METROBENCH: EXECUTION-GROUNDED MEASUREMENT FOR VISUAL CODE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual code generation couples interpretation of a reference artifact with program synthesis, yet artifact fidelity and calibrated capability are distinct measurement problems. We introduce MetroBench, an execution-grounded framework and diagnostic study toward their joint evaluation. The framework specifies Construct, Reflect, and Repair protocols, and separates a four-component chart scorer from an independently audited psychometric layer. On 200 procedural chart items and 1,000 program variants, mean fidelity declines from 1.000 to 0.658 across five perturbation levels; 197 item profiles decrease strictly, while three exhibit local reversals. Removing plotted-data agreement reduces the strictly ordered profiles to 145, exposing sensitivity to the scoring rubric. In a separate synthetic-response experiment with 20 repetitions, 40 Fisher-selected items yield held-out ability RMSE 0.104 versus 0.110 for random selection and 0.090 for the 200-item full test; the paired 40-item gain is uncertain. A matched-bank study shows that anchor linkage can recover scale despite nearly unchanged rank correlation. However, nominal 95% plug-in ability intervals cover only 45.8% of held-out abilities, compared with 95.8% when item parameters are known. These findings demonstrate why execution grounding, adaptive efficiency, and uncertainty calibration require separate evidence. The present studies evaluate procedural scoring and synthetic measurement, not model capabilities; a validated common scale across multimodal perception and code synthesis remains an open empirical requirement.

1 INTRODUCTION

Visual code generation asks a system to interpret an artifact and produce a program that realizes it. Chart reconstruction, screenshot-to-code generation, and interactive interface synthesis instantiate this pattern with different outputs and verification requirements (Yang et al., 2025; Tang et al., 2026; Si et al., 2025; Xiao et al., 2025). An executable output is necessary but insufficient: it may encode the wrong data, omit a meaningful annotation, or reproduce a static appearance while violating the intended behavior. Conversely, programs that differ substantially as text can render equivalent artifacts. These observations motivate evaluation grounded in execution rather than source-string agreement alone.

Execution grounding does not, by itself, solve measurement. A visual similarity of 0.8 on one task need not carry the same meaning as 0.8 on another, and a weighted composite can rank plausible errors inconsistently. Psychometric methods address a different layer: estimating item characteristics, adapting item selection, and linking measurements when an item bank changes (Liao et al., 2025; Habba et al., 2026; Balkır et al., 2026; Uebayashi et al., 2026). Connecting these layers requires evidence that a scoring rule captures relevant errors and that its observed responses support the chosen measurement model. Neither a continuous score nor a well-fitting synthetic example establishes these properties automatically.

MetroBench treats this connection as an evaluation design to be audited rather than an accomplished common scale. Figure 1 illustrates a reference artifact, candidate code and execution-based scoring. Construct asks for a program, Reflect asks about the visible effect of a code change, and Repair asks for a patch restoring a target. The current implementation executes trusted procedural chart variants and measures their rendered and plotted properties. An independent numerical study evaluates item

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.

calibration, adaptive testing, and version linkage under a specified synthetic response model. It does not fit that model to chart scores or report language-model performance.

The central empirical lesson is that apparently successful measurements can hide important failure modes. A high overall correlation between score and perturbation level coexists with local score reversals and sensitivity to component weights. Adaptive selection must be compared with random and full testing under the same response-generating process; otherwise, a difference in experimental conditions can be mistaken for efficiency. Finally, high rank correlation and low point-estimation error do not ensure that a scale or its uncertainty intervals are calibrated. These distinctions are consequential when benchmark scores support thresholds, model comparisons or deployment decisions. Sections 4 and 5 demonstrate each failure mode on stored numerical data rather than by argument alone.

Our contributions are threefold. First, we provide an inspectable execution-grounded scoring implementation and item-paired diagnostics for 200 charts with 1,000 variants. Second, we provide a matched-response audit of continuous measurement with held-out respondents, identical full-test endpoints, anchor linkage, and explicit interval-coverage checks. Third, we specify the requirements and remaining evidence for extending these components into a multimodal–coding benchmark, without presenting procedural variants as model outputs or synthetic respondents as evaluated systems. The accompanying appendices describe the scoring conventions, perturbation operators, numerical estimation, full repetition-level results and reproduction procedure.

2 Related Work

2.1 Visual understanding and executable synthesis

ChartMimic and Chart2Code evaluate chart-to-code generation (Yang et al., 2025; Tang et al., 2026), while ChartQA emphasizes understanding through visual and logical questions (Masry et al., 2022). OpusAnimation, ChartEditBench, DEEPCHART, IGenBench and ChartCoder explore dynamic generation, editing, chart fidelity and specialized synthesis (Li et al., 2025a; Kapadnis et al., 2026; tang et al., 2026; Tang et al., 2026; Zhao et al., 2025). These settings motivate testing several observable aspects of a generated artifact rather than execution success alone. MetroBench does not claim that render-based evaluation or visual repair is new; it investigates the behavior and limitations of a small explicit scorer before making stronger measurement claims. The separation has a practical consequence: a scorer can be replaced or extended without redefining the measurement layer, and measurement procedures can be audited without regenerating artifacts. MetroBench operationalizes this by storing component-level observations beside each composite, so later analyses can recombine channels without re-executing programs.

Screenshot and interface synthesis introduce additional challenges involving layout, interaction and code organization. Design2Code, Sketch2Code, Interaction2Code and Divide-and-Conquer study related transformations (Si et al., 2025; Li et al., 2025b; Xiao et al., 2025; Wan et al., 2025). Widget2Code, WebUIBench, WebCode2M and visual-feedback-based GUI work further motivate executable evaluation beyond a single image (Zhang et al., 2026a; Lin et al., 2025; Gui et al., 2025; Liu et al., 2026b). Our five-family chart pilot does not establish performance on these domains, and it should not be treated as a proxy for interactive correctness.

Unified multimodal evaluation and modeling address broader combinations of understanding and generation (Zou et al., 2026; Li et al., 2026d; Jiang et al., 2025; Li et al., 2026b; Fu et al., 2025; Yin et al., 2024). Task breadth is distinct from score comparability: collecting different tasks does not by itself identify a shared latent construct. MetroBench's proposed Construct/Reflect/Repair protocol is intended to expose this question on matched content. Its cross-task identifiability and invariance must still be established with actual model responses.

2.2 Psychometric evaluation and adaptive measurement

Item response theory and related approaches have been applied to efficient evaluation, bank extension and joint model–item analysis (Uebayashi et al., 2026; Liao et al., 2025; Habba et al., 2026; Li et al., 2026a;c; Zhou et al., 2026). Continuous-score adaptive evaluation is already studied by Balkır et al. (2026), and computerized adaptive testing also appears in medical language-model bench-

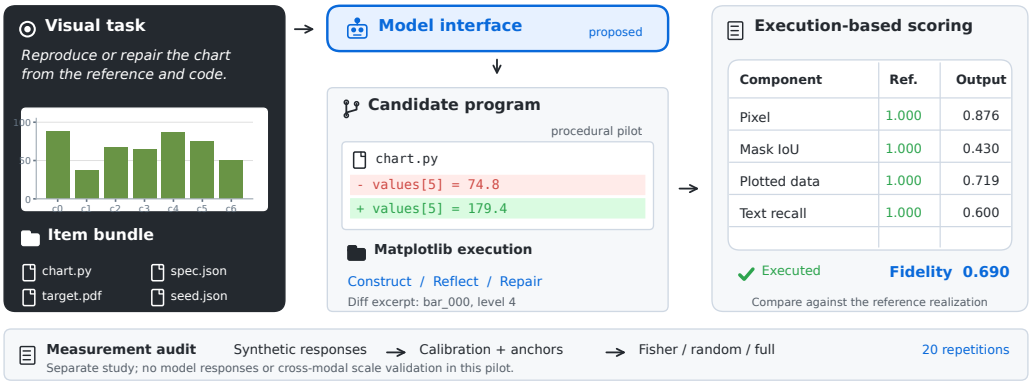

Figure 1: Task-oriented MetroBench overview, using the compact input–program–evaluation organization of the SWE-bench overview. The original vector icons identify the input bundle, model interface, code change and evaluator. The shown change is an excerpt from bar_000 at level 4, not the entire perturbation; the vector thumbnail displays reference data. The model interface is proposed, while actual scoring runs use procedural variants. The lower strip is an independent synthetic study, not a response model fitted to these chart scores.

marking (Zheng et al., 2026). Other work examines option-level responses, latency, uncertainty, contextual multidimensional structure, cognitive diagnosis and safety (Fei et al., 2026; Xu et al., 2026; Mandujano Reyes, 2026; Shang et al., 2026; Zhang et al., 2026b; Liu et al., 2026a; Rivera et al., 2026). We therefore claim neither a new IRT family nor a new adaptive-selection principle. Our numerical study audits implementation choices and comparison design in a transparent continuous model, including failures that would be obscured by reporting only a favorable point estimate.

3 Framework and Measurement Design

3.1 Observable artifacts and task scope

An item consists of a seeded specification, reference program and rendered artifact. The implemented bank contains bar, line, scatter, pie and area charts, with 40 instances per family. The input bundle in Figure 1 illustrates the intended task interface, while the code excerpt and evaluation values come from a concrete procedural variant. In this pilot, candidate programs are trusted template mutations, not responses from an evaluated model. The executor is a local Matplotlib process, not a security sandbox.

Construct presents the reference and requests a plotting program. Repair presents defective code and a target, then requests a corrective patch. Reflect instead requests a statement about a code mutation's visible effect; its output is not necessarily executable and requires a separate validator. Only the artifact-scoring path is implemented here. A future cross-protocol scale would require matched item content, protocol-specific scoring and empirical tests of measurement invariance, rather than assuming that one scorer applies unchanged to all three output types.

3.2 Four-component realization fidelity

The executor saves a 600-by-400-pixel reference or candidate chart. For image comparison, it converts the PNG to grayscale and resizes it to 128 by 96 pixels. Pixel agreement is one minus mean absolute grayscale difference; mask agreement is intersection over union of pixels darker than 0.98. Plotted-data agreement uses values recovered from Matplotlib artists and reference specification data, while text recall measures the fraction of reference text strings found in the candidate figure. Artist introspection is an extra observation channel, not information recoverable from pixels alone.

Writing these components as P, I, D, T, the fixed composite is

$$F = 0.35P + 0.25I + 0.30D + 0.10T. \tag{1}$$

The weights are engineering choices that were fixed for the original scoring run, not optimized against human judgments. Appendix A gives the exact normalization, unequal-length behavior and text conventions. Pixel agreement is sensitive to background area, mask IoU is not semantic structure, data matching is representation-dependent, and text recall does not penalize additional unsupported labels. The composite therefore needs empirical auditing even when every component is deterministic.

The plotted-data channel also operates at a small scale by construction. Reference data vectors contain 4–8 category values for bar charts, 5–12 points for line and area charts, 3–6 slices for pie charts and 20–60 points for scatter charts. The channel therefore compares tens of numbers at most, not dense pixel statistics, and a mutation that changes series length falls back to the explicit unequal-length penalty rather than value comparison.

3.3 Continuous-response numerical audit

The measurement study is deliberately independent of the chart-score matrix. For respondent ability θ_i and item parameters (a_j, b_j), it draws

$$y_{ij} \mid \theta_i \sim \mathcal{N}(b_j + a_j\theta_i, \sigma_{ij}^2), \qquad \sigma_{ij} = 0.5 + 0.15|\theta_i - b_j|. \tag{2}$$

Here $a_j > 0$ is a slope and b_j serves as both a mean intercept and a variance-location parameter. It is not the usual logistic-IRT difficulty. Because the normal distribution has unbounded support, this model is not automatically valid for bounded fidelity $F \in [0, 1]$. The synthetic study tests estimation and administration under specified assumptions, not a fitted multimodal capability scale.

Calibration maximizes a per-item marginal likelihood under a fixed normal ability prior using 15-point Gauss–Hermite quadrature. This factorized objective does not exploit the full cross-item joint response likelihood. Expected-a-posteriori (EAP) estimation then integrates the conditional item likelihoods over a 241-point ability grid on $[-6, 6]$. Greedy Fisher selection uses the information at the current ability estimate, excludes previously asked items and updates EAP after each response. The likelihood, information formula, parameter bounds and numerical conventions are derived in Appendix C.

3.4 Matched comparison and scale linkage

Before administration, each repetition generates one response matrix for its held-out respondents using the true item parameters. Fisher and random selection read entries from that same matrix and use identical estimated parameters for scoring. Full testing consumes every entry of each row. This construction isolates selection from response generation and provides a useful endpoint invariant: every 200-item administration must produce the same EAP result as the full test, up to floating-point summation error.

The linkage experiment likewise uses the same version-2 bank and responses for its two arms. It retains 80 items, introduces 120 new items and calibrates on a restricted population of 50 respondents. The linked arm holds estimated version-1 anchor parameters fixed and adjusts the new-item calibration prior using anchor-only EAP estimates; the free arm refits all items under a standard normal prior. Both arms evaluate the same held-out respondents with the same scoring prior. The contrast measures this combined linkage procedure, not the isolated causal effect of anchor retention apart from prior adjustment.

4 Artifact-Scoring Experiments

4.1 Design and statistical unit

The artifact study contains 200 items and five variants per item, including the unchanged reference at level 0. Nominal levels progress from cosmetic changes through text/grid changes to larger data and structural changes. The operator schedule is cumulative, but random mutation realizations may differ between levels because the level enters the seed. Level is therefore an intended damage ordering to test, not a guarantee of monotonically increasing perceptual error. Exact schedules and randomization are documented in Appendix B.

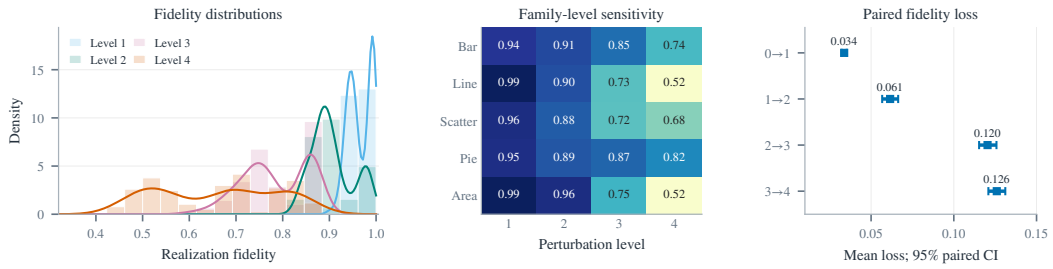

Figure 2: Compact artifact-scoring diagnostics. Left: score histograms and kernel-density estimates at levels 1–4 (200 items per level); level 0 is an exact point mass at 1. Center: family mean scores (40 items each). Right: adjacent-level mean losses with 95% paired bootstrap intervals, stratified by family. Density curves visualize existing observations; they are not intermediate severity measurements.

Each item contributes repeated observations, so it is the statistical block. We report pooled Spearman correlation descriptively, use a Friedman test for within-item ordering, and estimate adjacent-level losses with an item-paired, family-stratified bootstrap. Each of 5,000 resamples retains all five levels of a sampled item. The reported confidence intervals concern mean score changes under this procedural population; they are not uncertainty intervals for an individual model's capability.

4.2 Ordering, heterogeneity and reversals

Figure 2 shows full score distributions, family means and paired losses. Overall mean fidelity is 1.000, 0.966, 0.905, 0.784 and 0.658 at levels 0–4, with corresponding standard deviations 0.000, 0.025, 0.043, 0.070 and 0.128. The pooled fidelity–level Spearman correlation is -0.944. The repeated-measures test gives $\chi^2(4) = 797.636$, $p = 2.497 \times 10^{-171}$, and Kendall's $W = 0.997$ over 200 item blocks. Weak monotonicity, which permits ties, holds for the same 197 of 200 profiles as strict monotonicity; the three exceptions therefore contain genuine adjacent increases rather than ties. This is evidence of strong ordering under the chosen operator distribution, not human-validated severity.

Strict monotonicity holds for 197 of 200 items. The three exceptions are pie_001, pie_026 and scatter_003, whose fidelity increases from level 3 to 4 by 0.0132, 0.0003 and 0.0679, respectively. These are reversals, not ties. Appendix D decomposes their component changes, and Appendix F shows representative rendered artifacts. The exceptions show that an aggregate trend can conceal item-level reversals relative to the nominal perturbation ordering.

Family means differ substantially: at level 4, pie charts average 0.824 and area charts 0.521. The difference combines chart content, operator choices and component sensitivity. It cannot be identified as intrinsic family difficulty or as a causal property of any one scorer component. Likewise, a high correlation between two components does not demonstrate redundancy; the relevant question is whether removing a channel changes the behavior needed for the task.

4.3 Component overlap and channel behavior

The stored component matrix permits direct inspection of the four channels. Across the 1,000 scored rows, pairwise Spearman correlations between components are 0.713 between pixel and mask agreement, 0.754 between mask agreement and plotted data, 0.487 between pixel and plotted data, 0.641 between plotted data and text recall, and 0.430 and 0.480 between text recall and the two image-derived channels. The largest value, 0.754 between mask agreement and plotted data, indicates shared sensitivity to gross corruptions. Text recall is the least correlated channel, which is expected because it measures annotation strings rather than rendered geometry. This diagnostic complements the ablations in Table 1: correlation describes co-movement on these designed errors, whereas removal shows what the ordering loses without a channel. Neither statistic establishes that any channel measures perceptual quality as a human would judge it. The asymmetry also cautions against summarizing fidelity in a single number: channels that co-move on gross errors can still disagree about

Table 1: Score-weight sensitivity on the same 200 item profiles. Strict decrease counts profiles with all four adjacent differences negative. Rank vs. original is the Spearman agreement of the reweighted composite with the original composite over all 1,000 stored rows. No weighting is selected as a new optimum.

Weights	Severity ρ	Strict decrease	Rank vs. original	Level-4 mean
Original	-0.944	197/200	1.000	0.658
Equal	-0.936	196/200	0.965	0.620
Without pixel	-0.941	197/200	0.998	0.496
Without mask	-0.937	193/200	0.935	0.708
Without data	-0.902	145/200	0.982	0.740
Without text	-0.891	198/200	0.966	0.671

which specific defect occurred, and a downstream repair decision may depend on exactly that disagreement.

4.4 Sensitivity to the scoring rubric

Table 1 evaluates equal weighting and leave-one-component-out reweighting, with remaining weights renormalized to sum to one. These are post-hoc sensitivity analyses on saved component scores, not additional renders or a search for a better metric. Four-decimal component storage introduces at most 1.1×10^{-4} discrepancy from the stored composite, which we check before recombination. The analysis uses identical variants for every weighting rule.

Removing plotted-data agreement reduces strict monotonicity from 197 to 145 profiles, while equal weights retain 196. Removing text increases the strict-profile count to 198 but weakens pooled severity correlation to -0.891. The rank column of Table 1 adds a complementary view: it reports the Spearman agreement between each reweighted composite and the original composite over all 1,000 stored rows. Dropping the pixel channel barely disturbs item-level ranking (0.998), while dropping the mask channel perturbs it most (0.935), even though its severity correlation stays high. Absolute score levels are more rubric-dependent than ordering: the level-4 mean ranges from 0.496 to 0.740 across reweightings even when ordering statistics barely move. These criteria can therefore favor different rubrics, making it unsafe to tune weights against only one statistic. The original weights remain a transparent baseline; perceptual validation and task-specific utility judgments are still required before treating the composite as a calibrated quality measure.

5 Synthetic Measurement Experiments

5.1 Calibration and held-out administration

Each of 20 independent repetitions draws 150 calibration respondents and 60 held-out respondents from a standard normal population. The 200-item bank is fixed across repetitions: $a_j \sim \mathcal{U}[0.5, 2.5]$, and b_j is drawn from a standard normal and clipped to $[-3, 3]$. Repetition uncertainty therefore conditions on this bank; it does not quantify variation across all possible item banks. Random seeds, priors and parameter constraints are specified in Appendices C and E.

Discrimination and intercept recovery have RMSE 0.119 ± 0.029 and 0.199 ± 0.065, respectively. These values summarize error across 200 items within each repetition and then mean and standard deviation across repetitions. Figure 3 displays recovery-error distributions and held-out testing results, without interpolating performance between budgets. Every 200-item Fisher and random sequence is checked against the full-test EAP estimate to absolute tolerance 10^{-10}.

At 40 items, Fisher selection has RMSE 0.104 ± 0.049 versus 0.110 ± 0.041 for random selection and 0.090 ± 0.049 for the full test (Table 2). At the smallest budget of five items both policies remain above 0.19 RMSE, so no selection rule in this study recovers full-test accuracy from a fortieth of the bank. The mean paired 40-item gain is 0.0057, with 95% interval $[-0.0027, 0.0140]$. The interval is compatible with both a small loss and a modest benefit. Consequently, the present experiment does

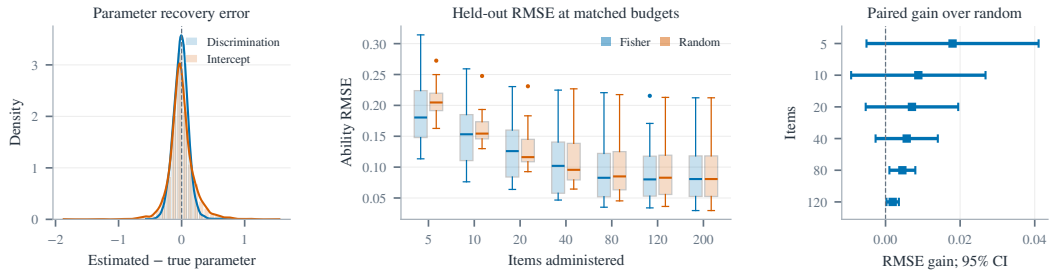

Figure 3: Matched-response measurement audit. Left: distributions of parameter errors across 200 fixed items and 20 repetitions. Center: repetition-level ability RMSE at each budget; boxes show quartiles, horizontal lines medians, whiskers 1.5 IQR and points outliers. Right: paired Fisher-over-random mean gains with 95% Student-t intervals across the 20 repetitions. The identical 200-item endpoint has zero gain and is omitted from the interval panel.

Table 2: Held-out ability RMSE, mean $\pm$ s.d. over 20 repetitions with 60 held-out respondents each. Both policies use the same response matrix in every repetition. The 200-item endpoint equals full testing.

Items	Fisher selection	Random selection
5	0.191 ± 0.055	0.209 ± 0.025
10	0.153 ± 0.049	0.162 ± 0.026
20	0.124 ± 0.045	0.131 ± 0.034
40	0.104 ± 0.049	0.110 ± 0.041
80	0.094 ± 0.050	0.098 ± 0.047
120	0.091 ± 0.049	0.093 ± 0.049
200	0.090 ± 0.049	0.090 ± 0.049

not establish a stable advantage at that budget, superiority to the full test, or a no-loss reduction in execution cost.

Appendix D reports all six nontrivial budget contrasts, their paired statistics and Holm-adjusted p-values. The 20 repetitions, not the 1,200 individual held-out respondents pooled across them, determine the inferential sample size for these contrasts. The full-test endpoint also prevents a misleading interpretation in which different policies are evaluated on responses generated from different parameter sets. Selection quality and response-generation difficulty remain separate factors.

5.2 Ability strata and full-test bias

Aggregates can hide where estimation errs. Pooling the 20 repetitions by true ability, held-out respondents with $\theta < -1$ have full-test RMSE 0.112 and mean error 0.0045, respondents with $|\theta| \leq 1$ have RMSE 0.095 and mean error 0.0001, and respondents with $\theta > 1$ have RMSE 0.119 and mean error 0.0090 (Appendix D). Central abilities are estimated nearly without bias, while both extremes carry small positive mean errors and wider spread. The pattern is consistent with the bank's discrimination and intercept ranges, but it is a descriptive summary: pooled respondents are not independent replications of calibration, and stratum differences confound ability with item coverage. Per-repetition full-test bias spans -0.211 to $+0.129$ (Table 11), so single-repetition bias estimates fluctuate far more than the pooled means suggest. We report these strata to characterize the error structure of the audited estimator, not to claim uniform measurement quality across the ability scale.

5.3 Linkage: scale recovery versus rank preservation

The version-2 study retains 80 items and replaces the other 120, calibrating with 50 new respondents drawn uniformly from $[-1, 1]$. The free and linked arms receive the same version-2 calibration and test responses. The linked arm fixes estimated version-1 anchor parameters and estimates the

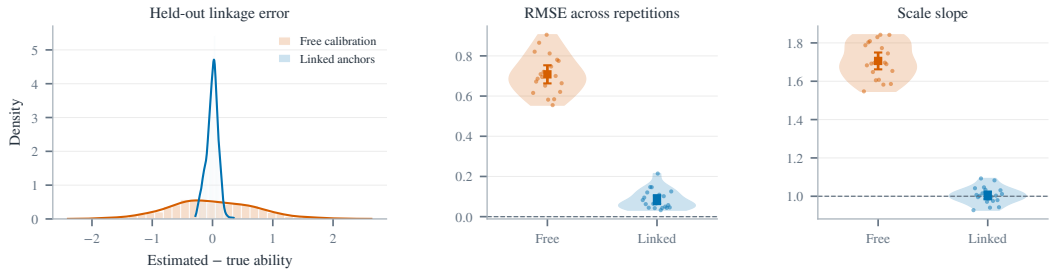

Figure 4: Matched-bank linkage diagnostics. Left: held-out ability-error distributions, pooled descriptively across 20 repetitions. Center and right: repetition-level RMSE and scale-slope distributions, individual repetition points, and mean 95% intervals. The reference lines mark zero RMSE and unit slope. No line connects two arm means as if they formed a trend.

Table 3: Scale and rank metrics for the same version-2 bank. Mean ± s.d. across 20 repetitions. The rank reference is true held-out ability, not version-1 estimates.

Metric	Free calibration	Linked anchors
Held-out RMSE	0.708 ± 0.096	0.085 ± 0.047
Scale slope	1.706 ± 0.094	1.004 ± 0.043
Spearman ρ vs. truth	0.9989 ± 0.0004	0.9989 ± 0.0004

restricted calibration prior from anchor-only EAP values; the free arm calibrates all items under a standard normal prior. Both use a standard normal scoring prior for the independent held-out respondents.

The free arm has scale slope 1.706 ± 0.094 and RMSE 0.708 ± 0.096; linkage yields slope 1.004 ± 0.043 and RMSE 0.085 ± 0.047. Both arms nevertheless have mean Spearman correlation above 0.998 (Table 3). This is consistent with the invariance of ranks to positive affine transformations: ordering alone cannot diagnose scale distortion. The result supports the combined procedure under this synthetic shift, not automatic cross-task invariance or an isolated causal claim about anchors independent of the prior adjustment.

5.4 Uncertainty audit and numerical sensitivity

Point-estimation accuracy does not establish interval validity. We construct equal-tailed nominal 95% ability intervals from the conditional posterior on a refined 0.005-spaced grid. The plug-in intervals hold estimated item parameters fixed, omitting their uncertainty. They cover only 0.458 ± 0.304 of held-out true abilities per repetition, whereas the known-parameter control covers 0.958 ± 0.027 (Appendix D.4). This substantial undercoverage rules out interpreting the plug-in posterior as calibrated benchmark confidence.

Halving the production EAP grid spacing from 0.05 to 0.025 changes held-out full-test posterior means by at most 0.00169. This measures sensitivity to EAP grid refinement conditional on the fitted item parameters. Appendix C.6 adds analytical reference checks and a calibration-quadrature diagnostic. The observed undercoverage characterizes the implemented estimator; the contributions of item-parameter uncertainty, estimation bias and calibration-quadrature error are not separated by these controls. Confidence-based adaptive stopping remains future work.

6 Limitations and Implications

The strongest limitation is the absence of evaluated model responses. Procedural mutations probe the sensitivity of a scorer to designed errors; they do not estimate the prevalence of those errors in actual systems. Likewise, the synthetic measurement audit assumes the family used for estimation and covers a fixed item bank. A real multimodal benchmark needs cross-model response variation,

unseen templates, protocol-specific validators, human checks of perceptual quality, and goodness-of-fit and invariance tests. Neither the score nor the proposed common scale has yet passed those checks.

Several implementation restrictions are also consequential. The plotted-data extractor expects specific Matplotlib artist arrangements, the text metric measures recall rather than precision, and grayscale comparison can miss semantically important color errors. The unequal-length data penalty is an explicit heuristic, while multivariate perception–coding abilities are not identified by the one-dimensional synthetic model. The executor is not isolated for untrusted code. All these restrictions must be addressed or bounded before running a public model leaderboard.

These limitations do not make component audits uninformative. They determine which conclusions the data support: strong but imperfect ordering under the chosen perturbation distribution; sensitivity to rubric weights; uncertain adaptive gains at some budgets; and substantial overconfidence when item uncertainty is ignored. The appropriate next stage is an empirical bridge between rendered-artifact scoring and actual model responses, not simply more synthetic respondents. A shared scale should be the outcome of that validation, rather than an assumption encoded into the benchmark name.

A staged validation path follows from these requirements. First, the composite scorer should be checked against human judgments on a sample stratified by chart family and perturbation type, large enough to estimate agreement with usable precision rather than anecdotal spot checks. Second, evaluation should move to held-out templates and then to open model responses, with raw outputs, execution failures and component observations preserved for later audit. Third, the bounded fidelity response distribution should be modeled directly, and interval calibration re-checked once item-parameter uncertainty is propagated rather than held fixed. Only after these steps would adaptive administration and cross-protocol linking be testable on real responses. The present study supplies the instrumentation and the failure examples for that program; it does not substitute for any of its stages.

7 Conclusion

MetroBench provides an execution-grounded scoring pilot and a matched-response measurement audit toward evaluating visual code generation. Its transparent experiments expose distinctions that aggregate leaderboard numbers can hide: score ordering versus local fidelity failures, selection efficiency versus unequal evaluation conditions, and rank preservation versus scale and interval calibration. The 200-item scoring study and 20-repetition numerical audit support these limited findings without establishing model capabilities or a universal multimodal scale. The released package stores raw programs, component-level scores, per-repetition estimation records and the scripts that produced every table, so each number in this paper can be recomputed rather than trusted. Readers should therefore treat absolute score levels as rubric-dependent, interpret adaptive gains only within the matched design, and regard interval calibration as the binding constraint for any future deployment.

AI USE STATEMENT

AI tools were used for language polishing.

ETHICS STATEMENT

The experiments use procedurally generated charts and synthetic respondents, with no recruited human participants, personal records or model-user data. No demographic or clinical conclusions are drawn. The principal risks are misleading benchmark interpretation and unsafe execution of submitted code: component validation must not be presented as model capability evaluation, and the local executor must not be used on untrusted programs without a separate security boundary. The manuscript reports negative and uncertain findings and describes the limitations of its scoring and uncertainty estimates.

REPRODUCIBILITY STATEMENT

Appendix A specifies the scorer, Appendix B the generators and operator schedules, and Appendix C the numerical estimation and administration procedures. Appendix D gives complete repetition-level results, while Appendix E describes the supplied source package, commands, seeds, software environment and verification invariants. Numerical tables are generated from saved outputs, and the main overview and statistical figures are provided as vector PDF and editable SVG files. The package contains the data needed to reproduce the reported analyses; it does not claim an external public repository or an independently reviewed artifact certification.

REFERENCES

Esma Balkır, Alice Pernthaller, Marco Basaldella, José Hernández-Orallo, and Nigel Collier. Confident rankings with fewer items: Adaptive llm evaluation with continuous scores, 2026. URL https://arxiv.org/abs/2601.13885.

Xiao Fei, Yang Zhang, Sarah Almeida Carneiro, and Michalis Vazirgiannis. Every wrong answer counts: Option-level psychometrics for llm multiple-choice benchmarks, 2026. URL https://arxiv.org/abs/2608.02966.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. Mme: A comprehensive evaluation benchmark for multimodal large language models, 2025. URL https://arxiv.org/abs/2306.13394.

Yi Gui, Zhen Li, Yao Wan, Yemin Shi, Hongyu Zhang, Bohua Chen, Yi Su, Dongping Chen, Siyuan Wu, Xing Zhou, Wenbin Jiang, Hai Jin, and Xiangliang Zhang. Webcode2m: A real-world dataset for code generation from webpage designs. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, pp. 1834–1845. ACM, April 2025. doi: 10.1145/3696410.3714889. URL http://dx.doi.org/10.1145/3696410.3714889.

Eliya Habba, Itay Itzhak, Asaf Yehudai, Yotam Perlitz, Elron Bandel, Michal Shmueli-Scheuer, Leshem Choshen, and Gabriel Stanovsky. Growing pains: Extensible and efficient llm benchmarking via fixed parameter calibration, 2026. URL https://arxiv.org/abs/2604.12843.

Lingjie Jiang, Shaohan Huang, Xun Wu, Yixia Li, Dongdong Zhang, and Furu Wei. Viscodex: Unified multimodal code generation via merging vision and coding models, 2025. URL https://arxiv.org/abs/2508.09945.

Manav Nitin Kapadnis, Lawanya Baghel, Atharva Naik, and Carolyn Rosé. Charteditbench: Evaluating grounded multi-turn chart editing in multimodal language models, 2026. URL https://arxiv.org/abs/2602.15758.

Aaron J. Li, Hao Huang, Youngmin Park, Yitong Ma, Wei-Lin Chiang, Li Chen, Cho-Jui Hsieh, Bin Yu, and Ion Stoica. Dualeval: Joint model-item calibration for unified llm evaluation, 2026a. URL https://arxiv.org/abs/2606.26429.

Bo Li, Yida Yin, Wenhao Chai, Xingyu Fu, and Zhuang Liu. Ueval: A benchmark for unified multimodal generation, 2026b. URL https://arxiv.org/abs/2601.22155.

Bozheng Li, Miao Yang, Zhenhan Chen, Jiawang Cao, Mushui Liu, Yi Lu, Yongliang Wu, Bin Zhang, Yangguang Ji, Licheng Tang, Jay Wu, and Wenbo Zhu. Opusanimation: Code-based dynamic chart generation, 2025a. URL https://arxiv.org/abs/2510.03341.

Peiyu Li, Xiuxiu Tang, Si Chen, Ying Cheng, Ronald Metoyer, Ting Hua, and Nitesh V. Chawla. Adaptive testing for llm evaluation: A psychometric alternative to static benchmarks, 2026c. URL https://arxiv.org/abs/2511.04689.

Ryan Li, Yanzhe Zhang, and Diyi Yang. Sketch2code: Evaluating vision-language models for interactive web design prototyping, 2025b. URL https://arxiv.org/abs/2410.16232.

Yanlin Li, Minghui Guo, Kaiwen Zhang, Shize Zhang, Yiran Zhao, Haodong Li, Congyue Zhou, Weijie Zheng, Yushen Yan, Shengqiong Wu, Wei Ji, Lei Cui, Furu Wei, Hao Fei, Mong-Li Lee, and Wynne Hsu. Unim: A unified any-to-any interleaved multimodal benchmark, 2026d. URL https://arxiv.org/abs/2603.05075.

Lele Liao, Qile Zhang, Ruofan Wu, and Guanhua Fang. Toward a unified framework for data-efficient evaluation of large language models, 2025. URL https://arxiv.org/abs/2510.04051.

Zhiyu Lin, Zhengda Zhou, Zhiyuan Zhao, Tianrui Wan, Yilun Ma, Junyu Gao, and Xuelong Li. Webuibench: A comprehensive benchmark for evaluating multimodal large language models in webui-to-code. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 15780–15797. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.815. URL http://dx.doi.org/10.18653/v1/2025.findings-acl.815.

Jia Liu, Zhiyu Xu, and Yuqi Gu. Scalable text-embedding-informed cognitive diagnosis of large language models, 2026a. URL https://arxiv.org/abs/2603.14676.

Zhilin Liu, Ye Huang, Ting Xie, Ruizhi Zhang, Wen Li, and Lixin Duan. Coding with eyes: Visual feedback unlocks reliable gui code generating and debugging, 2026b. URL https://arxiv.org/abs/2604.19750.

Juan Francisco Mandujano Reyes. Laplace-psn-irt: Uncertainty quantification for neural item response theory models of llm benchmarks, 2026. URL https://arxiv.org/abs/2607.25257.

Ahmed Masry, Do Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-acl.177. URL http://dx.doi.org/10.18653/v1/2022.findings-acl.177.

Joshua Fonseca Rivera, Neil Shah, David Demitri Africa, and Konstantinos Voudouris. Item response theory for ai safety, 2026. URL https://arxiv.org/abs/2608.05086.

Ergan Shang, Weijing Tang, and Yinqiu He. Llm evaluation on unseen questions: Contextual multidimensional irt model, 2026. URL https://arxiv.org/abs/2608.22295.

Chenglei Si, Yanzhe Zhang, Ryan Li, Zhengyuan Yang, Ruibo Liu, and Diyi Yang. Design2code: Benchmarking multimodal code generation for automated front-end engineering, 2025. URL https://arxiv.org/abs/2403.03163.

Jiahao Tang, Henry Hengyuan Zhao, Lijian Wu, Zijian Zhang, Yifei Tao, Dongxing Mao, Yang Wan, Jingru Tan, Min Zeng, Min Li, and Alex Jinpeng Wang. From charts to code: A hierarchical benchmark for multimodal models, 2026. URL https://arxiv.org/abs/2510.17932.

Jiahui tang, Kuicai Dong, Dexun Li, Hongchao Gu, Haocheng Yu, Wei Han, Chen Zhang, Yong Liu, Hao Wang, and Enhong Chen. Deepchart: How far are llms from faithful data-science chart generation?, 2026. URL https://arxiv.org/abs/2608.26757.

Yinghao Tang, Xueding Liu, Boyuan Zhang, Tingfeng Lan, Yupeng Xie, Jiale Lao, Yiyao Wang, Haoxuan Li, Tingting Gao, Bo Pan, Luoxuan Weng, Xiuqi Huang, Minfeng Zhu, Yingchaojie Feng, Yuyu Luo, and Wei Chen. Igenbench: Benchmarking the reliability of text-to-infographic generation, 2026. URL https://arxiv.org/abs/2601.04498.

Shunki Uebayashi, Kento Masui, Kyohei Atarashi, Han Bao, Hisashi Kashima, Naoto Inoue, Mayu Otani, and Koh Takeuchi. Evaluating cross-modal reasoning ability and problem characteristics with multimodal item response theory, 2026. URL https://arxiv.org/abs/2603.02663.

Yuxuan Wan, Chaozheng Wang, Yi Dong, Wenxuan Wang, Shuqing Li, Yintong Huo, and Michael R. Lyu. Automatically Generating UI Code from Screenshot: A Divide-and-Conquer-Based Approach, 2025. URL https://arxiv.org/abs/2406.16386.

Jingyu Xiao, Yuxuan Wan, Yintong Huo, Zixin Wang, Xinyi Xu, Wenxuan Wang, Zhiyao Xu, Yuhang Wang, and Michael R. Lyu. Interaction2code: Benchmarking mllm-based interactive webpage code generation from interactive prototyping, 2025. URL https://arxiv.org/abs/2411.03292.

Zhiyu Xu, Jia Liu, Yixin Wang, and Yuqi Gu. Latency-response theory model: Evaluating large language models via response accuracy and chain-of-thought length, 2026. URL https://arxiv.org/abs/2512.07019.

Cheng Yang, Chufan Shi, Yaxin Liu, Bo Shui, Junjie Wang, Mohan Jing, Linran Xu, Xinyu Zhu, Siheng Li, Yuxiang Zhang, Gongye Liu, Xiaomei Nie, Deng Cai, and Yujiu Yang. Chartmimic: Evaluating lmm's cross-modal reasoning capability via chart-to-code generation, 2025. URL https://arxiv.org/abs/2406.09961.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models, 2024. URL https://arxiv.org/abs/2306.13549.

Houston H. Zhang, Tao Zhang, Baoze Lin, Yuanqi Xue, Yincheng Zhu, Huan Liu, Li Gu, Linfeng Ye, Ziqiang Wang, Xinxin Zuo, Yang Wang, Yuanhao Yu, and Zhixiang Chi. Widget2code: From visual widgets to ui code via multimodal llms, 2026a. URL https://arxiv.org/abs/2512.19918.

Xu Zhang, Xudong Gong, Jiacheng Qin, Qiang Wang, JiaQi Liao, Zhe Wang, Dawei Feng, and Bo Ding. Beyond scores: Diagnostic llm evaluation via fine-grained abilities, 2026b. URL https://arxiv.org/abs/2604.12191.

Xuanle Zhao, Xianzhen Luo, Qi Shi, Chi Chen, Shuo Wang, Zhiyuan Liu, and Maosong Sun. Chartcoder: Advancing multimodal large language model for chart-to-code generation, 2025. URL https://arxiv.org/abs/2501.06598.

Tianpeng Zheng, Zhehan Jiang, Jiayi Liu, and Shicong Feng. Leveraging computerized adaptive testing for cost-effective evaluation of large language models in medical benchmarking. *npj Digital Medicine*, 9(1), May 2026. ISSN 2398-6352. doi: 10.1038/s41746-026-02671-w. URL http://dx.doi.org/10.1038/s41746-026-02671-w.

Hongli Zhou, Hui Huang, Ziqing Zhao, Lvyuan Han, Huicheng Wang, Kehai Chen, Muyun Yang, Wei Bao, Jian Dong, Bing Xu, Conghui Zhu, Hailong Cao, and Tiejun Zhao. Lost in benchmarks? rethinking large language model benchmarking with item response theory, 2026. URL https://arxiv.org/abs/2505.15055.

Kai Zou, Ziqi Huang, Yuhao Dong, Shulin Tian, Dian Zheng, Hongbo Liu, Jingwen He, Bin Liu, Yu Qiao, and Ziwei Liu. Uni-mmmu: A massive multi-discipline multimodal unified benchmark, 2026. URL https://arxiv.org/abs/2510.13759.

A Exact Scoring Conventions

A.1 Execution and image channels

The pilot executor compiles and executes the trusted program with the path variable `OUT` bound to a temporary PNG destination. It obtains the last Matplotlib figure, closes registered figures, draws the canvas and extracts properties from its active axes. The program templates create a single axes object; general multi-axes programs are outside this extractor's supported scope. The implementation is not an adversarial code runner and provides no filesystem, network, process or time-limit isolation.

Let $R, V \in [0,1]^{96\times128}$ be the reference and variant grayscale arrays after resizing the saved PNGs using the installed Pillow implementation. The image channels are

$$P(R,V) = 1 - \frac{1}{96\cdot128}\sum_{u,v}|R_{uv} - V_{uv}|, \tag{3}$$

$$I(R,V) = \frac{|M_R \cap M_V|}{|M_R \cup M_V|}, \qquad M_A = \{(u,v) : A_{uv} < 0.98\}. \tag{4}$$

If the union is empty, I is set to 1. The resize routine and threshold are fixed throughout the study, rather than adapted per family. A grayscale comparison intentionally discards chromatic information; two colors with similar luminance may receive similar pixel scores. Large white backgrounds also contribute agreement and can make P relatively insensitive to an important but spatially small defect.

A.2 Artist-based plotted-data agreement

For equal-length reference and candidate vectors $r, v \in \mathbb{R}^n$, the implemented helper is

$$g(v,r) = \max\left\{0, 1 - \frac{n^{-1}\sum_{k=1}^{n}|v_k - r_k|}{\max(\max_k|r_k|, 10^{-9})}\right\}. \tag{5}$$

When both vectors are empty, the helper returns 1. When their lengths differ, it instead returns $\max\{0, 1-0.25\,|\mathrm{len}(v)-\mathrm{len}(r)|\}$, without comparing the retained values. This length-only fallback is a limitation, not a principled semantic equivalence test. It can assign high agreement to unequal representations whose values are otherwise inaccurate.

Bar extraction reads patch heights, line extraction reads the first line's y-data, and scatter extraction reads the first collection's x and y offsets. Scatter agreement averages g on the two coordinates. Pie extraction reads wedge angular spans modulo 360 degrees and normalizes their sum before comparison with normalized reference fractions. Area extraction reads the first two line series; missing series are represented by empty vectors, and agreement averages the two series scores. The current line channel ignores x-values, while the area channel does not directly recover fill polygons. Consequently, D should be understood as a family-specific plotted-data diagnostic, not complete geometric or functional equivalence.

A.3 Text recall and composite precision

The text pool is the set of nonempty strings from the title, horizontal and vertical axis labels, tick labels and axes text artists. If the reference pool is $\mathcal{T}_R$, the text score is

$$T = \frac{|\mathcal{T}_R \cap \mathcal{T}_V|}{|\mathcal{T}_R|}, \tag{6}$$

with $T = 1$ when the reference pool is empty. Multiplicity and location are ignored; additional incorrect text is not penalized. Changes in automatic ticks can alter text recall even when an explicit label was not modified. The component therefore conflates some renderer-induced formatting differences with semantic annotation differences.

The composite is computed from unrounded components and then rounded to four decimal places for storage. Each component is also stored at four-decimal precision. Recombining these stored components for sensitivity analysis need not exactly reproduce the stored composite, although the observed discrepancy is bounded by 1.1×10^{-4}. All alternative weight results use the same rounded component matrix and are identified as post-hoc reweighting.

Table 4: Complete perturbation schedule. Operators within a cell are applied in order. Levels are cumulative in operator membership, not necessarily in random realization.

Family	Level 1	Level 2	Level 3	Level 4
Bar	Change color; flip width	Drop y label; toggle grid	Multiply one value by 1.2; scale all by 0.8	Multiply one value by 3; drop title
Line	Change marker; change width	Disable grid; append title suffix	Add 5 to even-indexed y; remove markers	Add 10 to all y; change color
Scatter	Change point size; change marker	Drop title; jitter x	Add y noise; scale x by 2	Permute y values
Pie	Rotate colors; two-decimal percentages	Remove slice labels; append title suffix	Multiply one size by 1.3; remove percentages	Set one size to 50; use one color
Area	Flip opacity; change second color	Disable grid; append title suffix	Add 2 to second series; remove fill	Add 8 to first series; remove second line

A.4 What the scorer does not establish

The observed ordering concerns deliberately designed corruptions, not human preferences or a natural distribution of model errors. Some operators remove content that the heuristic channels explicitly count, so sensitivity alone does not demonstrate external construct validity. Equivalent programs may use different artist structures or ordering, and an adversarial program may manipulate accessible artist data without producing a faithful result. Human evaluation, representation-invariance tests and adversarial robustness checks remain necessary for deployment.

B Procedural Items and Perturbation Schedule

B.1 Generation and coverage

The base generator produces 40 items for each of five families, for 200 total. Family order is bar, line, scatter, pie and area. The base seed for item i within family index f is $20260922 + 1000f + i$. Every program is rendered once during generation, and generation fails rather than silently accepting an unrenderable reference. Templates vary data, colors and selected graphical attributes, but they do not represent the full diversity of scientific figures.

Bar templates have 4–8 categories with values drawn uniformly from 5 to 100. Line templates contain 5–12 points formed from a Gaussian random walk. Scatter templates contain 20–60 correlated points with added Gaussian noise. Pie templates contain 3–6 categories from positive integer draws normalized to percentages. Area templates contain two related 5–12-point series, two lines and a filled region between them. All templates use a 6-by-4-inch Matplotlib figure at 100 dpi and save their source code and reference data.

B.2 Level-specific operators

Table 4 lists the full schedule. A level-s variant applies all operators with assigned level at most s, in the listed order. Its random seed is $77031 + 10i + s$, where i is the item's global index. As a result, operators shared across levels may receive different random draws; a level-4 variant is not necessarily obtained by editing the already-rendered level-3 variant. This design supports graded testing but does not justify a claim of strict severity nesting at the instance level.

Color and marker changes choose a different member of the template palette or marker set. Bar width switches between 0.6 and 0.8, line width among 1.0, 1.5 and 2.0, scatter size among 40, 70 and 110, and area opacity between 0.3 and 0.5. Scatter jitter has standard deviation 0.1 in x and 0.8 in y. Setting a pie size to 50 is not guaranteed to increase its original value; normalization and interactions with earlier changes can therefore reverse the intended damage ordering. These exact choices explain why the score ordering must be measured rather than assumed.

B.3 Reference checks and analysis matrix

Level 0 has no applied operators and reconstructs the base code through the same family builder used for other levels. The scorer assigns 1 to each reference comparison in the recorded run. The analysis matrix has 200 rows and five ordered columns, and the component tensor has shape $200 \times 5 \times 4$. An item's row is never split across the resampling scheme used for paired loss confidence intervals. Raw programs and component observations are retained so that scores can be independently recomputed rather than inferred from the final plots.

C Numerical Estimation and Administration

C.1 Per-item marginal calibration

For item j and a calibration population prior $\pi(\theta)$, the objective is

$$\ell_j(a, b) = \sum_{i=1}^{N} \log \int p(y_{ij} \mid \theta, a, b)\pi(\theta)\, d\theta. \tag{7}$$

Fifteen Gauss–Hermite nodes x_q with weights w_q approximate the integral under $\mathcal{N}(m, s^2)$ using $\theta_q = m + s\sqrt{2}x_q$ and normalized weights $w_q/\sqrt{\pi}$. The implementation evaluates the sum in the log domain with `logsumexp`. Parameters are fitted by L-BFGS-B, starting at $(1, 0)$, with bounds $a \in [0.05, 4]$ and $b \in [-4, 4]$ and a 200-iteration limit. If the first optimizer call reports failure, it retries from $(0.5, 0)$.

The objective is factorized over items. Unlike a full respondent-level marginal likelihood, it integrates each item's response separately and does not use joint response dependence to infer the calibration respondents' abilities. The legacy routine returns the second attempt's parameters without persisting its convergence status. We therefore do not claim certified optimizer convergence; bounds and finite outputs are checked, but a future implementation should retain termination diagnostics and verify optimality.

C.2 EAP and conditional uncertainty

For administered item set A and prior density π, the grid posterior is

$$\omega_g \propto \pi(\theta_g) \prod_{j \in A} p(y_j \mid \theta_g, \hat{a}_j, \hat{b}_j), \qquad \hat{\theta}_{\mathrm{EAP}} = \sum_g \bar{\omega}_g \theta_g, \tag{8}$$

where normalized weights $\bar{\omega}_g$ sum to one. The production grid is $\{-6, -5.95, \ldots, 6\}$. A finer 0.025-spaced grid is used only to assess posterior-mean sensitivity, while interval diagnostics use spacing 0.005. Quantiles interpolate the cumulative grid mass at cell midpoints, avoiding the half-cell displacement produced by assigning all mass to one grid edge. The refined diagnostics are distinct from the estimator used in the main adaptive-test runs.

The plug-in posterior conditions on estimated item parameters as though they were known. It does not integrate their uncertainty, so its credible intervals need not have nominal repeated-sampling coverage. The known-parameter control replaces $\hat{a}, \hat{b}$ with the generating a, b while leaving the held-out responses and interval procedure unchanged. Differences can reflect item estimation bias, omitted uncertainty and approximation error; the experiment does not isolate a single cause among them.

C.3 Fisher-information expression

Writing $Z = (Y - \mu)/\sigma$ for a standard normal variable, the score away from the variance cusp is

$$\partial_\theta \log p(Y \mid \theta) = \frac{\mu'}{\sigma} Z + \frac{\sigma'}{\sigma}(Z^2 - 1). \tag{9}$$

Since $\mathbb{E}[Z^2] = 1$, $\mathbb{E}[(Z^2 - 1)^2] = 2$, and $\mathbb{E}[Z(Z^2 - 1)] = 0$, its second moment is

$$I_j(\theta) = \frac{(\mu')^2 + 2(\sigma')^2}{\sigma^2} = \frac{a_j^2 + 2(0.15)^2}{(0.5 + 0.15|\theta - b_j|)^2}, \qquad \theta \neq b_j. \tag{10}$$

The expression is a property of the stated response model, not a newly introduced estimator. At the cusp $\theta = b_j$, ordinary differentiability does not hold; the implementation uses the same limiting expression for selection. No asymptotic efficiency theorem is claimed for this greedy rule under estimated parameters.

C.4 Matched administration and endpoint invariant

Fisher administration starts from ability estimate zero. At each step it chooses the highest-information unused item, reads its pre-generated held-out response, appends that observation to the administered set, and recomputes EAP. Ties are resolved by the smallest item index through `argmax`. Random administration uses a fresh permutation of the 200 indices, but reads the same respondent's response row. Both policies are evaluated at budgets 5, 10, 20, 40, 80, 120 and 200.

Proposition 1 (Order invariance of the full-test endpoint)**.** *For a fixed response row, item parameters, scoring prior and EAP grid, administration of every item exactly once gives the same full-test EAP estimate for every item order in exact arithmetic.*

Proof. At each grid point, the log likelihood after full administration is the sum of the same 200 item log densities, independent of the permutation. Adding the same log prior therefore gives the same unnormalized posterior weights. Normalization and the weighted grid average preserve this equality. Floating-point summation may differ slightly, so the implementation verifies agreement to absolute tolerance 10^{-10} rather than bitwise equality. □

This endpoint invariant is a comparison check, not a claim that every finite-sample point estimate improves monotonically with each additional item. Selection is adaptive and item parameters are imperfectly estimated. Individual trajectories can therefore fluctuate. What the invariant prevents is attributing incompatible full-test endpoints to an alleged selection benefit.

C.5 Anchor linkage and rank invariance

The restricted version-2 calibration respondents have true abilities drawn uniformly from $[-1, 1]$. The linked arm estimates their mean m and sample standard deviation s from anchor-only EAP values, floors s at 0.3, and fits the 120 new items under $\mathcal{N}(m, s^2)$. It retains the 80 estimated version-1 anchor parameters. The free arm fits all 200 version-2 items under $\mathcal{N}(0, 1)$, and both arms score held-out respondents with the same standard normal prior.

A positive affine transformation $z = \alpha\theta + \beta$, $\alpha > 0$, preserves the ordering of distinct abilities, but can produce large RMSE and a slope different from one. Spearman correlation therefore cannot alone certify scale comparability. This algebraic observation does not assert that the chosen heteroskedastic family is invariant under arbitrary affine reparameterization; its fixed variance constants generally are not. It explains the need to inspect rank, scale and absolute error separately.

C.6 Analytical reference checks and quadrature audit

We constructed a separate synthetic reference test with known parameters: $\theta \sim \mathcal{N}(0, 1)$ and $Y_j = b_j + a_j\theta + \varepsilon_j$, with independent $\varepsilon_j \sim \mathcal{N}(0, 1)$, $a = (0.5, 1, 1.5, 2)$ and $b = (-1, 0, 0.5, 1)$. For an administered set A, completing the square gives the exact posterior $\mathcal{N}(m_A, v_A)$, where

$$v_A = \left(1 + \sum_{j \in A} a_j^2\right)^{-1}, \qquad m_A = v_A \sum_{j \in A} a_j(Y_j - b_j).$$

An independent reference implementation checks 48 posterior calculations against these expressions to absolute tolerance 10^{-10} and verifies 72 full-test orderings. These provide reproducible tests of posterior arithmetic and ordering; calibration is examined separately below.

Here item information is a_j^2, so selection is static, following the order $4, 3, 2, 1$. Bayes MSE equals v_A, and the exact posterior intervals have 95% prior-average coverage under the known-parameter reference model.

Table 5: Analytical Bayes RMSE under a separate known-parameter Gaussian model. Entries integrate over the prior and observation noise in this reference model. The random column averages MSE over all equally likely K-item subsets before taking the square root.

K	Information-ranked	Random subset
1	0.447214	0.672252
2	0.371391	0.493377
3	0.348155	0.398989
4	0.342997	0.342997

Table 6: Full family-by-level scores. Mean $\pm$ sample s.d. across 40 items per cell.

Family	Level 0	Level 1	Level 2	Level 3	Level 4
Area	1.000 ± 0.000	0.992 ± 0.003	0.957 ± 0.033	0.752 ± 0.025	0.521 ± 0.034
Bar	1.000 ± 0.000	0.937 ± 0.006	0.906 ± 0.008	0.854 ± 0.009	0.742 ± 0.057
Line	1.000 ± 0.000	0.994 ± 0.003	0.898 ± 0.064	0.726 ± 0.056	0.523 ± 0.053
Pie	1.000 ± 0.000	0.952 ± 0.004	0.885 ± 0.004	0.865 ± 0.006	0.824 ± 0.025
Scatter	1.000 ± 0.000	0.956 ± 0.023	0.878 ± 0.019	0.724 ± 0.036	0.681 ± 0.039

A separate stress test examines the pilot's 15-node calibration quadrature. For the legal parameters $a = 2.5$, $b = 0$ at $y = 0$, it returns marginal density 0.255595, versus 0.163668 from cusp-split integration, a 56.2% overestimate. The reference is numerical: 256 and 512 Gauss–Legendre nodes per piece agree within 10^{-11}, with an independent refined trapezoid check. This identifies a source of calibration error distinct from EAP discretization. The reported measurement results characterize the original 15-node implementation; this diagnostic alone does not attribute its observed undercoverage to quadrature error.

D Complete Statistical Results

D.1 Family scores and item-paired differences

Table 6 reports all family-level means and sample standard deviations, including the unchanged reference. Every family has 40 items at every level. Table 7 reports the adjacent-level losses and their paired intervals. The bootstrap resamples 40 items with replacement within each family and retains the complete five-level row of each sampled item, using 5,000 replicates and seed 8101.

The Friedman statistic is computed from the five repeated scores per item, using the implementation's tie correction. With $n = 200$ items and $k = 5$ levels, Kendall's concordance effect size is $W = \chi^2/[n(k-1)]$. The reported small p-value is the asymptotic chi-squared tail probability, not a claim of exact finite-sample permutation inference. Balanced family composition is part of this experiment and should not be generalized to an unbalanced production workload without reweighting or separate reporting.

D.2 Component-level failure accounting

Table 8 decomposes the three local reversals. Each entry is the stored component score at level 4 minus its value at level 3; the final column is the corresponding composite difference. Components can improve when a different random mutation partly restores values, changes the visible mask or affects automatic text. The decomposition identifies which numerical channels contributed to a reversal, not an isolated causal effect of one operator in a multi-operator chain.

D.3 Paired adaptive-test contrasts

For each budget, the contrast is the repetition-level RMSE under random selection minus RMSE under Fisher selection. The paired confidence interval uses a Student-t critical value with 19 degrees of freedom. Table 9 also reports the standardized paired effect $d_z = \bar{d}/s_d$ and a Holm adjustment over the six nontrivial budget tests. These tests assume that repetition-level contrasts are sufficiently

Table 7: Paired mean loss $F_s - F_{s+1}$ with percentile 95% bootstrap intervals.

Transition	Mean loss	95% paired bootstrap interval
$0 \to 1$	0.0337	$[0.0322, 0.0353]$
$1 \to 2$	0.0615	$[0.0567, 0.0664]$
$2 \to 3$	0.1204	$[0.1153, 0.1258]$
$3 \to 4$	0.1260	$[0.1209, 0.1311]$

Table 8: All local score reversals and component changes, level 4 minus level 3.

Item	ΔP	ΔI	ΔD	ΔT	ΔF
pie_001	0.0020	0.0004	0.0413	0.0000	0.0132
pie_026	0.0009	-0.0004	0.0003	0.0000	0.0003
scatter_003	0.0007	0.0117	-0.0875	0.9091	0.0679

regular for the t approximation; the intervals should be interpreted together with effect magnitude rather than reduced to a binary superiority verdict.

D.4 Conditional interval coverage

Table 10 reports coverage of nominal 95% equal-tailed ability intervals. Each repetition contributes the fraction of its 60 held-out abilities falling within their intervals; means, standard deviations and confidence intervals summarize those 20 fractions. Known-parameter and plug-in estimates use identical responses and a refined 0.005 grid, so the comparison is not explained by different held-out data. Low plug-in coverage exposes a calibration problem despite relatively small point-estimation RMSE.

The known-parameter control is a diagnostic oracle, not a deployable baseline. Its near-nominal average coverage is conditional on the correctly specified synthetic model and sampled ability population. The plug-in result combines parameter uncertainty, estimation bias and numerical fitting assumptions. Joint uncertainty propagation, calibration-sample bootstrap or hierarchical posterior inference are possible remedies, but they are not implemented or claimed as results here.

D.5 Every repetition and ability-stratum diagnostics

Table 11 gives the 40-item and full-test errors for every repetition, together with full-test bias and plug-in coverage. This makes between-repetition variation visible instead of hiding it behind a single aggregate. Table 12 additionally pools full-test errors by true ability range: low is $\theta < -1$, central is $|\theta| \leq 1$, and high is $\theta > 1$. The stratum table is descriptive; pooled respondents are not independent replications of calibration.

E Implementation and Reproduction Guide

E.1 Files and execution order

The source package separates generators, scoring, numerical estimation, experiment drivers, analysis and figure generation. Base item specifications and programs are stored in `outputs/items.json`, and candidate programs in `outputs/variants.json`. The original scored rows are stored in `outputs/scoring_results.json`; the current synthetic repetitions and derived analyses are under `outputs/revision/`. The final manuscript uses the current corrected response study, not the earlier legacy CAT driver.

The README supplies a complete command sequence. Scoring is divided into two independent shards before merging, while each of the 20 numerical repetitions writes a separate file. This permits continuation after interruption without repeating completed independent work. The merge expects the specified repetition indices and fails if required files are absent. Supplementary tables are generated from result files rather than manually transcribed.

Table 9: All nontrivial paired budget comparisons across 20 repetitions. Positive gain favors Fisher. Intervals are marginal 95% intervals; Holm correction is applied to the reported p-values, not to interval widths.

Items	Gain	95% CI	t_{19}	p	Holm p	d_z
5	0.0180	$[-0.0051, 0.0410]$	1.630	0.11959	0.47837	0.364
10	0.0088	$[-0.0092, 0.0268]$	1.022	0.31983	0.51232	0.228
20	0.0071	$[-0.0053, 0.0195]$	1.195	0.24663	0.51232	0.267
40	0.0057	$[-0.0027, 0.0140]$	1.424	0.17077	0.51232	0.318
80	0.0045	$[0.0010, 0.0080]$	2.718	0.01365	0.08191	0.608
120	0.0019	$[0.0003, 0.0036]$	2.412	0.02615	0.13074	0.539

Table 10: Nominal 95% ability-interval coverage. The oracle control conditions on true item parameters; the plug-in method conditions on estimated parameters without propagating their uncertainty.

Item parameters	Mean coverage $\pm$ s.d.	95% interval across repetitions
Estimated (plug-in)	0.458 ± 0.304	$[0.316, 0.601]$
Known (oracle)	0.958 ± 0.027	$[0.946, 0.971]$

The separate synthetic-validation supplement provides fixed inputs, an independent checker, outputs and validation records for Appendix C.6.

E.2 Environment and constants

The recorded experiment environment uses CPU float64 computation on macOS arm64, Python 3.13.5, NumPy 2.2.6, SciPy 1.16.1 and Matplotlib 3.10.5 with the Agg backend. No GPU training or model inference is included. Timings saved per numerical repetition use a wall-clock timer around that repetition; they are execution records, not hardware-normalized speed comparisons. The software environment and measurement constants should be held fixed when checking deterministic reproduction.

E.3 Validation invariants and limitations

Every stored administration order must be a permutation without repeated items, and both full-budget endpoints must match direct full-test EAP to 10^{-10}. Arrays and posterior weights are required to be finite, and posterior weights must sum to one. Scoring rows must retain their item and level identifiers, and recombined component scores are checked against the stored composite within rounding tolerance. These invariants catch specific implementation errors but do not prove construct validity, optimizer optimality or correct behavior on untrusted programs.

The final overview and compact statistical figures are vector PDF files, with text fonts embedded and no raster image objects in those four figures. Editable SVG files accompany them. The appendix's rendering examples are necessarily raster observations of the saved chart images; they are explicitly distinguished from the vector summary graphics. Manuscript compilation checks the inclusion of all figures and the resolution of references, while page inspection checks readability and clipping.

E.4 Boundaries of any future model evaluation

A future deployment should freeze the task specifications, scoring rubric, bank-linkage rules and evaluation budget before collecting held-out model responses. It should use a separately secured executor, preserve raw outputs and error logs, and report execution failures rather than discarding them. The bounded fidelity response distribution must be modeled and diagnosed, and Reflect responses require their own validator. Human agreement studies, template-held-out evaluation, model-family-held-out calibration and cross-protocol invariance checks are necessary before claiming a common scale. None of those deployments is implied by the results in this manuscript.

Table 11: All 20 synthetic repetitions. Each has 150 calibration respondents and 60 independent held-out respondents. Coverage refers to nominal 95% plug-in full-test intervals.

Rep.	Fisher_{40}	Random_{40}	Full	Full bias	Coverage
0	0.1186	0.0936	0.0810	0.0113	0.4000
1	0.1016	0.0986	0.0829	-0.0714	0.3333
2	0.0584	0.0644	0.0441	-0.0369	0.7667
3	0.1559	0.1409	0.1333	0.1290	0.0167
4	0.0902	0.0798	0.0660	0.0477	0.5000
5	0.1707	0.1704	0.1688	0.1243	0.2833
6	0.1563	0.1476	0.1481	-0.1439	0.0333
7	0.0712	0.0976	0.0804	0.0323	0.5667
8	0.0467	0.0903	0.0553	-0.0111	0.7167
9	0.1024	0.1034	0.0855	0.0544	0.4833
10	0.2247	0.2268	0.2123	-0.2109	0.0000
11	0.1375	0.1377	0.1131	0.0872	0.2667
12	0.0520	0.0757	0.0441	0.0261	0.8167
13	0.0557	0.0778	0.0346	-0.0061	0.8667
14	0.1043	0.1209	0.1026	0.0806	0.3333
15	0.0773	0.0840	0.0583	0.0493	0.5833
16	0.0530	0.0751	0.0296	-0.0125	0.9500
17	0.1058	0.0825	0.0755	0.0235	0.4000
18	0.0488	0.0729	0.0407	0.0066	0.8333
19	0.1496	0.1543	0.1394	-0.1361	0.0167

Table 12: Descriptive full-test errors by true ability stratum, pooled over repetitions.

Ability stratum	Respondents	Pooled RMSE	Mean error
low	178	0.1119	0.0045
central	830	0.0951	0.0001
high	192	0.1189	0.0090

Table 13: Implementation responsibilities. Paths are relative to the experiment directory.

Module	Responsibility
`gen_items.py`	Five deterministic reference-chart families and source builders.
`perturb.py`	Ordered level-specific mutations and variant program reconstruction.
`fidelity.py`	Trusted execution, image comparison, artist extraction and text recall.
`run_scoring.py`	Independently runnable scoring shards and merged item statistics.
`irt.py`	Marginal fitting, EAP and Fisher information; legacy administration is not used for final comparisons.
`revision_analysis.py`	Matched held-out administration, anchor study and paired scoring analysis.
`supplemental_analysis.py`	Weight sensitivity, coverage, numerical sensitivity and paired tests.
`final_tables.py`	Numeric LaTeX tables generated from result JSON.
`compact_statistics.py`	User-reviewed compact vector statistical figures.
`swe_style_overview.py`	Original vector overview with artifact and scoring examples.

Table 14: Fixed seeds, dimensions and numerical controls used in the final study.

Quantity	Value
Chart generation seed	$20260922 + 1000f + i$
Perturbation seed	$77031 + 10i + s$
Synthetic repetition seed	$20260924 + 1000r$
Version-1 bank seed	4242
New version-2 items seed	4742
Bootstrap seed / resamples	8101 / 5,000
Reference items / variants	200 / 1,000
Numerical items / repetitions	200 / 20
Calibration / held-out respondents	150 / 60 per repetition
Restricted calibration / anchors	50 / 80
Noise constants (σ_0, σ_1)	$(0.5, 0.15)$
Quadrature / production EAP grid	15 nodes / 241 points
Production / interval grid spacing	0.05 / 0.005
Image comparison resolution / mask threshold	128×96 / 0.98

F RENDERED EXAMPLES

Figure 5 shows the first item by identifier from each family, selected independently of its score. The panels contain actual reference and candidate PNG renders at levels 0, 1 and 4, together with absolute RGB difference images on a common range. Score labels are read from the original scoring output. These examples illustrate the appearance of the prescribed mutations; they are not outputs from a language model or evidence of generalization to unseen chart styles.



Figure 5: Reference and perturbed renders across the five families. Columns show the reference, level-1 variant, level-4 variant and pixel difference between the reference and level 4. These are raster image observations inside a PDF, unlike the main text's vector overview and statistical plots. The scorer uses grayscale resized images and artist/text channels, so the RGB difference panel is a qualitative aid rather than the complete scoring function.