Bridging Research Gaps With AI: Expertise and Disciplinary Mobility

Ghina Abdul Baki, Juan P. Aparicio, Bruno Barbarioli, Abel Brodeur,
Lenka Fiala, Derek Mikola, David Valenta

2025-10-15

1 Abstract

We test whether providing large-language-model (LLM) assistance compresses performance gaps in research skills along two pre-specified dimensions. Vertically, we study expertise tiers using a three-category grouping (undergraduate, graduate student, professor/researcher) that governs both randomization and analysis while retaining self-reported titles for descriptive context. Horizontally, we study cross-discipline performance in three quantitative social sciences (Economics, Political Science, Psychology) to assess whether AI reduces the penalty from working outside one's primary discipline (out-of-discipline, OOD). Participants are randomized 1:1 to AI access (e.g. ChatGPT) versus human-only controls within the applicable expertise strata; tasks are assigned a single discipline tag, with undergraduates always receiving inside-discipline assignments and approximately the remaining participants randomly assigned to an outside-discipline paper. Outcome measures are organized into two families: coding skills (computational reproducibility and the detection of major/minor coding errors) and non-coding skills (referee appropriateness/overall scores and robustness-check execution). The study is designed to reveal whether AI acts as an equalizer within expertise tiers and whether it enables researchers to operate effectively across disciplines.

2 Registration and Funding

This pre-analysis plan is registered on the Open Science Framework (OSF) at https://osf.io/dkfzt/. Funded by Open Philanthropy and the Institute for Replication (I4R). The locked PAP together with the analysis scripts used to generate the mock tables and figures are mirrored there; live study code and data will be added only after the analysis lock.

3 Background and Rationale

Prior multi-site "AI Replication Games" documented measurable AI effects on success and speed in reproduction tasks, while also revealing substantial variation across teams and events. Building on that foundation, we pre-register two complementary dimensions of distributional impacts. First, the vertical dimension (expertise ladder): whether AI narrows performance gaps by disproportionately lifting less-experienced participants (equalizer) or widens gaps by enabling experts to better leverage tools (amplifier). Second, the horizontal dimension (cross-discipline): whether AI reduces the penalty from working outside one's primary discipline (out-of-discipline, OOD) when reproducing studies across Economics, Political Science, and Psychology. These questions matter for pedagogy, workforce development, and equity: vertical compression would support broad inclusion strategies; horizontal compression would support cross-field mobility and knowledge diffusion; in both cases, amplification would call for targeted training and governance to avoid widening disparities.

We keep tasks, instructions, and grading rubrics closely aligned with prior exercises to ensure comparability while tailoring the design to individual-level randomization within strata and a discipline-tagged task pool. The design isolates intent-to-treat effects within expertise tiers and introduces an orthogonal OOD contrast by allocating approximately 30% of non-undergraduate participants to an outside-discipline article while keeping all undergraduates inside their primary field. This enables transparent tests of heterogeneous effects across tiers (vertical) and along the OOD dimension (horizontal). To support interpretability, we pre-specify a compact outcome set split between coding skills (computational reproducibility; identification of major and minor coding errors) and non-coding skills (referee appropriateness/overall judgments and robustness-check execution) alongside a small list of precision-enhancing controls; discipline fixed effects are nested within article fixed effects and are thus absorbed in all models.

Several design features anchor the study and deserve to be unpacked carefully. We begin with randomization: participants enter one of three expertise strata—undergraduate, graduate (master's and PhD students together), or professor/researcher (postdocs and faculty). Within each stratum, participants are randomized 1:1 to either AI assistance or a human-only control condition. This stratification keeps comparisons within peer groups while preserving overall balance; it also fixes the reference point for the "vertical" estimands that trace how treatment effects evolve along the expertise ladder.

Task assignment introduces the horizontal dimension. Articles arrive with a single discipline tag (Economics, Political Science, or Psychology), enabling us to contrast performance inside versus outside a participant's primary field. We explicitly model two families of estimands—vertical (AI × tier) and horizontal (AI × out-of-discipline)—and we test each dimension on its own set of outcomes. This separation keeps the interpretation of compression tests transparent and avoids conflating discipline moves with tier moves.

Outcome measurement rounds out the design. Coding outcomes combine a binary indicator for successful computational reproducibility with the counts of major and minor coding errors flagged during replication. Non-coding outcomes focus on communication and judgment: human referees—blinded to treatment and discipline—and a parallel AI grader each record whether the submission is appropriate and assign an overall 0–5 score, while participants' robustness logs record whether they execute at least one or at least two qualifying checks. Section "Grading Rubric" details the anchors that guide human judgments, while Appendix "AI Referee Prompt" reproduces the corresponding instructions for the AI grader, ensuring that communication quality is assessed consistently across arms.

Our design choices address practical concerns. First, measurement: we standardize the classification of major/minor errors and use independent human and AI judges for referee outcomes to triangulate communication quality, with blinding to treatment and discipline. We also track the execution of robustness checks as part of the non-coding skill set to contextualize main effects. Second, scope and external validity: by spanning multiple events, software ecosystems (R/Stata/Python), a broad experience range (undergraduate to professor), and three disciplines, we gauge how AI assistance interacts with realistic heterogeneity in tools, backgrounds, and fields. These features, combined with preregistration, separate multiplicity control for the vertical and horizontal families, and a limited set of pre-specified estimands, aim to balance credibility with informativeness.

4 Research Questions and Hypotheses

This study addresses three linked questions. First, we ask whether AI assistance compresses or widens the expertise gradient by comparing treatment effects across the three pre-defined tiers; this provides the "vertical" lens that motivates the stratified randomization. Second, we investigate whether AI attenuates the penalty from tackling a problem outside one's primary discipline, thereby capturing "horizontal" mobility across Economics, Political Science, and Psychology. Third, we consider the intent-to-treat contrast averaged across all participants, which anchors the benchmark effect size against which heterogeneous responses are interpreted.

The corresponding hypotheses follow naturally. H1 posits that access to AI increases coding performance—raising the probability of a successful reproduction while lowering major and minor coding errors. H2 states that any gains are at least as large for undergraduates as for graduate students and professors/researchers, implying a compression of tier gaps in coding outcomes. H2b mirrors this logic across disciplines, anticipating that AI shrinks the outside-of-discipline penalty for the same coding skills. H3 focuses on non-coding capabilities, predicting that AI exposure improves referee appropriateness/overall judgments and increases the share of participants executing qualifying robustness checks. Each hypothesis is tested within the relevant estimand family, with multiplicity handled as described below.

Throughout the document we rely on several recurring definitions. "Reproduction success" means the participant's final output matches the pre-specified focal result within the documented tolerance; adjudication follows the rubric in Section "Grading Rubric." "Error detection" counts correctly identified coding errors and distinguishes between major issues—those that would alter the substantive result—and minor issues that affect presentation or reproducibility without changing the estimate. "Communication quality" focuses on the Appropriateness indicator (binary) and the overall 0–5 score derived from the referee rubric; the underlying rubric is available to judges but only these two statistics enter the preregistered estimands. "Robustness execution" records whether participants implement at least one or at least two robustness checks that meet the pre-specified criteria. Finally, "out-of-discipline (OOD)" labels any case in which the task's discipline tag differs from the participant's self-reported primary discipline.

5 Experimental Design

We recruit participants across the three canonical tiers—undergraduates, graduate students (master's and PhD), and professors/researchers (postdocs and faculty)—and observe each in a single, timed, one-day session. Randomization always occurs within these three strata while we continue to track individual titles for descriptive reporting. Every participant attempts to reproduce one pre-specified result using the same software ecosystems as the original studies (R, Stata, or Python) within a seven-hour working window. The window mirrors prior AI Replication Games, where most teams completed their submissions within seven hours while still allowing careful documentation. We randomly assign access to AI assistance (ChatGPT Plus with tools) within the relevant strata and events. Participants in the control arm pledge not to use AI tools. Any deviations are documented and, when material, addressed through the pre-specified per-protocol sensitivity analyses. Human referees are blinded to both treatment and discipline, and filenames/metadata that could reveal either are redacted.

Within each event, reproduction papers are randomly assigned from a pre-curated pool once participant rosters are finalized. After the random draw, the research team audits each assigned paper to confirm that (i) the replication package includes executable code together with a README or documentation, (ii) all requisite datasets and intermediate files are accessible without additional permissions, and (iii) the combined data footprint remains tractable for local execution (target: compressed package 500 MB and memory requirements within a standard 16 GB RAM laptop). Papers that fail any check are replaced before assignments are communicated to participants.

Task materials are version-controlled in `Papers/`, which stores a journal-level folder for each study together with the inventory workbook `Papers/papers.xlsx`. As of the lock, the workbook enumerates 15 studies (AJPS = 5, AEJ: Applied Economics = 5, Psychological Science = 5) with DOI identifiers and replication-package URLs. Each journal folder contains the published article (`paper.pdf`) and the original replication package supplied by the journal or data repository; where only code is distributed (for example, "The Willingness to Pay for a Cooler Day"), the folder retains the vendor-provided archive so teams receive the same assets that were audited. This structure lets us share identical bundles across events while keeping provenance and updates transparent.

The design studies vertical compression across expertise and horizontal compression across disciplines in a unified framework (run simultaneously). Each article carries a single discipline tag—Economics, Political Science, or Psychology—and participants self-report a primary discipline at registration. The task pool spans all three fields at every event. Undergraduates always receive inside-discipline papers; among graduate students and professors/researchers (postdocs and faculty) we randomly select enough individuals to ensure that approximately 30% of the overall roster (subject to feasibility) works outside their primary discipline (OOD), with the remainder kept inside. This prioritizes a fixed OOD exposure rate over per-cell balance and reflects our expectation that undergraduates benefit from staying within field during a timed replication task. We report realized counts by tier, the collapsed tier grouping described below, discipline, and OOD status prior to the analysis lock. Event×article fixed effects absorb site, tooling, and task heterogeneity; because each article maps to a single discipline, separate discipline indicators are redundant.

For descriptive summaries and replication outputs we rely on the same three-tier stratification encoded as `tier_3`: undergraduates form the first category, master's and PhD students comprise the graduate grouping, and postdocs plus faculty comprise the professor/researcher tier. Treatment assignment and inference reference this partition throughout; the finer self-reported titles enter only in descriptive appendices.

5.1 Pre-Event Training

Participants attend a 60-minute orientation the week before each event (slides in `Pregame/ai_research_webinar_codex_cli_v2.pdf`). The session focuses on logistics: how the games run, what deliverables to submit, the structure of the Excel tracking file, and expectations about documentation, timing, and referee reports. We provide a brief overview of ChatGPT Plus features so attendees understand the tools that treated participants will receive, but the emphasis is on workflow discipline, reproducibility reminders, and answering procedural questions. Randomization is revealed one day before each event so that ChatGPT Plus invitations can be activated for treated participants in advance.

5.2 Post-Pilot Focus Groups

To contextualize quantitative findings, we will run five parallel focus-group sessions (six participants per group) immediately after the pilot wave concludes. Scheduling them post-pilot minimizes contamination risk: participants cannot brief future cohorts about task structure or AI prompts before those sessions occur. Groups are stratified by treatment status (AI-assisted vs. human-only) and, when numbers permit, by inside/outside-discipline assignments so that discussions surface arm-specific workflows and cross-discipline frictions. Each 60-minute session is moderated by Institute for Replication staff using the standardized guide in `focus_groups/focus_group_guide.md`; facilitators remind attendees of confidentiality expectations and collect recorded consent before beginning. Discussion notes and transcripts are coded with the companion qualitative codebook (`focus_groups/codebook.md`), which maps themes on motivations, preparation, AI use or workarounds, cross-discipline challenges, and suggestions for future waves. Insights from these sessions inform protocol refinements before scaling beyond the pilot.

Design cells overview (what is crossed with what):

Table 1: Design factors and levels (crossed: Arm × Tier × OOD).

Factor	Levels	Notes
Arm	Human-Only; AI-Assisted	1:1 allocation within tier
Expertise tier	Undergraduate; Graduate; Professor/Researcher	Stratification (randomization blocks; Graduate = MA/PhD, Professor/Researcher = postdoc/faculty)
OOD status	Inside; Outside	Derived from participant vs task discipline

Randomization is implemented with a reproducible script and a fixed seed recorded in the registry; the assignment file is timestamped and stored read-only. Allocation is concealed until check-in, when the onsite coordinator reveals arm and task. No-shows remain in their assigned arm for intent-to-treat analyses. Replacements are permitted only before the event begins and are re-randomized using the same stratum-specific seed. The same concealment and documentation protocol applies to inside/outside assignments. Any late swaps or deviations are logged prior to accessing outcomes.

We plan to enroll roughly 300 participants across multiple events. For power, we assume a plausible tier composition (more undergraduates than postdocs/professors) rather than equal counts by tier. Simulations suggest that, with baseline success gaps between undergraduates and professors of 15–20 percentage points and AI compressing roughly 40% of that gap, we achieve at least 80% power for the vertical interactions at $\ = 0.05$. For the horizontal dimension, we assume a baseline OOD penalty of 20–25 percentage points in the control arm and target detectable AI-induced reductions of about 8–10 percentage points, while preserving the same overall sample size. Throughout, we rely on heteroskedasticity-robust (HC3) standard errors. Table 2 summarizes the core inputs; we will freeze any updates to these assumptions prior to registry lock.

The primary analysis set follows intent-to-treat principles and includes all randomized individuals with any outcome data. We do not impute outcomes. For success indicators, error counts, referee measures, and robustness checks, nonresponse results in missing outcomes that are excluded from that specific regression but retained for other outcomes. For covariates only, we handle missingness as follows: categorical covariates (tier, software, prior ChatGPT familiarity, and—where used descriptively—participant and task discipline) gain an explicit "Missing" category if needed; continuous covariates (years of coding) use within-stratum median imputation with a missingness indicator. We report outcome and covariate missingness by arm and verify robustness to listwise deletion.

Table 2: Prospective power and design inputs (pre-lock)

Quantity	**Value**
Participants (N)	300 (approximate)
Tier composition (assumed)	UG 35–40%, Graduate 35–40%, Professor/Researcher 20–25%
Discipline composition	≈50% Econ, 25% PolSci, 25% Psych
Allocation	1:1 within tier (AI vs Control)
Inside vs Outside discipline	Undergrads always inside; 30% of roster assigned outside (selected from non-UG)
Variance estimator	HC3 heteroskedasticity-robust SEs
Vertical: control success (UG; P)	40% ; 55–60% (assumed)
Vertical: baseline gap (UG vs Prof)	15–20 pp (assumed)
Vertical: detectable compression	≈ 40% of gap @ 80% power, $\alpha = 0.05$
Vertical: baseline major/minor error rates	Major 0.40–0.50; Minor 0.80–1.00 (counts per participant, illustrative)
Horizontal: OOD penalty (control)	20–25 pp (assumed)
Horizontal: detectable AI reduction (Success)	≈ 8–10 pp @ 80% power, $\alpha = 0.05$
Horizontal: detectable AI reduction (Major errors)	≈ 0.15–0.20 fewer errors @ 80% power (illustrative)
Non-coding baseline (Appropriate; Overall 0–5)	45–55% ; mean 3.0–3.3 (assumed)
Robustness execution baseline	55–60% 1 check; 30–35% 2 checks (assumed)
Multiplicity	Tests conducted separately within vertical and horizontal families
Small-cluster inference	Wild-cluster bootstrap if clusters < 30 (9,999 reps)

6 Outcomes and Measurement

Participants receive a discipline-tagged article, a standardized instruction sheet, and a data/code package (as available) mirroring the original study's setup. They are asked to reproduce a pre-specified focal result and to document their workflow. At the end of the seven-hour window, participants submit (i) a final result file (tables/figures or numeric outputs), (ii) executable code and a short README describing the steps taken, (iii) an error log listing any major and minor coding issues identified (with file and line references when possible), and (iv) a referee-style assessment of the credibility and clarity of the reproduced evidence. The referee narrative is capped at 1,000 words and deliberately omits standalone sections on data/code availability, computational reproducibility, or code quality; participants note any blockers inside the summary or recommendations instead. Human graders (blinded to treatment and discipline) use a rubric to classify success and error detection, and to score the referee Appropriateness indicator together with an overall 0–5 assessment. AI-assisted grading (for the AI referee outcomes) follows the same prompts and rubric; Appendix "AI Referee Prompt" reproduces the exact instructions, including required outputs and scoring anchors.

We standardize these deliverables with the empty workbook `Reports/Replication_Log_Referee_Template.xlsx`, which participants fill as they work. Sheet `00_Main` captures session metadata (participant name, article identifiers, software, event, discipline tags, and the out-of-discipline flag). Sheet `01_CodingErrors` provides the structured log for major/minor issues, including timestamp, affected element, narrative justification, evidence pointer, and a minor/major classification toggle. Sheet `02_Computation` records whether the focal result was reproduced and when. Sheet `03_Robustness` allocates parallel columns for up to two robustness checks—each with motives, specification changes, original versus reproduced estimates, and interpretation—enforcing the "maximum two" rule in the protocol. Sheet `04_Referee_Report` houses the scored rubric, pairing the binary/0–5 entries with prompts limited to five narrative sections (summary, design/identification, robustness, ethics/transparency, overall assessment) and a reminder of the 1,000-word cap so that the exercise remains focused on substantive evaluation rather than coding style. Aligning the reporting template with the PAP ensures fields are named consistently across sites and can be ingested without ad-hoc cleaning.

We organize outcomes into primary and secondary categories to align directly with our hypotheses and to keep inference focused. Primary outcomes capture whether participants reproduced the pre-specified result (level), how long it took to first achieve a reproduction (timing), their ability to detect coding errors (major and minor), and performance on the structured referee rubric (Appropriateness, five content components, and an overall score). These outcomes together reflect the core goals of the exercise: getting to the right result, getting there efficiently, avoiding substantive mistakes, and communicating clearly.

Secondary and exploratory outcomes provide contextual texture and mechanisms. In particular, we summarize robustness proposals (1 and 2 checks) alongside indicators for whether participants actually implemented at least one or two qualifying checks. These measures clarify whether AI support changes both planning and execution. Finally, two pre-specified moderators (self-reported years of coding and prior AI usage) enter as covariates in the main models and, where noted, as separate heterogeneity analyses; they are not outcomes themselves. The table below consolidates definitions, types, and assessment sources for quick reference; the analyses and figures throughout the plan use these exact definitions.

Table 3: Outcomes and measurements (primary and secondary).

Category	Outcome	Type	Measurement / Assessment
Primary (Coding)	Success	Binary	Core result reproduced by endline (yes/no).
	Error detection — major	Count	Number of major coding errors correctly identified (pre-defined rubric).
	Error detection — minor	Count	Number of minor coding errors correctly identified (pre-defined rubric).
Primary (Non-coding)	Referee appropriateness (human)	Binary	Appropriate vs. Not appropriate; human judges (blinded).
	Referee overall 0–5 (human)	0–5	Holistic 0–5 assessment; human score averaged across judges.
	Referee appropriateness (AI)	Binary	Appropriate vs. Not appropriate; AI judge using mirrored rubric.
	Referee overall 0–5 (AI)	0–5	Holistic 0–5 assessment from AI judge using mirrored rubric.
	Robustness planned 1	Binary	Indicator for proposing at least one qualifying robustness check (rubric-based).
	Robustness planned 2	Binary	Indicator for proposing at least two qualifying robustness checks.
	Robustness implemented 1	Binary	Indicator for executing at least one qualifying robustness check with code and outputs.
	Robustness implemented 2	Binary	Indicator for executing at least two qualifying robustness checks with code and outputs.
Secondary	Learning (treatment × event order)	Interaction	Assesses whether treatment effects evolve across repeated events.
Moderator	Years of coding	Continuous	Self-reported; used as moderator (not an outcome).
	Prior AI usage	Categorical	Self-reported; included as precision control.

We pre-define classification of “major” versus “minor” errors and use a standardized grading rubric. As a preview of magnitudes, the table below (Table 4) reports simple means (and standard deviations) by arm, together with Welch tests for the difference in means. Figures then visualize the core outcomes that the analysis will focus on (Figure 1).

The figures referenced here are predefined mock-ups of the panels we will produce once data are available; they are rendered using synthetic or placeholder data solely to illustrate the visual layout and expected content. Their purpose is to communicate the intended presentation of our preregistered estimands rather than to reveal any substantive pattern at this stage.

Table 4: Comparison of Human-Only and AI-Assisted Metrics

			Human-Only
			vs
Variable	**Human-Only**	**AI-Assisted**	**AI-Assisted**
	0.468	0.505	-0.036
Reproduction	(0.502)	(0.502)	[0.627]
	0.367	0.286	0.081
Number of minor errors	(0.644)	(0.514)	[0.357]
	0.139	0.200	-0.061
Number of major errors	(0.348)	(0.508)	[0.337]
	0.329	0.438	-0.109
Planned 1 robustness check	(0.473)	(0.499)	[0.132]
	0.165	0.162	0.003
Planned 2 robustness checks	(0.373)	(0.370)	[0.962]
	0.215	0.324	-0.109
Implemented 1 robustness check	(0.414)	(0.470)	[0.098]
	0.089	0.105	-0.016
Implemented 2 robustness checks	(0.286)	(0.308)	[0.714]
	0.367	0.429	-0.061
Appropriate (human)	(0.485)	(0.497)	[0.401]
	3.093	3.226	-0.133
Overall 0–5 (human)	(0.850)	(0.787)	[0.281]
	0.481	0.448	0.033
Appropriate (AI)	(0.503)	(0.500)	[0.655]
	3.081	3.211	-0.130
Overall 0–5 (AI)	(0.947)	(0.831)	[0.334]

Note: Columns 2–3 present means and standard deviations in parentheses for the two arms; column 4 presents the difference in means (Human-Only − AI-Assisted) and two-sided Welch p-values in brackets.

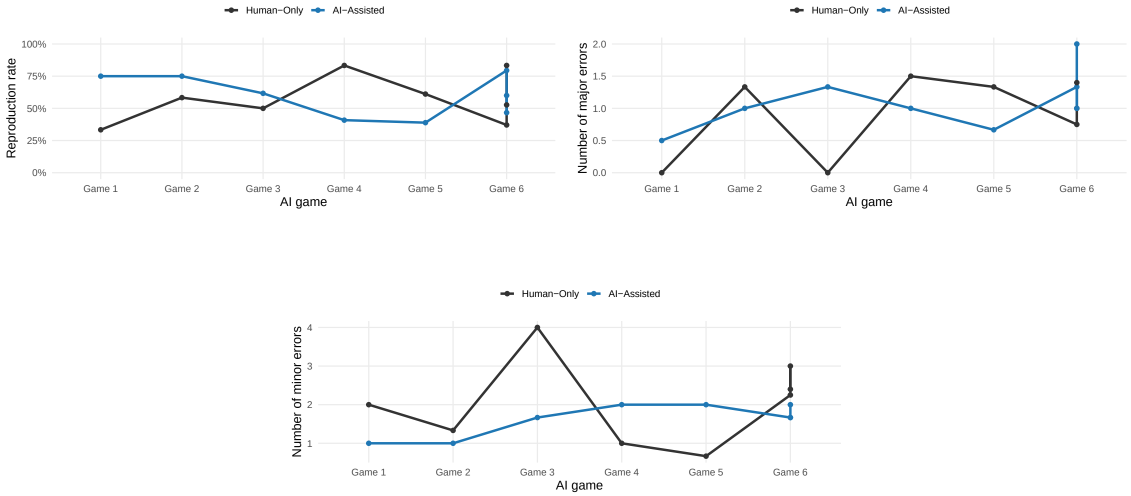

Figure 1: Primary outcomes (levels): reproduction and coding errors. Notes: Panels display reproduction rates (raw) alongside major and minor error counts. Difference-style plots are intentionally omitted.

6.1 Grading Rubric (Operational Definitions)

Coding skills - Success (binary): “Yes” when the participant’s final output matches the pre-specified focal result within documented tolerance (numeric threshold or visual equivalence for figures), with code that runs cleanly to produce the output. Partial or alternative results do not count. - Major error (count): Any coding/data/model issue that, once corrected, changes the focal result’s sign, statistical significance, or substantive interpretation; for example a wrong sample filter or an omitted transformation that alters the reported effect size. Issues that could matter but do not demonstrably affect the focal estimate are coded as minor. - Minor error (count): Formatting, non-substantive code, or reproducibility issues that do not change the focal result. Examples: mislabeled axes; non-deterministic seed; inefficient code without impact on estimate.

Non-coding skills - Appropriateness (binary): “Appropriate” when the report accurately identifies whether the reproduced evidence supports the original claim, flags substantive issues if present, and substantiates claims with code/snippets or references to outputs. - Overall (0–5): Holistic assessment of the referee report quality (not a mechanical average of component scores). Anchors follow the referee rubric’s 0–5 scale, where 0 denotes missing/incorrect and 5 denotes comprehensive, well-argued feedback. - Planned robustness checks: Indicators capturing whether the participant proposes at least one or at least two robustness checks that satisfy the pre-specified rubric (clear motivation, documented specification change, comparison to the focal estimate, and concise interpretation). - Implemented robustness checks: Indicators capturing whether the participant successfully executes at least one or at least two qualifying checks, documented with code and resulting estimates.

Human graders apply these definitions blind to treatment and discipline; disagreements are reconciled via consensus. The same rubric is prompted to the AI judge for the AI-referee outcomes.

We will interpret Figure 1 as the primary visualization of coding outcomes (reproduction and error rates) with accompanying usage context. Difference-style and cumulative milestone panels are omitted to focus attention on the preregistered estimands.

7 Controls (Covariates and Stratification)

Table 5 summarizes the covariates and fixed effects we pre-specify for precision and design alignment. We use a compact, stable set that mirrors the randomization scheme and absorbs systematic heterogeneity without overfitting.

We include a small, pre-specified set of controls to improve precision and absorb systematic differences that are not of direct interest. Stratification by expertise tier (Undergraduate, Graduate, Professor/Researcher) reflects our design and is included as dummies in the main specification so that treatment effects are identified within tier; the interaction with treatment captures heterogeneous effects along the expertise ladder. Event and article fixed effects absorb site- and task-specific differences. Software indicators (R/Stata/Python) capture baseline workflow differences across toolchains. Finally, self-reported years of coding and prior AI familiarity improve precision and help stabilize estimates across events. Years of coding also serves as the pre-specified continuous moderator in secondary analyses; the resulting interaction terms do not alter the main ITT estimands.

Table 5: Controls, fixed effects, and moderators (pre-specified).

Variable	Role	Type	Coding_or_Levels	Notes
Expertise tier	Stratification; control	Categorical	Undergrad, Graduate, Professor/Researcher	Tier dummies in main specs; Graduate = MA/PhD, Professor/Researcher = postdoc/faculty; interacted with AI (heterogeneity).
Event	Fixed effect	Categorical	One FE per event	Absorbs site/time differences (not a parameter of interest).
Article	Fixed effect	Categorical	One FE per task/article	Absorbs task-specific difficulty/fit (not a parameter).
Software	Control	Categorical	R, Stata, Python	Preferred software indicator (workflow baseline).
Years of coding	Control; moderator	Continuous	Self-reported years	Improves precision; interacted with AI in secondary (yrs×AI).
Prior ChatGPT familiarity	Control	Categorical	None, Some, Heavy	Self-reported familiarity with ChatGPT/AI tools.
Variance estimator	Estimation setting	—	HC3 heteroskedasticity-robust SEs	Used for all reported models.

8 Statistical Analysis Plan

This section lays out our estimands and modeling approach for the two preregistered dimensions of interest—vertical (expertise tier) and horizontal (out-of-discipline, OOD)—and clarifies how outcome types map to link functions and fixed-effect structure. We begin with intent-to-treat (ITT) specifications and then describe the interaction terms that capture compression: AI × tier for vertical and AI × OOD for horizontal. Event×article fixed effects absorb site/tool and task heterogeneity; all models report HC3 heteroskedasticity-robust standard errors. We control multiplicity separately within the vertical and horizontal families using Holm's method at = 0.05.

Let Y_i denote a pre-specified outcome (reproduction success; counts of minor or major coding errors; referee appropriateness; referee overall score; robustness-check indicators). We estimate ITT effects in a tier-interaction framework that allows the AI effect to vary with expertise. Formally,

$$Y_i \;=\; \beta_0 \;+\; \beta_1 A_i \;+\; \sum_s \gamma_s\, \mathbb{1}\{\text{Tier}_i = s\} \;+\; \sum_s \delta_s\, A_i\, \mathbb{1}\{\text{Tier}_i = s\} \;+\; X_i'\theta \;+\; \lambda_{e(i)\times s(i)} \;+\; \varepsilon_i,$$

where A_i is the treatment indicator, $\lambda_{e(i)\times s(i)}$ are event-by-software fixed effects, and X_i collects the pre-specified controls (years of coding and prior AI familiarity). Binary outcomes (reproduction, referee appropriateness, robustness indicators) are estimated with linear probability models; continuous outcomes (overall 0–5 scores) use OLS; and counts (minor/major errors) use a Poisson GLM with a log link:

$$\mathbb{E}[Y_i \mid \cdot] \;=\; \exp\!\left(\beta_0 + \beta_1 A_i + \sum_s \gamma_s\, \mathbb{1}\{\text{Tier}_i = s\} + \sum_s \delta_s\, A_i\, \mathbb{1}\{\text{Tier}_i = s\} + X_i'\theta + \lambda_{e(i)\times s(i)}\right).$$

Horizontal compression estimands. Let D_i denote an indicator for receiving a task outside the participant's primary discipline (OOD). We test whether AI reduces the OOD penalty by augmenting the main models with D_i and $A_i \times D_i$: $Y_i = \cdots + \beta_2 D_i + \beta_3(A_i \times D_i) + \varepsilon_i$. The coefficient β_3 captures horizontal compression (a smaller penalty, or a gain, when outside). Article indicators absorb discipline effects because tasks carry a single discipline tag.

Interpretation and outcome-specific notes. In all tables, the "AI-Assisted" row reports the ITT contrast for participants in the baseline tier or inside-discipline case; the interaction rows report how that contrast varies along tiers (vertical) or OOD (horizontal). For counts, we check over-dispersion (deviance/df) and, if material, re-estimate with a negative binomial link as a robustness check.

Multiplicity. We pre-register two primary outcome families and control the family-wise error rate within each using Holm's method at $\alpha = 0.05$: (i) Coding family: reproduction, minor errors, and major errors in the tier-interaction and OOD-interaction models (AI main and AI×moderator). (ii) Non-coding family: referee appropriateness (human and AI), referee overall scores (human and AI), and robustness execution (1, 2 checks). Years-of-coding heterogeneity, usage measures, and other ancillary analyses are secondary.

To assess the success of randomization and support the modeling choices, we report balance on all individual-level controls used in the specification. The table below (Table 6) shows means (standard deviations) by arm and Welch tests for the difference in means.

Table 6: Balance of Participant Characteristics by Arm

Variable	Human-Only	AI-Assisted	Human-Only vs AI-Assisted
Years of coding	4.787 (2.775)	4.675 (2.819)	0.111 [0.747]
Tier: Undergraduate	0.218 (0.414)	0.237 (0.427)	-0.019 [0.720]
Tier: Graduate	0.429 (0.497)	0.435 (0.498)	-0.007 [0.915]
Tier: Professor/Researcher	0.353 (0.480)	0.328 (0.471)	0.025 [0.668]
Software: R	0.353 (0.480)	0.344 (0.477)	0.010 [0.867]
Software: Stata	0.368 (0.484)	0.344 (0.477)	0.025 [0.674]
Software: Python	0.278 (0.450)	0.313 (0.465)	-0.035 [0.538]
Prior ChatGPT familiarity: None	0.391 (0.490)	0.344 (0.477)	0.047 [0.426]
Prior ChatGPT familiarity: Some	0.323 (0.470)	0.359 (0.481)	-0.035 [0.545]
Prior ChatGPT familiarity: Heavy	0.286 (0.453)	0.298 (0.459)	-0.012 [0.831]

Note: Means and standard deviations in parentheses by arm; difference column shows Human-Only − AI-Assisted and two-sided Welch *p*-values in brackets. All variables are individual-level controls used in the models.

In all models we use HC3 heteroskedasticity-robust standard errors. Binary outcomes (reproduction, referee appropriateness, robustness indicators) are estimated with linear probability models; counts of major/minor errors rely on Poisson specifications; and the continuous referee overall score uses OLS. We report coefficient estimates with 95% confidence intervals for the main effect β_1 and the tier interactions δ_s, and we conduct the pre-specified compression test on the interaction terms.

Heterogeneity and secondary analyses follow two paths. First, we replace tier dummies with a continuous moderator (years of coding) interacted with treatment to trace a dose–response. Second, within the AI arm only, we relate outcomes to self-reported usage intensity (prompts/files/images/words, entered as inverse hyperbolic sine transforms) to characterize under- and over-use; these are descriptive and do not alter the main ITT estimands. We also consider event-order interactions to gauge learning across events.

9 Results

For clarity and symmetry, we present vertical (tier-based) estimates first, followed by parallel horizontal (AI × out-of-discipline) estimates using the same outcomes and table layout. Figures and descriptives mirror this sequence.

We present results in two blocks that align with the preregistered outcome families—coding skills and non-coding skills—with vertical (tier) estimates first and horizontal (OOD) contrasts second. Throughout, standard errors rely on HC3 heteroskedasticity-robust corrections, and the coefficients are displayed with confidence intervals plus the pre-specified compression test for the interaction terms.

Coding outcomes (Table 7) collect reproduction success together with the counts of minor and major coding errors. The AI coefficient speaks to H1 (average effect), while the tier interactions speak to H2 (compression). We interpret effect magnitudes jointly to assess whether AI shifts both success rates and error detection in a consistent direction. Table 8 mirrors this layout for the out-of-discipline estimands (H2b).

Table 7: Coding outcomes: vertical ITT effects with tier interactions. HC3 heteroskedasticity-robust standard errors.

	(1)	(2)	(3)
	Reproduction	Minor errors	Major errors
AI-Assisted	0.187	-0.071	1.132*
	(0.119)	(0.465)	(0.658)
	[-0.047; 0.421]	[-0.982; 0.840]	[-0.158; 2.422]
AI × Graduate	-0.170	0.186	-1.879*
	(0.152)	(0.590)	(1.003)
	[-0.470; 0.130]	[-0.971; 1.343]	[-3.844; 0.087]
AI × Professor/Researcher	0.032	0.259	-1.462*
	(0.160)	(0.569)	(0.810)
	[-0.283; 0.346]	[-0.856; 1.374]	[-3.049; 0.124]
Controls	✓	✓	✓
Mean of dep. var	0.485	0.356	0.159
p-val (Monotonic compression)	0.918	0.436	0.975
Obs.	264	264	264

Note: Standard errors in parentheses; confidence intervals in brackets.

Controls: Event×article FE; years of coding; software; prior AI familiarity.

Compression (monotonic): one-sided p-value for increasing effects across the three tiers (baseline: Undergraduate).

$^{*}p < 0.10$, $^{**}p < 0.05$, $^{***}p < 0.01$

Definition of robustness columns (used below): “Planned 1/ 2” indicate the share of participants proposing qualifying checks under the rubric; “Implemented 1/ 2” track the share that executed those checks with documented code and estimates.

We complement the tier-based estimates with horizontal contrasts that interact AI access with the out-of-discipline indicator (Table 8). These estimates test whether AI mitigates the OOD penalty in coding outcomes (H2b).

Table 8: Coding outcomes: horizontal AI × outside-of-discipline effects. HC3 heteroskedasticity-robust standard errors.

	(1) Reproduction	(2) Minor errors	(3) Major errors
AI-Assisted	0.339***	0.156	0.230
	(0.096)	(0.309)	(0.505)
	[0.151; 0.528]	[-0.450; 0.761]	[-0.761; 1.221]
AI × Outside-discipline	-0.339***	-0.154	-0.382
	(0.121)	(0.425)	(0.652)
	[-0.577; -0.100]	[-0.987; 0.680]	[-1.659; 0.895]
Controls	✓	✓	✓
Mean of dep. var	0.485	0.356	0.159
Obs.	264	264	264

Note: Standard errors in parentheses; confidence intervals in brackets.
Controls: Event×article FE; years of coding; software; prior AI familiarity.
$^{*}p < 0.10$, $^{**}p < 0.05$, $^{***}p < 0.01$

Non-coding outcomes (Table 9) track the communication and robustness dimension—referee appropriateness and overall scores from both human and AI judges, alongside indicators for planning at least one or two robustness checks and actually implementing them (H3). Table 10 repeats the analysis for the OOD contrasts.

Table 9: Non-coding outcomes: vertical ITT effects on referee judgments and robustness execution. HC3 heteroskedasticity-robust standard errors.

	(1) Appropriate (human)	(2) Overall 0–5 (human)	(3) Appropriate (AI)	(4) Overall 0–5 (AI)	(5) At least 1 check	(6) At least 2 checks	(7) Implemented 1	(8) Implemented 2
AI-Assisted	0.044	0.210	-0.102	0.159	-0.052	0.176*	0.008	0.149**
	(0.129)	(0.226)	(0.133)	(0.243)	(0.131)	(0.091)	(0.114)	(0.068)
	[-0.210; 0.297]	[-0.236; 0.655]	[-0.364; 0.159]	[-0.321; 0.639]	[-0.311; 0.207]	[-0.002; 0.354]	[-0.216; 0.232]	[0.015; 0.284]
AI × Graduate	0.059	0.243	0.145	0.343	0.118	-0.155	0.131	-0.063
	(0.165)	(0.264)	(0.166)	(0.280)	(0.160)	(0.117)	(0.140)	(0.094)
	[-0.267; 0.384]	[-0.277; 0.763]	[-0.183; 0.473]	[-0.210; 0.895]	[-0.198; 0.434]	[-0.386; 0.076]	[-0.145; 0.406]	[-0.249; 0.122]
AI × Professor/Researcher	-0.145	-0.079	-0.043	-0.026	0.302*	-0.048	0.209	-0.133
	(0.166)	(0.281)	(0.172)	(0.295)	(0.170)	(0.122)	(0.157)	(0.095)
	[-0.472; 0.181]	[-0.633; 0.476]	[-0.382; 0.295]	[-0.607; 0.555]	[-0.032; 0.637]	[-0.288; 0.193]	[-0.101; 0.518]	[-0.320; 0.053]
Controls	✓	✓	✓	✓	✓	✓	✓	✓
Mean of dep. var	0.443	3.178	0.462	3.190	0.360	0.170	0.254	0.110
p-val (Monotonic compression)	0.927	0.932	0.906	0.949	0.270	0.953	0.306	0.918
Obs.	264	264	264	264	264	264	264	264

Note: Standard errors in parentheses; confidence intervals in brackets.
Controls: Event×article FE; years of coding; software; prior AI familiarity.
Compression (monotonic): one-sided p-value for increasing effects across the three tiers (baseline: Undergraduate).
$^{*}p < 0.10$, $^{**}p < 0.05$, $^{***}p < 0.01$

Table 10: Non-coding outcomes: horizontal AI × outside-of-discipline effects. HC3 heteroskedasticity-robust standard errors.

	(1) Appropriate (human)	(2) Overall 0–5 (human)	(3) Appropriate (AI)	(4) Overall 0–5 (AI)	(5) At least 1 check	(6) At least 2 checks	(7) Implemented 1	(8) Implemented 2
AI-Assisted	0.019	0.202	-0.093	0.258	0.133	0.204**	0.227**	0.119*
	(0.106)	(0.164)	(0.101)	(0.174)	(0.105)	(0.079)	(0.092)	(0.069)
	[-0.189; 0.228]	[-0.121; 0.525]	[-0.293; 0.106]	[-0.084; 0.600]	[-0.074; 0.341]	[0.048; 0.360]	[0.045; 0.409]	[-0.016; 0.255]
AI × Outside-discipline	-0.007	0.127	0.065	0.058	-0.047	-0.173*	-0.139	-0.070
	(0.130)	(0.203)	(0.129)	(0.212)	(0.129)	(0.100)	(0.117)	(0.085)
	[-0.263; 0.249]	[-0.272; 0.526]	[-0.190; 0.320]	[-0.359; 0.475]	[-0.301; 0.208]	[-0.370; 0.023]	[-0.369; 0.091]	[-0.237; 0.097]
Controls	✓	✓	✓	✓	✓	✓	✓	✓
Mean of dep. var	0.443	3.178	0.462	3.190	0.360	0.170	0.254	0.110
Obs.	264	264	264	264	264	264	264	264

Note: Standard errors in parentheses; confidence intervals in brackets.
Controls: Event×article FE; years of coding; software; prior AI familiarity.
$^{*}p < 0.10$, $^{**}p < 0.05$, $^{***}p < 0.01$

Finally, we probe robustness to alternative outcome definitions and specifications. For compactness, the main results table also reports robustness columns (1 and 2 checks), which implement the pre-specified threshold variations. We expect sign and order-of-magnitude stability, with shifts that are interpretable given the alternative codings. These checks complement the design-based safeguards (stratification, fixed effects, pre-specified controls) and help establish that the main conclusions are not an artifact of a particular functional form or threshold.

10 Data Management, Documentation, and Ethics

We will publish de-identified participant-level data, code, and grading rubrics on OSF and GitHub upon acceptance (view-only earlier when necessary). Sensitive logs will be redacted according to the consent form. The study will obtain institutional ethics approval prior to data collection. Any deviations from protocol will be preregistered before accessing outcome data.

Static study materials live inside this repository and will be version-locked before launch: curated task bundles in `Papers/`, the pre-game training deck in `Pre game/`, and the reporting workbook in `Reports/`. Repository history provides provenance for any updates (for example, refreshed replication packages or revised slides); the commit hash distributed to sites is recorded alongside the randomization seed. The same directory structure will be mirrored in the OSF archive so external teams can audit exactly what participants received.

Audio recordings and notes from the post-pilot focus groups are stored on encrypted I4R drives with filenames keyed to anonymous participant IDs and session arm (AI vs. control). Transcription software with local processing is used where feasible; otherwise, trained staff transcribe manually. Only the qualitative analysis team accesses the raw audio. De-identified excerpts linked to codebook categories are released alongside the quantitative replication package after redacting personally identifiable information and any mention of other participants' outputs.

11 Timeline and Deliverables

We plan six one-day events across partner institutions within the academic year, spaced to avoid overlap and allow consistent staffing. Events are pre-announced with a shared protocol: pre-generated randomization within tiers and a seven-hour work window. Each individual may participate in at most one event; duplicate registrations are blocked at check-in. We lock randomization and materials ahead of time and document any deviations (no-shows, substitutions) prior to analysis.

Event and article selection. Events are hosted by partner institutions with capacity for proctoring and secure data handling. Articles are selected from the I4R pipeline to span the three disciplines and software ecosystems (R/Stata/Python), drawing on the catalog documented in Brodeur et al. (2025). Selection prioritizes clarity of a focal result and a mix of tasks with and without known coding pitfalls; each article is tagged to a single discipline before the lock. Because assignments draw randomly from this pool, we do not target a fixed share of tasks with seeded errors within an event. We aim for temporal balance across the academic year to minimize tool-version drift; any updates to AI tooling are logged.

After the sixth event, we finalize the preregistration lock and freeze all code paths before accessing outcome data. The analysis phase proceeds in two stages. First, we produce the pre-specified main results and figures, checking internal coherence and documenting data lineage. Second, we generate the pre-specified secondary and appendix tables to illuminate mechanisms and robustness. All outputs are cross-validated against the preregistered estimands and data checks.

Deliverables include a public replication archive (de-identified individual-level data, code, and grading rubrics), a pre-analysis report that summarizes the locked design and main estimands, and a manuscript integrating results and interpretation. We aim to share preliminary results with partners quickly after the final event and proceed to manuscript submission once the replication archive is complete.

12 Limitations

Despite stratified randomization and event-by-software fixed effects, external validity remains a key limitation. Participating institutions, topics, and software ecosystems may not reflect the broader population of replication exercises or research teams. We minimize site-specific artifacts by controlling for event–software cells and by standardizing instructions and grading rubrics, but context still matters for both the baseline rates and the scope for AI assistance.

Measurement and compliance present additional challenges. Although we combine pledges, spot checks, and audits to monitor AI usage, some noncompliance in the control arm or heterogeneous usage quality in the treatment arm is inevitable. We pre-specify strategies to document and, when necessary, bound any bias (e.g., per-protocol and IV sensitivity), but these strategies trade robustness for different assumptions. Grading, while rubric-based, can also admit residual subjectivity; we address this with clear definitions, double-checks, and rater consensus when needed.

Finally, the evolving nature of AI tools introduces temporal drift. Model updates can affect both capability and interface, potentially shifting the level and composition of gains even with identical prompts. We log model versions and timing, keep baseline instructions identical across events, and emphasize design features (e.g., within-tier randomization) that stabilize inference. Nevertheless, any broader extrapolation should consider how quickly the technology landscape changes and whether the tasks studied here generalize to other domains or longer-horizon research workflows.

13 Appendix

These additional analyses extend and contextualize the main results without altering the primary estimands. We examine moderators beyond the tier-based interactions via a continuous years-of-coding interaction (Tables 11 and 12) and report both planned and implemented robustness activity in the appendices. These tables are not substitutes for the main ITT estimands; they are meant to clarify mechanisms and the consistency of patterns.

13.1 AI Referee Prompt

The AI grader receives a standalone instruction each time it evaluates a submission. The block below reproduces the full prompt, including the required output structure and scoring anchors.

Role and objective You are "AI Referee," an expert evaluator for the AI Replication Games. Your task is to read a participant's reproduction package and produce referee-style assessments that mirror the human rubric.

Context - You are judging a replication of a published empirical result. - The participant had access to the original paper, its replication package, and a seven-hour work window. - The materials you receive include the participant's referee report (plain text), a README describing the reproduction steps, a log of detected errors, and any supporting tables or figures they produced.

General instructions - Work independently—do not assume facts that are not present in the materials. - Keep your narrative within 1,000 words and avoid standalone sections on data/code availability, computational reproducibility, or code quality; note any such blockers inside the summary or recommendation text. - If critical evidence is missing, record that as a weakness and reflect it in the scores. - Base every judgment on the participant's reasoning and evidence; parroting the source paper without evaluation should be penalized. - When assigning numeric scores, use the full 0–5 range (no half points). - For each scored dimension, write a short justification (two to four sentences) that cites the relevant portion of the participant's submission (quote, paraphrase, or reference to a table or figure).

Evaluation tasks 1. Appropriateness (binary). Return "Appropriate" if the participant correctly characterizes whether the reproduced evidence supports the original claim, flags material discrepancies, and grounds the conclusion in the submitted code or outputs. Otherwise return "Not appropriate". 2. Summary accuracy (0–5). Score the fidelity of the participant's summary of the target paper and reproduced result. 3. Literature placement (0–5). Score how well the participant situates the study within the broader literature, including whether comparisons to related work are accurate and meaningful. 4. Weakness diagnosis (0–5). Score identification and prioritization of substantive weaknesses or threats to validity in the reproduced analysis. 5. Recommendations (0–5). Score the specificity, feasibility, and usefulness of suggested robustness checks or next steps. 6. Clarity (0–5). Score organization, tone, and readability of the write-up. 7. Overall assessment (0–5). Provide a holistic rating that reflects the report's usefulness to an editor deciding on publication, not a mechanical average of the components.

Anchors for 0–5 scales (apply to tasks 2–7) - 0 = Missing, wholly incorrect, or incoherent. - 1 = Substantially inaccurate, with major misunderstandings that would mislead an editor. - 2 = Partially correct but incomplete or containing notable inaccuracies that reduce usefulness. - 3 = Adequate: correct core points with limited depth or weak justification. - 4 = Strong: accurate, well-supported, and actionable with only minor omissions. - 5 = Exceptional: precise, comprehensive, and offering original insight or particularly valuable guidance.

Output format Return JSON with the following fields: `appropriateness` ("Appropriate" or "Not appropriate"), `summary_accuracy`, `literature_placement`, `weakness_diagnosis`, `recommendations`, `clarity`, `overall`, `justifications`, and `notes`. The `justifications` object must contain keys for each of the seven tasks above whose values are the explanatory paragraphs. Include a single string in `notes` describing any missing information, suspected hallucinations, or uncertainties.

```json
{
  "appropriateness": "Appropriate",
  "summary_accuracy": 4,
  "literature_placement": 3,
  "weakness_diagnosis": 5,
  "recommendations": 4,
  "clarity": 3,
  "overall": 4,
  "justifications": {
    "appropriateness": "...",
    "summary_accuracy": "...",
    "literature_placement": "...",
    "weakness_diagnosis": "...",
    "recommendations": "...",
    "clarity": "...",
    "overall": "..."
  },
  "notes": "..."
}
```

Quality control - Double-check that every field is filled. - If evidence is missing for a dimension: - set the score to the literal value `null`, - explain the omission in the notes field, and - flag the gap in the corresponding justification. - Do not invent references or cite external material. - Use only the materials provided with the submission. - Return only the JSON object—no additional prose after the closing brace.

Table 11: Coding outcomes with years-of-coding moderator (interaction with AI).

	(1)	(2)	(3)
	Reproduction (yrs)	Minor (yrs)	Major (yrs)
AI-Assisted	0.045	-0.908**	0.758
	(0.118)	(0.422)	(0.755)
	[-0.187; 0.277]	[-1.734; -0.082]	[-0.722; 2.237]
AI × Years of coding	0.017	0.207***	-0.153
	(0.022)	(0.071)	(0.134)
	[-0.027; 0.060]	[0.068; 0.346]	[-0.415; 0.109]
Controls	✓	✓	✓
Mean of dep. var	0.485	0.356	0.159
Obs.	264	264	264

Note: Standard errors in parentheses; confidence intervals in brackets.
Controls: Event×article FE; software; prior AI familiarity.
$^{*}p < 0.10$, $^{**}p < 0.05$, $^{***}p < 0.01$

Table 12: Non-coding outcomes with years-of-coding moderator (interaction with AI).

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
	Appropriate (human) (yrs)	Overall 0–5 (human) (yrs)	Appropriate (AI) (yrs)	Overall 0–5 (AI) (yrs)	Implemented 1 (yrs)	Implemented 2 (yrs)	Planned 1 (yrs)	Planned 2 (yrs)
AI-Assisted	-0.021	0.685***	-0.078	0.717***	0.017	0.161**	-0.089	0.080
	(0.119)	(0.196)	(0.125)	(0.212)	(0.112)	(0.065)	(0.123)	(0.082)
	[-0.255; 0.213]	[0.299; 1.070]	[-0.325; 0.169]	[0.300; 1.135]	[-0.203; 0.238]	[0.032; 0.290]	[-0.332; 0.153]	[-0.081; 0.242]
AI × Years of coding	0.008	-0.084**	0.005	-0.088**	0.025	-0.018	0.040*	0.003
	(0.022)	(0.034)	(0.023)	(0.038)	(0.021)	(0.013)	(0.022)	(0.016)
	[-0.035; 0.052]	[-0.152; -0.016]	[-0.040; 0.051]	[-0.163; -0.013]	[-0.017; 0.067]	[-0.043; 0.007]	[-0.004; 0.084]	[-0.030; 0.035]
Controls	✓	✓	✓	✓	✓	✓	✓	✓
Mean of dep. var	0.443	3.178	0.462	3.190	0.254	0.110	0.360	0.170
Obs.	264	264	264	264	264	264	264	264

Note: Standard errors in parentheses; confidence intervals in brackets.
Controls: Event×article FE; software; prior AI familiarity.
$^{*}p < 0.10$, $^{**}p < 0.05$, $^{***}p < 0.01$

14 References

References render from `references.bib`. We will cite prior AI Replication Games and related methodology upon registration finalization.

Brodeur, Abel, Ghina Abdul Baki, Juan P. Aparicio, Bruno Barbarioli, Lenka Fiala, Derek Mikola, and David Valenta. 2025. "AI-Replication Games." I4R-DP195. Institute for Replication. https://econstor.eu/bitstream/10419/308508/1/I4R-DP195.pdf.