COMPLEXITY LADDER: A CONTROLLED TEST OF SPECIFICATION COMPLEXITY IN CODE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

A previous study, ComplexityKink, found on a scraped corpus of about 5,000 prompts that the structural complexity of a programming prompt predicts whether a language model solves it, and that the relationship has a breakpoint. An observational corpus cannot say whether complexity itself did that, because prompts that are complex in the wild differ in many other ways at once. This study controls the prompt instead of sampling it. We take one hundred base problems, ninety-eight of them contest problems taken as written, and produce five further versions of each by adding one requirement at a time, holding the base task, the function signature and the test harness fixed: 600 prompts, each rated on the same frozen rubric by four judges. Sixteen models generate every prompt five times, 48,000 draws in all. For each model separately, a pre-registered and calibrated procedure tests whether family-adjusted correctness departs from a single straight line along the rated complexity composite and, where it does, locates the change and reports its direction and size. No model meets the registered criterion for a breakdown: one, GPT-5.6 Sol, rejects the straight line but its departure cannot be located on the grid, and the other fifteen do not reject at this design's precision. Every model is nonetheless lower at rung 5 than at rung 0, by 2.5 to 40.8 points of graded correctness. A descriptive split, chosen after the result, shows that for six models most of that decline arrives as answers cut off at the output limit, which score zero, while correctness conditional on a program running changes little, and that for the others the programs that run get worse. These results do not establish that breakpoints are absent, and they do not explain the earlier observational finding; they show what a controlled ladder, read with a calibrated procedure, can and cannot see.

1 INTRODUCTION

The question is simple to state. As a programming specification gets more complex, where, if anywhere, does a model stop being able to implement it? ComplexityKink (Hernandez & Zhao, 2026) asked it on about 5,000 scraped prompts and 21 models. That study scored every prompt before generation on six structural dimensions, branching, iteration, state, data structures, edge cases and composition, using four judge models, and found breakpoints in conditional reliability along the composite score. It also reported something that bothers us more than the breakpoints: the six dimensions did not agree about the sign of the effect, and overidentification tests rejected in every model-specific fit. Either the composite is not measuring one thing, or the dimensions are picking up different things that travel with complexity in a scraped corpus. Prompts that score high on structure in the wild are also longer, which by itself degrades reasoning (Levy et al., 2024), drawn from different domains, written by different people, and more likely to leave something unspecified. Any of those could produce a breakpoint, and an observational corpus cannot tell which did.

So this study does not sample prompts. It builds them. The idea is a ladder. Take a contest problem as written and call that rung 0. Add one requirement, recompute the worked examples, and call that rung 1. Add another for rung 2, and so on to rung 5. The base task, the function signature, the problem statement and the test inputs never change; the only thing that changes between rungs is a numbered list of requirements and the example answers computed from it, and the list at any rung is an exact prefix of the list at every rung above it. Within a family, what varies is the specification. Across families the base problem varies too, and a family intercept absorbs that. We built one hundred such families, 600 prompts, under eight mechanical checks that fail the build rather than

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.

warn, and had every prompt rated on the frozen rubric ComplexityKink used, so the two studies share their instrument. We want to be plain about what this identifies. Rung is assigned, but the analysis axis is the judge-rated composite C, and whatever else changes when a requirement is added changes with it: the prompt gets longer, the feasible set narrows, and the algorithm needed may change. A breakpoint on C therefore describes where a model's correctness changes along measured complexity within controlled families. It is not a randomized estimate of the effect of complexity, and we do not call it one.

Where this sits. Code-generation benchmarks measure correctness on fixed tasks (Chen et al., 2021; Austin et al., 2021; Hendrycks et al., 2021; Li et al., 2022; Jimenez et al., 2024; Jain et al., 2025; Zhuo et al., 2025); instruction-following benchmarks vary how many constraints a prompt carries (Zhou et al., 2023; Jiang et al., 2024; Qin et al., 2024; Wen et al., 2024), and compositional-depth studies report performance falling with the number of steps or clauses a task needs (Dziri et al., 2023; Mirzadeh et al., 2025; Shojaee et al., 2025). Whether such declines are smooth or discontinuous is itself contested: scaling curves have been fitted with breaks (Caballero et al., 2023), and apparent jumps can be artifacts of the metric (Wei et al., 2022; Schaeffer et al., 2023). We take the statistical machinery for that question from the structural-change literature, a sup-type test over unknown change points (Andrews, 1993; Bai & Perron, 1998; Hansen, 2000) with cluster-robust wild-bootstrap inference (Cameron et al., 2008; MacKinnon & Webb, 2017), we score complexity with judge models in the manner of Zheng et al. (2023), and we pre-register the whole procedure in the sense of Nosek et al. (2018). Models that reason at length before answering make the output budget part of the instrument (Snell et al., 2025; Muennighoff et al., 2025; Lawsen, 2025) (see also Appendix J).

What we test. Sixteen models each generate every rung of every family five times. For each model separately, we ask whether the family-adjusted relationship between graded correctness and C is adequately described by a single straight line, and whether the departure, where there is one, is downward and can be located: sixteen tests under Holm's procedure, no pooled verdict, and no prediction that every model breaks. The procedure is deliberately asymmetric and was calibrated before the run (Section 3); a located break is strong evidence, and the absence of one is weak evidence about any single model. We wrote into the registration, before any confirmatory draw existed, that many models would probably not break and that a panel of non-rejections had to be read as what it is.

What we found. No model earns a located break: GPT-5.6 Sol departs from a straight line after Holm but the departure cannot be placed on the grid, and the other fifteen do not reject at this design's precision, which is not evidence that their curves are straight. Every model is nonetheless lower at rung 5 than at rung 0, and a descriptive split of the draws, chosen after the result, shows two patterns behind that: for six models most of the decline arrives as answers cut off at the output limit, which score zero, while correctness conditional on a program running changes much less; for the rest, the programs that run get worse.

Pre-registration. The design, the corpus, the rating index, the analysis code and its calibration were frozen into one content digest over 2,394 artifacts and registered on OSF (link withheld for double-blind review) before the first confirmatory draw. The registration was updated eighteen times during the run, each time with the reason and the date, mostly because hosted endpoints dropped long responses; none of the updates changes the design, the hypotheses, the measures or the analysis, and Appendix H records them. The primary analysis ran once, on an admission artifact that recorded, for every one of the 48,000 planned draws, the checks the join performed; the registered secondary analyses are not reported in this submission, and we say so where they would appear.

Contributions.

1. A corpus of 100 six-rung ladders, 600 prompts, built under mechanical checks that fail the build rather than warn, with every retired base kept together with the measurement that retired it (Section 2, Appendix C).
2. A pre-registered, calibrated procedure for finding a model-specific breakpoint along a frozen judge-rated complexity index, with a location set, direction labels and simultane-

ous effect bands, and with its false-alarm and detection rates published in full (Section 3, Appendix G).

3. The confirmatory result on sixteen deployed models over 48,000 draws, with every draw retained: no model meets the registered criterion for a breakdown along C, every model is lower at rung 5 than at rung 0, and a descriptive split separates a decline that arrives as answers cut off at the output limit from one that arrives as wrong programs (Sections 4 and 5).

2 THE LADDER

Families and rungs. The unit of manipulation is the family: one contest problem at rung 0 and five further versions, each adding one requirement. All six rungs share one layout, the contest's statement and constraints, the task restated, a numbered list of additional requirements, and worked examples. Across the rungs of a family the function signature, the number of test cases and their inputs, and the problem statement are fixed. Only the requirement list and the worked-example answers vary, and the examples are recomputed from a reference function at every rung, rung 0 included, so no prompt contains an example that contradicts its own requirements. The unit of observation is the draw: one rung prompt sent, one response received, a program extracted from it, and that program run against the family's tests.

How a ladder is built. Each family provides more candidate requirements than the ladder needs, one reference function that computes the answer under any subset of them, and a pool of candidate inputs. Which five requirements are used, in which order, and on which inputs was chosen by search and fixed in the family module, because which rules can bind is a property of the base problem and is not visible in the wording. Eight checks then run before any artifact is written, and the build fails if any of them does: among them, each rung's expected answers must differ from the rung below's on at least a quarter of the shipped cases, no rung may be passable by a constant, no requirement may be redundant, and every input must satisfy the constraints the prompt states. Appendix C gives the checks in full, what the pilots taught us about bases that cannot carry a ladder, and the independent oracles against which every added requirement was checked. Requirements were chosen by measurement rather than by reading, because in piloting a requirement that changes no answer reads exactly like one that does. The candidate requirements, references and case generators were written by language models working in this project, the authoring model is recorded per family, and the authoring model that wrote most families is not on the panel.

Corpus. Ninety-eight bases come from LiveCodeBench (Jain et al., 2025), with contest dates from 2023-06-04 to 2025-03-29, and two were written for this study to fill harder difficulty bands. Selection was purposive: candidates were screened for solvability, measured, and chosen to spread difficulty, and bases that could not carry a ladder were replaced for properties of the base, never for an outcome above rung 0. Each rung carries between 10 and 20 small shipped test cases, chosen by search for how well they separate a rung from the one below; correctness in this study means correctness on these suites, which can pass wrong programs (Liu et al., 2023); 98 families also carry a held-out suite, a secondary outcome not reported here. On the preparation round, over its executed draws, mean graded correctness at rung 0 was 0.898 across families, so rung 0 sits near the top of the scale for most of this panel: the ladder's headroom is real, but it is headroom to fall.

The complexity index. Each of the 600 prompts is rated on the frozen ComplexityKink rubric, six dimensions scored as integers from 0 to 4, by four judge models (o4-mini, GPT-5.5, Llama 4 Maverick and Cohere Command A), 2,400 rating slots in all. For prompt p, judge j and dimension d,

$$Z_{p,d} = \tfrac{1}{4} \sum_{j} s_{p,j,d}, \qquad C_p = \sum_{d} Z_{p,d}, \tag{1}$$

so C lies in $[0, 24]$ in multiples of one quarter and is stored as an exact rational, so no split ever depends on rounding. Ratings are prompt-side only: judges see the prompt and the rubric, never a response or a test result, and the index was frozen on 2026-09-18 before any confirmatory outcome existed. On the realized index mean C rises from 11.35 at rung 0 to 15.56 at rung 5; it is not monotone in rung within most families, and those families stay in every analysis, because the index

is a measurement of the rung and a family whose index does not rise is a fact about the instrument to report. Between-judge agreement is modest (mean pairwise Spearman 0.51), and the uncertainty we report covers variation in generated responses, not judge error (Appendix E).

Panel and requests. Sixteen models were fixed before any confirmatory draw, in a frozen order that breaks Holm ties (Table 3 in Appendix D); none is a complexity judge. Every model receives the same system prompt, which asks for a complete Python implementation returned as code only, and the same rendered prompt. Every model has an explicit output limit, the largest number of tokens it may generate for one answer: 65,536 on fourteen routes, 64,000 for Mistral Medium 3.5 (a host limit on prompt and response together) and 131,072 for MiniMax M3 (raised on preparation-round evidence that 65,536 bound on the easiest rung), set that high because piloting showed a binding cap manufactures a complexity effect (Appendix A); where a model has a reasoning setting it is set to its highest level, temperature is omitted on thirteen routes and set to 1.0 on three, and the program is extracted from the answer alone as the longest fenced Python block. Every model generates every rung of every family five times, in a seeded order fixed before the first call, so rung is not aligned with position or drift at an endpoint. The exact request of every draw is retained and checked against the request policy before it goes out, and on the fifteen synchronous routes the served model identity each endpoint reports is checked on receipt; the native batch route reports none.

Outcomes. The primary outcome is graded correctness, the fraction of a rung's shipped tests that a draw's program passes; all-pass and the same two quantities on a held-out suite are secondary. A received response is final and is never requested again. A draw that reaches the output limit is recorded as truncated and scored zero: the limit is ours and has to sit somewhere, but spending all of it without answering is the model's. A refusal, an empty response and a program that exceeds the 30-second test limit also score zero, while a transport failure, where nothing is received, is missing data and is requested again under a protocol fixed in advance. Every draw keeps its response bytes, extracted program, per-test results, serialized request, finish reason, served identity and freeze digest (Appendix F).

3 THE REGISTERED ANALYSIS

For each model separately, the five draws in each family-by-rung cell are averaged to one graded-correctness value, 600 equally weighted observations per model. For model m, family f, rung k and candidate split g,

$$\bar{y}_{f,k,m} = a_{f,m} + b_m C_{f,k} + d_{m,g}\,\mathbb{I}(C_{f,k} > g) + h_{m,g}\,(C_{f,k} - g)\,\mathbb{I}(C_{f,k} > g) + \varepsilon_{f,k,m}, \qquad (2)$$

where the family intercept $a_{f,m}$ removes stable family differences, b_m is the low-regime slope, $b_m + h_{m,g}$ the high-regime slope, and $d_{m,g}$ permits a level discontinuity. Fits are least squares with family absorption, no covariates enter, and results are never pooled across models.

Candidates and the test. The grid of candidate splits was built once from the admitted C values, before any outcome existed and identically for all sixteen models: one candidate between each adjacent pair of distinct scores, admitted only where at least ten families sit on each side and cross the split and every reported contrast has at least ten informative families. That admits 25 contiguous candidates between C of 10.25 and 16.50; there is not enough corpus above 16.50 to estimate a high-side regime, so a change above that margin is not something this design can see. With RSS_0 the residual sum of squares of the family-adjusted straight line, the statistic is $T_m = \max_g \max(0, \mathrm{RSS}_0 - \mathrm{RSS}_g)$ over the admitted splits, calibrated by a restricted family-cluster wild bootstrap (Cameron et al., 2008) with a CR2-style adjustment (Bell & McCaffrey, 2002): one Rademacher sign per family per replicate, the whole rung residual vector multiplied by it, the null refitted and the complete split search repeated, with the same family sign stream shared across the sixteen models. The Monte Carlo p-value uses $B = 49{,}999$ replicates from a seed derived from the SHA256 of the frozen rating index. Holm's step-down procedure (Holm, 1979) at familywise $\alpha = 0.05$ runs over the sixteen p-values, so the first threshold is $1/320$, with ties broken by the frozen panel order.

Location, direction and the decision. Rejecting the straight line says the relationship is not one line; it does not say the departure is a break, and in calibration a smooth quadratic decline rejects

the line more often than a genuine five-point drop does. What separates the two is the location set, formed by inverting fixed-candidate profile tests over the grid at $\alpha_{\mathrm{loc}} = 0.025$ divided by sixteen. A real break concentrates it; smooth curvature leaves it spread across the grid, because every split fits about as badly as every other. Direction and size come from simultaneous family-cluster multiplier bands over all sixteen models, every admitted candidate and five contrasts (the two slopes, their change, and the gap between the fitted regimes at each bound of the candidate interval); direction at the fitted breakpoint alone is read from a pointwise interval from the same replicates. H1 is supported for a model only when its Holm-adjusted test rejects, its location set retains at most 6 of the 25 candidates, at least one downward label holds at the fitted breakpoint, and no upward label holds; otherwise the direction is reported as uncertain, and a model with both labels as a mixed regime change. The localization requirement is what makes the label informative rather than merely significant.

Calibration. Before the freeze the whole procedure ran on 21,500 simulated sixteen-model panels generated on the real 600 C values with the real test counts and draw-level noise shaped like the pilot's, across null, smooth, tied-support, heteroskedastic, top-of-scale, mixed and upward scenarios. Every gating bar was met with no remedial round (familywise false alarm 0.055 against a bar of 0.065, band coverage 0.964 and 0.984 against 0.955, wrong direction 0.002). Per panel, the full decision fires on 0.15 and 0.40 percent of straight-line panels and on 4.6 and 0.2 percent of smoothly declining panels; per model with a planted break, it fires on between 0 and 10.6 percent, so a large level change in the middle of the range is found about one time in ten, a change of half that size almost never, and a break at the edge of the support under one percent of the time because it cannot be localized. The maximum width of six retained candidates was itself chosen on these calibration runs, before any confirmatory outcome existed, as the largest width that met the false-alarm bar, so the calibration describes a rule selected on it rather than one validated independently. Appendix G gives the whole table with its denominators, because quoting its best row would misdescribe the design. Confirmatory analysis requires the complete planned roster, every draw admitted only after its identity, receipt, score, program and schedule slot are verified; a missing planned cell withholds the confirmatory result, no result is rerun with another seed, no B is increased, and nothing is extended because a result lies near a threshold.

4 RESULTS

The primary analysis ran once, on the admission artifact, with the registered budget of $B = 49{,}999$ replicates, and its output is marked confirmatory: 48,000 of 48,000 planned draws admitted, 25 candidate splits, sixteen tests. We report what it said in the order the registration fixed, and then what a descriptive split adds. The registered secondary analyses, the all-pass and held-out outcomes, the per-model comparison of a two-regime fit against a smooth cubic, the assigned-rung contrasts and the sensitivity analyses, are not reported in this submission; the artifacts to run them are released with the code.

H1 is not supported for any model. Table 5 in Appendix I gives the sixteen search p-values and Holm's step-down over them. One model's test rejects the family-adjusted straight line: GPT-5.6 Sol, with $p = 0.0008$ against a Holm threshold of 0.0031. Its location set holds 20 of the 25 admitted candidates and spans C from 11.50 to 16.50, nearly the whole grid. By the registered rule that is a relationship which is not one line but whose change cannot be placed, and it is not called a break. At the fitted candidate, the split between C of 15.75 and 16.00, the pointwise contrasts the direction rule reads show a downward level gap of -0.16 in graded score between the two working lines (intervals $[-0.31, -0.02]$ and $[-0.30, -0.03]$ at the two bounds) and a slope change indistinguishable from zero; across the twenty retained candidates the simultaneous-band envelope over all five contrasts spans $[-0.23, 0.18]$, so the departure is real, its direction at the best split is downward, and where it sits is not something these data say (Table 6; the full candidate set is in the released artifacts). Nine of the sixteen raw p-values are below 0.05 (Sol, Terra 0.0071, Luna 0.0092, gpt-oss-120b 0.025, Qwen3.8 27B 0.025, Qwen3.8 Max 0.033, MiniMax M3 0.040, Kimi K2.7 Code 0.047, DeepSeek V4-Pro 0.049), where under a global null fewer than one would be expected, and only Sol's survives Holm; the other seven range from 0.20 to 0.81. No model is labeled mixed and no upward label holds anywhere. One limitation bears directly on Sol. Seventy draws scored zero because the reply could not be parsed as an answer, 68 of them Sol's, rising from 2 at rung

0 to 15 at rung 5; reading their stored envelopes after the analysis, 65 were streams that ended in a provider-side failure event, which the registration classes as transport to be requested again and which the frozen harness instead recorded as a received reply and scored zero (Appendix H). We did not re-run the study; we bounded the effect. With those draws scored as full marks and the registered procedure otherwise unchanged, a sensitivity run that is not confirmatory, Sol's search p-value is 0.00028 against the registered 0.00084, its location set holds 18 of 25 candidates against 20, and no other model's decision changes. The departure is not an artifact of the zeros, and it fails to localize at both bounds, so the verdict for Sol is the same either way.

Every model is lower at rung 5 than at rung 0, most of them a lot. Between rung 0 and rung 5 every one of the sixteen has lower mean graded correctness, by amounts from 2.5 points (Gemini 3.8 Flash) to 40.8 points (Qwen3.8 27B); Table 7 gives every rung mean, the solid curves in Figure 2 show them, eight models have at least one rung above the one before it along the way, and the smallest differences are within family-level noise (Mistral Medium 3.5's 5.4 points comes entirely from rung 0 to rung 1 and its curve is flat after that). Three models stay near the top of the scale across the whole ladder: GPT-6 Astra (1.000 to 0.962), Gemini 3.8 Flash (1.000 to 0.975) and Grok 4.6 (0.994 to 0.930). Three start low and stay low: Llama-3.3-70B, Mistral Medium 3.5 and Mistral Large 3 score between 0.29 and 0.50 at every rung, with timeouts and wrong programs on the base problems accounting for most of their misses. The ten in between lose 13.5 to 40.8 points, and Figure 1 shows the raw cell means along C: a slow slide across most of the range, with the binned means falling fastest above C of about 15.5 for GPT-5.6 Terra and Luna, Kimi K2.7 Code, DeepSeek V4-Pro and both Qwen3.8 deployments. Those binned means average across a changing mix of families and are not the family-adjusted fit.

Where the decline comes from. We did not register the next split, we chose it after seeing the result, and we present it as description. Under the registered scoring, the mean over all draws equals the share of draws whose program ran times the mean correctness among those, because every other terminal outcome scores zero; Figure 2 and Table 1 show both factors by rung. Across the panel the share of draws cut off at the output limit rises from 0.7 percent at rung 0 to 14.3 percent at rung 5, and for six models it rises far more: GPT-5.6 Terra from 0.4 to 36.8 percent, GPT-5.6 Luna from 2.6 to 38.0, Qwen3.8 27B from 2.6 to 42.8, Qwen3.8 Max from 1.4 to 34.6, DeepSeek V4-Pro from 2.4 to 29.2 and Kimi K2.7 Code from 0.8 to 20.8. Among the draws of those six models whose program ran, mean correctness at rung 5 is close to rung 0: Terra 0.990 to 1.000, Luna 0.972 to 0.991, Qwen3.8 Max 0.998 to 0.988, Qwen3.8 27B 0.980 to 0.954, DeepSeek 0.992 to 0.946, Kimi 0.978 to 0.907. That conditional mean is over a selected and shrinking set of draws, so it does not estimate what the truncated draws would have scored, and it says nothing about time. Two facts limit it further: at rung 5 Terra produced no program in any of its five draws on 34 of the 100 prompts (Luna 33, Qwen3.8 27B 34, Qwen3.8 Max 25, DeepSeek 16, Kimi 9), and for Kimi and DeepSeek the prompts with a truncated draw are also the prompts whose finished draws score lower. The other ten models are cut off at the output limit on at most 6 percent of rung-5 draws, and for MiniMax M3, Gemma 4 31B, Mistral Large 3, gpt-oss-120b and Llama-3.3-70B the programs that run are worse at rung 5 than at rung 0 by 22.5, 16.3, 14.0, 12.9 and 10.1 points; for Llama and both Mistral models, timeouts rather than the output limit account for most of the excluded draws, and every one of the 116 gpt-oss-120b answers cut off at the output limit is among the 126 draws its alternate provider served, because the registered route closed those streams before any finish. The six are grouped by the observed share of their answers cut off at the output limit, not by architecture: all six expose extended reasoning, but so do models in the other group, and MiniMax's larger output limit makes its share not comparable.

Delivery. Hosted endpoints lost long responses often enough that 9,381 draws carry a final ledger attempt index above one, none above six; Appendix I gives the counts by model and the provider-served means, and Appendix H records how the losses were handled.

5 Discussion and implications

We went looking for a place where a model stops being able to implement a specification, and we did not find one that the registered procedure could locate. What we found instead is a panel in which correctness is lower at the top of the ladder than at the bottom, in most models substantially,

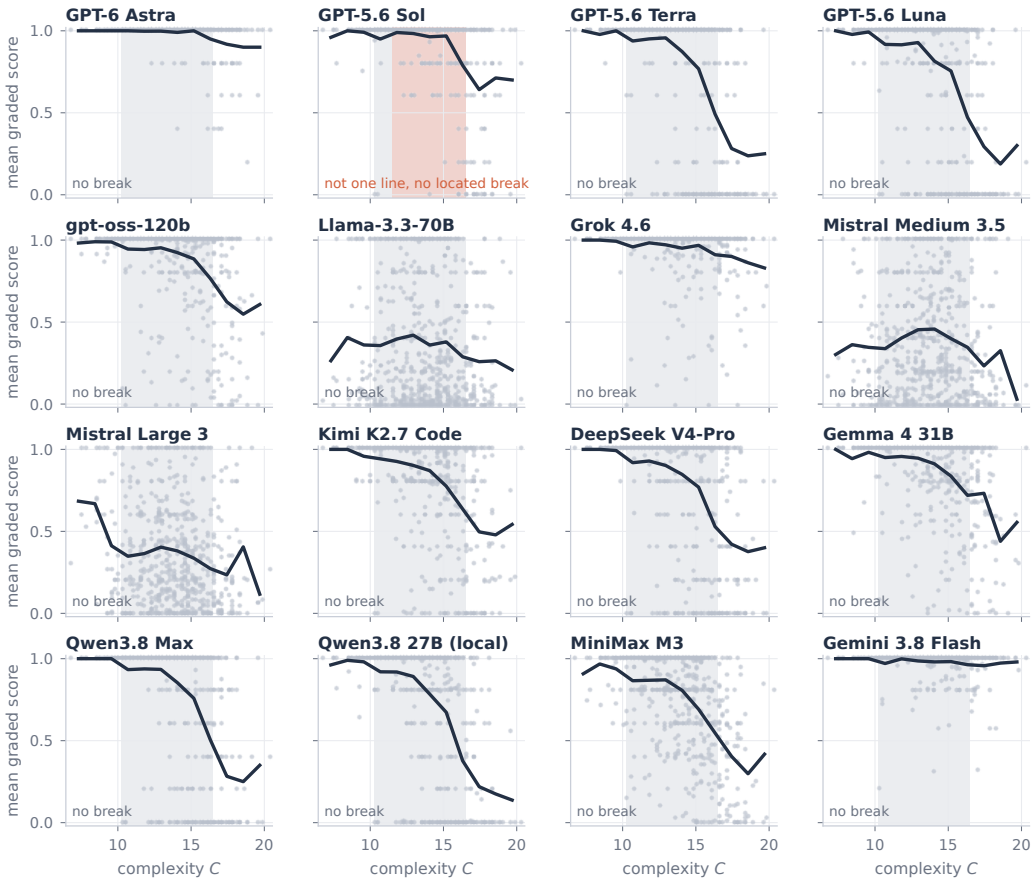

Figure 1: Mean graded correctness of each (model, prompt) cell against its complexity C (points) over the full range of the index, with the raw mean within twelve equal bins of C (line); these are not the family-adjusted fits. The gray band is the interval on which candidate splits are admitted, 10.25 to 16.50. The red band is GPT-5.6 Sol's location set, the candidate splits its test could not exclude; it covers nearly the whole admitted interval, which is why the departure is not called a break.

with no departure from a line that the procedure could place. Two readings of that are tempting, and we want to say clearly what the result does and does not support.

What the controlled setting shows. ComplexityKink found breakpoints along the same composite on a scraped corpus. Within controlled families, where the base task, the wording and the tests are held fixed and only the requirement list changes, the same instrument finds no locatable break in any of sixteen models. That is consistent with the observational breaks having been, in part, a property of what travels with complexity in the wild, and it is also consistent with breaks this design cannot see: the calibration says a genuine mid-range break of moderate size is found less than one time in ten, and the corpus, the panel and the protocol all differ between the two studies. On a controlled ladder read with a procedure that is quiet on smooth declines, nothing localized, and the earlier finding did not reappear.

Two patterns behind one number. The descriptive split in Table 1 is, to us, the most useful thing in the paper, and also the easiest to over-read. For six models, most of the registered decline arrives as draws cut off at the output limit, and that limit was set at 65,536 tokens precisely to avoid manufacturing an effect; it binds anyway on more than a third of rung-5 draws for four of them. The registered outcome scores those draws as failures and we stand by that scoring, since a program that never arrives is not a correct program. What the split adds is that, among the draws of those models

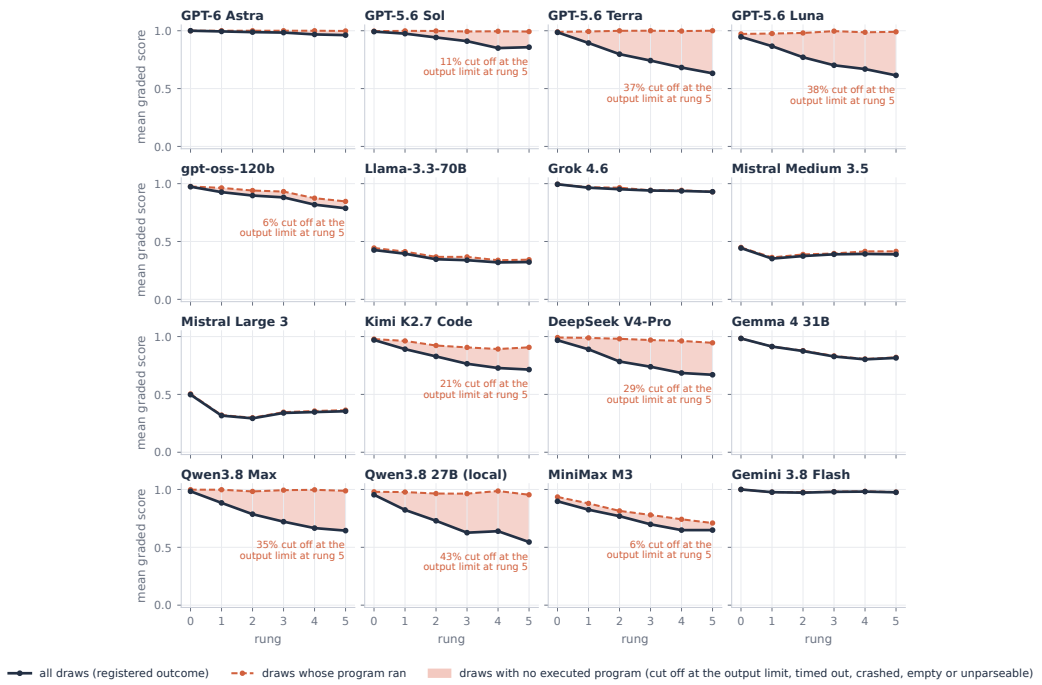

Figure 2: Registered mean correctness by rung over all draws (solid) and mean correctness conditional on the program having run (dashed). The shaded gap is the contribution of every draw excluded from that conditional mean, which is any draw cut off at the output limit, timed out, crashed, returned nothing or could not be parsed; the share of rung-5 draws cut off at the output limit is printed where it exceeds 5 percent, and Table 1 separates that cause from the others.

whose program ran, correctness changed little across the ladder. That is a statement about a selected subset, and it does not say what the truncated draws would have scored with more budget, or that the cost was time. For five other models the programs that ran are clearly worse at the top of the ladder, so the decline there is not an accounting effect of the output limit.

The top of the ladder. The binned raw means fall fastest above C of about 15.5 for several models, and that is also where answers are cut off at the output limit most often. Our grid of candidate splits ends at 16.50, because the corpus has few prompts above it, so a change that begins near the top of the range has almost no room on the right to be located. GPT-5.6 Sol's unlocatable departure, whose best split sits at 15.75 to 16.00, and the near misses of Terra and Luna are consistent with a change that starts where our support runs out. Whether that is a bend, a level shift or the beginning of a cliff is a question this ladder cannot answer; a ladder with eight or ten rungs, and an output limit high enough that it is not what stops the answer, is the study that could.

What this means for evaluation and for practice. Three things follow for anyone who benchmarks or deploys code generation. First, a correctness number on a specification-heavy task mixes failure modes that behave differently, and a benchmark that does not report answers cut off at the output limit, timeouts and wrong answers separately will describe different models with one number that means different things. Second, the output limit is part of the instrument: as long as it is reached, a decline in correctness with complexity is partly a statement about the limit, so every limit should be explicit, constant across conditions, and reported with the share of answers cut off at it; on this panel that share rose long before the correctness of the answers that did arrive changed, and whether more budget would have turned those answers into correct programs is something this study cannot say. Third, complexity is worth measuring rather than assuming. A rated index on the prompt, frozen before generation, let us ask a within-family question that a difficulty label or a rung count could not, and it let us say where on the scale our design stops being able to see.

Table 1: The decline decomposed. Share of draws cut off at the output limit at rung 0 and rung 5; share excluded from the conditional mean for any other reason (timeout, crash, empty or unparseable response); the drop in mean graded correctness from rung 0 to rung 5 over all draws (the registered outcome); and the mean correctness conditional on the program having run at rung 0 and rung 5 with its drop, in percentage points. The conditional mean is over a selected subset and does not estimate what the excluded draws would have scored.

	cut off at output limit (%)		other excluded (%)		all draws	program ran		
Model	rung 0	rung 5	rung 0	rung 5	drop	rung 0	rung 5	drop
GPT-6 Astra	0.0	3.4	0.0	0.2	3.8	1.000	0.998	0.2
GPT-5.6 Sol	0.0	10.6	0.4	3.0	13.5	0.996	0.993	0.4
GPT-5.6 Terra	0.4	36.8	0.0	0.0	35.4	0.990	1.000	-1.0
GPT-5.6 Luna	2.6	38.0	0.0	0.0	33.3	0.972	0.991	-1.8
gpt-oss-120b	0.0	5.6	0.2	1.4	18.6	0.975	0.846	12.9
Llama-3.3-70B	0.0	0.0	4.0	6.0	10.4	0.444	0.343	10.1
Grok 4.6	0.0	0.0	0.0	0.0	6.5	0.994	0.930	6.5
Mistral Medium 3.5	0.0	0.0	1.2	6.2	5.4	0.449	0.415	3.4
Mistral Large 3	0.0	0.4	1.2	2.2	14.4	0.504	0.364	14.0
Kimi K2.7 Code	0.8	20.8	0.0	0.4	25.6	0.978	0.907	7.1
DeepSeek V4-Pro	2.4	29.2	0.0	0.0	29.8	0.992	0.946	4.6
Gemma 4 31B	0.0	0.0	0.0	0.6	16.8	0.984	0.820	16.3
Qwen3.8 Max	1.4	34.6	0.0	0.2	34.0	0.998	0.988	1.0
Qwen3.8 27B (local)	2.6	42.8	0.0	0.0	40.8	0.980	0.954	2.5
MiniMax M3	0.6	6.0	3.4	2.6	24.9	0.935	0.710	22.5
Gemini 3.8 Flash	0.0	0.0	0.0	0.2	2.5	1.000	0.977	2.3

On pre-registering a result we did not want. The primary analysis ran once; everything above the descriptive split is that one run, and the split is a decision we made after seeing it, labeled as such.

6 LIMITATIONS

The axis is measured, not assigned. Rung is what we assigned, and C is what we analyze. A requirement added to a family raises C, but it also lengthens the prompt (Liu et al., 2024), narrows the set of programs that satisfy it, and sometimes changes which algorithm is needed. Our design removes the confounds a scraped corpus carries between prompts, domain, author and the base task's own wording; underspecification inside an added requirement can remain, one known case was left in place, and the confounds that arrive with a requirement itself remain too. A breakpoint here is a change in a model's correctness along judge-rated complexity within controlled families, and we have chosen that narrower reading over a causal one.

One language, one prompt style, one genre of problem. Every prompt asks for Python, every requirement arrives as one more item in a numbered list, and ninety-eight of the hundred base problems are contest problems, chosen because they come with a statement, examples and a checkable answer; we can say nothing directly about specifications written as prose, other languages, or tasks with no single correct output. The rubric that produces C measures what four judge models agree is structural complexity, and a break on C is a break on that.

Hosted endpoints, five draws, and what a non-rejection means. Fifteen of the sixteen models were reached through a provider. We checked the served identity on every synchronous row and validated every admitted score against the retained bytes, but we cannot see the weights behind a hosted name, the native batch route reports no identity, and three routes changed provider during the run for reasons the registration states. Five draws per prompt, with temperature omitted on thirteen routes and set to 1.0 on three and the local server's sampling pinned, is enough for the registered test at the calibrated rates and not enough to say much about any single prompt. Finally, the procedure is asymmetric by design: it finds a discrete change along C and refuses to call a gradual decline a break, so non-rejection is not evidence that complexity does not matter to that model.

AI USE STATEMENT

All AI assistance in this work was supervised by the authors, who specified, reviewed and tested everything it produced. We used generative AI tools to generate the dataset, to help build the software, and for feedback. The rung requirements and the reference functions that compute each family's worked examples were drafted by an authoring model (the registration records which), then built under the eight mechanical checks of Appendix C, which fail the build rather than warn, and reviewed by the authors before any family entered the corpus. The generation harness and the pipeline code were written with a coding assistant to the authors' specification and reviewed by them; they are covered by fourteen test suites and a 372-check registration checker. The harness was frozen and hashed before the confirmatory run and every draw records the digest it ran under; the draw-admission join and the reporting scripts were added during and after the run and are recorded in the registration's updates. AI tools also gave feedback on the methodology, on drafts of the text and on the reading of the results, which the authors treated as review: the hypotheses, the registered design, the analysis, which was fixed before the run and executed once without change, and the conclusions are the authors' own. We have not used generative AI tools to formulate or prove mathematical claims or for translation; qualitative or thematic analysis is not applicable to this work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The registration was filed on OSF before the first confirmatory draw and updated eighteen times with reasons and dates (link withheld for double-blind review). Every artifact the study runs on or cites is hashed into one content digest over 2,394 files; the digest in force at filing and the one in force at analysis are both quoted in Appendix H, and every draw records the digest it ran under. Every draw retains its response bytes, the program extracted, the pass or fail of every test and the serialized request, and every synchronous draw its served identity and freeze digest; the join validated every admitted row's stored scores against the score and response contracts and rescored from bytes the 169 draws recovered from journal lines or receipts. The admission artifact, the admitted draws file and the analysis output are released with the code, and the scripts in the repository regenerate every table and figure in this paper from them. The registered analysis is deterministic given its seed rule and index digest, both recorded in its output. The original responses of most judge ratings were not retained; the index is recomputable from the admitted score vectors, which are.

The software artifacts of this paper are available at `https://anonymous.4open.science/r/complexity-ladder-3108/README.md`.

REFERENCES

Donald W. K. Andrews. Tests for parameter instability and structural change with unknown change point. *Econometrica*, 61(4):821–856, 1993.

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, 2021.

Jushan Bai and Pierre Perron. Estimating and testing linear models with multiple structural changes. *Econometrica*, 66(1):47–78, 1998.

Jushan Bai and Pierre Perron. Computation and analysis of multiple structural change models. *Journal of Applied Econometrics*, 18(1):1–22, 2003.

Robert M. Bell and Daniel F. McCaffrey. Bias reduction in standard errors for linear regression with multi-stage samples. *Survey Methodology*, 28(2):169–181, 2002.

Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, Assya Trofimov, Brennan Nichyporuk, Justin Szeto, Nazanin Mohammadi Sepahvand, Edward Raff, Kanika Madan, Vikram Voleti, et al. Accounting for variance in machine learning benchmarks. In *Proceedings of Machine Learning and Systems*, volume 3, 2021.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling, 2024.

Ethan Caballero, Kshitij Gupta, Irina Rish, and David Krueger. Broken neural scaling laws. In *International Conference on Learning Representations*, 2023.

A. Colin Cameron, Jonah B. Gelbach, and Douglas L. Miller. Bootstrap-based improvements for inference with clustered errors. *The Review of Economics and Statistics*, 90(3):414–427, 2008.

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, 2021.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do NOT think that much for 2+3=? On the overthinking of long reasoning models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 9487–9499, 2025.

Robert B. Davies. Hypothesis testing when a nuisance parameter is present only under the alternative. *Biometrika*, 74(1):33–43, 1987.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, et al. Faith and fate: Limits of transformers on compositionality. In *Advances in Neural Information Processing Systems*, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z.

Maurice H. Halstead. *Elements of Software Science*. Elsevier North-Holland, 1977.

Bruce E. Hansen. Inference when a nuisance parameter is not identified under the null hypothesis. *Econometrica*, 64(2):413–430, 1996.

Bruce E. Hansen. Sample splitting and threshold estimation. *Econometrica*, 68(3):575–603, 2000.

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, and Jacob Steinhardt. Measuring coding challenge competence with APPS. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2021.

Michael Hernandez and Tian Zhao. The complexity kink: A prompt-side structural complexity index for code-generation reliability, 2026. URL `https://arxiv.org/abs/2609.19616`.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

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*, 2025.

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. FollowBench: A multi-level fine-grained constraints following benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, 2024.

A. Lawsen. Comment on The Illusion of Thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity, 2025.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. Same task, more tokens: The impact of input length on the reasoning performance of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. Competition-level code generation with AlphaCode. *Science*, 378(6624):1092–1097, 2022.

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*, 2023.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024.

James G. MacKinnon and Matthew D. Webb. Wild bootstrap inference for wildly different cluster sizes. *Journal of Applied Econometrics*, 32(2):233–254, 2017.

Thomas J. McCabe. A complexity measure. *IEEE Transactions on Software Engineering*, SE-2(4): 308–320, 1976.

Evan Miller. Adding error bars to evals: A statistical approach to language model evaluations, 2024.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. GSM-Symbolic: Understanding the limitations of mathematical reasoning in large language models. In *International Conference on Learning Representations*, 2025.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20275–20321, 2025.

Brian A. Nosek, Charles R. Ebersole, Alexander C. DeHaven, and David T. Mellor. The preregistration revolution. *Proceedings of the National Academy of Sciences*, 115(11):2600–2606, 2018.

James E. Pustejovsky and Elizabeth Tipton. Small-sample methods for cluster-robust variance estimation and hypothesis testing in fixed effects models. *Journal of Business & Economic Statistics*, 36(4):672–683, 2018.

Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. InFoBench: Evaluating instruction following ability in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

David Roodman, Morten Ørregaard Nielsen, James G. MacKinnon, and Matthew D. Webb. Fast and wild: Bootstrap inference in Stata using boottest. *The Stata Journal*, 19(1):4–60, 2019.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? In *Advances in Neural Information Processing Systems*, 2023.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity. In *Advances in Neural Information Processing Systems*, 2025.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, 2025.

Shiqi Wang, Zheng Li, Haifeng Qian, Chenghao Yang, Zijian Wang, Mingyue Shang, Varun Kumar, Samson Tan, Baishakhi Ray, Parminder Bhatia, et al. ReCode: Robustness evaluation of code generation models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022.

Bosi Wen, Pei Ke, Xiaotao Gu, Lindong Wu, Hao Huang, Jinfeng Zhou, Wenchuang Li, Binxin Hu, Wendy Gao, Jiaxin Xu, et al. Benchmarking complex instruction-following with multiple constraints composition. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2023.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 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, et al. BigCodeBench: Benchmarking code generation with diverse function calls and complex instructions. In *International Conference on Learning Representations*, 2025.

A WHAT THE PILOT WORK TAUGHT US

Nothing in this section is part of the confirmatory sample. It is here because the design in Section 2 looks the way it does for reasons that are only visible in what went wrong first.

Authored families had no headroom. Our first corpus was a set of authored branching families. Over 640 draws on a four-family pilot, 638 executed and 15 failed at least one test. One 20B model applied a Tuesday discount on every day of the week and scored 6 of 17, so failures were rare rather than absent. A corpus that close to the top of the scale cannot locate a breakdown, because there is nothing to fall from.

Many of ComplexityKink's failed prompts were broken rather than hard. We then tried to draw base problems from the ComplexityKink corpus, which is public and comes with pass rates on 5,000 prompts, and where the moderator looked free. It was not. Screening the corpus for mechanically decidable defects, a prompt that models failed was about four times more likely to be broken than a prompt they passed. Repairing forty candidates, without rewriting any prose or changing any expected value, drove their mean pass rate to 0.960, with 29 of 40 at or above 0.95, and the repaired difficulty correlated $r = +0.06$ with the difficulty that corpus had been stratified on. The clearest single case was a bank account task on which every model, every draw, passed exactly seven of ten tests. That is not what difficulty looks like. Difficulty produces a spread. A deterministic split is what a shared reading of one ambiguous sentence produces, and one added note took it from 0.70 to 1.00 on all thirty draws. Contest problems from LiveCodeBench, measured the same way over a 255-problem pool, had a mean of about 0.84 with a standard deviation of about 0.16, and just over half of them fell inside a 0.20 to 0.90 band against 10 percent for the repaired corpus. The corpus is drawn from there for that reason.

Truncation manufactures a complexity effect. Generation length rises with prompt complexity, so a binding output cap produces missing or failed draws that are graded by the very variable under study. In piloting, gpt-oss-120b at a 4,096-token cap hit it on 24 percent of draws, none of them on easy problems and over half of the hard ones. Kimi-K2.5 at a 24,000-token cap returned empty completions on hard problems, and at 32,768 the cap still bound on 6 of 56 hard draws and on none of 128 easy or medium ones. With the cap omitted, one provider silently imposed a 16,384 default. Every output limit in this study is therefore explicit, recorded per request, set as a limit rather than a target, and checked on every draw. An explicit constant limit prevents hidden allocation. It does not prevent censoring that rises with complexity, which is why a draw cut off at the output limit is scored as a failure of the model rather than dropped (Appendix F).

A requirement that cannot bind looks exactly like one that can. In one family, four successive rules each changed zero of sixteen answers while reading correctly on the page. Requirements therefore have to be chosen by measurement, not by reading, and that is what the checks in Appendix C enforce.

Which base problems can carry a ladder is itself a finding. Of the bases we tried, 18 were built and retired with their modules kept. They failed in four measured ways. The statement demands an exact match, so a rule about how the match is made attacks what the base already needs. The statement fixes the size of the structure, leaving a thin slice of the space. The answer has a narrow range or a floor, so a constraint removes a choice without changing the reported number. Or one decision absorbs rules aimed at another. Nine of the fourteen imported retirements were the narrow-range case.

Ambiguity is a rival explanation, and we tested it. A behavioral measure of ambiguity, validated on a controlled one-sentence manipulation, correlated with measured complexity at $+0.051$ raw and $+0.079$ after controlling for probe count and prompt length, against a pre-specified floor of 0.15. Failing that criterion is not an equivalence test. It does not establish what caused the overidentification rejection, and a composite correlation can hide dimension-specific effects. It did tell us to write added requirements so that they narrow acceptable behavior rather than widen it.

Hosted providers refuse benign prompts. A canary request through one provider's batch route returned an apparent classifier refusal on a benign function request. That is why a refusal has a registered treatment in Appendix F rather than being handled when it appears.

B DESIGN DETAILS

Units. The unit of manipulation is the family: one contest problem at rung 0 and five further versions, each adding one requirement. All six rungs are rendered in one layout, the contest's statement and constraints, the task restated, a numbered list of additional requirements, and worked examples. Rung 0's requirements section reads "None." rather than being omitted, so that the presence of the section is not itself a difference between rung 0 and every other rung. The unit of observation is the draw: one rung prompt sent, one response received, a program extracted from it, and that program run against the family's tests. A call is only the request going out. A request that returns nothing, a timeout, a rate limit, a provider error, is sent again, so one draw can take more than one call. The analysis counts draws. Calls are counted where cost is reported.

Complete block and order. Every model generates every rung of every family, five times. There is no selection into dose. For synchronous routes, the order in which requests go out is a seeded shuffle of the whole job list, written to disk before the first call, so rung is not aligned with position, throttling or drift at a hosted endpoint. The seed, 20260918, was the harness default and was fixed in the panel manifest before the panel was adopted, so it could not be chosen against an outcome. The one native batch route carries the same seeded order in its batch file; the provider decides the processing order inside a batch, which is a property of the route and is reported rather than controlled.

The analysis axis. The primary axis is not the rung index. It is the judge-rated composite C of Appendix E. We chose that because rung is a count of additions, and additions are not equal: one

Table 2: The eight build checks. Each runs before any rung artifact is written; a failure stops the build.

Check	Criterion
1. Semantics	The rung-0 reference reproduces the contest's own assertions. Every one of the 100 families reproduced all of its checked assertions, between 4 and 10 per family.
2. Separation	For each rung above 0, the rung's expected answers differ from the rung below's on at least $\max(3, \lfloor n/4 \rfloor)$ of the family's n shipped cases.
3. Monotonicity	Where every rung-0 answer is an integer being optimized, the optimum moves only in the direction adding a constraint permits. Not applicable in 5 families whose requirements append transformation steps.
4. Fixed signature	Every rung presents the same function signature.
5. Non-degeneracy	No rung is passable by a constant: at least three distinct answers, none covering more than half the cases, and no more than a quarter of cases with the impossible-case answer.
6. No redundant requirement	Removing any one requirement, with the other four applied, changes at least one answer.
7. Rendered prompt	The prompt a model sees contains exactly one worked-example block and never gives one input two different answers.
8. Input domain	Every test input and every worked example satisfies the constraints the prompt itself states.

requirement can add a case to check while another changes the algorithm. Within a family, variation in C comes from the requirements. Across families, C also reflects the base problem, and the family intercept absorbs that. Section 1 states what this does and does not identify.

A preparation round before the freeze. Before the freeze, every model generated three draws on the rung-0 prompt of every family, 4,800 draws. We call this R0. It served three purposes. It verified each of the sixteen request routes end to end under the exact registered settings. It measured, per model, how often the output limit or a provider-side limit is reached on the easiest rung, and how often refusals and empty responses occur. And it gave a descriptive base-difficulty figure per family. R0 draws never enter the confirmatory analysis. R0 was allowed to correct a request setting, and only a request setting, on one criterion applied to every model alike: recorded evidence that the setting was stopping a model's answer from being delivered. Nothing else could change on R0 evidence, not the corpus, the rubric, the index, the candidate grid, the support rules or any part of the analysis, and no setting could change because a model's scores were disappointing. Appendix D records the corrections that were made.

C How a ladder is built

Each family provides a set of candidate requirements written by the authoring model, one reference function that computes the answer under any subset of them, and a pool of candidate inputs. More candidates are written than the ladder needs, because which rules can bind is a property of the base problem and is not visible in the wording. Which five are used, in which order, and on which inputs was chosen by search against the criteria in Table 2 and fixed in the family module. Eight checks then run before any artifact is written, and the build fails rather than warns if any of them does.

Two of the checks exist because of specific defects we found. Check 7 exists because a defect can live entirely in the gap between the family definition and the rendered text. Check 8 exists because three families were found shipping 66 assertions on inputs their own prompts ruled out, each from a one-character bug in a case generator. Across the 100 families, the weakest rung's separation has a median of 100 percent, and in 57 families every rung changes every shipped answer. The minimum is 25 percent.

Held fixed, and recomputed. Across the rungs of a family, the function signature, the number of test cases and their inputs, and the problem statement are fixed. Only the requirement list and the

worked-example answers vary. Worked examples are recomputed from the reference at every rung, rung 0 included, so no prompt contains an example that contradicts its own requirements. Where a contest example has no valid answer at the top rung, the example input is replaced. Of the 98 imported families, 75 use exactly the examples of their contest prompt, 4 carry the contest's full example set, 6 replace one example and 13 replace both.

What the checks are and are not. They are internal consistency checks. The same reference supplies every expected answer and every separation result, so they cannot establish that the reference implements the intended specification above rung 0. Check 1 covers rung 0 only. For that reason, all 100 families had all five added requirements checked against an independently written oracle, with the evidence preserved. For 23 families the oracle was written from the rendered prompt text alone, which checks the stored answers against an independent reading of what the models see; across those 23, 4,416 stored assertions agree with no disagreement. For the other 77 the oracles come from audits and builder proof scripts, several of which implement the builder's own reading of the rules, and we say so. Every oracle was written by a language model. All checks are exhaustive on small inputs only.

Who wrote the ladders. The candidate requirements, references and case generators were written by language models working in this project, and the authoring model is recorded per family. Claude Opus 5 authored 91 families and GPT-6 Astra 9. For 47 of the 91 the attribution rests on retained build records; for the original 44 it rests on the first author's recollection, confirmed on 2026-09-21, and the two statuses are kept distinct in the corpus file rather than flattened, because a recollection and a build record are not the same kind of evidence. Of the two authoring models, GPT-6 Astra is on the panel and Claude Opus 5 is not. The Anthropic models were removed from the panel for cost, but their exclusion also removes the corpus's largest authorship conflict: a panel containing Opus 5 would be asking a model to solve ladders it wrote across at least nearly half the corpus, and nothing in this design could separate that from ability. Four further tested models share Astra's lab. That overlap is deliberate. The remaining families were directed to Astra so that a same-lab comparison would exist at all, and that comparison is registered as exploratory.

D CORPUS, PANEL AND REQUESTS

Corpus. One hundred families, 600 prompts. Ninety-eight bases come from LiveCodeBench (Jain et al., 2025), with contest dates from 2023-06-04 to 2025-03-29. Two were written for this study to fill harder difficulty bands and take no part in any date figure. Each rung carries between 10 and 20 small shipped test cases, chosen by search for how well they separate a rung from the one below, and small enough that the reference enumerates rather than approximates. Correctness in this study means correctness on these suites. They are not the contest's full suites and do not test performance at contest-scale inputs. Ninety-eight families also carry a held-out suite built from the same generators, twelve inputs in ninety-seven families and ten in family 3739, disjoint from the shipped cases, taken in generator order and never examined for what it separates; some held-out inputs also appear as worked examples. Held-out separation is weaker than shipped separation, as it should be for suites never selected for it, and six rung-5 held-out suites are passable by a constant; they are reported with those properties stated beside them.

Selection was purposive. Candidates were screened for solvability, measured, and chosen to spread difficulty across bands, and bases that could not carry a ladder were replaced for properties of the base, never for an outcome above rung 0. These 100 families are not a random sample of programming problems. On the preparation round, over its 4,631 executed draws, each family's mean graded correctness at rung 0 has mean 0.898 and standard deviation 0.065, with a minimum of 0.644, and 49 of the 100 fall inside the 0.20 to 0.90 band. Rung 0 sits near the top of the scale for most of this panel. The ladder's headroom is real, but it is headroom to fall. One known specification ambiguity, in family 3674, was left in place, because rewording it would have changed the corpus after preparation outcomes existed, and specifications outside this study are routinely less clear than that one. A sensitivity analysis excluding the family is registered.

Panel. Sixteen models, fixed before any confirmatory draw, in the frozen order of Table 3. The order breaks ties in Holm's procedure. None of them is a complexity judge. Every Azure deployment

Table 3: The panel. Slots are in the frozen order that breaks Holm ties.

Slot	Model	Route
1	GPT-6 Astra	Azure OpenAI, Responses API
2	GPT-5.6 Sol	Azure OpenAI, Responses API
3	GPT-5.6 Terra	Azure OpenAI, Responses API
4	GPT-5.6 Luna	Azure OpenAI, Responses API
5	gpt-oss-120b	Azure AI Foundry
6	Llama-3.3-70B-Instruct	Azure AI Foundry
7	Grok 4.6	Azure AI Foundry
8	Mistral Medium 3.5	Azure AI Foundry
9	Mistral Large 3	Azure AI Foundry
10	Kimi K2.7 Code	Azure AI Foundry
11	DeepSeek V4-Pro	Azure AI Foundry
12	Gemma 4 31B Instruct	OpenRouter, pinned provider
13	Qwen3.8 Max 0902	OpenRouter, pinned provider
14	Qwen3.8 27B Q4_K_M	Local llama.cpp servers, pinned artifact
15	MiniMax M3	OpenRouter, pinned provider
16	Gemini 3.8 Flash	Google native batch

is pinned to a served version with automatic upgrades disabled, and the control plane was read before the run to confirm it. For the OpenRouter routes, the provider order is a single listed provider, fallbacks are disabled and parameters are required to be honored; those fields establish the requested routing, not the physical weights or precision served, and the provider and model identity each endpoint reports are retained on every draw and checked. One slot, Qwen3.8 27B, is served by our own llama.cpp servers from a pinned quantized artifact whose weights, chat template and engine commit are verified by hash before a host serves a single draw.

One slot was replaced before filing. Slot 6 was MAI-Thinking-1. In the preparation round it delivered 2 of 122 attempts; the other 120 reached the registered request wall, and the two that arrived spent over 11,000 output tokens each on a rung-0 problem. A slot that returns nothing measures nothing, and it consumes a sixteenth of the multiplicity budget to do it. It was replaced by Llama-3.3-70B-Instruct on 2026-09-20, before filing and before any confirmatory draw, and the replacement is recorded as a panel decision rather than as one of the request-setting corrections the preparation round was allowed to make. What MAI-Thinking-1 did on this corpus is reported as an observation about the model, outside the panel.

Requests. Every model receives the same system prompt and the same rendered prompt. The system prompt asks for a complete Python implementation of the specification exactly as written, returned as code only in a single code block. Every model has an explicit output limit of 65,536 tokens, with two exceptions: Mistral Medium 3.5's deployment limits prompt and response together to 65,536, so its limit is 64,000; and MiniMax M3's limit was raised to 131,072 on preparation-round evidence that 65,536 was binding on the easiest rung, which is the one criterion that round could act on, applied to every route and selecting only that one. Where a model has a reasoning setting, it is set to its highest level; where a provider returns reasoning separately, it is retained apart from the answer, and the program is extracted from the answer alone as the longest fenced Python block. Temperature is not sent on thirteen routes, so the provider's default applies and is recorded as not sent rather than assumed; three routes send 1.0. No decoder seed is sent anywhere. Settings differ between models, and no comparison of absolute correctness between models is a primary claim. The exact serialized request of every draw is retained and checked against the request policy before it goes out, every client-library retry is disabled so that every attempt is logged, and the served model identity each endpoint reports is checked against the pinned expectation on receipt.

Corrections the preparation round forced. Seven corrections were made on R0 evidence, all before filing: three settings, three transport changes and the local server's recorded launch vector, with two further corrections to how a response is read. Three were settings. A pinned aggregator route for DeepSeek returned nothing in 150 calls and was replaced by the first-party deployment.

Mistral Medium 3.5 with reasoning effort set returned its whole answer in the reasoning field and nothing in the content, so the setting was removed, matching Mistral Large 3 on the same account. MiniMax M3's limit was raised as above. The other four were transport. The Azure endpoints close a long single response from the server side, and the loss rose with how much a model generated, which is censoring by the study's own variable: Grok 4.6 lost 71 of 262 preparation draws that way, and our local server lost 97 of 171 because a non-streamed request delivers nothing until the whole answer exists. Every first-party route and the local route were therefore switched to streaming, with the rule that a streamed answer is scored only if the server said why it stopped; a transcript that ends without a finish event is a prefix of an answer and is treated as a transport loss that can be filled, never as a short answer that failed its tests. Two more corrections concerned reading a response correctly: the local route writes its reasoning inline, and until 2026-09-21 the extractor could execute a draft the model wrote while thinking rather than the answer it gave, which changed the extracted program on 18 of 74 preparation draws and the score on five when corrected; and an aggregator's normalized "error" finish, returned inside an ordinary 200, was read as an answer until it was recognized as a provider failure partway through generation. Each correction is recorded in the registration with its evidence and its date.

E MEASURING COMPLEXITY

Each of the 600 prompts is rated on the frozen ComplexityKink rubric, on six dimensions scored as integers from 0 to 4: branching, iteration, state, data structures, edge cases and composition. The rubric is the system message and the exact prompt bytes are the user message. Four judges rate every prompt: o4-mini, GPT-5.5, Llama 4 Maverick and Cohere Command A, for 2,400 rating slots. The four are distinct models but not four independent measurements. They may share training data and conventions, and two of them come from OpenAI, the lab of a tested model, and carry half of the composite's weight. That overlap concerns the primary instrument and we state it here rather than in a footnote.

The composite. For prompt p, judge j and dimension d, with the complete four-judge panel,

$$Z_{p,d} = \tfrac{1}{4} \sum_{j} s_{p,j,d}, \qquad C_p = \sum_{d} Z_{p,d}, \tag{3}$$

so C lies in $[0, 24]$ and takes values in multiples of one quarter. A composite exists only when all four judges supply one admitted rating with all six scores valid. A missing rating is not a zero, does not authorize averaging the surviving judges, and does not remove the prompt. C is stored as an exact rational, so ties and candidate splits never depend on rounding. No rung number, generated-code measure, rank transform, family-normalized score or outcome-selected subset of dimensions is substituted for it.

Timing. Ratings are prompt-side only. Judges see the prompt and the rubric, never a model's response or a test result. All 2,400 ratings were complete and the index frozen on 2026-09-18, before the preparation round and before any confirmatory outcome existed. Two reasoning judges started under a 4,000-token completion limit and spent it before answering on a few prompts; the limit was raised to 16,384 for the slots not yet rated, and four slots that were still missing were retried once each after their cause was diagnosed, under a rule that a failed rating is retried only when the cause is transient. A valid rating was never resampled and no slot received more than one retry.

The realized index. C ranges from 6.75 to 20.25 over the 600 prompts, with a median of 13.75 and a standard deviation of 2.37, and takes 49 distinct values. Mean C by rung rises from 11.35 at rung 0 through 12.55, 13.68, 14.36 and 14.95 to 15.56 at rung 5. Within families, C rises from rung 0 to rung 5 by a median of 4.38. It is not monotone in rung within most families: 77 of the 100 have at least one rung whose C is below the rung before it, and 3 have C at rung 5 no higher than at rung 0. Those families stay in every analysis. The rung was assigned by construction, and the index is a measurement of it, so a family whose index does not rise with rung is a fact about the instrument to report, not a reason to drop the family. Between-judge agreement is modest. The mean pairwise Spearman correlation between judges is 0.51 for the six-dimension total, and between 0.30 and 0.52 on the individual dimensions.

What C is and is not. It is a realized measurement of specification complexity on the instrument ComplexityKink used, which is what keeps the two studies comparable. Freezing the scores prevents outcome-dependent measurement choices. It does not make them unbiased, and equal weighting of ordinal dimensions assumes they are comparable in scale. Every inferential statement conditions on the realized ratings. The uncertainty we report covers variation in generated responses; it does not cover judge error, alternative judge panels or repeated ratings.

F OUTCOMES AND WHAT COUNTS

The primary outcome is graded correctness, the fraction of a rung's shipped tests that a draw's program passes. All-pass, whether a draw passes every shipped test, is secondary, and the same two quantities on the held-out suite are secondary alternates for the 98 families that have one. The two behave differently. A requirement that costs a solution one test of twenty moves graded correctness slightly and all-pass to zero, so they can disagree in size and in which models look affected, and both are reported for every analysis.

A received response is final. The moment any response is received, the draw is final and is never requested again; in the final ledger no draw has more than one returned response. Under that rule, a draw whose provider reports a length finish, or whose token count reaches the output limit, is recorded as truncated and scored zero. The limit is ours and it has to sit somewhere; spending all of it without answering is the model's, and it is a different kind of failure from an absent one. A refusal, including a provider classifier refusal, is a received response and is scored zero, because this study measures deployed systems rather than weights: a user who sends a benign programming problem and receives a refusal has not been served, whatever produced it. A scan of all 600 prompts found no vocabulary of harm in any category; the only flagged words are the combat language of three game-simulation problems. An empty or unparseable response scores zero. The shipped tests run in one sandboxed process with a 30-second limit, and a run that exceeds it scores zero on that suite with the reason retained, because a specification that forces a more expensive search makes a timeout more likely and the rate is reported by rung and C.

Nothing received is missing data. A transport failure, where nothing is received, is missing data and not a failed draw. It is requested again. Which failures count as transport was fixed in advance: a provider that refused us funds or access halts the route without spending the draw's budget; a throttle, a server error or a dropped connection is retried after a pause without spending it; and our own timeout limit is the one transport outcome that does spend it, because a model thinking past the wall is a property of the model on that prompt. A stream the server cut before a finish event is transport. A response that arrives is never reclassified by this rule. Re-requesting is not guaranteed to be outcome-neutral, since latency and response length can be associated with difficulty, so the estimand is performance under this fixed delivery and retry protocol, and delivery failures are reported by model, rung and C.

Every draw is retained. For every draw the harness keeps the original response bytes, the program extracted, the pass or fail of every test in both suites, the serialized request, the route, the output limit, the provider settings, the finish reason, the model identity the endpoint reports having served, hashes of the prompt and tests it was scored against, and the digest of the frozen tree it ran under. Any draw can be rescored without asking a model again.

G ANALYSIS DETAILS AND CALIBRATION

Data entering the primary analysis. For each model separately, the five draws in each family-by-rung cell are averaged to one graded-correctness value, giving 600 equally weighted observations per model. Confirmatory analysis requires the complete planned roster: every family, rung, model and draw, each admitted only after its identity, receipt, score, program and schedule slot are verified. If the roster is incomplete, no confirmatory result is issued. We wrote that rule knowing it might bite, and it did; Appendix H says how.

Model. For model m, family f, rung k and candidate split g,

$$\bar{y}_{f,k,m} = a_{f,m} + b_m C_{f,k} + d_{m,g}\,\mathbb{I}(C_{f,k} > g) + h_{m,g}\,(C_{f,k} - g)\,\mathbb{I}(C_{f,k} > g) + \varepsilon_{f,k,m}. \quad (4)$$

The family intercept $a_{f,m}$ removes stable family differences. The low-regime slope is b_m, the high-regime slope $b_m + h_{m,g}$, and $d_{m,g}$ permits a level discontinuity. Fits are least squares with family absorption. No covariates and no family-specific slopes enter the primary model, and results are never pooled across models.

Candidate splits and support. The grid of candidate splits was built once from the admitted C values, before any generation outcome existed, and is the same for all sixteen models. The distinct values define one candidate partition per adjacent pair, so tied scores are never split, and a partition identifies an interval between observed scores rather than a point. A candidate is admitted only if it meets a support policy fixed from the outcome-blind design report: cells from at least 10 families on each side, at least 10 families whose rungs cross the split, and at least 10 effective informative families for each of the five contrasts reported below. On the realized index that admits 25 contiguous candidates, between C of 10.25 and 16.5. The cutoff at 16.5 is a property of the corpus and not a conservative choice. There is not enough corpus above it to estimate a high-side regime: 68 of the 600 prompts exceed it, and above 18 there are 15 prompts in 10 families. If a model's behavior changes above that margin, this design cannot see it, and the result is reported as no change detected within the admitted range.

Test statistic and calibration. With RSS_0 the residual sum of squares of the family-adjusted straight line and RSS_g that of each admitted split, the statistic is

$$T_m = \max_{g\ \text{admitted}} \max\big(0,\ \mathrm{RSS}_0 - \mathrm{RSS}_g\big). \quad (5)$$

It is calibrated by a restricted family-cluster wild bootstrap (Cameron et al., 2008; Roodman et al., 2019). Family intercepts are absorbed through orthonormal within-family contrasts, the null is fitted in that space, and each family's null residual vector is adjusted by the symmetric inverse square root of $I - H_{ff}$, a CR2-style adjustment (Bell & McCaffrey, 2002; Pustejovsky & Tipton, 2018). In each replicate one Rademacher sign is drawn per family, the family's whole rung residual vector is multiplied by it, the null is refitted and the complete admitted split search is repeated. C, the rung labels and the grid are never permuted, individual draws are never resampled as independent clusters, and the same family sign stream is used for all sixteen models within a replicate, which preserves the dependence between models that share families. The Monte Carlo p-value is $(1 + \#\{T^* \geq T_m\})/(B+1)$ with $B = 49{,}999$. No fixed-breakpoint F or χ^2 reference is used. Random numbers come from a seed derived from the SHA256 of the frozen rating index, so no person chose it.

Multiplicity. Holm's step-down procedure (Holm, 1979) at familywise $\alpha = 0.05$ over the sixteen p-values, so the first threshold is $1/320$. Ties are broken by the frozen panel order. The family of sixteen is never reduced after outcomes are seen; a model without an admissible analysis enters Holm with $p = 1$ and is displayed with its actual status.

Location. Rejecting the straight line says the relationship is not one line. It does not say the departure is a break, and the calibration is explicit that it cannot: a model with no break whose decline is a smooth quadratic rejects the line on 97.4 percent of occasions, more often than a genuine five-point level drop does. What separates the two is the location set. It is formed by inverting fixed-candidate profile tests over the grid: candidate g is retained when the Monte Carlo profile p-value of $\mathrm{RSS}_g - \min_q \mathrm{RSS}_q$ is at least $\alpha_{\mathrm{loc}}/16$ with $\alpha_{\mathrm{loc}} = 0.025$, and the set is the union of retained intervals with any disconnection preserved. A real break concentrates it. Smooth curvature leaves it spread across the grid, because every split fits about as badly as every other. An empty set is reported as a model-consistency failure, never replaced by the best split.

Direction and size. Simultaneous family-cluster multiplier bands are built over all sixteen models, every admitted candidate and five contrasts: the low slope, the high slope, the slope change, and the difference between the two fitted regimes at each of the two score bounds of the candidate gap. One critical value, from the maximum absolute standardized error over the whole contrast family, covers every band at $\alpha_{\mathrm{eff}} = 0.025$. Those are the intervals every reported effect carries. For deciding

Table 4: Frozen calibration of the full H1 decision. "Search rejects" is the rate at which the straight-line null is rejected; "H1 fires" additionally requires a localized, downward change.

Scenario	Planted change	Search rejects	H1 fires
A1, A2	none (two null designs)	0.053, 0.051	0.0015, 0.0040
D1, D2	smooth sigmoid, smooth quadratic	0.448, 0.974	0.0460, 0.0020
B1	level, low	0.0187	0.0007
B2	larger level, low	0.1927	0.0137
B3	slope, low	0.0250	0.0007
B4	level and slope, low	0.0117	0.0023
B5	level, middle	0.0237	0.0030
B6	larger level, middle	0.2943	0.1063
B7	slope, middle	0.0953	0.0000
B8	level and slope, middle	0.1840	0.0020
B9	level, high	0.0207	0.0010
B10	larger level, high	0.2417	0.0243
B11	slope, high	0.0253	0.0000
B12	level and slope, high	0.1640	0.0017
B13	larger level and slope, edge	0.3927	0.0083
B14	larger level and slope, narrow support	0.0543	0.0000
All fourteen breaks	pooled		0.0117

direction at the fitted breakpoint alone we use a pointwise interval from the same replicates, because a band that covers 16 models by 25 candidates by 5 contrasts at once is the wrong instrument for a question about one contrast at one candidate. A model whose standardization is degenerate, which is what a model that never varies across the range produces, has no estimable contrast and is reported as unavailable rather than given a zero-width band.

The decision. At the fitted breakpoint, the admitted candidate with the smallest residual sum of squares, a downward level change holds when the upper pointwise bounds of both gap-endpoint contrasts are below zero, and a steeper downward slope holds when the upper bounds of the slope change and the high slope are below zero. Upward labels are the mirror image. H1 is supported for a model only when its Holm-adjusted test rejects, its location set retains at most 6 of the 25 admitted candidates, at least one downward label holds, and no upward label holds. Otherwise the direction is reported as uncertain, even when the point estimate is negative, and a model with both a downward and an upward label is reported as a mixed regime change, not as a breakdown. The localization requirement is what makes the label informative rather than merely significant. It requires the study to be able to answer the question in its own title before it claims a breakdown.

Calibration. Before the freeze, the whole procedure was run on 21,500 simulated sixteen-model datasets generated on the real 600 C values with the real test counts and draw-level noise shaped like the pilot's, across scenarios that include no change, weak changes, tied and narrow support, high-leverage families, heteroskedastic and correlated rung errors, responses at the top of the scale, smooth curves, and mixed and upward changes. Every gating bar was met with no remedial round: familywise false alarm 0.055 against a bar of 0.065, effect-band coverage 0.964 and location coverage 0.984 against 0.955, and a wrong-direction rate of 0.002. Table 4 gives the full-decision rate against every planted alternative. We give the whole table because the rate varies by more than two orders of magnitude across alternatives, and quoting the maximum would misdescribe the design. Read it as the honest statement of what this design detects. A large level change in the middle of the range is found on about one occasion in ten. A change of half that size is found on almost none, wherever it sits, and a break at the edge of the support is found on under one percent of occasions even though the search rejects a straight line there more often than anywhere else, because a break at the edge cannot be localized. The claim this study can support is therefore that a large mid-range breakdown is detectable, not that breakdowns in general are. A located break is strong evidence. The absence of one is weak evidence about any single model, and is reported as such.

What the analysis refuses. Before any estimate, every row must come from a corpus family and a registered model, be scored against the frozen instrument, and be the first returned response for

its draw slot. A contrast the data do not identify is reported as not estimable, never as a number. A missing planned cell withholds the confirmatory result and exits with an error. No result is rerun with another seed, no B is increased, and nothing is extended because a result lies near a threshold.

H PRE-REGISTRATION, THE FREEZE, AND WHAT CHANGED DURING THE RUN

The order of operations was fixed and followed. The ratings were completed and the index frozen. The candidate grid, the support policy and the calibration were computed from C alone. The preparation round ran. Every artifact the study runs on or cites, the corpus and every stage that produced it, every family module including the retired ones, the harness, the rubric and the rating index, the panel manifest and request policy, and the analysis code, was hashed into one content digest over 2,394 artifacts. The registration was filed on OSF quoting the digest then in force, c0de4b164b7ec56b0bbd473788bac0c7de3ca5db054c1d50d4cab52e6309107d; the frozen tree and the filed text were committed together under an annotated tag, and only then did generation begin. The digest in force at analysis, after the updates below, is 546d7f221131ca80f5e84c315e2f2f90038b6daa946a4c3ab8a3fc938d5278a3. The harness checks the freeze before every job and every draw records the digest it ran under. Re-freezing is permitted but never silent: every superseded digest is appended to a history with the reason, and the superseded manifest is archived for all but nine early freezes, so those later freezes can be reconstructed file by file.

What changed during the run. The registration was updated eighteen times between filing and analysis, and each update states the reason and the date. None changes the design, the hypotheses, the measures or the analysis, and none touches a draw already answered. The pattern of what changed is worth recording, because nearly every change concerned the delivery of long answers, which are the answers the study is about.

Hosted endpoints lost long responses. Azure's chat and Responses endpoints close a long single response from the server side, so every first-party route was switched to streaming before filing, with the rule that a streamed answer is scored only if the server said why it stopped. Azure Foundry then closed Grok 4.6's streams at about 29 minutes whether or not the model had finished, and Grok sends nothing while it thinks, so a draw that needed longer came back as keepalive comments and nothing else. Azure also cut gpt-oss-120b's long streams. A stream of keepalive comments was at first read as unreadable bytes and treated as an identity failure, which stops a job; 22 Grok jobs stopped that way before the reading was corrected. Under those conditions a draw could get no response through no fault of the model on four requests in a row, so the limit of four requests was raised to eight, each request consumed in its own ledger and only after the one before it, and the draws that still had no response were sent through OpenRouter: gpt-oss-120b to CoreWeave at its release precision, Grok 4.6 to xAI, each with its registered settings, from a copy of the panel manifest that differs only in that row's route fields, with the serving provider named in every stored response. Gemma 4's registered provider ran out of capacity on the second day, and its cleanup requests went to a second provider serving the same weights at the same precision through the same endpoint.

The local Qwen route could not deliver 3,000 draws from one card in the time available and was served by a fleet, forty-five servers for the main pass and sixty over the run once fifteen more were added for the last draws, every one running the registered engine commit and the registered weights and verifying the weight hash before serving, behind a proxy at the registered endpoint that records which server produced every row; the execution host's own server is attested from its process table and the containers from what they reported, and the difference is stated rather than smoothed over. One of the last fifteen ran at a seventeenth of the others' speed and was dropped from the proxy with three draws open, which the job then sent again to servers that answered.

The harness's own census of which draws had been answered over-counted. A returned receipt or a journal line holding a stream of keepalive comments, a receipt whose body capture had failed, a whitespace-only reply, and a row cut short by a kill with nothing behind it had all been counted as answers, and 33 DeepSeek draws that had never been sent were held out of every cleanup pass because one line of their run was cut short. The join that builds the admission artifact reads the bytes rather than the labels; 121 draws were requested again under the corrected reading before the

roster was complete, and 67 more were admitted from a journal line or receipt that had been on disk all along. Other changes the updates record: the harness needed six repairs before the confirmatory phase could start; the within-job concurrency limit rose from 4 to 160 and the sandbox address-space limit from 8 to 64 GiB; eleven Gemini batches accepted with incorrect roster metadata were never read and were replaced by corrected batches; replies possibly lost when a worker died or a stream closed before its first event were classed as missing and requested again; and one Qwen3.8 27B job was stopped by hand after 18 of its 59 draws had answered, its remaining 41 requested again at higher concurrency, so it has no close capture. Twenty-one draws, two of gpt-oss-120b and nineteen of Grok 4.6, were served by the alternate provider at a lower ledger attempt index than the stated rule, from the fifth and sixth request respectively, because earlier attempts that had received only keepalive comments were counted as no response when the rule was applied; the artifact records each of them. The ledger's attempt index is not a count of provider requests: 6,693 of the 9,381 draws with an index above one had no recorded earlier request, their earlier attempts having been reservations by jobs stopped before sending, and one gpt-oss-120b draw needed 32 calls before an answer arrived. Seventy draws, 68 of them GPT-5.6 Sol's, scored zero because the reply could not be parsed as an answer. Reading their stored envelopes after the analysis, 65 of Sol's carried the terminal event 'response.failed', a failure on the provider's side with an empty completion in 63 of them, two were content-filter incompletes, which the registration treats as refusals scored zero, and one was recovered from a journal line. The frozen harness classed a failed terminal event as a received reply it could not parse rather than as the transport failure Section 10.3 of the registration describes, so those 65 draws were scored zero instead of being requested again; they rise with rung, and Sol is the one model whose test rejects. Because the roster rule refuses a cell with a missing draw, we bounded the effect instead of re-running the study: with those 68 rows scored as full marks and the registered procedure otherwise unchanged, a sensitivity run that is not confirmatory, Sol's search p-value is 0.00028 against the registered 0.00084, its location set holds 18 of the 25 candidates against 20, and no other model's decision changes; the departure is not an artifact of the zeros, and H1 remains unsupported for Sol at both bounds because the set does not localize. An independent audit of the join found that two of its gates, the binding of a row's freeze digest and the identity check on native rows, are weaker than the registration describes; no admitted row changes status under either, and both are disclosed here. Every one of these is stated in the registration with the digests of the documents involved.

I Additional results tables and delivery

Delivery. Hosted endpoints lost long responses often enough that 9,381 draws carry a final ledger attempt index above one, none above six (Table 8); the index counts the harness's attempts for a draw, not provider requests, and Appendix H records how the losses were handled. Where a draw was finally served through a different provider than the registered one, the provider groups differ substantially: the 126 gpt-oss-120b draws CoreWeave served after Azure had cut them score 0.052 on average against 0.917 for the rest; the 195 Grok 4.6 draws xAI served score 0.764 against 0.966; and Gemma 4's 1,766 draws served by Novita score 0.846 against 0.903 for the 1,234 Crusoe served. Provider and delivery-based selection changed together, so these means neither identify a provider effect nor rule one out; the analysis does not distinguish them, and the registered host-versus-container comparison for the local route is not reported here.

Table 5 gives the registered test for every model, Table 6 the effects at the retained candidates for the model whose test rejected, Table 7 the rung means behind Figure 2, and Table 8 how many requests each model's draws needed.

J Related work in more detail

Code-generation evaluation. Code benchmarks score a model's programs against tests on a fixed set of tasks, from hand-written function problems (Chen et al., 2021; Austin et al., 2021) through contest problems (Hendrycks et al., 2021; Li et al., 2022; Jain et al., 2025) to repository issues and library-heavy tasks (Jimenez et al., 2024; Zhuo et al., 2025). Two lines of work bear directly on how such scores are read. EvalPlus (Liu et al., 2023) showed that extending an existing benchmark's tests marks as wrong a substantial number of programs the original tests had passed, which is why correctness in this study is stated as correctness on our shipped suites. ReCode (Wang et al., 2023)

Table 5: H1 by model: the search p-value against the family-adjusted straight line, its rank and Holm-adjusted value at $\alpha = 0.05$ over the sixteen, whether the test rejects, the size of the location set out of 25 admitted candidates with its span on C, and the H1 decision.

Model	p	rank	adjusted p	rejects	|set|	location on C	H1
GPT-6 Astra	0.2707	11	1.0000	no			no
GPT-5.6 Sol	0.0008	1	0.0134	yes	20	11.50 to 16.50	no
GPT-5.6 Terra	0.0071	2	0.1062	no			no
GPT-5.6 Luna	0.0092	3	0.1291	no			no
gpt-oss-120b	0.0247	4	0.3214	no			no
Llama-3.3-70B	0.6501	14	1.0000	no			no
Grok 4.6	0.8110	16	1.0000	no			no
Mistral Medium 3.5	0.5914	13	1.0000	no			no
Mistral Large 3	0.2046	10	1.0000	no			no
Kimi K2.7 Code	0.0468	8	0.4208	no			no
DeepSeek V4-Pro	0.0486	9	0.4208	no			no
Gemma 4 31B	0.2902	12	1.0000	no			no
Qwen3.8 Max	0.0330	6	0.3630	no			no
Qwen3.8 27B (local)	0.0250	5	0.3214	no			no
MiniMax M3	0.0405	7	0.4048	no			no
Gemini 3.8 Flash	0.6992	15	1.0000	no			no

Table 6: Effects at the three lowest retained candidates for the model whose test rejected: estimate [simultaneous band] for the slope below the split, the slope above it, the change in slope (per unit of C, in graded score), and the gap between the fitted regimes at the two bounds of the candidate gap (graded score).

Model	candidate gap	low slope	high slope	slope change	gap, lower bound	gap, upper bound
GPT-5.6 Sol	11.50 to 11.75	-0.029 [-0.141, 0.082]	-0.053 [-0.096, -0.010]	-0.024 [-0.141, 0.094]	0.082 [-0.074, 0.237]	0.076 [-0.098, 0.249]
GPT-5.6 Sol	11.75 to 12.00	-0.027 [-0.124, 0.071]	-0.054 [-0.099, -0.009]	-0.028 [-0.134, 0.078]	0.074 [-0.074, 0.223]	0.067 [-0.096, 0.230]
GPT-5.6 Sol	12.00 to 12.25	-0.025 [-0.107, 0.057]	-0.057 [-0.104, -0.009]	-0.031 [-0.125, 0.062]	0.074 [-0.066, 0.214]	0.066 [-0.084, 0.216]

perturbs docstrings, names and code formatting while holding the task fixed and measures how far performance moves; the ladder also holds the task fixed, but changes what is required rather than how it is phrased.

Instruction following and added constraints. IFEval and FollowBench (Zhou et al., 2023; Jiang et al., 2024) vary the constraints an instruction carries, FollowBench adding them one level at a time; InFoBench (Qin et al., 2024) scores a response by the share of decomposed requirements it satisfies; and ComplexBench (Wen et al., 2024) studies how constraints compose, through conjunction, chaining, selection and nesting. The ladder borrows the incremental structure and applies it to programs whose correctness is checked by execution rather than by rules over text or by a judge.

Complexity, composition and collapse. Faith and Fate (Dziri et al., 2023) reports accuracy falling as the compositional depth of a task grows. GSM-Symbolic (Mirzadeh et al., 2025) instantiates grade-school mathematics problems from templates and finds accuracy dropping as clauses are added, and a single irrelevant clause costing a large share of it, in line with earlier evidence that irrelevant context distracts language models (Shi et al., 2023) and that longer inputs alone degrade reasoning (Levy et al., 2024). The closest analog to our question is Shojaee et al. (2025), who raise the complexity of controllable puzzles and report that reasoning models collapse beyond a threshold; Lawsen (2025) replies that part of that collapse reflects output-token limits and unsolvable instances rather than failed reasoning. Our descriptive split meets the same distinction in code generation, with limits that were set explicitly, held constant across rungs and recorded on every draw. It does not settle that debate, because correctness conditional on a program running is computed over a selected set of draws (Section 4). Whether declines of this kind are smooth or abrupt depends on the functional form fitted (Caballero et al., 2023) and on the metric (Wei et al., 2022; Schaeffer et al., 2023), which is one reason the registered analysis uses graded rather than all-pass correctness.

Table 7: Mean graded correctness by rung (cell means averaged over the 100 families) and the drop from rung 0 to rung 5 in percentage points.

Model	rung 0	1	2	3	4	5	drop
GPT-6 Astra	1.000	0.994	0.988	0.984	0.968	0.962	3.8
GPT-5.6 Sol	0.992	0.975	0.942	0.910	0.850	0.858	13.5
GPT-5.6 Terra	0.986	0.894	0.798	0.742	0.682	0.632	35.4
GPT-5.6 Luna	0.947	0.866	0.771	0.702	0.669	0.614	33.3
gpt-oss-120b	0.974	0.927	0.897	0.881	0.818	0.787	18.6
Llama-3.3-70B	0.426	0.394	0.347	0.338	0.319	0.322	10.4
Grok 4.6	0.994	0.965	0.952	0.940	0.937	0.930	6.5
Mistral Medium 3.5	0.443	0.352	0.374	0.389	0.392	0.389	5.4
Mistral Large 3	0.498	0.317	0.293	0.340	0.347	0.354	14.4
Kimi K2.7 Code	0.970	0.891	0.828	0.765	0.728	0.715	25.6
DeepSeek V4-Pro	0.968	0.890	0.784	0.739	0.685	0.670	29.8
Gemma 4 31B	0.984	0.913	0.875	0.828	0.803	0.815	16.8
Qwen3.8 Max	0.984	0.884	0.786	0.722	0.666	0.644	34.0
Qwen3.8 27B (local)	0.954	0.823	0.729	0.626	0.639	0.546	40.8
MiniMax M3	0.898	0.825	0.769	0.699	0.649	0.649	24.9
Gemini 3.8 Flash	1.000	0.976	0.972	0.979	0.982	0.975	2.5

Table 8: Admitted draws by model: how many were answered on the first request, how many needed more than one, and the most any draw needed.

Model	admitted	first request	more than one	most
GPT-6 Astra	3000	2885	115	2
GPT-5.6 Sol	3000	2912	88	2
GPT-5.6 Terra	3000	2883	117	2
GPT-5.6 Luna	3000	2992	8	2
gpt-oss-120b	3000	2644	356	6
Llama-3.3-70B	3000	3000	0	1
Grok 4.6	3000	1577	1423	6
Mistral Medium 3.5	3000	3000	0	1
Mistral Large 3	3000	3000	0	1
Kimi K2.7 Code	3000	2896	104	2
DeepSeek V4-Pro	3000	2881	119	2
Gemma 4 31B	3000	1234	1766	4
Qwen3.8 Max	3000	2459	541	2
Qwen3.8 27B (local)	3000	2301	699	4
MiniMax M3	3000	1955	1045	5
Gemini 3.8 Flash	3000	0	3000	3

Measuring complexity. Software engineering has long measured the complexity of programs, through control-flow counts (McCabe, 1976) and operator and operand counts (Halstead, 1977). The ComplexityKink rubric (Hernandez & Zhao, 2026) applies a similar decomposition, branching, iteration, state and the rest, to the specification before any code exists, and rates it with judge models (Zheng et al., 2023). The modest agreement between judges (Appendix E) is one reason we treat C as a measurement with error rather than as ground truth.

Reasoning length and the output budget. Reasoning models trained to produce long chains of thought (Guo et al., 2025) and methods that scale test-time compute (Snell et al., 2025; Muennighoff et al., 2025) make the number of generated tokens a quantity that responds to the problem. Such models can spend many tokens even on trivial problems (Chen et al., 2025), and repeated sampling keeps raising the share of problems solved as the number of samples grows (Brown et al., 2024). The first bears on why a fixed output limit can bind more often as specifications grow; the second on why five draws per prompt estimate a cell mean rather than what a model could eventually produce.

Change points and inference. The sup-type statistic over an unknown split follows the structural-change literature (Andrews, 1993; Bai & Perron, 1998; 2003; Hansen, 2000). Because the split is defined only under the alternative, the usual fixed-breakpoint reference distributions do not apply (Davies, 1987; Hansen, 1996), and the procedure is calibrated by bootstrap instead. Inference is clustered on families with the wild cluster bootstrap (Cameron et al., 2008; MacKinnon & Webb, 2017; Roodman et al., 2019) and a CR2-type small-sample adjustment (Bell & McCaffrey, 2002; Pustejovsky & Tipton, 2018). In machine-learning evaluation, the case for reporting uncertainty that respects how items are grouped, and for accounting for run-to-run variance, has been made by Miller (2024) and Bouthillier et al. (2021); the pre-registration follows Nosek et al. (2018).