International Journal of Advanced Trends in Engineering and Technology (IJATET)
International Peer Reviewed - Refereed Research Journal, Website: www.dvpublication.com
Impact Factor: 5.965, ISSN (Online): 2456 - 4664, Volume 10, Issue 2, July - December, 2025

AGENT ARCHITECTURE, LANGUAGE MODEL, AND TASK DIFFICULTY AS PREDICTORS OF COMPUTATIONAL REPRODUCIBILITY IN RESEARCH BENCHMARKS

Divya Vishwanath* & Paul Johnson Asamoah**
* Janardhan Rai Nagar Rajasthan Vidyapeeth, Udaipur, Rajasthan, India
** Power Life University College, Accra, Ghana, West Africa

Cite This Article: Divya Vishwanath & Paul Johnson Asamoah, "Agent Architecture, Language Model, and Task Difficulty as Predictors of Computational Reproducibility in Research Benchmarks", International Journal of Advanced Trends in Engineering and Technology, Volume 10, Issue 2, July - December, Page Number 203-211, 2025.
Copy Right: © DV Publication, 2025 (All Rights Reserved). This is an Open Access Article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium provided the original work is properly cited.
DOI: https://doi.org/10.5281/zenodo.24040374

Abstract:

This analysis examines how evaluation configurations shape observed computational reproducibility performance, informing international debates on autonomous research verification. Using 1,080 CORE Bench evaluation attempts across 45 papers recorded during August 2024, it reconstructs binary task success and calculates configuration specific percentages, pair wise contrasts and trial ranges. Overall, 30.83 percent of attempts succeed. CORE Agent exceeds AutoGPT within every corresponding model and difficulty category, with advantages ranging from 14.07 to 35.56 percentage points. GPT 4o exceeds GPT 4o mini across all corresponding comparisons, although model differences narrow on hard tasks. Hard task completion is lowest across configurations, including one tie at zero, indicating that performance advantages depend on operational demands. These descriptive patterns distinguish successful interpretation of supplied outputs from completion requiring greater execution independence without identifying causal mechanisms. The analysis refines capability assessment by treating reproducibility as conditional system performance and separating repeated attempts from independent research cases. Its contribution is an auditable reanalysis supporting configuration specific reporting, transparent scoring and workflow matched evaluation when research institutions assess automated verification systems.

Key Words: Computational Reproducibility; Autonomous Research Agents; Benchmark Auditing; Descriptive Analysis; Research Transparency.

1. Introduction:

An autonomous research agent can return an answer without successfully reproducing the computation that should support it. This distinction matters when researchers, journals and institutions consider using agents to verify scientific results. An evaluation must establish what the agent completed, under which computational conditions, and how success was calculated. Otherwise, a benchmark percentage can obscure differences between interpreting supplied outputs, executing configured code and independently preparing an execution environment. Computational reproducibility provides an appropriate starting point because it connects reported results to inspectable data, software and procedures. Peng, 2011, identifies this connection as an attainable standard for assessing computational research. The availability of research materials does not guarantee that another user or system can execute them successfully. Stodden et al., 2018, demonstrate the distinction between journal requirements for sharing computational materials and the practical reproducibility of published analyses. Trisovic et al., 2022, further document execution difficulties in shared research code. Together, these contributions establish that reproducibility depends on operational conditions as well as access. For autonomous agents, the problem includes selecting relevant instructions, managing software dependencies, executing commands and interpreting outputs. Failure at any required operation can prevent successful completion even when the underlying research materials are available.

Agent architecture, language model identity and task difficulty represent distinct dimensions of this operational problem. Architecture organizes the instructions, tools and checks surrounding the model. Language model identity specifies the system generating interpretations and actions within that architecture. Task difficulty distinguishes the assistance supplied for completing the evaluation. Resource rational analysis offers a useful conceptual lens: performance depends on selecting and executing operations under computational constraints. Lieder and Griffiths, 2020, develop this explanation for cognition; its application here concerns the allocation of limited computational resources rather than an assumption that artificial agents reproduce human cognitive processes. The resulting expectation is conditional performance, because the usefulness of a model or architecture may depend on the operations that a task requires. The closest empirical contribution is CORE Bench, introduced by Siegel et al., 2024. The benchmark contains tasks derived from 90 papers across computer science, social sciences and medical sciences, with separate training and test sets. Its three difficulty categories distinguish interpretation of supplied outputs, execution with configured computational assistance and execution requiring independent environment preparation. The original evaluation compares AutoGPT and CORE Agent using GPT 4o and GPT 4o mini, and already establishes substantial differences across architectures, models and difficulty categories. Consequently, reproducing those rankings alone would not constitute a distinct contribution. The relevant extension is to examine their reconstruction from released evaluation fields and to characterize the configuration differences without treating repeated attempts as independent papers.

Outcome definition is central to that extension. A successful benchmark task requires correct answers to the questions associated with a research capsule, but this does not independently establish the truth of the paper's substantive conclusions. Goodman et al., 2016, distinguish dimensions of research reproducibility that should not be conflated. This analysis therefore uses a narrow outcome: every recorded text and visual question must be answered correctly, and each evaluation must contain at least one question. Architecture, model and difficulty remain observed categories rather than latent capabilities. The analysis does not introduce a moderator or mediation pathway because those relationships were not estimated. This study aims to describe how computational reproducibility task success varies across agent architectures, language models and task difficulty categories using

a consistently reconstructed outcome. The central question is how observed completion percentages differ across these configurations when the same research papers recur in the evaluations. The objectives are to compare architectures within equivalent model and difficulty categories, compare models within equivalent architecture and difficulty categories, and examine difficulty differences within each architecture and model combination. These objectives support descriptive comparisons rather than causal attribution.

The empirical analysis uses 1,080 evaluation attempts involving 45 test papers, recorded from 8 August to 20 August 2024. Each observation identifies a paper capsule, architecture, model, difficulty category and trial. CORE Agent contributes three trials per configuration, while AutoGPT contributes one. The analytical strength lies in retaining these distinctions and calculating configuration specific percentages from the underlying answer counts. All required workbook fields are populated, and the observation keys contain no duplicates. A scoring audit also identifies a difference between success totals reconstructed from answer fields and those supplied in archive summaries. This makes the consistency of evaluation reporting an empirical concern alongside agent performance. The findings show higher observed completion percentages for CORE Agent than AutoGPT and for GPT 4o than GPT 4o mini within every corresponding comparison. Hard tasks have the lowest completion percentages, including one tie with medium tasks at zero. These patterns qualify a single score interpretation of agent capability by showing how performance depends on the evaluated configuration and supplied assistance. The contribution is an auditable descriptive reanalysis that separates operational task completion, repeated evaluation structure and score construction. Its practical relevance concerns how benchmark maintainers, researchers and journals report and interpret agent evaluations. Transferability rests on these evaluation principles rather than an assumption that the curated papers represent all disciplines, countries or research workflows.

2. Theory, Conceptual Framework and Research Questions Development:

This analysis treats computational reproducibility as a coordination problem between an agent, its computational environment and the information required to complete a research task. Success depends on connecting instructions, software execution and result interpretation into a functioning sequence. Peng, 2011, establishes computational reproducibility as an attainable standard through which researchers can inspect and repeat analyses using the original computational materials. This perspective directs attention to observable task completion rather than general claims about autonomous scientific capability. The theoretical foundation combines computational reproducibility with resource rational analysis. Lieder and Griffiths, 2020, explain how computational constraints shape the selection and execution of cognitive operations. Applied here as an analytical lens, this reasoning suggests that agent performance depends on how effectively a configuration allocates limited resources across searching, interpreting instructions and executing actions. The extension does not assume that agents reproduce human cognition. It identifies a shared constraint: additional operations consume resources and introduce opportunities for failure. Architecture, language model and task difficulty therefore represent distinct observed conditions under which successful completion can vary. Agent architecture concerns the organization of instructions, tools and output checks around the language model. A configuration that guides file inspection and verifies report structure may reduce avoidable omissions and direct attention toward relevant outputs. Wilson et al., 2014, connect reliable scientific computing with explicit workflows, automation and systematic checking. These principles provide a rationale for examining whether structured agent configurations display more consistent task completion. However, additional procedural guidance may also introduce unnecessary actions or fit some task formats better than others. Architecture is consequently interpreted as an observed configuration difference rather than an independently measured planning capability.

The available records distinguish CORE Agent from AutoGPT while retaining language model and difficulty identifiers. This permits descriptive comparisons between architectures within equivalent model and difficulty categories. Because architecture bundles prompting, tools and report checks, the analysis does not attribute any observed difference to one component of that bundle. Research question 1 asks: How do observed task success proportions differ between CORE Agent and AutoGPT within each language model and difficulty category?

Language model identity represents a second dimension of the evaluation configuration. Its conceptual role differs from architecture because it concerns the system generating interpretations and actions within the surrounding workflow. A model may identify relevant instructions accurately yet produce an incorrect command, or execute commands successfully yet misinterpret the resulting output. These possibilities make overall task success dependent on several connected operations rather than a single reasoning score. The benchmark developed by Siegel et al., 2024, evaluates both GPT 4o and GPT 4o mini within the two agent architectures. Its design permits model comparisons without treating model identity as a direct measurement of reasoning capacity, parameter count or computational efficiency. A shared expenditure ceiling also means that model comparisons reflect performance under a specified resource constraint rather than equal numbers of computational operations. Research question 2 asks: How do observed task success proportions differ between GPT 4o and GPT 4o mini within each architecture and difficulty category?

Task difficulty concerns the operational demands imposed by the materials supplied to the agent. Easy tasks provide outputs from a successful execution, medium tasks provide a configured route for executing the code, and Hard tasks require agents to establish the execution environment. The theoretical mechanism is the accumulation of dependent operations. When successful completion requires environment preparation, execution and interpretation, failure at an earlier operation can prevent subsequent operations from producing a correct answer. Sandve et al., 2013, identify software versions, computational procedures and the preservation of intermediate information as essential elements of reproducible workflows. Their argument supports distinguishing difficulty according to the assistance available for completing those operations. Nevertheless, supplying additional materials does not guarantee success because agents must still select and interpret them correctly. Difficulty is therefore represented by its three observed categories rather than an assumed numerical scale with equal intervals. Research question 3 asks: How do observed task success proportions vary across Easy, Medium and Hard tasks within each architecture and language model combination?

The conceptual framework links these three configuration dimensions to computational reproducibility task success. Each dimension is observed directly, and the outcome is a binary indicator requiring every recorded text and visual question to be

answered correctly, with at least one question present. Success proportions summarize this indicator within configuration categories. Stodden et al., 2018, demonstrate why access to research materials and the practical achievement of computational reproducibility require separate assessment. Accordingly, the framework evaluates completion under supplied benchmark conditions and does not equate correct benchmark answers with independent validation of the underlying scientific conclusions. The three research questions organize an exploratory descriptive analysis of 1,080 task runs involving 45 papers. Comparisons retain the architecture, model and difficulty combinations because pooled averages would conceal differences in task composition and repetition. Repeated runs remain observations of the same papers rather than independent research cases. The empirical comparisons concern evaluations conducted in 2024 within the eligibility period from 2010 to 2024. The framework therefore supports configuration comparisons without implying historical trends or causal effects.

3. Data:

This analysis establishes the provenance, coverage and construction of the benchmark records used to compare agent configurations. It defines the observed measures, documents sample selection and verifies the calculations supporting descriptive comparisons.

3.1 Data Sources, Population, and Sample Construction:

The primary source is the CORE Bench agent results archive maintained by researchers at Princeton University. Siegel et al., 2024, describe the benchmark, its evaluation procedures and the distinction between training and test papers. The supplied workbook consolidates observations from the original baseline configurations using CORE Agent and AutoGPT with GPT 4o and GPT 4o mini. These configurations were evaluated under a nominal expenditure ceiling of USD 4 per task. The target population comprises computational research tasks that autonomous agents attempt to reproduce using supplied research materials. The accessible sampling frame is narrower: curated CORE Bench capsules associated with reproducible papers in computer science, social sciences and medical sciences. The retained sample contains 18 computer science papers, 14 social science papers and 13 medical science papers. It includes 22 Python capsules and 23 R capsules. Country identifiers are absent, so geographical coverage is not classified. The observation unit is a paper capsule evaluated under a specific architecture, language model, difficulty category and trial. The data constitute a repeated benchmark evaluation structure rather than a calendar panel. Evaluations occurred from 8 August to 20 August 2024, within the eligibility period from 2010 to 2024. Each observation records a completed evaluation attempt rather than a flow, stock or annual average.

Table 1: Sample construction

Stage	Distinct Papers Remaining	Evaluation Files Remaining	Task Records Remaining	Inclusion or Exclusion Rule
Source archive containing evaluation results	90	47	2,046	Files containing capsule evaluation records
Training evaluations excluded	45	32	1,371	Remove 15 training files containing 675 records
Other test configurations excluded	45	26	1,170	Remove six files containing 201 records outside the four selected baseline configurations
Baseline evaluations outside the observation window excluded	45	24	1,080	Remove two files containing 90 records outside the permitted period
Consolidated analytical sample	45	24	1,080	Retain all eligible observations
Descriptive comparison sample	45	24	1,080	No additional record losses

The sample contains every combination of the 45 papers, two architectures, two models and three difficulty categories. Repetition differs by architecture: CORE Agent contributes three trials per combination, while AutoGPT contributes one. Accordingly, the configuration structure is complete but trial replication is unequal. The workbook preserves archive member identifiers and cryptographic hashes, providing traceability consistent with the data stewardship principles articulated by Wilkinson et al., 2016. Its extraction date and repository commit are not recorded, so archive identity is established through the stored hash rather than an asserted immutable release. The dataset card specifies an MIT licence; rights associated with underlying capsule materials require separate attention. This analysis reuses public evaluation records without collecting new participant data. Attribution and preservation of source identifiers support transparent reuse, as discussed by Nosek et al., 2015. Public availability nevertheless defines an accessible research resource rather than a representative sample of all computational science, consistent with the sharing constraints documented by Tenopir et al., 2011. The contribution is a traceable secondary analysis, without assuming the citation benefits associated with data sharing documented by Piwowar and Vision, 2013.

3.2 Data Integration, Cleaning, and Quality Assurance:

The construction process concatenated capsule records from 24 eligible evaluation files. It did not merge independent datasets or match observations to external sources. Architecture, model and difficulty labels were obtained from the evaluation configuration and filename, while capsule identifiers, disciplinary fields, programming languages and answer counts were retained from the source records. This produces the observation structure described by Wickham, 2014, with one evaluation attempt per row and one recorded attribute per column. The observation key combines capsule identifier, architecture, language model, difficulty and trial. An audit found no duplicate keys among the 1,080 retained records. Repeated capsule identifiers represent evaluations under different configurations or trials rather than duplicate observations. Configuration summaries group these records without expanding their number, following the grouping logic described by Wickham, 2011. All 18 workbook fields are populated in every retained record. Missingness is therefore zero across the recorded variables, papers and evaluation dates. Zero text question counts occur in 648 records, while zero visual question counts occur in 264 records. These values indicate that the corresponding modality was absent from the task. They are not missing responses, and every record contains at least one question

across the two modalities. No deletion, imputation or interpolation was required. This distinction between missing information and substantively meaningful values is consistent with the treatment principles discussed by Hughes et al., 2019.

The audit checked that correct answer counts were nonnegative and did not exceed the corresponding question totals. All records satisfied these conditions. Recalculation also reproduced every stored binary task score. No observations were trimmed or winsorized because the outcome is bounded and the answer counts passed their validity checks. The observed question totals range from one to eight, and these differences were retained as task characteristics. Identifiers and run dates were inspected directly rather than relying on their displayed spreadsheet appearance. This separation between stored values and presentation follows the organizational principles discussed by Broman and Woo, 2018. It also addresses the general risk of spreadsheet conversion errors demonstrated by Ziemann et al., 2016, without implying that gene identifier errors occurred in these data. The archive summary fields report 356 successful tasks, whereas the question count rule produces 333. Both quantities remain identifiable, but descriptive calculations use the reproducible question count rule applied uniformly to every record. No observations or scores were changed to force agreement with the archive summaries. The archive represents a retrieved historical results collection, not a preserved sequence of releases available at each evaluation date.

3.3 Variable Measurement, Construction, and Validation:

The outcome is computational reproducibility task success. It records whether an agent answered every text and visual question correctly in an evaluation containing at least one question. This operational definition measures successful benchmark completion. It does not independently establish the validity of the underlying scientific claim. That distinction follows the separation between computational procedures and broader scientific reproducibility discussed by Peng, 2011, and Goodman et al., 2016. Equation 1 defines the implemented scoring rule.

$$Y_i = 1\{C_{ti} = T_{ti} \ \wedge \ C_{vi} = T_{vi} \ \wedge \ T_{ti} + T_{vi} > 0\}$$

Here, Y_i denotes task success for evaluation i, C_{ti} and T_{ti} denote correct text answers and total text questions; C_{vi} and T_{vi} denote correct visual answers and total visual questions. The indicator equals one when every condition holds and zero otherwise. Architecture, language model and task difficulty are categorical observed measures. They are not latent constructs or components of a weighted index. Their source labels remain intact in the workbook. Difficulty is not assigned an equally spaced numerical score because the categories differ in supplied assistance rather than measured increments of complexity. Documentation of computational inputs and procedures is consistent with the reproducibility requirements described by Sandve et al., 2013.

Table 2: Variable dictionary

Variable or Indicator	Analytical Role	Definition and Construction	Source and Identifier	Unit and Frequency	Transformation and Direction	Missing Value Rule	Coverage and Limitation
Task success	Outcome	All recorded answers correct and a positive question total	Correct and total answer fields; workbook score formula	Binary value per evaluation	One indicates successful completion	All records complete	1,080 evaluations; benchmark completion only
Agent architecture	Configuration dimension	CORE Agent or AutoGPT	Evaluation configuration and archive member	Category per evaluation	No numerical ordering	All records complete	810 CORE Agent and 270 AutoGPT records
Language model	Configuration dimension	GPT 4o or GPT 4o mini	Evaluation configuration and archive member	Category per evaluation	No numerical ordering	All records complete	540 records per model
Task difficulty	Configuration dimension	Easy, Medium or Hard	Evaluation filename	Category per evaluation	Categories retained without equal spacing	All records complete	360 records per category
Capsule identifier	Repeated observation identifier	Original paper capsule identifier	Capsule identifier field	Identifier per evaluation	No transformation	All records complete	45 distinct papers
Trial	Replication identifier	Sequential run index within configuration and difficulty	Sorted source filenames	Integer per evaluation	Not interpreted as capability	All records complete	Three trials for CORE Agent and one for AutoGPT
Discipline	Sample descriptor	Computer science, social sciences or medical sciences	Field attribute	Category per capsule	No transformation	All records complete	Three disciplinary groups
Programming language	Sample descriptor	Python or R	Language attribute	Category per capsule	No transformation	All records complete	22 Python and 23 R capsules
Question counts	Scoring inputs	Correct and total questions by modality	Four answer count fields	Counts per evaluation	Raw counts retained	Modality absence remains zero	Positive combined total in every record
Run date	Timing descriptor	Date parsed from evaluation filename	Archive member filename	Date per evaluation	No temporal aggregation	All records complete	8 to 20 August 2024

Validation focused on field completeness, count bounds, key uniqueness and exact agreement between recalculated and stored task scores. It did not apply factor analysis or internal consistency statistics to categorical configurations. Stodden et al., 2018, demonstrate why available computational materials and successful reproduction require separate assessment; the National Academies of Sciences, Engineering, and Medicine, 2019, likewise distinguish computational reproducibility from replication using new data. These distinctions delimit the interpretation of the outcome.

Table 3: Descriptive statistics for numerical measures

Measure	Valid Records	Mean	Sample Standard Deviation	Median	Minimum	Maximum
Binary task success	1,080	0.3083	0.4620	0	0	1
Total questions per evaluation	1,080	1.7556	1.1192	2	1	8

The overall success percentage is 30.83. This is an observation weighted description of the retained records, whose architectural trial counts differ.

3.4 Analytical Framework, Identification, and Reproducibility:

The analysis is exploratory and descriptive. It compares task success percentages across architecture, model and difficulty combinations. It does not estimate causal effects or treat the available records as a random sample of scientific research. The questions were developed after the benchmark results were available and therefore do not constitute preregistered confirmatory tests. This distinction follows Nosek et al., 2018, who separate hypothesis generation from hypothesis testing. Equation 2 defines the percentage calculated for each configuration group.

$$P_g = 100 \frac{\sum_{i \in g} Y_i}{n_g}$$

Here, P_g denotes the success percentage in configuration group g, Y_i denotes the binary score defined in Equation 1, and n_g denotes the number of retained evaluations in that group. Each group is defined by one architecture, one language model and one difficulty category. Denominators are 135 for CORE Agent groups and 45 for AutoGPT groups. Comparisons retain all 12 configuration groups rather than relying on pooled architecture or model averages. The same 45 papers appear across these groups, while multiple trials remain repeated evaluations of those papers. Reporting papers separately from evaluation attempts avoids the inflation of independent sample size described by Lazic, 2010. No regression coefficients, significance tests, confidence intervals or predictive validation results are reported. Differences are expressed as observed percentages and percentage points. Their magnitude describes the retained benchmark evaluations without establishing a population effect, consistent with the distinction between substantive interpretation and statistical significance emphasized by Wasserstein and Lazar, 2016.

The implemented quality checks retain the full eligible sample. No alternative eligibility specification or outlier adjustment was selected after inspecting comparative performance. Reporting the scoring rule and source summary discrepancy makes the analytical choices explicit rather than selecting whichever definition produces a preferred result. This disclosure addresses analytical flexibility concerns identified by Simmons et al., 2011. Reproducibility rests on the retained source paths, archive and member hashes, field definitions and editable workbook formulas. These materials permit reconstruction of the binary outcome and configuration percentages from the supplied records. Their preservation supports the transparency principles discussed by Munafò et al., 2017. The resulting comparisons remain bounded by the curated papers, selected configurations and observed evaluation window.

4. Methods:

4.1 Research Design and Comparative Strategy:

This analysis uses an exploratory descriptive design to examine differences in computational reproducibility task success across recorded evaluation configurations. The quantities of interest are success percentages within each configuration and percentage point differences between comparable configurations. The unit of analysis is one evaluation attempt identified by research capsule, agent architecture, language model, difficulty category and trial. The comparative strategy retains the same 45 research capsules across configurations. Architecture comparisons hold model identity and difficulty category constant. Model comparisons hold architecture and difficulty constant. Difficulty comparisons retain the same architecture and model. These restrictions improve descriptive comparability by avoiding comparisons between groups containing different papers.

The design examines observed performance rather than treatment effects. Configuration differences combine several operational features, and the records do not isolate their individual contributions. Repeated attempts also share research materials and task questions. They therefore provide evidence about performance variation within the benchmark without constituting independent scientific cases. This distinction addresses the pseudoreplication problem described by Lazic, 2010. The analysis concerns evaluations conducted during August 2024. It does not estimate temporal trends, forecast future performance or attribute observed differences to causal mechanisms. The research questions guide descriptive comparisons developed after the evaluation results were available, rather than prespecified confirmatory tests. This separation between exploration and confirmation follows Nosek et al., 2018.

4.2 Outcome Construction and Comparison Measures:

The outcome follows Equation 1: an evaluation succeeds only when every recorded text and visual question is answered correctly and the combined question total is positive. The construction uses the source answer counts rather than the aggregate success fields. A modality containing no questions does not constitute missing information and does not independently determine success. The configuration percentage follows Equation 2, dividing successful evaluations by the number of retained attempts in the corresponding architecture, model and difficulty group. Every attempt receives equal weight within its group. Because each CORE Agent group contains three evaluations per paper and each AutoGPT group contains one, papers receive equal representation within each configuration percentage. Pairwise comparisons subtract the percentage for a specified comparison configuration from that for the focal configuration. Equation 3 defines the contrast.

$$D_{ab} = P_a - P_b$$

Here, D_{ab} denotes the percentage point difference between configurations a and b, while P_a and P_b denote their success percentages calculated using Equation 2. A positive value indicates higher observed completion in the focal configuration. Architecture contrasts designate CORE Agent as the focal configuration, and model contrasts designate GPT 4o as the focal configuration. Difficulty remains categorical. Comparisons examine Hard against Easy and Medium separately because the categories represent different supplied materials and execution requirements rather than equally spaced numerical increments. The benchmark distinguishes interpretation of existing outputs, execution with configured assistance and execution requiring environment preparation, as documented by Siegel et al., 2024. The numerical summaries comprise valid counts, means, sample standard deviations, medians and observed ranges. Categorical characteristics are summarized using counts. Calculations retain

full precision, with percentages and percentage point differences rounded to two decimal places for presentation. The method does not fit a regression model or construct latent variables, composite indices, interactions or adjusted effects.

4.3 Sample Alignment and Analytical Procedures:

The analysis uses the eligibility rules documented in Table 1 and the measurement definitions in Table 2. All configuration comparisons retain the complete analytical sample of 1,080 attempts involving 45 papers. No additional restrictions arise from missing covariates, lag construction or estimator requirements. Processing first identifies the eligible evaluation files and retains their capsule records. It then assigns architecture, model, difficulty and trial identifiers, preserves the original answer counts and calculates the binary outcome. Grouping by architecture, model and difficulty produces 12 configuration percentages. The percentage point contrasts are calculated from these unrounded percentages. The procedure also calculates success percentages separately for each of the three CORE Agent trials. The minimum and maximum trial percentages summarize observed variation around the configuration average. These ranges describe the available repetitions and are not confidence intervals. Trial labels identify runs within configurations rather than matched random seeds across configurations. The workbook supplies editable formulas for outcome construction and configuration summaries. Independent checks read the spreadsheet fields directly using Python standard library tools for archive access, XML parsing, numerical summaries and grouping. No stochastic estimation, optimization routine or random sampling enters the reported calculations. No imputation, interpolation, trimming or winsorization is applied. All scoring inputs are present, and zero question counts are retained when a modality is absent. This treatment distinguishes substantive zeros from unavailable observations, consistent with the missing data considerations discussed by Hughes et al., 2019.

4.4 Validation, Reproducibility and Interpretive Limits:

Validation checks address the integrity of the implemented descriptive calculations. The observation key combines capsule identifier, architecture, model, difficulty and trial. Uniqueness checks distinguish genuine repetitions from duplicated records. Count checks require nonnegative answer totals and correct answer counts no greater than their corresponding question totals. Every retained evaluation must contain at least one question. The binary outcome is recalculated independently from its component fields and compared with the stored workbook score. Configuration totals are then checked against the retained record counts. A separate audit compares reconstructed successes with the archive summary fields. The descriptive analysis consistently uses the field-based scoring rule; it does not modify observations to reproduce the alternative source totals. The interpretation distinguishes computational completion from scientific validity. Correct benchmark answers establish completion under the specified evaluation conditions, while independent confirmation of substantive research conclusions requires additional evidence. Goodman et al., 2016, explain why these dimensions of reproducibility require separate treatment.

The reported percentages characterize the retained benchmark records. Repeated evaluations, unequal architectural replication and curated paper selection limit extrapolation beyond those records. No standard errors, significance tests or confidence intervals are attached to the descriptive contrasts. Observed magnitude therefore remains separate from inferential claims, consistent with Wasserstein and Lazar, 2016. Replication materials comprise the source archive identifiers, archive member paths and hashes, measurement dictionary and workbook formulas. Together, these materials permit reconstruction of the outcome, configuration percentages and reported contrasts. The methodological contribution is transparent application of established descriptive procedures rather than a new estimator or causal identification design.

5. Findings:

This analysis compares computational reproducibility task success across agent architectures, language models and task difficulty categories. The findings answer the three research questions through configuration specific success percentages and percentage point differences. Interpretation remains descriptive and accounts for repeated evaluations of the same research papers. The analytical sample contains 1,080 evaluation attempts involving 45 papers, with observations recorded from 8 to 20 August 2024. Computer science contributes 18 papers, social sciences 14 and medical sciences 13. The sample includes 22 Python capsules and 23 R capsules. All retained records contain the fields required to calculate task success, and the observation key identifies no duplicate evaluations. CORE Agent contributes three trials per configuration and difficulty category, whereas AutoGPT contributes one. Consequently, the 1,080 evaluations represent repeated observations of 45 papers rather than independent research cases. This distinction prevents the interpretation of replication counts as an increase in independent sample size, consistent with Lazic, 2010. Task success equals one when every recorded text and visual question is answered correctly and the evaluation contains at least one question. Recalculation reproduces all stored task scores. Across the retained observations, 333 evaluations succeed and 747 fail, producing an overall success percentage of 30.83. The binary outcome has a sample standard deviation of 0.462 and a median of zero. This overall percentage describes the observed evaluation mix, which contains more CORE Agent trials than AutoGPT trials. Configuration comparisons therefore provide the primary evidence.

Table 4: Computational reproducibility task success by evaluation configuration

Agent Architecture	Language Model	Difficulty	Evaluations	Successful Evaluations	Success Percentage
CORE Agent	GPT 4o	Easy	135	79	58.52
CORE Agent	GPT 4o	Medium	135	75	55.56
CORE Agent	GPT 4o	Hard	135	26	19.26
CORE Agent	GPT 4o mini	Easy	135	57	42.22
CORE Agent	GPT 4o mini	Medium	135	41	30.37
CORE Agent	GPT 4o mini	Hard	135	19	14.07
AutoGPT	GPT 4o	Easy	45	15	33.33
AutoGPT	GPT 4o	Medium	45	16	35.56
AutoGPT	GPT 4o	Hard	45	2	4.44

Agent Architecture	Language Model	Difficulty	Evaluations	Successful Evaluations	Success Percentage
AutoGPT	GPT 4o mini	Easy	45	3	6.67
AutoGPT	GPT 4o mini	Medium	45	0	0.00
AutoGPT	GPT 4o mini	Hard	45	0	0.00

Research question 1 concerns differences between agent architectures within equivalent model and difficulty categories. CORE Agent records a higher success percentage in all six comparisons. With GPT 4o, its advantage is 25.19 percentage points on Easy tasks, 20.00 on medium tasks and 14.81 on Hard tasks. With GPT 4o mini, the corresponding advantages are 35.56, 30.37 and 14.07 percentage points. Architecture differences are therefore consistently positive in the retained sample, although their magnitude depends on the accompanying model and task category. The architecture advantage is largest for GPT 4o mini on Easy tasks and smallest for that model on Hard tasks. Structured configuration differences coincide with improved completion across the benchmark, but their observed advantage narrows when agents must establish the execution environment. The records do not isolate the contributions of prompting, tools or report checks. The result therefore concerns the architecture bundle, consistent with the bundled modifications documented by Siegel et al., 2024. Research question 2 concerns language model differences within each architecture and difficulty category. GPT 4o records higher success percentages than GPT 4o mini in all six comparisons. Within CORE Agent, the advantages are 16.30 percentage points on Easy tasks, 25.19 on Medium tasks and 5.19 on Hard tasks. Within AutoGPT, they are 26.67, 35.56 and 4.44 percentage points. The largest model differences occur on Medium tasks under both architectures.

The smaller model differences on Hard tasks reflect low completion percentages across both models rather than equivalent performance at a high level. CORE Agent reaches 19.26 percent with GPT 4o and 14.07 percent with GPT 4o mini, while AutoGPT reaches 4.44 percent and zero, respectively. These results identify model specific performance differences under the recorded evaluation conditions. They do not separately measure reasoning ability, computational efficiency or resource allocation. Research question 3 concerns variation across difficulty categories. Hard tasks have lower success percentages than Easy tasks in all four architecture and model combinations. The reduction is 39.26 percentage points for CORE Agent with GPT 4o, 28.15 for CORE Agent with GPT 4o mini, 28.89 for AutoGPT with GPT 4o and 6.67 for AutoGPT with GPT 4o mini. Relative to Medium tasks, Hard task success is lower in three combinations and unchanged at zero for AutoGPT with GPT 4o mini.

Table 5: Differences in success percentages across configurations

Comparison	Easy	Medium	Hard
CORE Agent advantage over AutoGPT with GPT 4o	25.19	20.00	14.81
CORE Agent advantage over AutoGPT with GPT 4o mini	35.56	30.37	14.07
GPT 4o advantage over GPT 4o mini within CORE Agent	16.30	25.19	5.19
GPT 4o advantage over GPT 4o mini within AutoGPT	26.67	35.56	4.44

The difficulty pattern is not uniformly declining between Easy and Medium. AutoGPT with GPT 4o increases from 33.33 percent to 35.56 percent, equivalent to one additional successful paper. CORE Agent with GPT 4o declines by 2.96 percentage points, whereas CORE Agent with GPT 4o mini declines by 11.85. These differences show why the difficulty categories should retain their operational meanings rather than be interpreted as equal increments on a numerical scale. Easy tasks assess interpretation of supplied outputs, while Medium and Hard tasks add execution requirements. Repeated CORE Agent trials reveal variation around the configuration averages. With GPT 4o, trial success percentages range from 57.78 to 60.00 on Easy tasks, 53.33 to 57.78 on Medium tasks and 17.78 to 20.00 on Hard tasks. With GPT 4o mini, the corresponding ranges are 35.56 to 48.89, 24.44 to 35.56 and 13.33 to 15.56. GPT 4o exceeds GPT 4o mini even when its lowest observed trial percentage is compared with the latter's highest percentage within each difficulty category. These ranges describe observed trial variation and are not confidence intervals.

The scoring audit identifies a distinction between reproducible field calculations and archive summaries. Summary fields total 356 successes, while the uniform question count rule produces 333, a difference of 23 evaluations. The reported findings consistently use the latter rule, preserving a traceable calculation rather than combining scoring definitions. Explicit construction rules support the computational transparency emphasized by Sandve et al., 2013. The evidence answers the research questions descriptively: CORE Agent outperforms AutoGPT within every recorded comparison, GPT 4o outperforms GPT 4o mini within every comparison, and Hard tasks display the lowest success percentages, including one tie at zero. These observed magnitudes do not establish statistical significance or causal effects, consistent with the interpretive distinctions emphasized by Wasserstein and Lazar, 2016. The outcome remains benchmark task completion rather than independent confirmation of scientific conclusions, following the distinction articulated by Goodman et al., 2016.

6. Discussion:

Successful result extraction does not imply successful autonomous reproduction. Tables 4 and 5 show that architecture and language model advantages persist across the recorded configurations, yet completion remains substantially lower on Hard tasks. Equation 1 defines success consistently, while Equation 2 makes these differences comparable within configuration groups. The central interpretation is therefore conditional: stronger observed performance depends on the combination of agent configuration and operational demands. The contribution of this reanalysis is its explicit separation of configuration differences and scoring definitions, rather than the discovery of an architecture ranking not previously reported. Collberg and Proebsting, 2016, position repeatability as a practical systems problem; the present comparisons extend that perspective to the interaction between automated execution and benchmark assistance. They shift attention from a single capability score toward the conditions under which successful completion occurs. The difficulty comparisons identify an operational boundary without establishing a causal transmission pathway. Table 4 distinguishes tasks requiring interpretation of supplied outputs from tasks that additionally require code execution and environment preparation. The lower completion percentages on Hard tasks are consistent with the

proposition that additional dependent operations create more opportunities for failure. However, the recorded variables do not identify which operation caused an unsuccessful evaluation. Boettiger, 2015, explains how containers preserve computational environments, providing a technical basis for treating supplied execution support as consequential. The present evidence adds a qualification: access to assistance and the capacity to use that assistance belong to the same evaluation system. Reproducibility assessments should therefore distinguish output interpretation from execution under configured and independently prepared environments, rather than treat these activities as interchangeable indicators of autonomy.

The configuration contrasts also resist a simple account in which one component dominates performance universally. Table 5 shows larger architecture advantages for GPT 4o mini on Easy and Medium tasks, whereas model differences are largest on Medium tasks under both architectures. These patterns suggest that the practical value of a component depends on the surrounding workflow, although the descriptive design does not establish statistical interaction. Table 4 further shows that Easy and Medium performance does not decline uniformly across configurations. Clyburne Sherin et al., 2019, distinguish sharing computational materials from making their execution reproducible. The present analysis sharpens that distinction by showing that assistance categories do not form a uniformly spaced performance ladder. Architecture, model identity and task demands should consequently remain separate evaluation dimensions rather than be collapsed into an assumed general capability index. The scoring audit reveals that evaluation infrastructure is itself part of the reproducibility problem. Equation 1 yields a result that differs from the archive summary fields, demonstrating why an aggregate performance figure requires a traceable construction rule. This observation does not establish which source implementation produced the discrepancy, but it makes the analytical consequence visible: comparisons must use a consistent outcome definition. Trisovic et al., 2022, document execution and code quality problems in shared research materials. The present audit extends the practical focus from the execution of research code to the reconstruction of benchmark scores. For benchmark maintainers and reviewers, the implication is specific: published totals should be reproducible directly from released evaluation fields, with scoring versions and denominators documented alongside the results.

The international relevance lies in a transferable evaluation problem rather than a comparison between countries. The sample spans three disciplines, but it does not identify national settings or support contrasts between advanced and emerging economies. Its broader implication is that institutions assessing research agents should report performance separately by architecture, model and operational assistance, using the configuration percentages in Table 4 rather than a pooled headline score. Kapoor and Narayanan, 2023, show that computational research can generate misleading scientific conclusions through methodological failures such as data leakage. Read alongside that evidence, the present findings reinforce a distinction with practical consequences: successful benchmark completion is evidence about task execution, not sufficient certification of scientific validity. Research governance should therefore preserve separate checks for computational completion, scoring consistency and the validity of substantive conclusions.

7. Conclusion and Implications:

Automating research verification requires evidence that the verification process itself can be reproduced. This analysis shows that successful completion varies across combinations of computational support, underlying capability and operational demands. Advantages associated with more structured assistance persist, yet diminish when evaluations require greater execution independence. The contribution is an auditable reanalysis that separates performance comparisons from the construction of the scores supporting them. It qualifies interpretations of autonomous capability by showing why aggregate rankings require explicit operational conditions and consistent evaluation rules. Theoretically, the evidence supports treating reproducibility as conditional system performance rather than an intrinsic property of an agent. The descriptive design does not establish causal pathways. For research managers, the findings support selecting systems against the demands of intended workflows and evaluating execution separately from output interpretation. Benchmark governance should require traceable denominators, documented scoring procedures and reproducible links between released records and reported totals. Operational teams should retain evaluation histories and distinguish repeated attempts from independent research cases. These practices can strengthen accountability when institutions assess automated verification tools. Their broader social value lies in supporting more transparent judgments about scientific automation while preserving the distinction between computational completion and the validity of scientific conclusions. The curated sample and short observation window provide a foundation for broader validation across disciplines, computational environments and successive system releases. Balanced replication across configurations would improve comparability, while controlled changes to individual workflow components could distinguish their contributions. Recording execution stages, resource consumption and failure locations would permit direct examination of mechanisms that the present records cannot identify. Independent reconstruction of evaluation summaries would clarify scoring differences. Future investigations could also pair benchmark completion with expert assessment of scientific validity, establishing when successful automated execution provides useful evidence for substantive research verification across institutional settings.

References:

1. Boettiger, C. 2015. An introduction to Docker for reproducible research. ACM SIGOPS Operating Systems Review, volume 49, issue 1, pages 71-79. https://doi.org/10.1145/2723872.2723882
2. Broman, K. W., & Woo, K. H. 2018. Data organization in spreadsheets. The American Statistician, volume 72, issue 1, pages 2-10. https://doi.org/10.1080/00031305.2017.1375989
3. Clyburne Sherin, A., Fei, X., & Green, S. A. 2019. Computational reproducibility via containers in psychology. Meta Psychology, volume 3. https://doi.org/10.15626/MP.2018.892
4. Collberg, C., &Proebsting, T. A. 2016. Repeatability in computer systems research. Communications of the ACM, volume 59, issue 3, pages 62-69. https://doi.org/10.1145/2812803
5. Goodman, S. N., Fanelli, D., & Ioannidis, J. P. A. 2016. What does research reproducibility mean? Science Translational Medicine, volume 8, issue 341, Article 341ps12. https://doi.org/10.1126/scitranslmed.aaf5027

6. Hughes, R. A., Heron, J., Sterne, J. A. C., & Tilling, K. 2019. Accounting for missing data in statistical analyses: Multiple imputation is not always the answer. International Journal of Epidemiology, volume 48, issue 4, pages 1294-1304. https://doi.org/10.1093/ije/dyz032
7. Kapoor, S., & Narayanan, A. 2023. Leakage and the reproducibility crisis in machine learning based science. Patterns, volume 4, issue 9, Article 100804. https://doi.org/10.1016/j.patter.2023.100804
8. Lazic, S. E. 2010. The problem of pseudoreplication in neuroscientific studies: Is it affecting your analysis? BMC Neuroscience, volume 11, Article 5. https://doi.org/10.1186/1471-2202-11-5
9. Lieder, F., & Griffiths, T. L. 2020. Resource rational analysis: Understanding human cognition as the optimal use of limited computational resources. Behavioral and Brain Sciences, volume 43, Article e1. https://doi.org/10.1017/S0140525X1900061X
10. Munafò, M. R., Nosek, B. A., Bishop, D. V. M., Button, K. S., Chambers, C. D., Percie du Sert, N., Simonsohn, U., Wagenmakers, E. J., Ware, J. J., & Ioannidis, J. P. A. 2017. A manifesto for reproducible science. Nature Human Behaviour, volume 1, Article 0021. https://doi.org/10.1038/s41562-016-0021
11. National Academies of Sciences, Engineering, and Medicine. 2019. Reproducibility and replicability in science. The National Academies Press. https://doi.org/10.17226/25303
12. Nosek, B. A., Alter, G., Banks, G. C., Borsboom, D., Bowman, S. D., Breckler, S. J., Buck, S., Chambers, C. D., Chin, G., Christensen, G., Contestabile, M., Dafoe, A., Eich, E., Freese, J., Glennerster, R., Goroff, D., Green, D. P., Hesse, B., Humphreys, M., … Yarkoni, T. 2015. Promoting an open research culture. Science, volume 348, issue 6242, pages 1422-1425. https://doi.org/10.1126/science.aab2374
13. Nosek, B. A., Ebersole, C. R., DeHaven, A. C., & Mellor, D. T. 2018. The preregistration revolution. Proceedings of the National Academy of Sciences, volume 115, issue 11, pages 2600-2606. https://doi.org/10.1073/pnas.1708274114
14. Peng, R. D. 2011. Reproducible research in computational science. Science, volume 334, issue 6060, pages 1226-1227. https://doi.org/10.1126/science.1213847
15. Piwowar, H. A., & Vision, T. J. 2013. Data reuse and the open data citation advantage. PeerJ, volume 1, Article e175. https://doi.org/10.7717/peerj.175
16. Sandve, G. K., Nekrutenko, A., Taylor, J., & Hovig, E. 2013. Ten simple rules for reproducible computational research. PLOS Computational Biology, volume 9, issue 10, Article e1003285. https://doi.org/10.1371/journal.pcbi.1003285
17. Siegel, Z. S., Kapoor, S., Nadgir, N., Stroebl, B., & Narayanan, A. 2024. CORE Bench: Fostering the credibility of published research through a computational reproducibility agent benchmark. Preprint, arXiv. https://doi.org/10.48550/arXiv.2409.11363
18. Simmons, J. P., Nelson, L. D., &Simonsohn, U. 2011. False positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. Psychological Science, volume 22, issue 11, pages 1359-1366. https://doi.org/10.1177/0956797611417632
19. Stodden, V., Seiler, J., & Ma, Z. 2018. An empirical analysis of journal policy effectiveness for computational reproducibility. Proceedings of the National Academy of Sciences, volume 115, issue 11, pages 2584-2589. https://doi.org/10.1073/pnas.1708290115
20. Tenopir, C., Allard, S., Douglass, K., Aydinoglu, A. U., Wu, L., Read, E., Manoff, M., & Frame, M. 2011. Data sharing by scientists: Practices and perceptions. PLOS ONE, volume 6, issue 6, Article e21101. https://doi.org/10.1371/journal.pone.0021101
21. Trisovic, A., Lau, M. K., Pasquier, T., &Crosas, M. 2022. A large scale study on research code quality and execution. Scientific Data, volume 9, Article 60. https://doi.org/10.1038/s41597-022-01143-6
22. Wasserstein, R. L., & Lazar, N. A. 2016. The ASA statement on p values: Context, process, and purpose. The American Statistician, volume 70, issue 2, pages 129-133. https://doi.org/10.1080/00031305.2016.1154108
23. Wickham, H. 2011. The split apply combine strategy for data analysis. Journal of Statistical Software, volume 40, issue 1, pages 1-29. https://doi.org/10.18637/jss.v040.i01
24. Wickham, H. 2014. Tidy data. Journal of Statistical Software, volume 59, issue 10, pages 1-23. https://doi.org/10.18637/jss.v059.i10
25. Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J. W., da Silva Santos, L. B., Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., Mons, B. 2016. The FAIR guiding principles for scientific data management and stewardship. Scientific Data, volume 3, Article 160018. https://doi.org/10.1038/sdata.2016.18
26. Wilson, G., Aruliah, D. A., Brown, C. T., Chue Hong, N. P., Davis, M., Guy, R. T., Haddock, S. H. D., Huff, K. D., Mitchell, I. M., Plumbley, M. D., Waugh, B., White, E. P., & Wilson, P. 2014. Best practices for scientific computing. PLOS Biology, volume 12, issue 1, Article e1001745. https://doi.org/10.1371/journal.pbio.1001745
27. Ziemann, M., Eren, Y., & El Osta, A. 2016. Gene name errors are widespread in the scientific literature. Genome Biology, volume 17, Article 177. https://doi.org/10.1186/s13059-016-1044-7