Vol. 16 (2026): Proceedings of the 2026 International Conference on AI and Technological Innovation Applications (AITIA 2026)
E-ISSN: 3105-0522 P-ISSN: 3105-0514
DOI: https://doi.org/10.70267/aitia.2026456464

SurveyUQBench: Recovering Complex-Survey Designs and Design-Based Uncertainty from Official Documentation

Junxi Zhou*

Department of Computer Science and Technology, Guizhou Normal University, Guiyang, China

**Corresponding author: Junxi Zhou.*

Abstract

Reliable public-survey analysis combines plausible coding and weighted point estimation with the recovery of the correct weights, strata, primary sampling units, replicate-weight convention, domain rule, degrees of freedom, and confidence-interval rule from official documentation. We introduce SurveyUQBench, a benchmark/resource that evaluates this complete path to design-based uncertainty. It contains 360 tasks drawn from 12 public survey programs and five variance mechanisms, together with official-document evidence qrels, evaluation-hidden design manifests, reference R analyses, and deterministic graders for evidence, design recovery, and estimate/standard-error/confidence-interval correctness. Reference analyses and clean replays cover the full resource, and mutation tests confirm that the deterministic grader distinguishes valid references from targeted mutations. Direct model responses achieve broad compliance with the response contract; strict NumericUQ and FullTask both record 0 under the joint criteria. On a fixed task-model subset, submission-native execution achieves 23 successful invocations; deterministic interface normalization increases this to 89, yielding 9 finite estimate/SE pairs and 2 NumericUQ passes—gold-input interventions further separate evidence, design, interface, and numerical stages. SurveyUQBench therefore provides a traceable evaluation resource, a reproducible baseline, and stage-specific signals for measuring future progress.

Keywords

benchmark evaluation, complex survey analysis, design-based uncertainty, large language model agents, survey design recovery

1. Introduction

Complex-survey design recovery is a central capability for language-model data agents. A system analyzing NHANES, ACS PUMS, CPS ASEC, SCF, or another public survey must respect the released table's complex sampling design. The official analysis weight determines population representation; strata and primary sampling units determine Taylor-linearized variance; replicate weights require program-specific scaling; and domain, multiple imputation, degrees of freedom, and confidence interval rules affect inferential conclusions. These details are usually distributed across technical reports, codebooks, data dictionaries, and release-specific notes.

The resulting task joins document evidence and executable statistical reasoning. A model must first locate official support for design fields, then express those fields as a coherent survey-design object, and finally run the requested estimand with design-based uncertainty. Reliable uncertainty additionally requires the correct full-sample weight, replicate family, scaling constant, and domain operation, because each can preserve a plausible point estimate while changing its standard error and confidence interval. Existing complex-survey software provides the statistical machinery [1-3]; SurveyUQBench supplies the evaluation layer for document-to-design translation.

SurveyUQBench isolates that translation. Each task supplies an estimand, official documentation, and task-ready data or a derivation recipe. A system returns evidence, a design manifest, executable analysis, and numeric uncertainty. Evaluation-hidden qrels and reference artifacts support deterministic main scores. The benchmark uses a whole-program development/held-out split so that test programs, rather than merely test phrasings, remain unseen.

We formulate official-document complex-survey design recovery as an end-to-end task with separate evidence, design, code-interface, and NumericUQ measurements. Second, we construct a multi-program resource spanning Taylor linearization, successive difference replication (SDR), standard BRR, Fay BRR, and bootstrap plus multiple imputation. Third, we provide traceable gold artifacts, a deterministic compiler/grader, reference replay, and mutation checks. Fourth, we provide descriptive diagnostics that quantify the stage-specific gap between contract compliance, design recovery, and correct design-based numerical uncertainty.

2. Related Work

Long-document analysis. LongDA is the closest benchmark-level comparison, as it evaluates agents on long, heterogeneous documents from U.S. national surveys [4]. SurveyUQBench narrows the target to official support for survey-design slots and deterministic grading of design-based estimates, standard errors, and confidence intervals.

Applied survey analysis. Mathematica's complex statistical data framework evaluates LLMs using ACS PUMS-style weighted analysis [5], while LAPS automates hypothesis-driven analysis of public surveys [6]. Building on this motivation for public-survey automation, SurveyUQBench withholds the design manifest, covers five variance mechanisms, and scores the recovered design and uncertainty against executable references.

Statistical reasoning and code generation. StatLLM, StatABench, and StatEval evaluate statistical analysis, programming, and reasoning across broader task families [7-9]. SurveyUQBench focuses specifically on the official-document-to-design-to-uncertainty chain, including replicate and multiple-imputation conventions that complement ordinary code-generation accuracy.

Data agents and benchmark validity. Data-agent benchmarks study end-to-end question answering over data and broader agent workflows [10, 11]. Work on rigorous agentic benchmarks and Evaluation Cards emphasizes leakage control, explicit outcome validity, diagnostic reporting, and reproducibility [12, 13]. SurveyUQBench responds with whole-program splits, evaluation-hidden gold artifacts, open executable outputs, deterministic grading, duplicate- and final-number-exposure audits, fresh-environment replay, and mutation tests. The main paper centers on complex-sampling design recovery, while the supplement situates survey-response simulation, questionnaire analysis, generic model-confidence uncertainty, and additional data-agent work around that focus.

3. Methodology

3.1 Task and Output Contract

A task contains a natural-language estimand, an official-document set, task-ready data, or a derivation recipe, and evaluation-hidden reference artifacts. The system must return four connected objects.

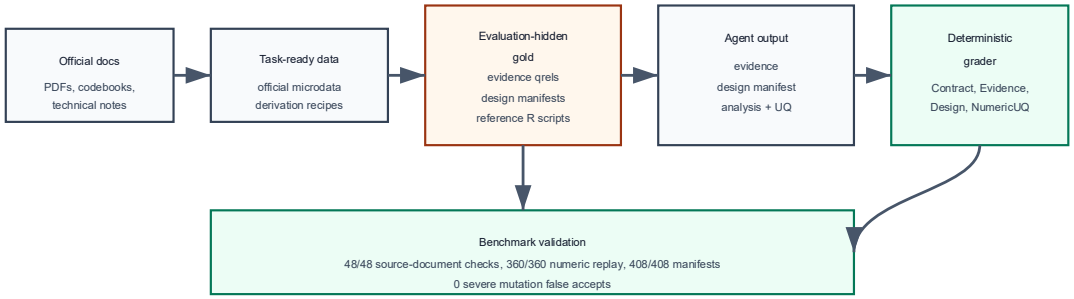
Table 1: SurveyUQBench output contract and grading targets

Output object	Required content	Primary check
Evidence	Official document, locator, and supported design slot	Evidence qrels / recall
Design manifest	Weight, strata/PSU or replicate specification, domain/df/CI rules	Schema and slot exactness
Analysis	Executable analysis over the constructed survey design and task data	Interface and isolated-execution checks
Numeric uncertainty	Estimate, SE, lower CI, and upper CI	NumericUQ tolerance checks

The manifest is the intermediate representation. It records **design_family**, **analysis_weight**, **strata**, **psu**, **replicate_method**, **replicate_columns**, **scale**, **rscales**, **rho**, **mse**, multiple-imputation rules, domain handling, lonely-PSU handling, degrees of freedom, and the confidence-interval rule. FullTask requires a contract, evidence/design, and NumericUQ success. The decomposition measures performance separately across evidence localization, design construction, code compatibility, and numerical execution.

Contract checks require a parseable response with all four objects and an executable analysis component. Evidence Recall awards matches between submitted document identifiers and locators and slot-specific qrels. Design Slot Exact reports exact correctness for applicable manifest fields, while design-all requires that every required field match. NumericUQ success requires that the estimate, SE, and both interval bounds meet task-specific, pre-specified tolerances. These definitions preserve partial diagnostic information while making FullTask a joint end-to-end measure.

Figure 1: SurveyUQBench pipeline from official documents and task-ready data through evaluation-hidden references, agent outputs, and deterministic component and end-to-end grading



3.2 Worked Example

Consider a task that asks for a design-based mean age from the 2019 ACS person PUMS. The evidence qrel points to the ACS design methodology and PUMS technical documentation sections describing SDR and PUMS replicate weights. For this person-level estimand, the documentation-level manifest resolves the full-sample weight to **PWGTP**, identifies 80 replicate columns **PWGTP1** through **PWGTP80**, uses the ACS replicate-variance scale **4/80**, retains the full design before domain estimation, and applies a normal 95% interval.

The example uses the same scoring evidence as the benchmark, with the exact supporting passages documented separately from scoring. ACS Design and Methodology Chapter 12, Section 12.2 (PDF p.1) names SDR; pp.2 and 6 state that ACS and PUMS use 80 replicate weights; and p.4 defines the multiplier as **4/R**, with **R = 80**. The Census ACS PUMS variables page identifies **PWGTP** and **PWGTP1** through **PWGTP80**. Chapter 12, Section 12.3 (pp.5-6) discusses official ACS margins of error at 90% using 1.645. We therefore treat the normal 95% interval as a benchmark task specification distinct from the official ACS reporting guidance.

A compact representation of the relevant manifest fields is:


```
{
"DESIGN_FAMILY": "SDR",
"ANALYSIS_WEIGHT": "PWGTP",
"REPLICATE_COLUMNS": {"PATTERN": "PWGTP1-PWGTP80", "COUNT": 80},
"REPLICATE_METHOD": "SDR",
```

```
 "SCALE": "4/80",
 "DOMAIN_RULE": "RETAIN FULL DESIGN BEFORE DOMAIN ESTIMATION",
 "CI_RULE": {"DISTRIBUTION": "NORMAL", "LEVEL": 0.95}
}
```

The compiler and reference script implement the same contract. The following expository excerpt focuses on design construction after benchmark-provided data preparation.

```
REP_COLS <- GREP("^PWGTP[0-9]+$", NAMES(DF), VALUE = TRUE)
DESIGN <- SURVEY::SVREPDESIGN(
 WEIGHTS = ~PWGTP, REPWEIGHTS = DF[, REP_COLS],
 TYPE = "OTHER", SCALE = 4 / 80, RSCALES = REP(1, 80),
 COMBINED.WEIGHTS = TRUE, MSE = TRUE, DATA = DF
)
FIT <- SURVEY::SVYMEAN(~AGEP, SUBSET(DESIGN, DOMAIN), NA.RM = TRUE)
```

The reference output is **estimate = 38.98558**, **SE = 0.13596**, and 95% CI **[38.71910, 39.25205]**. Evidence, manifest, code, and output are graded separately, so end-to-end credit requires both supported design recovery and numerically correct execution.

3.3 Program, Configuration, and Task Construction

SurveyUQBench covers 12 public U.S. survey programs. Four configurations per program were selected through an explicit four-criterion protocol: public-use files, retrievable official documentation, verifiable source records, and a runnable task-ready recipe. Selection favored consecutive recent releases when the official cadence allowed; programs with biennial, multi-year, or disrupted release schedules use spaced cycles. The resulting criterion-guided longitudinal roster supports controlled cross-cycle and cross-mechanism evaluation.

Within each program, the four releases provide temporal coverage of available cycles; across programs, the roster covers different variance mechanisms. The selected releases span 2011-2024. Mechanism diversity is principally between programs, while repeated releases within a program test cycle-specific evidence, file naming, weight selection, and pooling conventions. Balancing every program at four configurations prevents high-frequency annual programs from dominating the resource. The complete cycle roster and selection rationale appear in the supplement.

Each program contributes 25 controlled-core and five natural-extension tasks. Core tasks hold data preparation relatively fixed to isolate design recovery; extensions add file or weight selection, joins, domain derivation, pooling, validation, or wrong-year-document distractors. Across the core bank, 156 tasks also carry a pre-specified decision rule so that estimate/SE/CI errors can be related to a null-value decision while preserving the primary score. Difficulty labels serve as overlapping operational controls for construction.

The split is by whole program: ACS PUMS, AHS, BRFSS, and NHANES for development; the other eight programs form a held-out evaluation set. Development queries draw exclusively from the four development programs. CE and SCF introduce mechanism families that are absent from development, making standard BRR and bootstrap-plus-MI reconstruction challenging, and introducing new estimands for unfamiliar programs.

Table 2: Benchmark and task diversity. Counts are computed from the finalized benchmark inventory

Dimension	Coverage
Programs / configurations / years	12 / 48 / 2011-2024
Variance mechanisms	Taylor 20; SDR 8; standard BRR 4; Fay BRR 12; bootstrap + MI 4 configurations
Controlled-core estimands	Mean 70; proportion 41; total 24; difference 55; ratio 35; linear coefficient 42; logistic coefficient 33
Core difficulty controls	200 design-sensitive; 100 design-neutral; 120 domain; 60 multi-file; 60 pooled/cross-period
Natural extensions	60 total; five per program
Program split	4 development programs / 8 held-out programs

3.4 Gold Artifacts, Compiler, and Safeguards

Gold construction has three linked layers: official-document evidence qrels, design manifests, and reference R scripts with numeric outputs: a schema validator checks required and mechanism-specific slots. The compiler maps accepted manifests to Taylor, replicate-weight, or multiply imputed survey objects. Controlled execution enforces time limits and records completion status. NumericUQ applies pre-specified absolute/relative tolerances to estimate, SE, and both interval bounds; FullTask combines component checks.

Three groups of safeguards shape the protocol. Whole-program splitting, hidden gold artifacts, duplicate analysis, and a final-number exposure audit implement leakage controls. A gold/reference upper bound, mechanism-specific mutation tests, and deterministic scoring establish grader robustness. Source hashes, reference-script replay, clean-environment numeric replay, and explicit model-output provenance support reproducibility. Together, these safeguards establish resource validity across leakage control, grader robustness, and reproducibility within the recorded protocol.

4. Resource Validation

The construction checks establish that the released benchmark can be executed end-to-end and that the strict grader can both accept correct outputs and reject severe mutations.

Table 3: Resource-validation results.

Validation item	Result
Official configuration sentinels	48 / 48
Controlled-core reference numeric runs	300 / 300
Natural-extension reference numeric runs	60 / 60
Schema validation	408 valid / 0 invalid manifests
Design compiler probes	48 / 48
Reference-script and clean numeric replay	360 / 360 each
Targeted mutations accepted	0
Evidence locatable rate	1.000
Exact, lexical, or semantic duplicate matches	0
Potential exposed gold-number matches	0 / 360

The resource-validation counts characterize authored references and deterministic checks; model performance is reported separately in Section 6. A programmatic consistency analysis additionally covered 48 configurations, 768 essential slots, and 120 sampled tasks, providing procedural consistency evidence across official evidence, design specifications, numeric traces, and execution records. This separation keeps resource-validation and model-performance evidence directly traceable.

5. Model Evaluation Protocol

Strict raw/extension grading covers 1,920 model-task rows from 22 evaluation runs spanning two closed and six open-weight model configurations. The evaluated models are gpt-5.4, gpt-5.5, four Qwen configurations, Mistral-Small-3.2-24B, and Gemma-3-27B. All rows use the same response-contract and component-grading criteria. The recorded model versions, serving backends, decoding, and runtime conditions support descriptive component-level analysis and specify the conditioning variables for standardized comparison designs.

Conditioned diagnostics provide gold evidence, a gold manifest, or a gold design object to isolate pipeline stages. Pre-specified component contrasts use the same 100 tasks across eight model configurations so that evidence, design, code-interface, and direct numeric cells share a denominator. Retrieval diagnostics distinguish document-card selection from page/span localization. Shortcut perturbations remove data or documents, substitute adjacent-year material, shuffle evidence, redact answer-like text, or paraphrase prompts; two-seed repeats measure output stability. Oracle inputs and static code-interface analyses provide stage-specific measurements alongside the raw-agent scores.

The layered raw metrics grade numeric fields returned directly in model responses. An isolated-execution diagnostic complements them by running the submitted analysis code for the pre-specified 100-task-by-8-model subset and grading the values produced by that execution. All reported analyses are computed from the

same fixed archive of model outputs, preserving a common basis across direct-response and execution diagnostics.

6. Results

6.1 Strict and Layered Raw Diagnostics

Across all strict raw/extension evaluations, 91.8% of responses satisfy the output contract, establishing a strong baseline for parseability. The joint NumericUQ and FullTask criteria record 0; the layered measurements resolve this aggregate result into 123 rows with finite estimate/SE pairs, and the field-level correctness counts in Table 4.

Table 4: Layered diagnostics over 1,920 evaluated raw/extension model-task rows.

Layer	Count	Rate
Contract pass	1,763	0.918
Finite estimate and finite SE	123	0.064
Finite lower and upper CI	118	0.061
Estimate correct	0	0.000
SE correct	2	0.001
Both CI bounds correct	0	0.000
NumericUQ / FullTask	0 / 0	0.000 / 0.000

For the 123 finite estimate outputs, the median tolerance-normalized estimate error is 8,824 (interquartile range 1,674-10,000), and the corresponding SE error is 1,503 (222-9,999). These conditional distributions characterize the finite-output subset, while Table 4 retains the all-row success rates. Model-level values and complete error summaries are in the supplement.

6.2 Conditioned, Retrieval, and Component Diagnostics

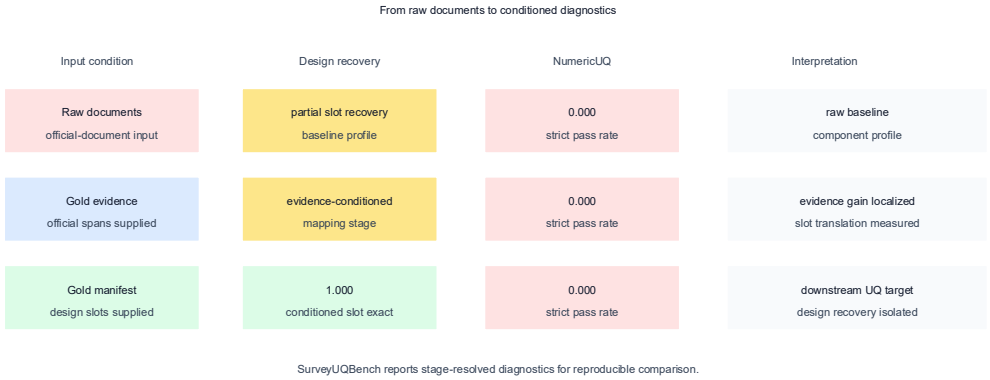
Providing the gold manifest raises design-slot exactness to 1.000 for the reported closed-model information-ladder rows; its paired NumericUQ result is 0. A synthetic gold/reference output meets all strict metrics across all 360 tasks, confirming grader attainability and identifying numerical execution as the next measured stage after design recovery.

Table 5: High-signal component diagnostics; detailed rows are in the supplement.

Diagnostic	Key result	Interpretation
Gold/reference upper bound	FullTask 360 / 360	Grader-attainability reference
Gold manifest condition	Design exact up to 1.000; NumericUQ 0	Design recovery can be isolated from numeric execution
Document-card BM25	R@5 0.995	Document-level retrieval
Calibrated page/span dense + rerank	R@5 0.688	48 queries over PDF-backed sources
All-reference component path	800 / 800	Reference path
Predicted strict evidence/design/numeric cells	0 / 800	Stage-specific improvement signals
Strict predicted-code execution	Invocation 23 / 800; finite estimate/SE 0	Submission-native interface; separate diagnostic
Deterministic compatibility execution	Invocation 89 / 800; finite estimate/SE 9; NumericUQ 2	Deterministic interface normalization

All 100 reference scripts first pass invocation, schema, and NumericUQ under the same isolated-execution protocol. Each predicted row is then evaluated independently. Submission-native execution records 23 successful invocations, one schema-complete row, and zero finite estimate/SE pairs. Deterministic interface normalization records 89 invocations, 17 schema-complete rows, nine finite estimate/SE/CI outputs, and two NumericUQ passes. This sequence quantifies the interface effect while preserving the submitted statistical logic. The supplement reports the corresponding raw-agent condition and the earlier 67/800 and 631/800 static interface diagnostics alongside these execution results.

Figure 2: Conditioned diagnostic matrix. Gold evidence and manifests improve upstream components and identify downstream execution as the next stage for strict NumericUQ improvement



Together, the diagnostics identify four separable improvement targets: page/span evidence localization, exact design-manifest recovery, conformance to the required code interface, and production of finite, accurate numerical outputs. The supplement provides the detailed error taxonomy, retrieval settings, counterfactuals, and stability results.

The error taxonomy is consistent with this decomposition. The highest-frequency improvement opportunities involve confidence-interval and degrees-of-freedom rules, domain handling, design family, analysis weight, replicate columns and scale, PSU, and lonely-PSU policy. Gold-design counterfactuals execute on all 200 selected tasks and show that simplified weight-only or iid analyses can alter uncertainty even when a point estimate is close. These counterfactuals provide contract-validation evidence, while raw-model success and formal model comparison use their dedicated measures.

7. Discussion and Scope

SurveyUQBench's main contribution is the traceable loop from official documentation to a design object and design-based uncertainty. This focus matters because replicate scale, weight, domain rule, and imputation convention can produce distinctions that ordinary code or point-answer checks leave unresolved. The layered metrics resolve the aggregate strict end-to-end result into stage-specific evidence by measuring contract compliance, finite-output production, evidence recovery, design reconstruction, and interface compatibility separately.

The empirical scope comprises 12 U.S. public programs, five variance mechanisms, and four configurations per program, selected using explicit feasibility criteria, thereby supporting conclusions about the releases and task recipes represented. The shared task schema also supports transportability studies across countries, agencies, survey modes, restricted-data designs, and additional estimators. Locator-derived relevance labels and within-study calibration provide evidence for page/span retrieval; independently annotated relevance judgments and external survey-statistician review are treated as distinct, complementary validation layers. The recorded serving conditions support descriptive component analysis and define the basis for standardized, powered rankings and version-to-version comparisons.

NumericUQ specifically measures the correctness of design-based estimates, SEs, and CIs. This construct complements model-confidence calibration, new survey estimators, general data-agent methods, and comparative method evaluation. Detailed ethics, reproducibility, source-use, and artifact records, together with the responsible-use guidance that motivates them [14], appear in the supplement.

8. Conclusion

SurveyUQBench evaluates whether language-model agents can recover complex survey designs from official documentation and produce correct design-based uncertainty. Its validated references and deterministic grader cover 360 tasks across five variance mechanisms. Current evaluated outputs show strong response-contract compliance and partial design recovery, while the strict results identify executable analysis and

numerical uncertainty as the principal targets for improvement. The resource and its component diagnostics provide a reproducible basis for measuring progress and conducting future standardized comparisons.

References

[1] Lumley, T. (2004). Analysis of complex survey samples. *Journal of statistical software*, *9*, 1-19.

[2] Lumley, T. (2020). Survey: analysis of complex survey samples. *R package version*, *4*(1).

[3] Li, Y., Zhang, Z., Ma, T., Wang, Z., Murugesan, K., Zhang, C., & Ye, Y. (2026). LongDA: Benchmarking LLM Agents for Long-Document Data Analysis. *arXiv preprint arXiv:2601.02598*.

[4] Kim, J., Jeong, D., Son, B., Kim, H., Kim, B., & Han, K. (2026, April). LAPS: Automating Hypothesis-Driven Statistical Analysis of Public Survey Using Large Language Models. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems* (pp. 1-19)

[5] Song, X., Lee, L., Xie, K., Liu, X., Deng, X., & Hong, Y. (2026). Statllm: A dataset for evaluating the performance of large language models in statistical analysis. *Scientific Data*.

[6] Zhu, Y., Ding, Y., Lai, P., Wang, L., Jing, B., & Chen, G. (2026). StatABench: Dataset and Framework for Evaluating Statistical Analysis Capabilities of LLMs. *arXiv preprint arXiv:2606.22977*.

[7] Lu, Y., Yang, R., Zhang, Y., Yu, S., Dai, R., Wang, Z., ... & Zhou, F. (2025). Stateval: A comprehensive benchmark for large language models in statistics. *arXiv preprint arXiv:2510.09517*.

[8] Ma, R., Shankar, S., Chen, R., Lin, Y., Zeighami, S., Ghosh, R., ... & Parameswaran, A. G. (2026). Can ai agents answer your data questions? a benchmark for data agents. *arXiv preprint arXiv:2603.20576*.

[9] Sun, Z., Zhong, S., Wen, D., Han, J., Li, G., Yan, Y., ... & Ruan, H. (2026). AgenticDataBench: A Comprehensive Benchmark for Data Agents. *arXiv preprint arXiv:2607.01647*.

[10] Zhu, Y., Jin, T., Pruksachatkun, Y., Zhang, A., Liu, S., Cui, S., ... & Kang, D. (2026). Establishing best practices in building rigorous agentic benchmarks. *Advances in Neural Information Processing Systems*, *38*.

[11] Ghosh, A., Reuel, A., Chim, J., Kennedy, W. M., Yadav, S., Mickel, J., ... & Solaiman, I. (2026). Evaluation Cards: An Interpretive Layer for AI Evaluation Reporting. *arXiv preprint arXiv:2606.09809*.

[12] Rothschild, D. M., Marla, J., Amaya, A., Barari, S., Buskirk, T. D., Cobb, C., ... & Webb, B. (2026). Responsible AI Integration in Survey Research.

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgment

This paper is an output of the science project.

Open Access