6 Sol — reasoning on.

Pair	A	B	Joint	AB	Raw PPF	Raw-PPF rate
P1 recursion × branching	20	20	20	20	0	0.0%
P2 nesting × delimiters	12	11	11	10	1	9.1%
P3 aggregation × partial failure	12	12	12	12	0	0.0%
P4 nesting × type dispatch	15	15	15	15	0	0.0%
P5 state × rollback	15	15	15	15	0	0.0%
Overall			**73/74**		**1**	**1.4%**

Instrumentation: 222/222 without fault, 0 truncations, 0 infrastructure errors, 0 answerless responses.

Claude Sonnet 5 — reasoning on.

Pair	A	B	Joint	AB	Raw PPF	Raw-PPF rate
P1 recursion × branching	20	20	20	20	0	0.0%
P2 nesting × delimiters	12	10	10	10	0	0.0%
P3 aggregation × partial failure	12	12	12	12	0	0.0%
P4 nesting × type dispatch	15	15	15	15	0	0.0%
P5 state × rollback	15	13	13	14	1	7.7%
Overall			**70/74**		**1**	**1.4%**

Instrumentation: 222/222 without fault, 0 truncations, 0 infrastructure errors, 0 answerless responses.

Matched transitions. Eligibility transitions are reported alongside the AB transitions, because an apparent repair is not mitigation if the family merely became ineligible.

Transition	GPT-5.6 Sol	Claude Sonnet 5
eligible → eligible	71	63
eligible → ineligible	0	2
ineligible → eligible	2	7
ineligible → ineligible	1	2
AB outcome, over families jointly eligible in BOTH conditions		
pass → pass	70	61
pass → fail	0	1
fail → pass	1	1
fail → fail	0	0

The single `pass` → `fail` transition is `c5_001` and is reported rather than set aside. The other new raw PPF, `c9_011`, comes from a family that became eligible under reasoning. Neither previously failing family changed eligibility, so both AB passes are not artifacts of a family dropping out.

`rt_003` (GPT-5.6 Sol) — AB now passes. Reasoning off: A pass, B pass, AB fail at 16/17. Reasoning on: A pass, B pass, AB pass at 17/17. The reasoning-off program propagates a lossy `(subtree_min, subtree_max)` summary that is exact only when a child's subtree holds a single value. The reasoning-on program abandons that strategy for an ancestor-aware traversal: a depth-first enter/exit walk maintaining the live ancestor set in an indexed structure, querying true predecessor and successor. Reasoning tokens were 116 for A, 0 for B and 1,024 for AB.

`c5_010` (Claude Sonnet 5) — AB now passes. Reasoning off: A pass, B pass, AB fail at 4/16. Reasoning on: A pass, B pass, AB pass at 16/16. The reasoning-off program collapses the undo stack and the visitation record into one `history` list, so `undo` pops values out of the count. The reasoning-on program keeps rollback state and count state separate: a stack that is popped on `undo`, and an independent counter incremented after every operation. No thinking block was exposed for any role; output tokens were 226, 1,425 and 1,977 for A, B and AB.

The two new raw PPFs, described. No mechanistic label is assigned to either. The table records only mechanical facts from the same screens applied to the reasoning-off panel (Appendix D). A precursor-domain input on which the AB program fails locates where the frozen defect can manifest, not why it was introduced, and is neither positive nor negative evidence for integration attribution.

Mechanical fact	`c9_011` (GPT)	`c5_001` (Claude)
instrumentation or extraction failure	none	none
failing AB hidden tests	18/19	3/16
byte-identity screening flag	did not fire	did not fire
precursor-domain inputs where AB fails	yes, 1,856	yes, 10
smallest such input	`()`	`[('add',-2),('undo',0),('undo',0)]`
matched precursor program correct there	yes	yes
precursor-coverage gap	no	no
orthogonal sentinel	no	no
representation shared across roles	yes	yes
mechanistic label	not assigned	not assigned

`c9_011`: the AB program counts parentheses at depth ≤ 1, which the model's own role-B program excludes. An exhaustive search over balanced strings to length 6 found 1,856 role-B-domain inputs on which the AB program fails; the smallest, `()`, contains no comma. `c5_001`: the AB program implements `undo` by appending the penultimate state instead of popping, so a second `undo` cannot roll back further; it fails role B's own hidden test `many_undos`, and the smallest such input carries a single state-changing operation.

Reasoning exposure, descriptive only. GPT-5.6 Sol reported non-zero reasoning tokens on 148 of 222 generations (median 152 when present, range 16–1182), rising by role: A 22/74, B 55/74, AB 71/74. Claude Sonnet 5 exposed **no thinking block on any of its 222 generations** and reported no thinking-token usage, while the accepted configuration is recorded in the run provenance. **No inference is drawn that the setting was ignored**, and reasoning quantity is never correlated with correctness: it was not randomised.

Provenance. The raw layer was fingerprinted before analysis and tagged `phase4-raw-frozen`. Manifest covering all 444 records: `8513df142f8b02ac26f542d0b2c8aed7...` Integrity audit: 33/33 checks pass — one scored response per task, no duplicate or missing task record, unique response identifiers throughout, zero semantic retries, zero infrastructure errors, zero truncations. All 444 generations completed before analysis; formal hashing and the freeze commit followed the first analysis pass, and no raw response, task, configuration, evaluator or benchmark artifact changed in between.

One orchestration defect is recorded for completeness. A monitoring loop polled for the sweep with a pattern that also matched the monitor's own command line, so three such loops outlived the completed sweep. This is a **non-scientific orchestration bug with no effect on requests, outputs, scoring or configuration**: it issued no request, reran no task and changed no result.

H COMPLETE FRONTIER RESULTS

Table 6 gives the reasoning-disabled frontier panel per pair: precursor and composed pass counts, joint eligibility, raw PPF, and the instrumentation outcome of every generation. *Elig.* is the joint count of families where A and B both pass, not an individual precursor pass rate. Each model produces one raw PPF — `rt_003` (GPT-5.6 Sol, P1) and `c5_010` (Claude Sonnet 5, P5) — and no refusal, truncation, extraction or API failure occurred. The reasoning-enabled panel is reported separately in Appendix G.

Table 6: Frontier external validation, reported as a *separate* panel. Not comparable cell-for-cell with Table 1: neither provider exposes sampling control, so these are single draws from a provider-default sampler while the open panel is pinned. The rate denominator is the eligible count in the same row.

		Passes						Artifacts			
Model	Pair	A	B	AB	Elig.	Raw PPF	Rate	Ref.	Trunc.	Extr.	API
GPT-5.6 Sol	P1	20	20	19	20	1	5%	0	0	0	0
GPT-5.6 Sol	P2	12	9	10	9	0	0%	0	0	0	0
GPT-5.6 Sol	P3	12	12	12	12	0	0%	0	0	0	0
GPT-5.6 Sol	P4	15	15	15	15	0	0%	0	0	0	0
GPT-5.6 Sol	P5	15	15	15	15	0	0%	0	0	0	0
Claude Sonnet 5	P1	19	19	20	18	0	0%	0	0	0	0
Claude Sonnet 5	P2	12	10	10	10	0	0%	0	0	0	0
Claude Sonnet 5	P3	12	12	12	12	0	0%	0	0	0	0
Claude Sonnet 5	P4	15	15	15	15	0	0%	0	0	0	0
Claude Sonnet 5	P5	15	10	11	10	1	10%	0	0	0	0

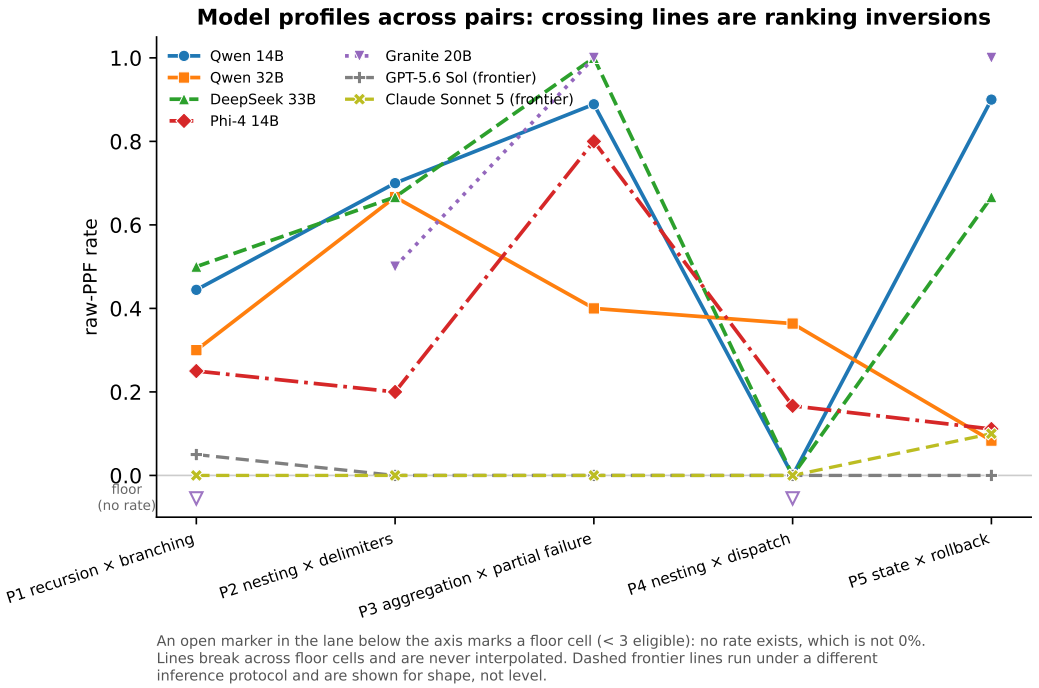

Figure 1: Model profiles across pairs; crossing lines are ranking inversions. The y axis is the raw-PPF rate, denominator the per-cell eligible count. An open marker in the lane below the axis marks a floor cell (fewer than three eligible families): no rate exists, which is not 0%. Lines break across floor cells. Dashed frontier lines run under a different inference protocol and are shown for shape, not level.

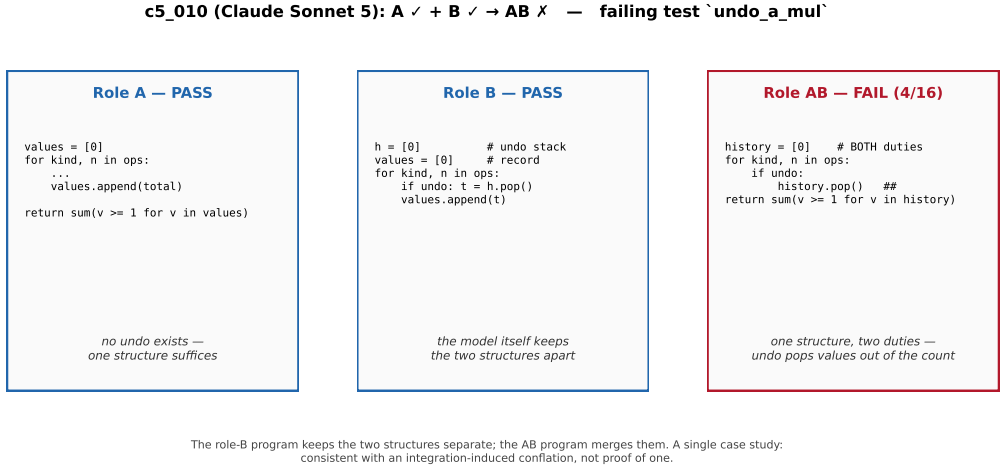

Figure 2: Role programs for `c5_010` (Claude Sonnet 5): A passes, B passes, AB fails. In role B the model kept the undo stack and the visitation record as separate structures; in AB one list serves both duties. No rate is shown; this is a single case study, and the programs are consistent with, not proof of, an integration-induced conflation.

I Additional figures

Four figures moved from the main text for length: model profiles across pairs (Figure 1), the three role programs of the `c5_010` case study (Figure 2), pair-stratified rates with intervals (Figure 3), and the model $\times$ pair heatmap (Figure 4). Every plotted rate is a raw-PPF rate.

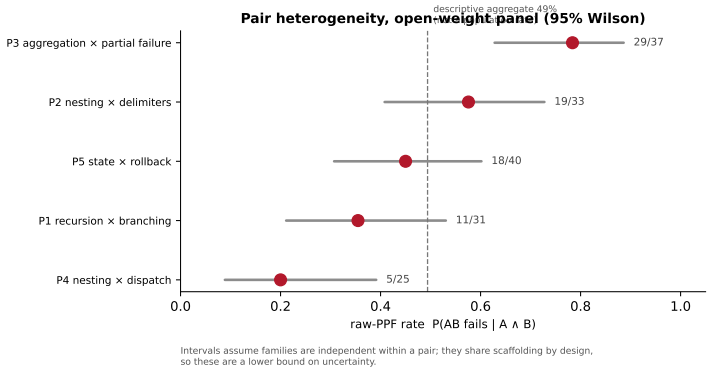

Figure 3: Pair heterogeneity in the open-weight panel. Points are raw-PPF rates over all five models; the denominator (raw PPF / eligible families) is printed at right. Bars are 95% Wilson intervals, which assume within-pair independence and are therefore a lower bound on uncertainty.

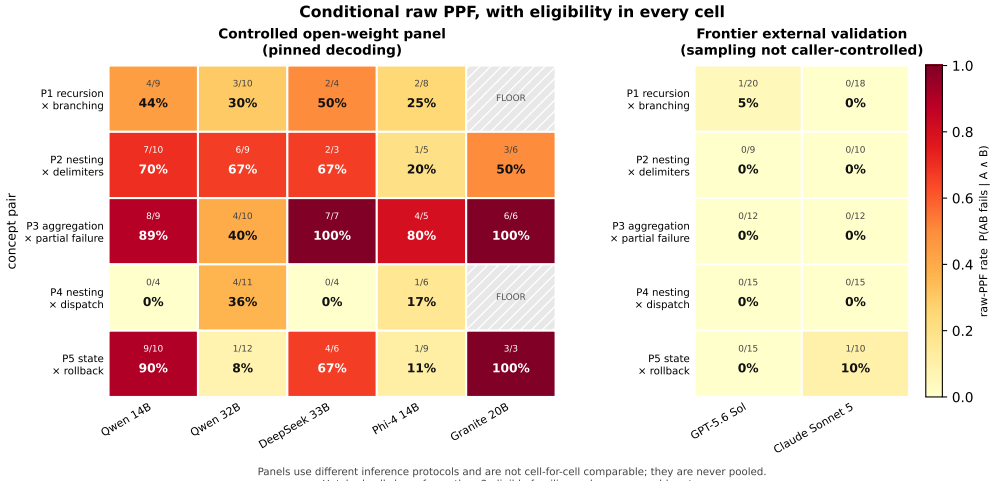

Figure 4: Raw-PPF rate by model and concept pair. Colour encodes the **raw-PPF rate**, whose denominator is the eligible-family count printed in each cell as `raw PPF/eligible`. Hatched cells have fewer than three eligible families: no rate exists there, which is not 0%. The two panels use different inference protocols and are never pooled.