CLEAR IDEAS

CLEAR IDEAS RESEARCH

Clear Ideas AI Capability Index

Measuring frontier model performance on complex enterprise reasoning

A technical whitepaper on structured output reliability, stateful aggregation, workflow abstraction, and price/performance across frontier AI models.

Date: April 27, 2026

Prepared by: Clear Ideas Research

clearideas.com

Date: April 27, 2026 **Prepared by:** Clear Ideas Research

Executive Summary

The Clear Ideas AI Capability Index measures how well AI models can transform a compact, plain-language business request into a complete, structured, validated plan for work.

The test is deliberately practical. The model has to behave like an expert operator: infer intent, decompose the request, identify inputs, define intermediate variables, preserve state, sequence dependent steps, and return a structured output that can be inspected and validated.

The hard part is reasoning about values that do not exist yet. Future outputs become later inputs. Intermediate results need names, storage, and scope. Repeated work needs a collection point. The final answer has to assemble those pieces without losing the thread.

In enterprise settings, this is the difference between a model that can talk about work and a model that can design work.

The results form a distinct frontier band. Leading models complete this class of task reliably and produce outputs that are structurally valid, reasonably sophisticated, and ready for inspection. The clearest separator is stateful aggregation: collecting intermediate outputs, preserving them across dependent steps, and synthesizing them into a final result. Below the frontier band, failures show up as partial completion, broken structure, omitted variables, weak decomposition, or confused dependencies.

The index separates first-shot success, validation repair, malformed structured output, and genuine completion. A one-off demo can look competent while the same model remains unreliable as a generator of executable enterprise structure.

This capability has only recently become dependable. Prior model generations could often describe a workflow-like solution in prose, but they struggled to produce machine-readable, internally coherent, execution-ready structures consistently. The 2026 frontier marks a shift from fluent assistance toward reliable system design.

The Benchmark Task

The task is simple to state and hard to perform:

> *Given a compact business objective, produce a complete structured design for accomplishing it.*

Each prompt describes a business problem at a high level. A strong model must infer the operational shape of the task and produce a structured result that could plausibly be executed by a system. The benchmark is domain-varied, but the underlying challenge is consistent: convert ambiguous human intent into a coherent, machine-readable plan.

The expected output includes:

- a correct interpretation of the user's objective
- a decomposition into meaningful steps
- required inputs and variables
- intermediate outputs that are named and reused consistently
- appropriate dependencies between steps
- repeated or nested work where the task requires it

- a final output contract that satisfies the original intent
- structured data that survives automated validation

The core question is whether the model can build a durable abstraction from a small prompt.

What Constitutes A Good Result

A good result is an expert-like design expressed in validated structure.

1. Correctness

The output must solve the task that was actually requested. It should include the essential work required to reach the intended result, avoid irrelevant steps, and preserve the business meaning of the prompt.

Correctness includes coverage of the core requirements. If the request implies extraction, comparison, synthesis, review, aggregation, or recommendation, those operations should appear in the design in the right relationship to one another.

2. Structural Discipline

The output must be structured enough to validate automatically. That means valid fields, consistent object shapes, and references that point to values that actually exist.

Many weaker models can produce plausible prose. Fewer can preserve a consistent internal representation across a multi-step output.

Common structural failures include:

- referencing variables that were never defined
- producing outputs that are never consumed
- omitting required final outputs
- flattening nested work into vague single steps
- returning malformed or truncated structured data
- confusing user-provided inputs with model-generated intermediate values

3. State Management

The strongest outputs define state and carry it through the process.

State is the heart of the benchmark. The model has to understand that a value can be:

- supplied by the user
- generated by an earlier step
- reused by a later step
- aggregated across repeated work
- included in a final answer

This requires the model to reason about theoretical values before those values exist. That is a much harder task than answering a direct question.

4. Stateful Aggregation

The hardest scenarios require more than passing a single intermediate value forward. They require aggregation across repeated work.

A strong design can:

- iterate over multiple source items
- preserve per-item results
- aggregate those results into a collection
- compute or synthesize across the collection
- carry the aggregate into a final answer

Many partial models break here. They may define a loop, then leave the child steps empty. They may create per-item outputs without collecting them. They may create a collection, then forget to use it in the final synthesis. Those failures reveal whether the model has an internal representation of the work or a surface-level template.

5. Sophistication

A good result uses the right amount of structure: specific enough to execute, restrained enough to avoid unnecessary machinery.

Sophistication can show up through:

- well-named variables
- clear intermediate outputs
- appropriate use of iteration
- nested child steps when a task naturally requires repeated sub-work
- multiple step types where they add value
- output contracts that make downstream use easier
- tags or other metadata that make the design more reusable

Sophistication means disciplined design over decoration.

6. Autonomy

The best models complete the task on the first attempt. They produce usable structure without repair of malformed syntax, deletion of extraneous variables, or repeated prompting.

In production, this difference is decisive. Complete, validated designs can move downstream. Malformed, incomplete, or inconsistent outputs need corrective tooling before they can become work.

Clarifying questions are treated differently from correction attempts. A useful clarifying question can be good behavior when the prompt is genuinely underspecified. A correction attempt is different: the first output was malformed, incomplete, or invalid, and the system had to guide it back into a valid output shape. The benchmark tracks that distinction where the run data makes it available.

Why This Is A Hard Reasoning Problem

The task combines several forms of reasoning that are often evaluated separately.

Abstracting from sparse instruction

The prompts are intentionally compact. The model receives a business objective and must infer the missing implementation structure.

This tests whether the model can move from intent to design. The model has to ask, implicitly:

- What is the user really trying to accomplish?
- What information is required?
- What steps are naturally implied?
- Which outputs need to be preserved for later use?
- What should be included in the final result?

Decomposing the task

Real enterprise work is rarely a single action. It usually involves extraction, transformation, comparison, synthesis, review, and presentation. The model must find the task boundaries and sequence them.

A weak model may produce one broad step: "Analyze the documents and summarize the result."

A stronger model will separate input handling, extraction, intermediate analysis, synthesis, final formatting, and follow-up recommendations.

Maintaining dependencies

The model must track which step produces which value and where that value is needed later. In practice, this is a dependency graph problem expressed through language and structured data.

Failures here are easy to miss and costly to execute. A reference to a missing variable or an unused output shows that the model lost its internal map of the process.

Producing validated structure

The final output must be usable by software. That forces discipline. The model cannot rely on rhetorical fluency alone.

Many models separate at this boundary. They understand the task in general, then fail at the interface between reasoning and structured production.

Acting like an expert

The strongest results resemble the work of an experienced workflow architect. They infer the operating model behind the prompt and design a reusable process around it.

That is the test: whether a model can perform abstract professional labor, not just describe it.

Why This Benchmark Matters

Enterprise AI adoption is moving from chat to systems.

Chat helps, but organizations ultimately need repeatable work:

- structured analysis
- governed document workflows
- recurring reports
- review processes
- evidence-backed summaries
- validated outputs
- handoffs between people, documents, and systems

In that environment, the model must do more than produce a good answer. It must produce something that can become part of a process.

The Clear Ideas AI Capability Index focuses on this transition. It asks whether a model can create a structured representation of work from a simple instruction.

That capability sits at the center of the next phase of enterprise AI.

Benchmark Design Principles

The benchmark is built around practical enterprise use rather than isolated puzzle solving. It emphasizes outputs that can become part of repeatable work.

The design principles are:

- **Low saturation:** The scoring scale preserves headroom so future models can improve meaningfully above today's frontier.
- **Stateful aggregation:** The benchmark stresses whether models can collect intermediate values, preserve them across dependent steps, and synthesize them into final outputs.
- **Structured-output discipline:** Valid structure is necessary, but the benchmark also evaluates whether that structure is complete, internally coherent, and operationally valuable.
- **Correction-aware scoring:** A model receives more credit for first-shot valid completion than for an output that requires repair after malformed or incomplete structure.
- **Measured configurations:** Scores apply to the specific model configuration tested, including reasoning level and other execution settings where available.
- **Operational relevance:** Speed and price/performance are tracked because production model choice depends on raw capability and operating profile.

These principles matter because frontier models are likely to converge on many capability dimensions over time. As that happens, price/performance, latency, reliability, and configuration behavior may become the deciding factors for production use.

Task Corpus

The public benchmark uses a compact set of enterprise-reasoning scenarios. The scenarios are intentionally domain-varied, but they share a common requirement: transform a sparse business objective into a structured, machine-readable design.

The scenario set covers:

Scenario family	Capability stressed
Market reasoning	Comparing external signals, preserving evidence, and producing a structured recommendation
Contract analysis	Extracting obligations, risks, and decision points from governed business text
Stateful aggregation	Iterating over multiple items, preserving per-item values, and synthesizing an aggregate result
Action synthesis	Turning analysis into a sequenced plan with explicit dependencies
Research synthesis	Combining multiple strands of evidence into a coherent output contract
Financial reasoning	Following quantitative and qualitative constraints while producing a usable business conclusion

The public paper summarizes the scenario families while preserving prompt-level detail. The goal is to describe the capability being measured without turning the benchmark into a prompt-memorization exercise.

Scoring Model

The index is normalized and calibrated with headroom for future progress. A leading score in the 70s represents strong frontier performance on this task while leaving room for better future systems.

The overall score combines several dimensions.

Dimension	What it measures
Capability	General task understanding and solution quality
Reliability	First-shot completion, structural health, and resistance to validation failure
Stateful aggregation	Whether intermediate values are reused, collected, synthesized, and carried into final outputs
Sophistication	Useful complexity, variable design, child steps, step variety, and output contracts
Coverage	Whether the important parts of the prompt were addressed
Autonomy	Whether the model succeeded without additional help
Speed	Relative execution speed
Price/performance	Capability delivered relative to cost

The benchmark intentionally distinguishes validity from quality. A model can produce a structurally valid output and still score poorly if the design is shallow, incomplete, or weakly useful.

Evaluation Reliability

Evaluation combines deterministic validation with judged quality assessment.

Deterministic checks verify whether the output can be parsed, whether required structure is present, whether references are well formed, whether child steps are populated when expected, and whether final outputs are defined. These checks catch failures that are hard to see from prose alone.

Quality scoring uses multiple AI judges. The judges assess usefulness, completeness, decomposition quality, variable discipline, state management, and whether the result resembles expert workflow design.

The scoring workflow is designed for automated reruns as new models and model configurations appear. Human review remains useful as a calibration layer, especially when refining rubrics or investigating surprising results, while official scoring relies on repeatable validation and multi-judge assessment.

Model outputs are non-deterministic, and repeated runs can vary. In ad hoc repeat testing, model rankings and failure patterns have been directionally consistent, but broader repeated-run confidence intervals are a future enhancement. The current official scores should therefore be read as configuration-level benchmark measurements, not immutable properties of a model family.

Current Results

The public index includes 20 models and model configurations. Nineteen produced non-zero scores. Nine completed the full benchmark set. One failed across the full set.

All aggregate statistics in this section are calculated from the rounded public index values shown on the benchmark page. The sample definitions, formulas, and reconciliation tables are included in Appendix A: Statistical Analysis.

Overall leaderboard

Rank	Model	Overall	Capability	Stateful	Reliability	Sophistication	Price/perf.
1	GPT-5.5	79.1	72.3	78.2	92.6	86.3	61.1
2	Claude Opus 4.7	77.0	70.5	80.1	91.3	80.8	60.9
3	GPT-5.4 Mini	75.7	69.0	78.0	89.5	85.8	76.5
4	GPT-5.3 Codex	75.5	68.0	77.0	90.8	80.6	73.4
5	GPT-5.4	75.1	68.0	77.0	91.6	83.7	68.2
6	Grok 4.20	74.0	67.4	77.5	90.1	77.7	72.1
7	Grok Code Fast 1	68.9	64.0	75.2	73.2	78.4	76.1
8	GPT-5.4 Mini Low	68.2	62.4	76.1	77.1	79.6	73.3
9	Claude Haiku 4.5	67.1	62.4	78.9	71.0	74.6	72.4
10	GPT-5.4 Nano	64.1	58.3	63.8	76.3	67.6	68.2

Chart: Overall capability index

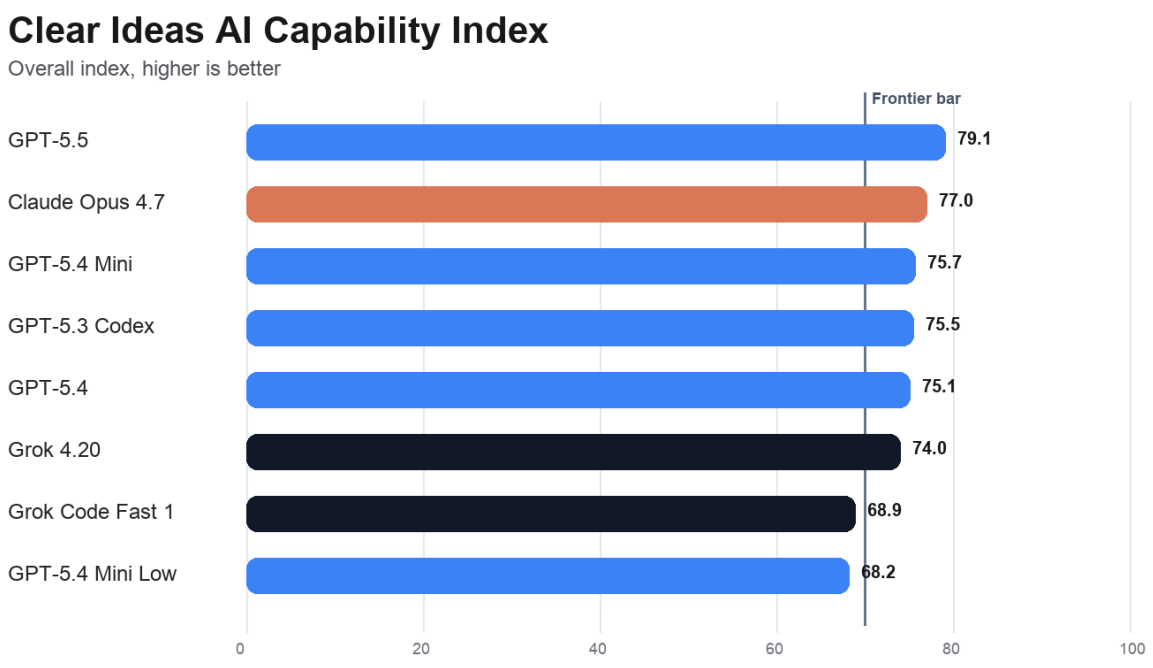

Figure 1. Overall capability index, higher is better.

Completion bands

Completion band	Model count	Average overall score
Full completion	9	73.4
Partial completion	10	37.9
Full failure	1	0.0

Completion bands do real work in the analysis. The strongest models score higher on subjective quality and are materially more likely to produce complete, usable, validated outputs. Once a model slips from full completion into partial completion, the average score falls sharply.

That gap is the operational threshold: completion quality matters most when structured outputs become part of repeatable work.

Relationships In The Data

The dimensions interact. Some move together strongly, while others reveal trade-offs.

Overall score is primarily a capability and structure signal

Across non-zero scoring models, overall score is highly correlated with the main quality and structure dimensions.

Relationship	Correlation with overall score
Capability	0.999
Sophistication	0.995
Judged quality	0.994
Reliability	0.988
Stateful aggregation	0.987
Structural discipline	0.987
Coverage	0.981
Autonomy	0.975
Price/performance	0.953
Speed	0.827

Overall score tracks capability most closely, which is expected. The more revealing relationships are stateful aggregation, sophistication, and structural discipline. High-scoring models create well-formed, internally coherent systems of work.

Speed matters only when paired with valid completion. A fast failure is still a failure, and a slow invalid output is worse.

Capability vs. price/performance

Across the full set of non-zero models, capability and price/performance show a strong positive relationship.

The correlation between capability and price/performance is approximately **0.95** across non-zero models. In broad terms, models that reason better also tend to deliver better effective value because failure is expensive. A low-cost model that fails or produces unusable structure carries hidden operational cost. The exact public-value correlation is **0.954**.

However, the frontier band tells a more nuanced story. Among models that complete the full benchmark, the relationship reverses. The correlation between capability and price/performance inside the full-completion group is **-0.662**.

Two patterns appear in the data:

- Across the whole market, capability improves price/performance because stronger models avoid failure.
- At the frontier, the most capable models often pay a premium for marginal capability gains.

This produces two trajectories:

- an upward capability/value trend as models move from unreliable to reliable
- a frontier premium where the highest capability models separate from the efficiency frontier

Chart: Capability vs. price/performance

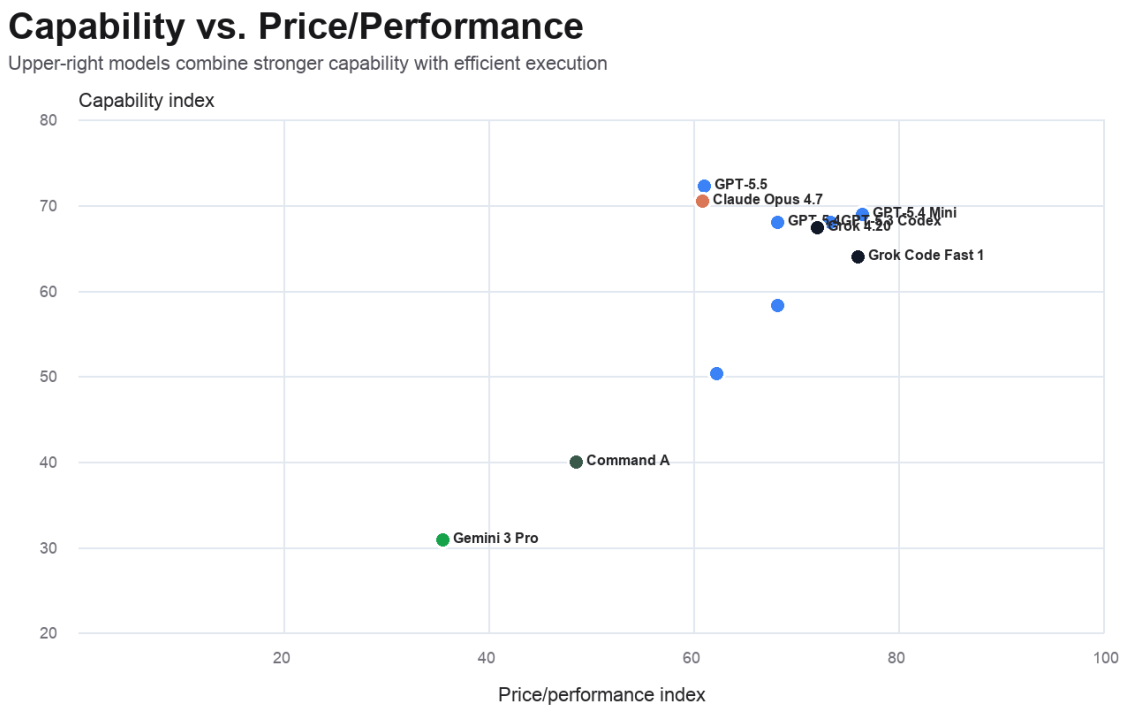

Figure 2. Capability vs. price/performance.

The relationship between capability and price/performance requires more than a single leaderboard.

GPT-5.5 leads the capability index, while GPT-5.4 Mini remains close to the capability frontier with a stronger price/performance profile. GPT-5.3 Codex and Grok 4.20 also sit in the upper portion of the value range, illustrating why production model selection should consider capability and operating efficiency separately.

Stateful aggregation is the core capability

The stateful aggregation score directly measures whether a model can design work instead of merely describing work.

Stateful aggregation has a **0.987** correlation with overall score across non-zero models. As models become operationally useful on this task, they almost always become better at preserving and reusing intermediate values.

Enterprise work frequently has this shape:

- collect evidence from multiple sources
- extract comparable fields
- preserve per-source observations
- compare or aggregate them
- synthesize a final result
- produce an auditable output

A model with weak state management may still answer direct questions well or produce a plausible outline, while struggling to create repeatable processes that software can execute.

Reliability, state, and sophistication move together

Reliability in this benchmark is contextual. A model receives more credit when it produces complete results that are also structurally healthy and sophisticated.

Naive reliability can mislead. A minimal valid object can be less valuable for enterprise work than a richer structure with occasional edge-case failures.

The benchmark therefore treats reliability as a blend of:

- completion
- first-shot success
- validation health
- structural consistency
- stateful aggregation
- sophistication of the output

That better reflects production reality. The goal is to produce the right thing in a form that can be used.

Quality vs. structure exposes misleading success

Some models can produce outputs that look structurally acceptable but receive low quality scores. This is an important failure mode.

For example, Command A completed much of the benchmark mechanically, but its judged output quality was low. That suggests the model could often satisfy the outer shape of the task while failing to design something truly useful.

The frontier requires both qualities at once:

- structure without quality is shallow automation
- quality without structure is hard to operationalize
- strong results combine both

Structured-output discipline exposes hidden fragility

Some models retain fluent surface behavior while failing the structured-output requirements needed for execution. The fast Grok 4.1 variants illustrate this pattern: their common failure mode was malformed or truncated structured payloads that could not become usable work.

Gemini 3 Pro and Gemini 3 Flash showed a related pattern. They could complete some simpler or flatter scenarios, but often failed loop-heavy tasks by emitting empty child-step structures, invalid loop configuration, or placeholders where executable steps were required.

Command A showed a different failure shape. It passed five scenarios structurally, but judged quality remained weak and provider errors appeared in more demanding cases. It illustrates why pass rate needs to be interpreted alongside quality, state management, and structural discipline.

GPT-OSS 120B on Groq remained notable as a low-cost partial performer: it passed five of six scenarios, but failed the hardest stateful aggregation scenario. GPT-OSS 20B remained a hard failure, repeatedly using output budget without returning usable final content.

Operational capability means reliably expressing the right idea in valid, complete, machine-usable form.

Reasoning configuration affects usable output

Reasoning effort is a configuration control surface. In structured generation tasks, too much hidden reasoning without enough final-output budget can reduce reliability.

The benchmark therefore treats reasoning settings as model configurations to be measured directly.

Why 2026 Is Different

Earlier model generations often failed before higher-level distinctions could be evaluated.

Prior models often showed one or more of these failure modes:

- malformed structured output
- inconsistent schemas
- references to values that did not exist
- confusion between inputs, outputs, and intermediate values
- poor handling of nested or repeated work
- inability to finish within the required format
- plausible prose wrapped around unusable structure

The frontier has changed. Several leading models reliably produce complete, structured, internally coherent designs. Quality differences can now be evaluated above the basic validity threshold.

That is a major shift.

The most capable systems are beginning to move from:

- answering questions
- summarizing content
- drafting text

to:

- designing processes
- preserving state
- generating reusable structures
- creating system-ready outputs

The Clear Ideas AI Capability Index tracks this progression from answer generation toward process design.

Interpretation Of Current Model Groups

Capability frontier

The frontier group contains the six models at or above the 70-point frontier bar, with an average overall score of 76.1. These models generally complete the benchmark, maintain structure, and produce executable designs.

GPT-5.5 leads the index at 79.1, 2.1 points ahead of Claude Opus 4.7. Its advantage combines raw task quality, strong reliability, sophistication, and coverage.

The next cluster includes Claude Opus 4.7, GPT-5.4 Mini, GPT-5.3 Codex, GPT-5.4, and Grok 4.20. These models are all practically capable on the benchmark, but their trade-offs differ.

Efficiency frontier

GPT-5.4 Mini, Grok Code Fast 1, GPT-5.3 Codex, GPT-5.4 Mini Low, and Grok 4.20 are strong on price/performance. They are especially relevant for production systems where repeated execution matters.

The efficiency frontier answers a different question from the capability frontier:

> *Which model gives the most useful capability for the cost profile?*

For many enterprise workflows, that may matter more.

This distinction should matter more as frontier capability converges. If several models can produce strong structured designs, the deciding factors shift toward price/performance, speed, reliability under load, and behavioral stability across settings. The index tracks capability and efficiency separately rather than collapsing them into one ranking.

Partial capability band

Models in the partial band can be useful while remaining less dependable for this class of autonomous structured design. They may complete easier scenarios while failing on tasks that require deeper state tracking, nested decomposition, or stricter output discipline.

These models are often good enough for assistance, drafting, or simple extraction. They are less suitable as independent designers of repeatable enterprise processes.

Below-bar models

Below-bar models are still informative. Their failures reveal where the benchmark is hard:

- structured output consistency
- variable discipline
- dependency tracking
- nested reasoning
- final output completeness

The benchmark preserves these results because failure shape matters. It helps distinguish near misses from models fundamentally misaligned with the task.

Methodological Notes

The public index intentionally reports normalized scores rather than raw execution mechanics.

This makes the benchmark easier to compare over time and avoids overfitting interpretation to any single run detail. The scoring scale is calibrated with headroom so future models can improve meaningfully above today's frontier.

The scoring framework emphasizes:

- expert-like task decomposition
- structured validity
- variable and output discipline
- use of appropriate intermediate state
- completeness of the final result
- autonomy on the first attempt
- practical speed and price/performance

The benchmark can be rerun as new models emerge. Historical scores can be maintained as official records, while scoring weights and public presentation can evolve carefully when they better reflect the task.

Limitations

The benchmark measures one important class of enterprise reasoning. It should be read as a focused operational benchmark rather than a universal intelligence test.

The following areas are outside the direct scope of this benchmark:

- long-context retrieval over very large corpora
- factuality in open-ended web research
- multimodal visual reasoning
- conversational helpfulness
- creative writing quality
- safety policy behavior

- bias, toxicity, or protected-class fairness
- privacy, data-governance, or access-control behavior
- jailbreak resistance or misuse resistance

Higher scores should be interpreted in context. Model selection remains task-dependent.

The benchmark is strongest when interpreted as a measure of structured enterprise reasoning: the ability to turn a compact instruction into a validated, reusable process.

Responsible AI evaluation remains a separate requirement. A model that scores well on this benchmark may still require additional safety, privacy, governance, and compliance review before deployment in a high-stakes environment.

Conclusion

The Clear Ideas AI Capability Index measures a frontier that matters for enterprise AI: whether models can turn simple business intent into structured, executable work.

This capability has become practical. Leading models can produce complete, validated, expert-like process designs from compact prompts. Prior generations reached that bar inconsistently.

There is still headroom in sophistication, robustness, efficiency, and expert judgment.

The next wave of AI progress will be about better answers and better systems: models that can preserve state, design processes, reason through dependencies, and produce outputs that can be trusted by software and people.

That is the frontier measured by the Clear Ideas AI Capability Index.

Appendix A: Statistical Analysis

This addendum supports the statistical claims made in the paper and reconciles them to the public benchmark values.

Source data and sample definitions

The statistical analysis uses the rounded public index values displayed on the Clear Ideas AI Capability Index page. Each row is a model configuration. A model tested at a different reasoning level is treated as a separate configuration when it appears in the public index.

Sample	Definition	Count
Full public sample	All public model configurations in the index	20
Non-zero sample	Configurations with overall score above 0	19
Full-completion sample	Configurations that completed all six scenarios	9
Partial-completion sample	Configurations that completed at least one, but fewer than six, scenarios	10
Full-failure sample	Configurations with zero completed scenarios	1
Frontier-bar sample	Configurations with overall score at or above 70	6

Descriptive statistics

Overall score distribution across the full public sample:

Statistic	Overall score
Mean	52.0
Median	61.8
Sample standard deviation	25.4
Minimum	0.0
Maximum	79.1

Completion-band averages:

Completion band	Count	Average overall score
Full completion	9	73.4
Partial completion	10	37.9
Full failure	1	0.0
Frontier bar, overall >= 70	6	76.1

The full-completion group scores 35.5 points higher than the partial-completion group on average. This supports the paper's claim that reliable completion is a major capability separator as well as a binary status marker.

Leader gap

Rank	Model	Overall
1	GPT-5.5	79.1
2	Claude Opus 4.7	77.0

GPT-5.5 leads Claude Opus 4.7 by 2.1 overall index points in the rounded public data.

Correlation method

Correlation claims use Pearson's correlation coefficient:

```
r = cov(X, Y) / (sd(X) * sd(Y))
```

Unless otherwise specified, correlations with overall score are calculated over the non-zero sample (`n = 19`). The full-failure row is excluded from these correlations because a zero across every dimension can artificially inflate relationships between dimensions. Completion-band averages include the full-failure row where appropriate because they describe outcome groups rather than dimension relationships.

Correlations with overall score

Dimension	Pearson r with overall score	Sample
Capability	0.999	Non-zero, n=19
Sophistication	0.995	Non-zero, n=19
Judged quality	0.994	Non-zero, n=19
Reliability	0.988	Non-zero, n=19
Stateful aggregation	0.987	Non-zero, n=19
Structural discipline	0.987	Non-zero, n=19
Coverage	0.981	Non-zero, n=19
Autonomy	0.975	Non-zero, n=19
Price/performance	0.953	Non-zero, n=19
Speed	0.827	Non-zero, n=19

These correlations support two central claims in the paper:

- Overall score is mostly a capability and quality signal, with speed and cost contributing as secondary operational dimensions.
- Stateful aggregation is one of the strongest measurable separators, with a correlation to overall score of 0.987.

Capability vs. price/performance

Relationship	Pearson r	Sample
Capability vs. price/performance	0.954	Non-zero, n=19
Capability vs. price/performance	-0.662	Full-completion, n=9

The positive relationship in the non-zero sample supports the broad-market claim that better reasoning often improves effective value because failed outputs are costly. The negative relationship inside the full-completion sample supports the frontier-premium claim: once models are reliable enough to complete the benchmark, marginal capability gains often come with lower price/performance.

Statistical limitations

The index is an operational benchmark with a small but carefully structured public sample. Correlations should therefore be read as descriptive statistics for the current benchmark set rather than universal laws about model performance.

The use of rounded public values can create small differences from internal calculations made on unrounded run data. The paper reports rounded public-value statistics so the claims are reproducible from the visible index.