

The Complexity Kink

LLM Rubric Instruments for Causal Inference on Code Generation Reliability

Michael Hernandez and Tian Zhao

University of Wisconsin-Milwaukee

5,000 Python prompts 21 models 105,000 generations

COMPUTING SUPPORT

Paper: [arXiv:2609.19616](https://arxiv.org/abs/2609.19616)



THE REQUEST

THE ASSIGNMENT

PYTHON / CONDENSED EXCERPT

Implement a **BinarySearchTree**

class in Python that supports the following operations:

<code>insert(value)</code>	Inserts a new node with the given value
<code>search(value)</code>	Returns True if a node with the given value exists
<code>in_order_traversal()</code>	Returns a list of values in in-order traversal order
<code>delete(value)</code>	Deletes the node with the given value if it exists

Source: recorded benchmark prompt 6b0f97bc...

THE RECORDED RESPONSE

THE RESPONSE

KIMI K2.5 / RECORDED CODE

```
1 class Node:
2     def __init__(self, value):
3         self.value = value
```

Complete extracted code • One observed benchmark generation

Generated-code cyclomatic complexity: 1

Recorded tests: 0 passed / 10 failed

Did the task get easier, or did the answer get shorter?

Actual recorded Kimi K2.5 response: 0/10 tests. Its generated-code cyclomatic complexity is 1. The requested task still includes insert, search, traversal and deletion.

Measure the request.

Four judges outside the evaluated panel

THE ASSIGNMENT

PYTHON / CONDENSED EXCERPT

Implement a

BinarySearchTree

class in Python that supports the following operations:

`insert(value)`

Inserts a new node with the given value

`search(value)`

Returns True if a node with the given value exists

`in_order_traversal()`

Returns a list of values in in-order traversal order

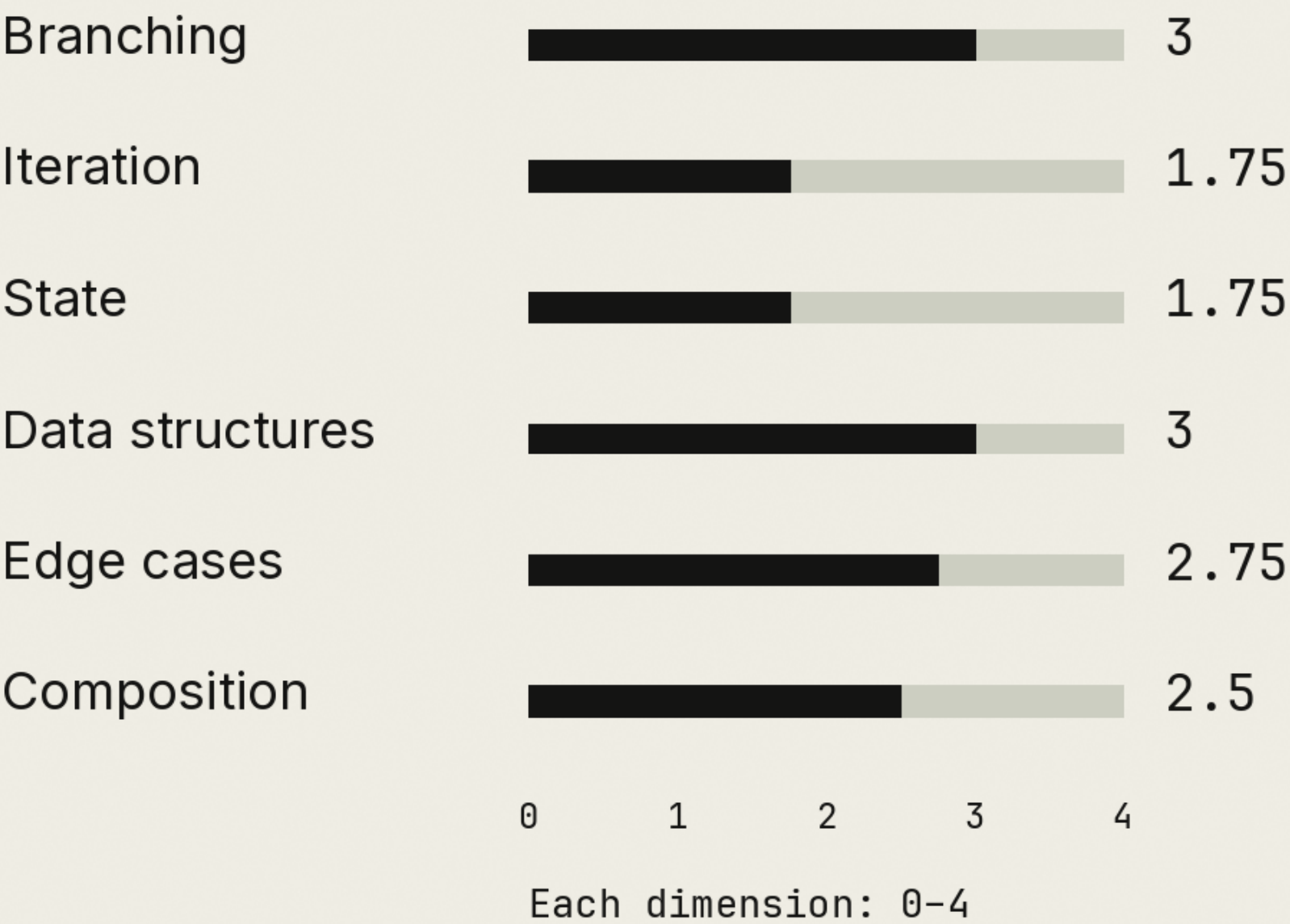
`delete(value)`

Deletes the node with the given value if it exists

Source: recorded benchmark prompt 6b8f97bc...

14.75

Prompt-only index / 0–24



Same task. Different answers.

THE REQUEST

GENERATED CODE / CC

AUDIT

THE ASSIGNMENT

PYTHON / CONDENSED EXCERPT

Implement a
BinarySearchTree

class in Python that supports the following operations:

- `insert(value)` Inserts a new node with the given value
- `search(value)` Returns True if a node with the given value exists
- `in_order_traversal()` Returns a list of values in in-order traversal order
- `delete(value)` Deletes the node with the given value if it exists

Source: recorded benchmark prompt 6b8f97bc...

14.75

Prompt-only index / 0–24

Kimi K2.5

■ 1

Failed

GPT-5.4

■ 26

Correct

Claude Opus 4.7

■ 23

Correct

The prompt index stays fixed at 14.75/24 across evaluated models. Prompt rubric index and generated-code cyclomatic complexity use different definitions and scales.

Calibration and stability

CALIBRATION / PROMPT-SIDE INDEX

0.872

LLM inter-rater ICC

Agreement among four judges

≈ 0.41

Human-ensemble Pearson r

First author; disagreement-enriched n=200

0.963

Paraphrase rank correlation

Useful proxy. Visible uncertainty.

ICC(2,1): n=4,998 complete prompts. Human Pearson r=0.408: n=200, disagreement-enriched. Paraphrase Spearman correlation: 150 pairs, equivalence unverified. Agreement does not establish correctness.

Failure can obscure requested structure

14,247 FAILURES WITH MEASURABLE OUTPUT CC



25.0%

of these failures

Prompt score > 8
Output CC ≤ 10

969

without measurable output CC

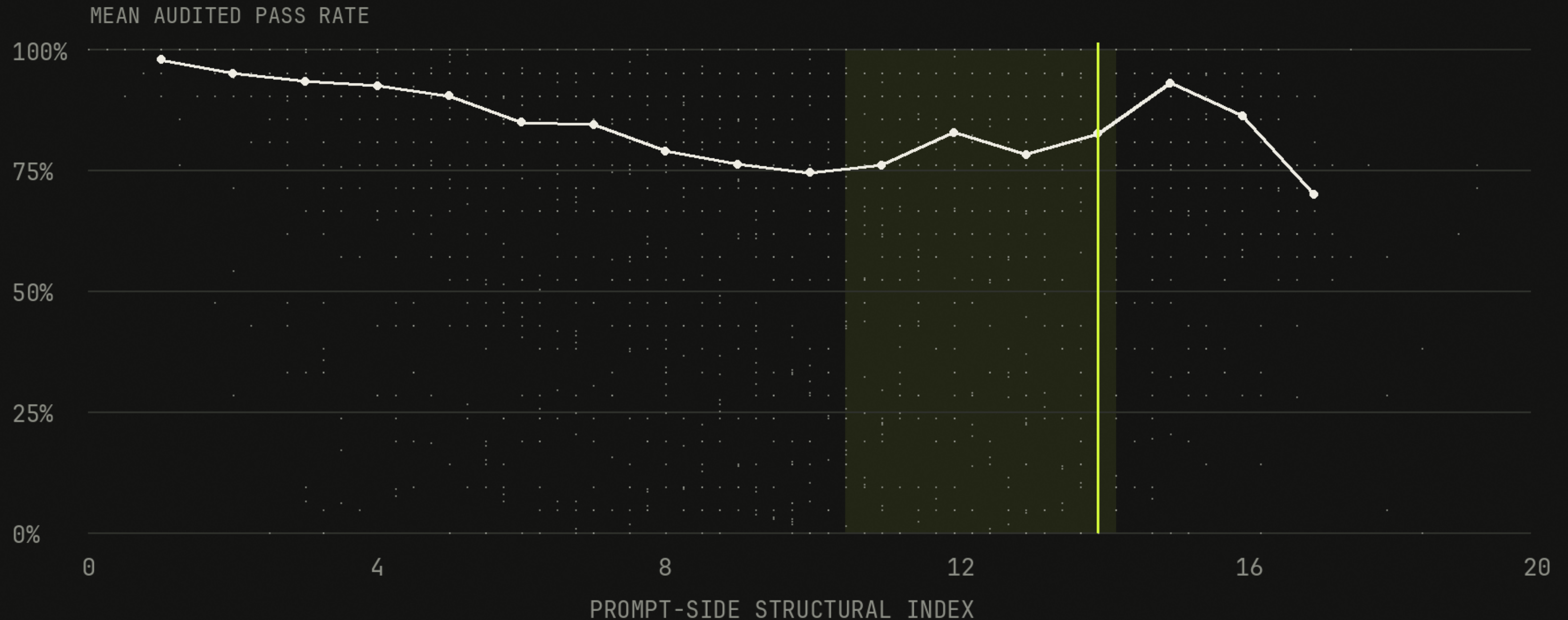
Passing complete cases: 12.2%

3,561/14,247 audited failures with computable output CC meet both conditions: prompt index >8 and output CC ≤10. The comparable share among passing complete cases is 12.2%. Counts refer to generations.

Pooled audited reliability

83.4% ≤ 14 90.1% > 14

Split 14.0



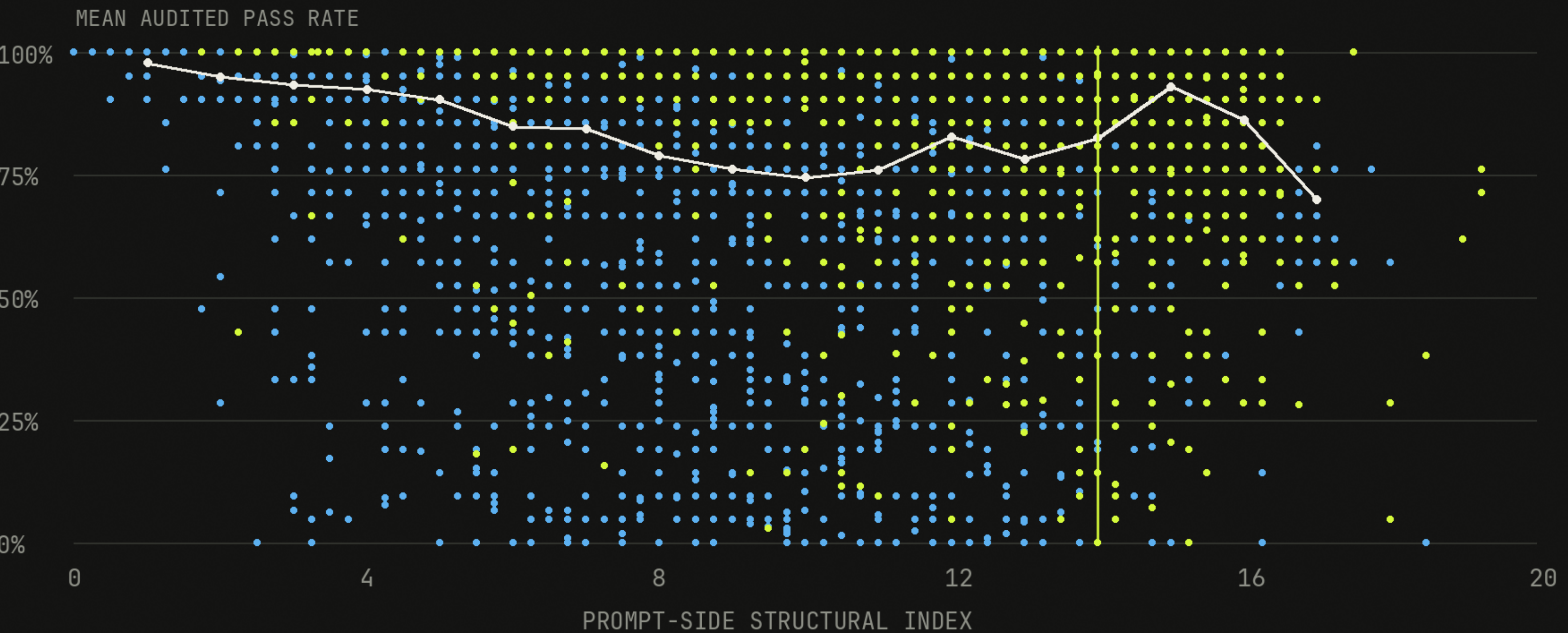
Each dot: one prompt mean across 21 models in a 5,000-prompt, selection-stratified Python benchmark. White line: binned means 1-17. 95% split-location interval [10.5, 14.25]. Bin 17: n=45.

● Algorithms / data structures

● Other task types

Algorithms / data structures share

≤ 14: 22.3% > 14: 90.9%



Algorithms / data-structures shares count prompts, not pass rates: 826/3,703 below or at 14, and 1,179/1,297 above. Each point is one prompt mean across 21 models. Points overlap. White line: binned means.

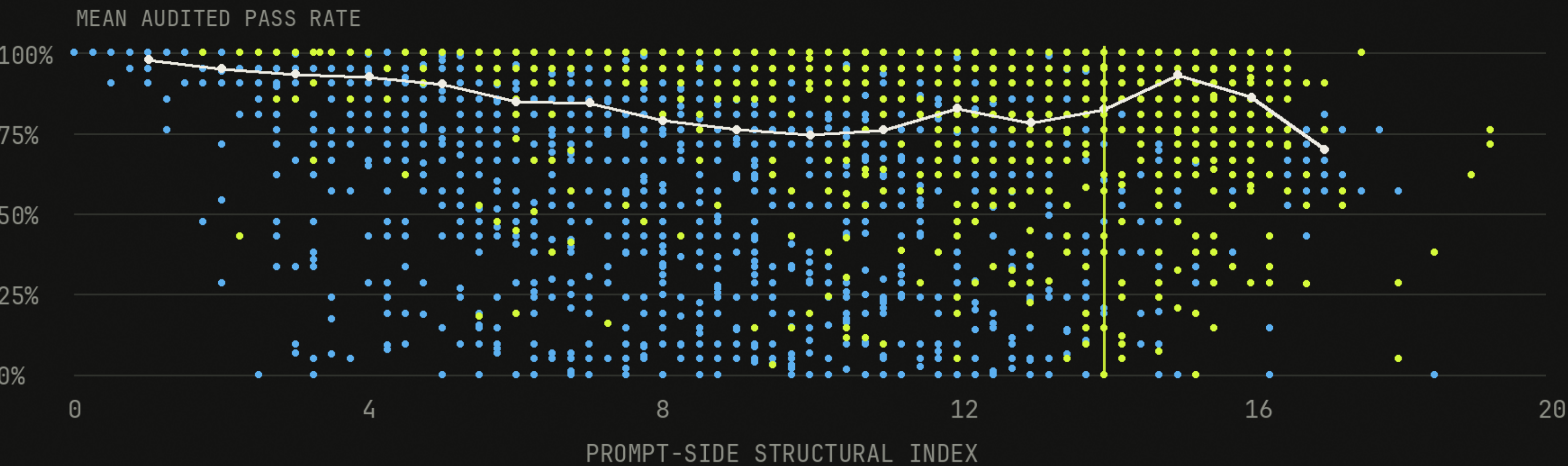
08

● Algorithms / data structures

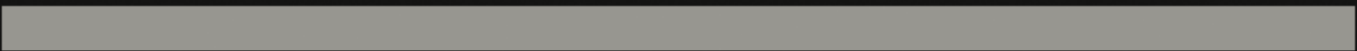
● Other task types

Algorithms / data structures share

≤ 14: 22.3% > 14: 90.9%



Unadjusted



+6.8 pp

Task adjusted

■ +0.4 pp

p = 0.66

Same split, 14.0: raw contrast +6.8 pp; additive task-adjusted contrast +0.4 pp. Adjusted contrast coefficient p=0.66. The separate piecewise search still detects a break.

Construction frames change the comparison

Same split: 14.0. Raw mean pass above minus at or below.

Pooled

+6.76

percentage points

Earlier frame

-0.05

percentage points

Later frame

+1.70

percentage points

Neither frame reproduces the large pooled rise. The earlier frame has only **60 prompts above 14.0**.

Descriptive contrasts. Construction frames also differ in test quality and generation runs.

Outcome scoring changes the pooled split

SAME 5,000 PROMPTS × 21 MODELS



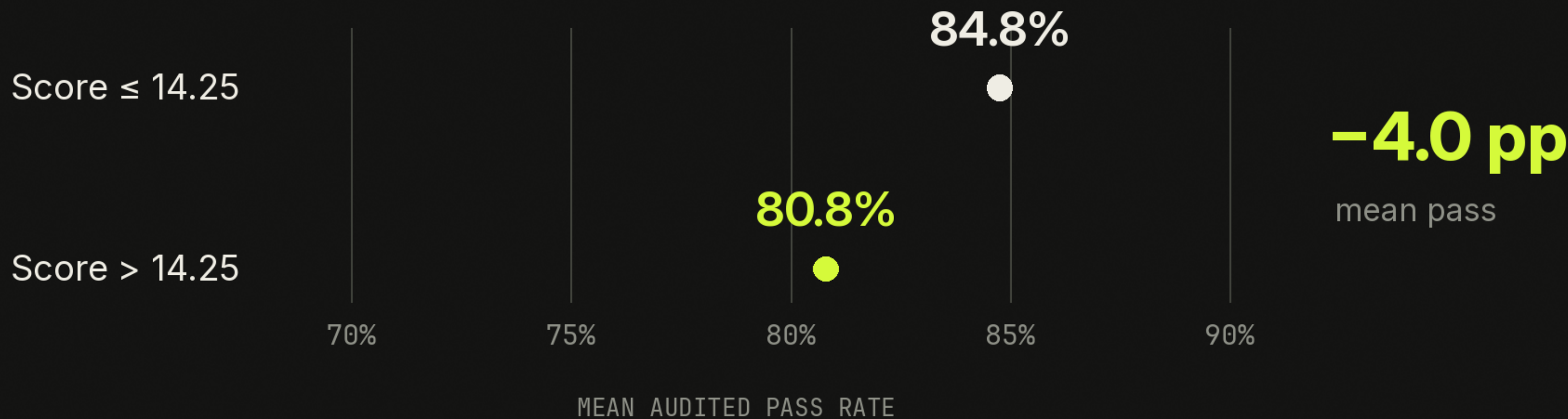
Even the breakpoint needs a methods section.

The same 5,000 prompts and 105,000 generation attempts, evaluated under three outcome definitions. The independent LLM audit is primary. Auditor judgments also carry uncertainty.

Kimi K2.5 in the later frame

KIMI K2.5 / LATER CONSTRUCTION FRAME

All three scoring rules select 14.25



Split interval: [14.00, 14.75]

Python only. Sparse extreme tail.

Post hoc, later frame only. Raw-test and partial-audit outcomes coincide in this frame. Sensitivity checks reuse the same prompts. The 95% interval concerns split location and reaches the upper search boundary.

Prompt-side measurement

Observational reliability diagnosis

The prompt index stays separate from evaluated answers.

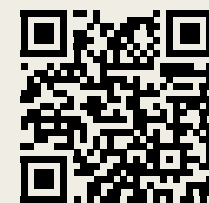
Task mix and outcome scoring shape pooled reliability patterns.

Model-specific findings retain their benchmark and protocol scope.

Tests reject the candidate instrumental-variable restrictions. These results do not identify a causal effect or a universal deployment cutoff.

Python-only, selection-stratified sample. Moderate, grader-dependent human alignment. Sparse extreme tail. Outcome-auditor uncertainty.

The assignment is still a tree.



Paper



Code + data