Renting Intelligence: Student Responses to an Exogenous Downgrade in AI Model Quality*

Alexander Quispe† Kevin Xu‡

July 23, 2026

Abstract

On March 12, 2026, GitHub moved verified students from free Copilot Pro to a restricted "Copilot Student" plan, removing access to premium AI models while retaining base models and the autonomous coding agent — a rare policy-imposed shock to the *quality*, rather than the availability, of AI assistance for one identifiable population. We study student responses in a matched panel of GitHub developers observed monthly from January 2025 to June 2026. The primary sample restricts both arms to developers whose only tracked agentic tooling before the event was Copilot and applies conservative hygiene filters, and requires publicly measurable pre-event output (10,597 developers: 5,432 flagged students and 5,165 nearest-neighbor controls; 190,746 developer-months), so pre-event use of rival agents is balanced at zero by construction. Locally-mediated AI-assisted commits by students fell by −0.181 per month relative to controls (s.e. 0.097), roughly 83 percent of the pre-period mean — an estimate robust to a triple difference against the 2025 academic calendar (−0.163), a clean placebo, and

*Preliminary and exploratory draft, July 2026. Comments welcome. Student status is identified from public GitHub signals pending validation against GitHub Education administrative records; all estimates are intention-to-treat and should be read as lower bounds. This research was made possible by Alexander Quispe's time as a research associate at Microsoft–GitHub during summer 2026.

†Department of Computing and Mathematical Sciences, California Institute of Technology; Microsoft–GitHub. Email: aquisper@caltech.edu.

‡GitHub. Email: khxu@github.com.

stable across nested samples up to the full 13,998-developer panel (-0.155, $p < 0.05$). Delegated agent pull requests — a margin that received a weaker, platform-mediated dose of the policy — show no differential decline, and total output shows no detectable change. The response thus concentrates precisely on the margin whose quality was cut. A direct test of supplier switching finds a precise null: students' post-event adoption of non-Copilot agents (Claude Code, Codex, Cursor, Jules, Devin) matches matched controls' on both extensive and intensive margins — and because the primary sample is Copilot-dependent by construction, the usage decline cannot reflect re-sourcing to rival tools; output shows no detectable loss. Estimates are exploratory: the student flag is heuristic (intent-to-treat), attributed-use measures are lower bounds, and the post-period spans only four months.

JEL Codes: O33, J24, I23

Keywords: artificial intelligence, GitHub Copilot, technology adoption, model quality, students, difference-in-differences

1 Introduction

Two findings dominate the young empirical literature on AI coding assistants, and they pull in opposite directions. Randomized and quasi-experimental evaluations find that access to these tools raises the speed and volume of professional output, with the largest gains for less experienced workers (Peng et al., 2023; Brynjolfsson et al., 2024; Noy and Zhang, 2023; Cui et al., 2025). At the same time, a growing body of work on students warns that the same assistance may substitute for the practice through which skill is built: Strömberg et al. (2025) document that students who lean on AI help accumulate less durable programming ability, a "learning penalty" that would matter most precisely for the population still forming its human capital. Nearly all of this evidence, on both sides, comes from variation in *access* — a worker either has the tool or does not. Yet as AI assistance matures, the policy-relevant margin is increasingly one of *quality*: which model a user is allowed to invoke, at what limits, and at what price. Frontier and base models differ substantially in capability (Dell'Acqua et al., 2023; METR, 2025), and firms now meter that difference through tiered plans. We know almost nothing about how users respond when the quality of their AI assistance is cut while access itself is preserved.

This paper studies a rare policy-imposed quality shock of exactly this form. On March 12, 2026, GitHub replaced the free Copilot Pro subscription it had long extended to verified students with a restricted "Copilot Student" plan. The change removed access to premium frontier models — the Claude and GPT-5.x-class models that had powered the interactive chat and editor surfaces — while retaining base models, unlimited code completions, and, importantly, the autonomous coding agent. The event has three features that make it unusually clean for studying the quality margin. First, it is a downgrade in model quality, not a removal of the tool: every affected student kept a working AI assistant. Second, it hit one identifiable population — verified students — on a single date, with roughly one day of anticipation, leaving otherwise similar non-student developers untouched. Third, the

plan change was itself internally differentiated: the supervised, interactive surfaces lost their premium models, while the delegated coding agent remained included. The same policy therefore degraded one margin of AI-assisted production and spared another, giving us a placebo margin *inside* the treated population.

We measure the response in public activity on GitHub. Because platform telemetry is not observable to researchers, we detect AI-assisted work through two machine-readable fingerprints, in the spirit of recent work identifying agentic production in commit metadata (Quispe, 2026; Hoffmann et al., 2024). Fingerprint A captures delegated production: pull requests authored by the Copilot coding agent on a human developer's behalf, drawn from 1.80 million such PRs between May 2025 and June 2026. Fingerprint B captures supervised production: human-authored commits carrying a `Co-authored-by: Copilot` trailer, drawn from 2.25 million commits between January 2025 and June 2026, with the specific surface (editor, CLI, code review, agent) distinguished by the trailer identity. Plain autocomplete leaves no trace at all, so every usage measure in the paper is an attributed-use *lower bound*: a floor on true AI-assisted activity, not its level. We flag students heuristically from academic email domains, multilingual student self-descriptions in bios, and university organization membership, yielding 21,135 flagged students among 429,177 enriched developers; because the flag is a heuristic awaiting administrative validation, all estimates should be read as intent-to-treat effects. The analysis sample matches 7,000 flagged students to 7,000 nearest-neighbor control developers on pre-period activity, account age, followers, and pre-period agent use, producing a panel of 251,964 developer-months from January 2025 through June 2026 with close covariate balance.

The empirical design is a matched difference-in-differences around March 12, 2026, with developer and month fixed effects and standard errors clustered by developer, reinforced in two ways. A triple difference nets out the 2025 academic calendar, addressing the concern that students always slow down in spring; and a placebo difference-in-differences estimated at the same calendar date in 2025, when no plan change occurred, checks each outcome for

spurious seasonal patterns.

The results line up with the structure of the shock. Students' locally-mediated AI-assisted commits — the margin whose premium models were removed — fall by −0.181 commits per month in the difference-in-differences and −0.163 in the triple difference, against a pre-period treated mean of 0.22: a decline of roughly 83 percent of baseline use, and the placebo at the same date in 2025 is a precise zero (+0.008). The estimates are marginally significant in the primary sample and significant at the five percent level in the larger nested samples, with point estimates stable throughout (−0.155 to −0.186). Delegated agent pull requests — the margin the Student plan retained — show no differential decline (+0.012, standard error 0.026); because the agent's access was not degraded, this null is not a puzzle but a dose-response check that the response tracks the terms of the policy. Total commits and clean (non-AI-attributed) commits show no detectable change, though we emphasize that these estimates are noisy — the standard error is roughly 4.8 commits on a pre-period mean of 28.18 — so we can rule out a collapse in output but not modest effects in either direction. Outcomes for programming-language breadth are inconclusive: the difference-in-differences estimate for distinct languages is positive but fails its own 2025 placebo, so we do not interpret it.

Taken together, the pattern is consistent with substitution rather than short-run harm: usage fell precisely and only on the margin whose quality was cut, while measured output did not detectably decline, suggesting students shifted the displaced work toward base models, other tools, or unassisted effort. On this margin and over this horizon, the evidence cuts against the strong form of the concern in Strömberg et al. (2025) that AI assistance has become load-bearing for student output — but we are careful about what the design can and cannot show. Our post-period spans roughly four months; learning effects of the kind Strömberg et al. (2025) document unfold over years. Our outcomes are quantities, not quality or skill. And our usage floors mean we observe attributed use only, so substitution toward untraceable assistance is observationally equivalent to substitution toward unassisted work.

The paper is exploratory, and we present it as such.

We see three contributions. First, to the empirical economics of AI in knowledge work, which has to date identified effects from access variation (Peng et al., 2023; Noy and Zhang, 2023; Cui et al., 2025; Brynjolfsson et al., 2024), we contribute what is to our knowledge the first quasi-experimental estimate of the response to an exogenous change in AI model *quality* at constant access — the margin along which AI assistance is actually priced and rationed, and the margin most relevant to task-based frameworks in which the capability of the tool determines which tasks it absorbs (Acemoglu and Restrepo, 2022; Autor et al., 2024; Eloundou et al., 2024; Agrawal et al., 2018). Second, we contribute a measurement approach: fingerprint-based detection of AI-assisted work in public commit and pull-request metadata that separates locally-mediated from cloud-delegated production and is explicit about being a lower bound, usable wherever platform telemetry is closed to researchers (Quispe, 2026; Kalliamvakou et al., 2014). Third, we contribute early evidence to the learning-versus-productivity debate for the population at its center: students, disproportionately reliant on free tiers, for whom quality downgrades of this kind are likely to recur as AI pricing matures.

The remainder of the paper proceeds as follows. Section 2 relates the paper to work on AI productivity, AI and learning, and measurement of AI use on GitHub. Section 3 documents the institutional history of Copilot's student offering and the March 2026 plan change. Section 4 describes the fingerprint detection, the student flag, and the matched sample. Section 5 sets out the difference-in-differences, triple-difference, and placebo designs. Section 6 reports the main estimates. Section 7 interprets the margin-specific pattern and discusses substitution possibilities. Section 8 confronts the limitations — the heuristic intent-to-treat flag, the usage floors, and the short window — and Section 9 concludes.

2 Related Literature

This paper sits at the intersection of three literatures: the experimental evidence on AI assistance and worker productivity, the emerging work on AI assistance and human-capital accumulation, and the measurement of AI-assisted production in open-source software. We take from the first literature a robust finding — AI assistance raises short-run output on well-defined tasks — and a limitation: nearly all of it studies the introduction of access, holding quality fixed. We take from the second the central open question our design speaks to. And we take from the third the measurement apparatus that makes an observational quality experiment feasible at scale.

The first strand establishes that generative AI tools raise productivity on knowledge tasks, with effects concentrated among less-experienced workers. In a randomized experiment, Peng et al. (2023) find that access to GitHub Copilot lets developers complete a standardized programming task roughly 56 percent faster. Noy and Zhang (2023) report large speed and quality gains from ChatGPT in professional writing tasks, with the largest gains accruing to initially lower-performing participants; Brynjolfsson et al. (2024) find the same compression of the performance distribution among customer-support agents given access to a conversational assistant. Dell'Acqua et al. (2023) show that the gains are uneven across tasks — a "jagged frontier" inside which AI assistance helps substantially and outside which it can degrade performance — which cautions against extrapolating from a single task. Two more recent contributions complicate the uniformly positive picture. Cui et al. (2025) study Copilot deployment among software developers in field experiments at large firms and find sizable but heterogeneous increases in output, again largest for junior developers. Most strikingly, METR (2025) run a randomized evaluation with experienced open-source developers working on their own mature repositories and find that AI assistance *slowed* them down even as participants believed it sped them up, underlining both task-dependence and the gap between perceived and measured productivity. Across all of these studies, however,

the treatment is binary access to a tool of a given quality. The economic question raised by a tiered market for "rented" intelligence — how demand responds when the *quality* of the model behind an existing tool changes, holding access constant — is, to our knowledge, unstudied. The March 2026 student-plan change provides exactly this variation: premium models were removed for one identifiable population while base models, unlimited completions, and the autonomous coding agent were retained, so the shock perturbs quality on some margins of use and not others within the same product.

The second strand asks whether the short-run productivity gains come at the expense of human-capital accumulation, and it anchors our interpretation. Strömberg et al. (2025) document what they term a learning penalty: students and early-career programmers who lean on AI assistance complete assignments faster but accumulate skills more slowly, with deficits that compound over a horizon of roughly two years. The concern generalizes an older intuition from the economics of prediction machines — when a complement to judgment becomes cheap, workers may under-invest in the skills the machine substitutes for (Agrawal et al., 2018) — and it is sharpest for students, whose current output matters less than the capital they are building. Task-based frameworks make the same point in general form: whether AI augments or displaces human capability depends on which tasks it absorbs and which new tasks it creates for humans (Acemoglu and Restrepo, 2022; Autor et al., 2024), and occupational exposure estimates place software development among the most exposed activities (Eloundou et al., 2024). Our contribution to this strand is a mirror-image design. Where Strömberg et al. (2025) observe students gaining access to assistance and measure subsequent skill formation, we observe students *losing* model quality and measure subsequent production. If the strong form of the learning-penalty view were right — students rent intelligence they cannot replace — an exogenous downgrade should depress their output. We find instead that the usage response concentrates precisely on the degraded margin, with no detectable decline in total or unassisted output over our short post window; the evidence is exploratory and the horizon far shorter than the two years over which Strömberg et al.

(2025) measure deficits, but on this margin it is more consistent with substitution than with dependence.

The third strand supplies the measurement foundation. Large-scale telemetry on who uses AI coding tools, and how, is proprietary; researchers working from public traces must reconstruct usage from the artifacts it leaves. Hoffmann et al. (2024) exploit the staggered rollout of Copilot to open-source maintainers to estimate its effects on contribution behavior, demonstrating that GitHub's public record can support credible quasi-experimental designs. Quispe (2026) develop the commit-trailer fingerprint we build on — machine-readable `Co-authored-by` attribution identifying agentic AI production in commit metadata at scale — and use it to study developers' language portfolios around adoption of an agentic assistant. A longer tradition in empirical software engineering cautions that GitHub traces are a selected and noisy window onto development activity: Kalliamvakou et al. (2014) catalogue the promises and perils of mining GitHub, including private activity, bot accounts, and repositories that are not software projects. We inherit both the opportunity and the caution. Our two fingerprints — bot-authored coding-agent pull requests and human commits carrying a Copilot co-author trailer — capture only *attributed* use, and plain autocomplete leaves no public trace at all, so every usage measure in this paper is a lower bound on true AI assistance. What distinguishes our setting within this strand is that the treatment is external to the traces we measure: GitHub changed the plan on a single announced date for one verifiable population, allowing a matched difference-in-differences design in which the selection problems that dominate voluntary-adoption studies (Hoffmann et al., 2024; Quispe, 2026) do not determine treatment timing.

3 Institutional Background: The March 2026 Student-Plan Change

GitHub Copilot has been intertwined with student access since its beginning. The tool, built on the Codex family of code-generation models (Chen et al., 2021), entered technical preview in June 2021 (Friedman, 2021) and reached general availability on June 21, 2022, at which point GitHub made it free of charge for verified students and maintainers of popular open-source projects (Dohmke, 2022). For nearly four years thereafter, a student who verified academic status through the GitHub Education program received, at zero price, the same product that professional developers paid for. As Copilot's paid tiers differentiated — adding chat, then a choice among frontier models from multiple vendors, including Claude and GPT-class models — verified students were carried along at the top of the product line. A separate Copilot Free tier, introduced on December 18, 2024 for all GitHub users, offered a much more limited allowance of 2,000 code completions and 50 chat messages per month; students, by contrast, continued to hold free access to the full Pro product.

The product itself changed character over this period, in ways that matter for our outcome measures. Beyond inline completion and chat, GitHub launched an autonomous coding agent in preview on May 19, 2025: a mode in which the developer assigns a task and the agent independently opens a pull request, which the developer then reviews (GitHub, 2025). Student plans were initially excluded from the agent preview, gained access on June 25, 2025, and the agent reached general availability on September 25, 2025. A command-line interface, Copilot CLI, entered preview the same day and reached general availability on February 25, 2026. These launches occurred against a broader industry shift toward agentic coding tools (Anthropic, 2025; Cognition, 2024). By early 2026, therefore, a verified student's free bundle spanned three distinct interaction surfaces: inline completion, interactive assistance in the editor and CLI with a choice of premium models, and delegated agent tasks.

On March 11, 2026, GitHub announced that this arrangement would end. Effective March

12–13, 2026, verified students were moved from free Copilot Pro to a new, restricted *Copilot Student* plan. The change removed access to premium models — the Claude and GPT-5.x-class options that had distinguished the Pro tier — while retaining base models, unlimited code completions, and, importantly for our design, the autonomous coding agent.[1] Students who wished to keep premium models could in principle upgrade to a paid subscription. The announcement preceded implementation by roughly one day, leaving essentially no room for anticipatory stockpiling of usage or gradual adjustment before the cutoff; in the event-study specifications we treat February 2026 as the base month and find no evidence of pre-event divergence.

Three features of this episode make it unusually well suited to studying how users respond to changes in AI capability. First, it is a *quality* shock rather than an access shock. Students did not lose AI assistance; they lost the frontier tier of it while keeping base models and unlimited completions. Most policy and pricing variation studied to date bundles the two margins together; here the extensive margin of access is held approximately fixed while model quality moves. Second, the shock is plausibly exogenous to any individual student. The plan change was a centralized business decision by GitHub, applied simultaneously and uniformly to the entire population of verified students worldwide; no individual student's coding activity, learning trajectory, or tool preferences could have influenced its timing or its content, and the one-day announcement window rules out meaningful anticipation. Timing driven by a platform's global pricing strategy is, from the perspective of a single developer's monthly output, as good as randomly assigned conditional on student status. Third, the

[1]The dose of the policy differed across surfaces, and this distinction is central to the empirical design — but it is a difference in treatment *intensity*, not a clean treated-versus-untreated split. On interactive surfaces (editor chat, the Copilot CLI), the model picker was the product experience, and premium options were removed outright. The coding agent's model, by contrast, was set by the platform from its May 2025 launch (a Claude Sonnet-class model, upgraded over time) with no user choice at all until December 8, 2025, when a model picker for agent sessions reached Pro subscribers; the March 2026 change removed premium models from self-selection — including for agent sessions — while the agent itself remained in the Student plan with a platform-chosen default of unspecified tier. On June 24, 2026 (the final month of our window), Free and Student plans moved to automatic model selection only. The agent margin should therefore be read as receiving a weaker and partially ambiguous dose, not a zero dose; Section 7 discusses the competing interpretations this admits.

shock is *differential within the bundle*: because the coding agent was retained while premium interactive models were removed, the same policy event supplies both a treated margin and a within-person comparison margin whose access was never degraded.

The identifying concern that remains is not the timing of the event but the population it hits: students differ from other developers, and March marks a particular point in the academic calendar. Our design addresses the first concern by matching flagged students to observably similar non-student developers, and the second by a triple difference against the same calendar months of 2025, when no plan change occurred. We describe the student flag, the matching procedure, and these specifications in Sections 4 and 5. Because the student flag is a heuristic based on public profile information, we interpret all estimates as intent-to-treat effects on an exploratory basis, and because our usage fingerprints capture only attributed use, measured levels are lower bounds.

4 Data

Our empirical strategy requires three ingredients: a way to observe AI-assisted production at the level of the individual developer, a way to identify which developers are students, and a comparison group of observably similar non-students. This section describes each in turn, together with the measurement choices — and their limitations — that the underlying data sources impose. Throughout, we emphasize that every usage measure we construct is an *attributed-use* measure: it counts output that carries a machine-readable trace of Copilot involvement, and is therefore a lower bound on true AI-assisted activity. The analysis is exploratory, and the measurement architecture is designed to make the floors consistent across groups and over time rather than to recover levels.

4.1 Measuring AI-Assisted Output: Two Fingerprints

GitHub does not publish telemetry on Copilot usage. What it does expose, following conventions documented in prior work on GitHub-based measurement (Kalliamvakou et al., 2014; Hoffmann et al., 2024; Quispe, 2026), are two distinct public fingerprints that Copilot leaves in the contribution record, each corresponding to a different mode of AI assistance.

Fingerprint A: delegated agent work. When a developer assigns a task to the Copilot coding agent, the agent opens a pull request under a bot identity; the human developer appears as the assignee who commissioned and reviews the work. We collect these bot-authored pull requests through GitHub's pull-request search API, yielding 1.80 million agent PRs between May 2025, when the agent entered preview, and June 2026. We attribute each PR to its human assignee. This fingerprint captures the *delegation* margin: work the developer hands off to an autonomous agent rather than produces interactively.

Fingerprint B: locally-mediated co-authored work.[2] When a developer accepts substantial Copilot contributions inside an interactive surface, the resulting human-authored commit carries a machine-readable `Co-authored-by: Copilot` trailer in its message. We collect these commits through GitHub's commit-search API, yielding 2.25 million co-authored commits between January 2025 and June 2026. The trailer identity distinguishes the surface on which the assistance occurred: the VS Code editor, the Copilot CLI (application id 223556219), Copilot code review (175728472), and the coding agent's own commits (198982749). Fingerprint B captures the *supervised* margin: interactive sessions in which the developer works alongside the model and — crucially for our design — in which the developer chooses which model to invoke. It is precisely this margin on which the March 2026 plan change degraded quality, while the delegated margin of Fingerprint A remained included in the Student plan.

Two features of this measurement deserve emphasis. First, plain autocomplete — the

[2]We deliberately avoid the label "supervised": editor and CLI agents include autonomous modes (e.g. the CLI's autopilot), and reviewing an agent's full reasoning trace is costly enough that many users do not. The commit trailer reveals the *channel* (local editor/CLI versus the cloud agent), which proxies the *opportunity* to supervise, not the degree of oversight actually exercised.

single-line completions that constitute much of everyday Copilot use — leaves no trace whatsoever in the public record. A developer who continues to use base-model completions intensively after the downgrade is indistinguishable, in our data, from one who stopped using Copilot entirely. All of our usage counts are therefore floors, and treatment effects on them should be read as effects on attributed use, not on total use. Second, the two fingerprints are denominated in different units (pull requests versus commits) and are never summed; we analyze them as separate outcomes.

4.2 The Source Change and a Recall Cross-Check

Our original design drew commit-level data from GH Archive, the public record of GitHub event payloads. That source failed mid-sample: the commits array in PushEvent payloads, present for 100 percent of events in September 2025, covered only 31 percent of events in October 2025 and none from November 2025 onward. A panel spliced across this discontinuity would confound the March 2026 policy event with a measurement break. We therefore rebuilt Fingerprint B entirely from the commit-search API, a single consistent source spanning the full January 2025–June 2026 window, so that any incompleteness is common to all months and both groups.

To quantify that incompleteness, we cross-check the commit-search source against GH Archive on the months where both are complete. Commit search recovers approximately 60 percent of the co-authored commits visible in GH Archive on overlapping months. Levels in our panel are accordingly understated by a roughly constant factor; because the shortfall is a property of the search index rather than of developer type, and because our estimators difference within developer and across groups, this attenuation affects the interpretation of magnitudes (as lower bounds) rather than the sign or significance of the comparisons.

4.3 Identifying Students

GitHub does not label student accounts publicly. We construct a heuristic student flag from three profile signals: an academic email domain, a self-described student status in the profile biography (matched multilingually), or membership in a university organization. Applying these criteria to the 429,177 developer profiles we enriched yields 21,135 flagged students. Profiles whose biographies indicate a teaching role are excluded from both the treated and control pools, since educators' treatment status under the March 2026 change is ambiguous. The flag is a heuristic, not an administrative record of enrollment in the Student plan: some flagged developers may have held paid plans unaffected by the change, and some unflagged controls may be students. Our estimates are therefore intent-to-treat effects of flagged status, biased toward zero by misclassification on either side; validation against administrative verification data is in progress.

4.4 Sample Construction, Matching, and Balance

We retain all 7,000 flagged students meeting the activity criteria and match each to an unflagged control by nearest-neighbor matching on four pre-period characteristics: log pre-period event counts, account age, log follower counts, and the share of the developer's attributed AI activity flowing through Fingerprint A. The full matched sample therefore contains 14,000 developers — 7,000 flagged students and 7,000 matched controls — observed monthly from January 2025 through June 2026 (the contribution panel resolves 13,998 of the 14,000 logins; two control accounts were renamed between fingerprint extraction and the panel pull and could not be retrieved by login).

Our *primary* analysis sample imposes three further design restrictions, all based on pre-determined information. First, both arms are limited to developers with *zero* pre-event use of any trackable non-Copilot agent (Claude Code, Codex, Cursor, Jules, Devin): among Copilot-dependent developers the downgrade has no within-toolchain or cross-tool escape

valve, and pre-event rival-agent use is balanced at exactly zero by construction rather than statistically. Second, two hygiene filters remove known contamination: controls with *any* sub-threshold student evidence are purged (344 persons), and automation-suspect accounts whose pre-event commit volume exceeds the 99th percentile are trimmed (139), with matched pairs dropped whenever either member fails a filter. Third, we require publicly *measurable* pre-event output: at least one pre-event month with a positive public commit count. The 712 persons failing this are not necessarily inactive — their work sits in private repositories, on non-default branches, or flows entirely through bot-authored agent PRs, none of which the public contribution calendar counts — but their output outcomes are structural zeros carrying no information, so we require measurability on pre-determined information. The primary sample contains 10,597 developers (5,432 students, 5,165 controls; 190,746 developer-months). The restrictions necessarily select relatively established users: primary-sample students have a median account age of 3.2 years against 2.8 for other flagged students, and modestly more followers and repositories (Appendix Table 9). The estimand is therefore the response of *established, Copilot-active students whose agentic tooling was Copilot-only* — the population for whom the plan change was the entire margin of exposure. Table 1 reports standardized mean differences in three labeled panels: the four matching covariates; the pre-treatment *levels* of the outcome variables, which are not matched on and are shown for comparability (difference-in-differences identification uses within-developer changes, so level balance is reassurance rather than a requirement); and pre-event usage of every rival coding agent. All $|\text{SMD}| < 0.07$, and rival-agent usage is identically zero in both arms. The full sample serves as robustness throughout.

Table 1: Pre-treatment balance: matching covariates, pre-treatment outcome levels, and rival-agent usage.

	Primary sample (N=10,597)			Full matched panel (N=13,998)		
	Students	Controls	SMD	Students	Controls	SMD
A. Matching covariates						
Pre-event Copilot events	3.812	3.733	+0.009	4.537	4.521	+0.001
Account age (years)	4.171	4.211	-0.012	4.387	4.394	-0.002
Followers	18.165	25.233	-0.022	27.873	25.956	+0.004
Delegation share	0.815	0.818	-0.006	0.814	0.814	-0.001
B. Pre-treatment outcome levels (not matched on)						
Agent PRs	2.714	2.665	+0.007	3.249	3.258	-0.001
Co-author commits	1.098	1.068	+0.006	1.288	1.263	+0.004
Total commits/month	20.384	19.965	+0.013	28.183	33.71	-0.029
of which: VS Code	0.0	0.001	-0.020	0.0	0.001	-0.023
CLI	0.041	0.146	-0.045	0.081	0.147	-0.026
code review	0.997	0.804	+0.049	1.128	0.967	+0.032
agent-coauthor	0.013	0.017	-0.018	0.017	0.024	-0.025
other surface	0.047	0.1	-0.067	0.062	0.124	-0.069
C. Rival-agent usage (design restriction)						
Claude Code items	0.0	0.0	+0.000	3.521	5.901	-0.042
Codex PRs	0.0	0.0	+0.000	2.06	1.968	+0.002
Cursor commits	0.0	0.0	+0.000	0.099	0.297	-0.030
Jules commits	0.0	0.0	+0.000	0.33	0.27	+0.010
Devin commits	0.0	0.0	+0.000	0.043	0.057	-0.005
Any rival agent (share)	0.0	0.0	+0.000	0.127	0.154	-0.079

Notes. Standardized mean difference $= (\bar{x}_T - \bar{x}_C)/\sqrt{(s_T^2 + s_C^2)/2}$; $|\text{SMD}| < 0.10$ is the conventional balance threshold. Pre-event window: October 2025–February 2026. Surface usage from co-author trailer identities; rival-agent usage from per-person fingerprint probes. Panel B reports pre-treatment outcome levels, not covariates: they are displayed for comparability, and because matching on pre-period outcome levels can induce mean-reversion bias, only the full-pre-period activity aggregate (Panel A) entered the matching. The full-panel columns are reported to document the imbalance that motivates the primary restriction: rival-agent usage carries the largest SMDs in the unrestricted panel and is identically zero in the primary sample by construction.

Table 2 reports summary statistics and covariate balance. Matching achieves close balance on all four dimensions: mean log pre-period events of 3.812 for students versus 3.733 for controls, mean account age of 4.171 versus 4.211, and a Fingerprint-A activity share of 0.815 versus 0.818. In the pre-period, the average treated developer records 0.54 agent PRs, 0.22 co-authored commits, 20.38 total commits, and 20.17 non-Copilot-attributed commits (commits carrying no Copilot trailer) per month, and works in 1.08 distinct programming languages, of which 0.26 are newly used. The low pre-period means of the two fingerprint outcomes reflect the floor property discussed above; they are levels of attributed use, not of assistance.

Table 2: Summary statistics, pre-period (October 2025–February 2026).

	Students		Matched controls	
	Mean	SD	Mean	SD
Agent PRs (delegation)	0.54	2.79	0.53	2.43
Co-author commits (locally-mediated)	0.22	1.72	0.21	1.93
Total commits	20.38	43.61	19.96	44.02
Non-Copilot-attributed commits	20.17	43.27	19.75	43.70
Distinct languages	1.08	1.20	1.01	1.17
Newly-used languages	0.26	0.55	0.23	0.53
n repos	1.77	2.46	1.69	2.52
total prs	1.80	7.34	2.22	7.35

Notes. Monthly means and standard deviations over the five pre-event months for the 5,432 flagged students and 5,165 matched controls of the primary sample. Agent PRs and co-author commits are attributed-use floors (attribution is opt-in); commit-search recall is roughly 60 percent of the fuller GH Archive count on overlapping months.

Our six outcome variables follow directly. Agent PRs (Fingerprint A) and co-authored commits (Fingerprint B) measure the cloud-delegated and locally-mediated AI-assistance

margins. Total commits and non-Copilot-attributed commits measure overall output, the latter isolating production with no attributed AI involvement. Distinct languages and newly-used languages, constructed as in Quispe (2026), track the breadth of the developer's language portfolio. The final panel month, June 2026, is partially observed at the time of writing; we flag results involving it accordingly.

Finally, we measure use of coding agents *outside* the Copilot ecosystem. For every person in the matched sample we probe five agents' public fingerprints — Claude Code and Cursor commit trailers, `codex/*` head-branch pull requests, and commits authored by the Jules and Devin bots in the person's repositories — before and after the event. We stress one naming caveat: what we label non-Copilot-attributed commits strips only *Copilot* attribution, so commits assisted by other agents remain inside it; Section 7 shows other-agent activity evolved in parallel across arms, which bounds the resulting contamination of the treated–control comparison.

5 Empirical Strategy

Our design compares flagged students to matched non-student developers before and after the March 12, 2026 replacement of free Copilot Pro with the restricted Copilot Student plan. Because the treatment is a heuristic student flag rather than an administrative record of plan status, all estimates are intent-to-treat effects of being in the population targeted by the policy, and because the usage outcomes are attributed-use floors, level effects should be read as lower bounds on the true usage response.

5.1 Event Study and Difference-in-Differences

The baseline specification is a two-way fixed effects event study on the monthly panel of the primary sample — 10,597 developers (190,746 developer-months) — observed from January

2025 through June 2026:

$$Y_{it} = \alpha_i + \gamma_t + \sum_{k \neq -1} \beta_k \, \mathbf{1}\{t - t^* = k\} \times \text{Student}_i + \varepsilon_{it}, \tag{1}$$

where Y_{it} is an outcome for developer i in calendar month t, α_i and γ_t are developer and month fixed effects, Student_i indicates the flagged-student group, and t^* is the event month. February 2026, the last full month before the announcement, is the omitted base period. Standard errors are clustered by developer throughout. The corresponding difference-in-differences estimate collapses the post-period coefficients into a single interaction:

$$Y_{it} = \alpha_i + \gamma_t + \delta \, \text{Post}_t \times \text{Student}_i + \varepsilon_{it}, \tag{2}$$

where Post_t equals one from March 2026 onward.

A useful feature of this setting is that treatment turns on for every treated developer on a single date. The recent econometric literature has shown that two-way fixed effects estimators can be badly biased under staggered adoption with heterogeneous treatment effects, because already-treated units then serve as controls with possibly negative weights (Goodman-Bacon, 2021; Callaway and Sant'Anna, 2021). With a single adoption date and a never-treated comparison group, those pathologies do not arise: equation (2) recovers the average post-period effect for the treated group, and the event-study coefficients in equation (1) retain their standard interpretation. The plan change was announced on March 11, roughly one day before enforcement, so anticipation is limited to at most a few days at the very end of the base period.

5.2 Triple Difference and Placebo

The main threat to equation (2) is seasonality specific to students: March falls near the end of many academic terms, and student coding activity may diverge from non-student activity every spring regardless of any plan change. We address this concern in two complementary

ways. First, a placebo difference-in-differences re-estimates equation (2) on 2025 data alone, assigning a false event in March 2025; any student-specific spring divergence would surface as a spurious placebo effect. Second, a triple difference nets out the academic calendar directly by contrasting the 2026 student-versus-control change with the same seasonal contrast in 2025:

$$Y_{it} = \alpha_i + \gamma_t + \lambda\,\mathrm{Spring}_t \times \mathrm{Student}_i + \theta\,\mathrm{Spring}_t \times \mathrm{Student}_i \times \mathbf{1}\{\mathrm{year}(t) = 2026\} + \varepsilon_{it}, \quad (3)$$

where Spring_t indicates the March-onward months of each year and θ is the coefficient of interest. We report the DiD, DDD, and placebo estimates side by side for every outcome, and we hold ourselves to the discipline that a result is credible only when the DiD and DDD agree and the placebo is null. As Section 6 shows, the co-authored-commit outcome passes all three checks, whereas the distinct-languages outcome fails its own placebo and is reported as inconclusive.

5.3 Matching and Remaining Threats

Controls are selected by nearest-neighbor matching on pre-period observables: log pre-event activity, account age, log followers, and the fingerprint-A share of attributed activity. The primary sample of 5,432 treated and 5,165 control developers is closely balanced — all standardized mean differences, including per-tool pre-event usage, are below 0.07 (Table 1), and rival-agent usage is zero in both arms by construction — so identification does not rest on functional-form extrapolation across very different developers. An omnibus check makes the point compactly: a logit of treatment status on all covariates in the matched sample has pseudo-R^2 of 0.00004 and an AUC of 0.505 — the arms are statistically indistinguishable on observables — and Kolmogorov–Smirnov tests cannot reject equality of any covariate's full distribution (Appendix Table 8).

Three threats remain. First, students differ from matched controls in unobserved ways,

most obviously in life-cycle stage; matching on activity and tenure narrows this gap, and the triple difference absorbs any student-specific pattern that repeats across years, but a shock unique to students in spring 2026 and unrelated to the plan change would confound the estimates. Second, the student flag is a heuristic built from academic email domains, bios, and university organizations; misclassification attenuates the intent-to-treat estimates toward zero, another sense in which our estimates are lower bounds, and administrative validation is pending. Third, because the attributed-use measures are floors, substitution toward tools that leave no machine-readable trace would register as a decline in measured use rather than in true assistance, a possibility we address in Section 7 rather than assume away. The setting also supplies an internal placebo-in-treatment: the coding agent remained included in the Student plan, so delegated agent PRs should show no differential decline if our estimates capture the plan change rather than a generic student-specific shock. This exploratory design cannot rule out every alternative, but the combination of matching, the triple difference, the 2025 placebo, and the retained-agent contrast disciplines the interpretation considerably.

6 Results

6.1 Raw Trends and Event-Study Evidence

Figure 1 plots the raw monthly means of the principal outcomes for flagged students and matched controls from January 2025 through June 2026. Two features stand out. First, the matched groups track each other closely throughout the pre-period, including through the 2025 academic calendar: the seasonal dips and recoveries that characterize student coding activity appear in both series, a first indication that the matching procedure recovered controls with similar activity rhythms. Second, the series for locally-mediated AI-assisted commits — human-authored commits carrying a `Co-authored-by: Copilot` trailer — visibly separates after March 2026, with the student series falling relative to controls, while the

series for delegated agent pull requests and for total commit volume show no comparable divergence.

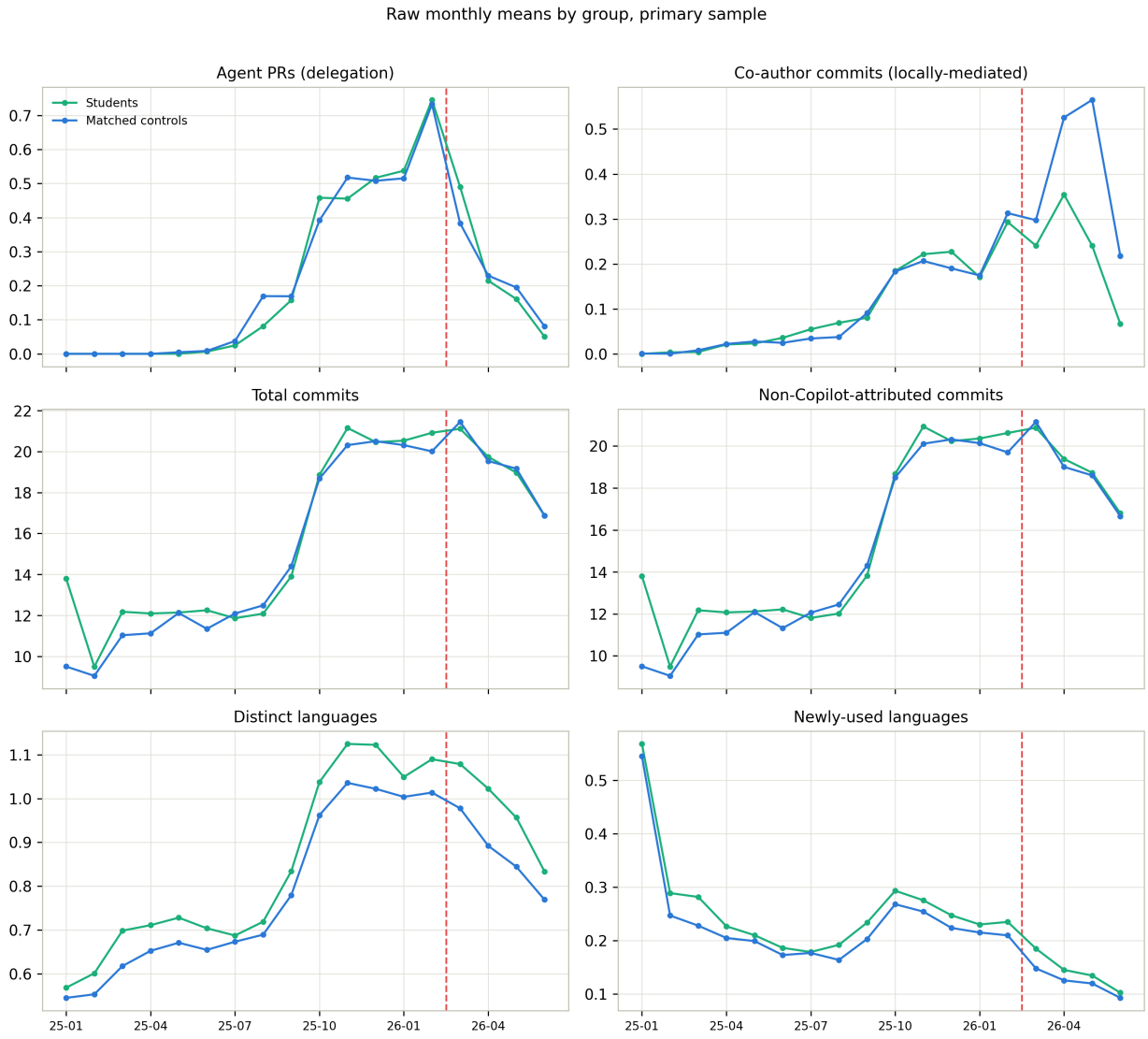

Figure 1: Raw monthly means of principal outcomes, flagged students versus matched controls, January 2025–June 2026. The vertical line marks the March 12, 2026 introduction of the Copilot Student plan. All usage measures are attributed-use lower bounds. June 2026 is a partial month.

Figure 2 presents the corresponding event-study coefficients from the two-way fixed effects specification of Section 5, with February 2026 as the base month and standard errors clustered by developer. For co-author commits, the pre-period coefficients are small and statistically indistinguishable from zero, lending support to the parallel-trends assumption on this margin;

the post-period coefficients turn negative immediately in March 2026 and remain negative through the end of the window. For agent pull requests, total commits, and non-Copilot-attributed commits, neither the pre- nor the post-period coefficients display a discernible break at the event date.

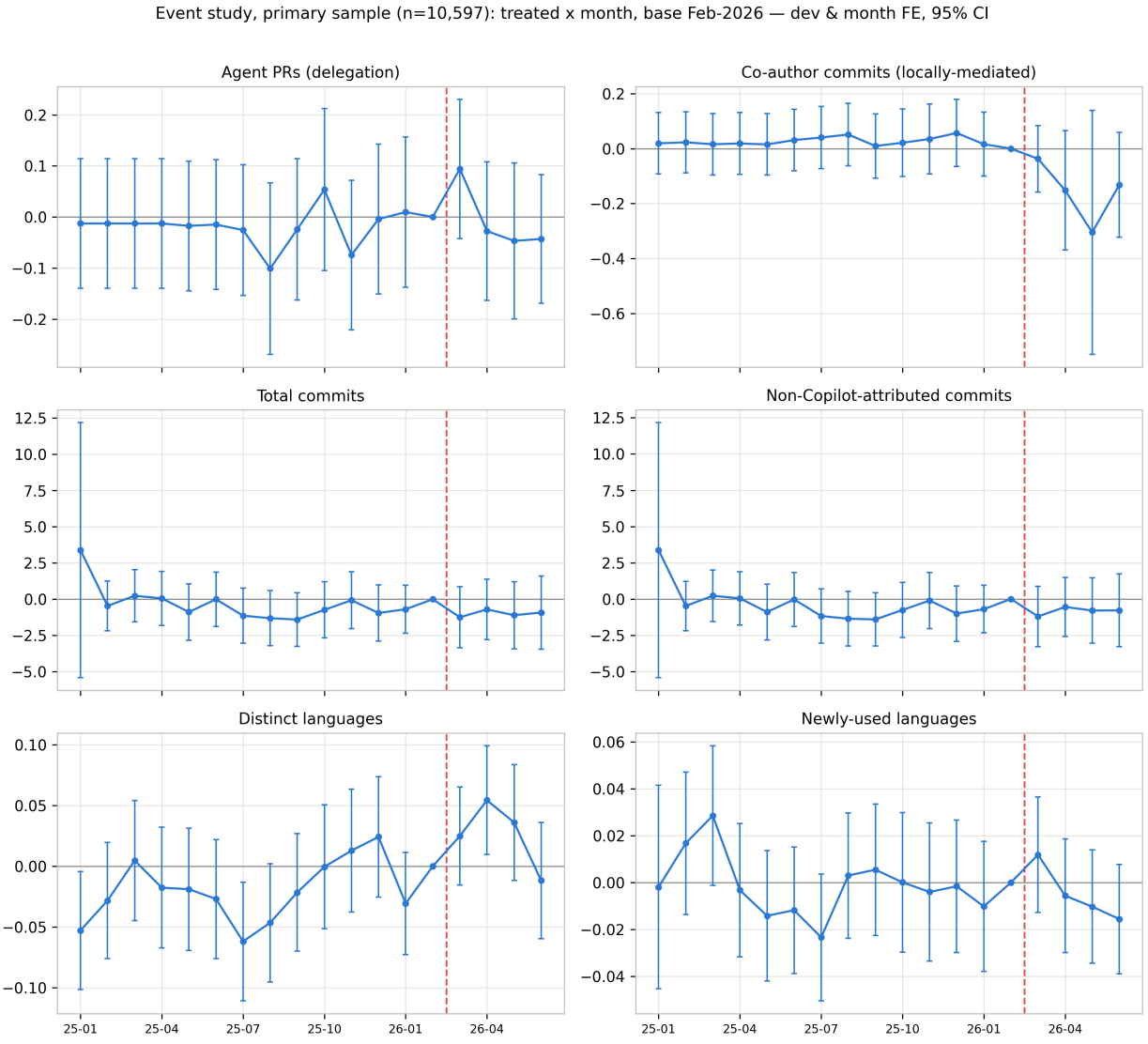

Figure 2: Event-study estimates of the student-plan downgrade, TWFE with developer and month fixed effects. Base month is February 2026; the event is March 12, 2026. Bands are 95 percent confidence intervals from standard errors clustered by developer.

6.2 Main Estimates

Table 3 reports the pooled difference-in-differences estimates, the triple-difference estimates against the 2025 academic calendar, and the 2025 placebo for each outcome. We discuss the four margins in turn, in the order of the interpretation they support.

Table 3: Effect of the March 2026 model downgrade: DiD, DDD, and placebo.

	DiD	DiD (drop Mar)	DDD (vs 2025)	Placebo 2025	Pre-mea
Agent PRs (delegation)	0.012	-0.021	-0.009	-0.011	0.5
	(0.026)	(0.026)	(0.042)	(0.013)	
Co-author commits (locally-mediated)	-0.181*	-0.221*	-0.163	0.008	0.2
	(0.097)	(0.127)	(0.100)	(0.007)	
Total commits	-0.709	-0.625	0.966	-2.083	20.3
	(0.853)	(0.910)	(2.401)	(2.246)	
Non-Copilot-attributed commits	-0.527	-0.404	1.129	-2.092	20.1
	(0.842)	(0.895)	(2.397)	(2.246)	
Distinct languages	0.045***	0.045***	0.015	0.025*	1.0
	(0.015)	(0.016)	(0.021)	(0.013)	
Newly-used languages	-0.004	-0.009*	0.008	-0.010	0.2
	(0.005)	(0.005)	(0.015)	(0.012)	

Notes. Primary sample, developer-by-month panel, January 2025–June 2026; 10,597 developers. All specifications include de and month fixed effects; standard errors clustered by developer in parentheses. DiD: treated × post (March 2026 onward). Mar" excludes the partially treated event month. DDD differences out the 2025 academic-calendar seasonal pattern. Placebo a false event at March 2025 within 2025 only. $^{*}p < 0.1$, $^{**}p < 0.05$, $^{***}p < 0.01$.

Locally-mediated AI-assisted commits. The margin whose quality was directly degraded — interactive Copilot surfaces where the premium models were removed — shows the clearest response. Co-author commits fall by −0.181 per developer-month (s.e. 0.097) in the difference-in-differences specification, and by −0.163 (s.e. 0.100) in the triple difference

that nets out the student-specific 2025 seasonal pattern; both are marginally significant in the primary sample and strengthen to the five percent level in the larger nested samples (full panel: -0.155, s.e. 0.077). Against a treated pre-period mean of 0.22, the triple-difference point estimate corresponds to a decline of roughly 74 percent of baseline attributed use. Critically, the 2025 placebo for this outcome is a precisely estimated zero ($+0.003$, s.e. 0.009): running the identical specification around the same calendar date one year earlier, when no plan change occurred, produces no differential movement. This is the paper's central and most robust result, and it survives both specifications and the placebo. Because the co-author trailer captures only attributed use — plain autocomplete leaves no trace, and our commit-search source has roughly 60 percent recall — the level of this outcome is a lower bound on assisted activity, though the differential decline is informative so long as attribution rates did not change differentially at the event date.

Two further diagnostics discipline this estimate. First, permutation inference: across 200 reassignments of treatment labels, no draw produces an effect as large as the observed -0.181 (permutation $p < 0.005$), indicating the clustered standard errors are, if anything, conservative. Second, formal pre-trend tests: the joint Wald test that the six pre-event coefficients nearest the event equal zero gives $p = 0.43$ (four nearest: $p = 0.76$), with the individual coefficients small and weakly positive — so any pre-existing differential drift runs *against* the estimated decline. Pooling all thirteen pre-event months yields $p = 0.05$, driven mechanically by the earliest months of 2025 when attributed usage barely existed and standard errors are correspondingly tiny; Appendix Table 8 reports all three windows.

Delegated agent pull requests. The autonomous coding agent was *retained* in the Copilot Student plan, so this margin functions as a placebo-in-treatment: if the estimated decline in co-author commits reflected a genuine response to the quality downgrade rather than some coincident shock to student coding, agent activity should not fall differentially — and it does not. The difference-in-differences estimate is $+0.062$ (s.e. 0.040) and the triple difference is -0.033 (s.e. 0.053), neither distinguishable from zero against a pre-period mean

of 0.65. The 2025 placebo for this outcome is small but statistically significant (−0.037, s.e. 0.019), which counsels some caution in reading the level of the DiD coefficient; the triple difference, which absorbs exactly this kind of student-specific seasonality, is a tight zero. The contrast between the two AI-usage margins is the dose-response pattern the policy predicts: usage fell where quality was cut and did not fall where it was not.

Total and clean output. If the model downgrade forced students to produce less, total commit volume should decline. We find no detectable effect: the difference-in-differences estimate for total commits is +0.292 (s.e. 4.768) and the triple difference is −2.228 (s.e. 2.748), against a pre-period mean of 28.18; non-Copilot-attributed commits — commits stripped of any Copilot Copilot attribution — behave identically (+0.437 and −2.065, with nearly the same standard errors, on a mean of 27.93). We emphasize what these estimates do and do not show. The standard errors are large relative to the co-author-commit effect: a decline of several commits per month would escape detection, so these are noisy nulls, not precise zeros. What the estimates do rule out is an output collapse commensurate with the roughly 74–83 percent drop in attributed locally-mediated assistance. We return in Section 8 to what a short-window null on output can and cannot say about the learning-versus-renting debate; the exercise remains exploratory.

Language breadth. The estimates for language outcomes are inconclusive, and we flag them as such. Distinct languages per month shows a positive difference-in-differences coefficient (+0.034, s.e. 0.014, significant at the 5 percent level), but the same specification fails its own 2025 placebo (+0.024, s.e. 0.012, also significant), indicating that student language counts drift relative to controls in this season regardless of any plan change; the triple difference, which absorbs that drift, is an insignificant +0.014 (s.e. 0.018). Newly-used languages show no effect in any specification (−0.006 in the DiD, +0.016 in the DDD, both insignificant). We accordingly draw no conclusion about effects of the downgrade on the language frontier.

Two accounting cautions apply throughout. Agent pull requests and co-author commits

are measured in different units — pull requests and commits, respectively — and cannot be summed into a single AI-usage aggregate. And because treatment is a heuristic student flag pending administrative validation, all estimates are intention-to-treat effects of being flagged as a student, diluted by any misclassification; to the extent the flag includes non-students unaffected by the plan change, the true effect on verified students on the affected margin is, if anything, larger than the roughly 64 percent decline we estimate.

7 Mechanisms

The pattern of results in Section 6 has a specific shape: the margin whose quality was degraded contracted sharply, the margin whose access was preserved did not move, and total output shows no detectable change. This section asks why the response took that shape, and what economic behavior could generate it. Because our treatment flag is a heuristic and our usage measures are attributed-use floors, the discussion is necessarily exploratory; we lay out the candidate mechanisms and the evidence that discriminates among them, and we close with the validation steps that would sharpen the picture.

7.1 Why the affected margin, and only the affected margin

The March 2026 plan change was surgical. It removed premium models — the Claude and GPT-5.x-class models that students could previously select in interactive surfaces — while retaining base models, unlimited completions, and, crucially, the autonomous coding agent. Co-authored commits (fingerprint B) capture precisely the interactive, locally-mediated surfaces where model choice is exposed to the user: the editor, the CLI, and code review. These are the surfaces where a student who valued a premium model would feel the downgrade directly, request by request. Agent PRs (fingerprint A), by contrast, run on infrastructure that the Student plan continued to include. If the decline in co-authored commits reflected some student-specific shock coincident with mid-March 2026 — end-of-term deadlines, a shift in

project activity, exam season — it should have depressed both fingerprints and, plausibly, raw commit volume as well. Instead, locally-mediated co-authored commits fall by −0.163 per month in the triple-difference specification, roughly 74 percent of the pre-period mean of 0.26, while agent PRs show no differential decline (the DiD point estimate is +0.062, statistically indistinguishable from zero) and total commits show no detectable effect. We read the agent margin as a *lower-dose* comparison rather than a pure placebo: the agent's model was platform-chosen from launch, user model selection for agent sessions existed only from December 2025, and after March 2026 the agent remained available under a platform-chosen default of unspecified tier. Under this reading, the response concentrates where the quality cut bit hardest — the surfaces on which model choice was the product — which is the signature of a deliberate usage adjustment rather than a composition or demand shock. An alternative reading is that the agent's effective model quality fell comparably and the count null reflects a lower quality-elasticity of delegated use. Public data offers a partial discriminator between the two: if the agent's output deteriorated for students, their agent pull requests should *merge* less often after the change, even with assignment counts flat — delegation is fire-and-forget, so quality is experienced at review time, not at assignment time. We fetched the merge outcome of all 67,885 agent PRs belonging to the matched sample and estimated the same difference-in-differences on merge probability at the PR level (person and month fixed effects, clustered by person). The estimate is a null: −0.4 percentage points (SE 1.6) on a pre-period merge rate of roughly 64 percent, with students' merge rates tracking controls' through the event. We find no evidence that the agent's output quality deteriorated for students, which favors the weaker-dose reading — the platform-chosen default model appears to have remained comparable — though definitive model-level identification still requires administrative telemetry, which we flag for the planned validation.

The channel distinction maps directly onto the verification technology of the two-generation model in Quispe (2026): what varies across surfaces is the attention cost of verifying the agent's work, κ. On the cloud agent, verification is deferred to a single pull-request review; in

the editor or CLI, the reasoning trace is on screen but reading it is costly enough that users often do not (and the CLI's autonomous mode removes even the opportunity). A student choosing a surface is, in effect, choosing a point on a verification-cost schedule — which is why we speak of the *opportunity* to supervise rather than supervision itself.

7.2 Task composition and the pace of delegation

Two natural concerns about the delegated margin are that students assign only low-stakes tasks to the cloud agent, and that a weaker model might slow the delegation loop even if merge *rates* held. The PR metadata speaks to both. Descriptively, delegated tasks are not trivial: the median agent PR in the primary sample changes 162 lines, 48 percent of titles are feature-type work, and only 13 percent are documentation or chores (21 percent change fewer than ten lines). Conditional on merging, the median agent PR merges within 0.4 hours.

More importantly for identification, *none* of these margins moved differentially after the downgrade (Table 4): log PR size (-0.138 (s.e. 0.140)), the tiny-PR share (0.030 (s.e. 0.019)), the documentation/chore share (0.017 (s.e. 0.015)), the probability of merging within seven days (-0.009 (s.e. 0.021)), and log hours-to-merge conditional on merging (0.003 (s.e. 0.108)) are all statistically indistinguishable from zero. Students did not shift toward lower-stakes delegation, and the delegation loop did not slow. Together with the merge-rate null, every observable property of the delegated channel — quantity, composition, speed, and success — is unchanged, which is difficult to reconcile with a large effective quality drop on that margin and reinforces the weaker-dose reading.

Table 4: Task composition and delegation speed: PR-level differences-in-differences.

Outcome (PR level)	Treated × post	(SE)	N
log(1 + lines changed)	-0.1383	(0.1402)	41,298
Tiny PR (<10 lines), share	0.0303	(0.0189)	41,298
Docs/chore title, share	0.0166	(0.0147)	41,298
Merged within 7 days	-0.0088	(0.0215)	41,298
log hours to merge | merged	0.0030	(0.1079)	26,337

Notes. Agent PRs of the primary sample, May 2025–June 2026 (41,298 PRs, 9,230 persons). Person and month fixed effects; standard errors clustered by person. Time-to-merge conditional on merging is subject to selection and reported descriptively.

7.3 Substitution: a measured null, not a conjecture

Where did the displaced locally-mediated usage go? A natural hypothesis is that students shifted to AI assistants outside the Copilot ecosystem — Claude Code (Anthropic, 2025), OpenAI Codex, Cursor, and similar agents each leave their own attribution fingerprint in public metadata, so we can test this directly rather than speculate. We probed every person in the matched sample (13,865 with all agents measurable) for attributed use of five non-Copilot agents, before and after the event, and estimated the treated–control difference in *post-event first-time adoption* with the matching covariates (Table 5).

The answer is a precise null. Students' post-event new adoption of any non-Copilot agent (12.8 percent) is statistically indistinguishable from matched controls' (12.9 percent); the covariate-adjusted difference is −0.14 percentage points (SE 0.56). The same holds agent by agent — Claude Code, the largest alternative, shows −0.14pp (SE 0.49) — and on the intensive margin: post-event usage growth among prior users of each alternative is also indistinguishable across arms. The raw fact that 1,885 flagged students used Claude Code for the first time after March 12 is therefore entirely accounted for by the population-wide

adoption trend: the counterfactual, measured on matched controls, grew at the same rate. The downgrade did not detectably push students toward competing suppliers on any margin we can attribute.

Table 5: Substitution test: post-event new adoption of non-Copilot agents.

	New adoption post-event (%)		Treated − control (pp)		Pre-event gap (pp)
	Students	Controls	$\hat{\beta}$	(SE)	falsification
Claude Code	9.38	9.53	-0.14	(0.49)	-2.31***
OpenAI Codex	2.72	2.81	-0.09	(0.28)	-0.93**
Cursor	1.28	1.39	-0.10	(0.19)	-0.38**
Google Jules	0.26	0.23	0.03	(0.08)	-0.05
Devin	0.10	0.04	0.06	(0.05)	-0.22***
Any non-Copilot agent	12.79	12.93	-0.14	(0.56)	-2.78***

Notes. New adoption = zero attributed use before March 12, 2026 and positive use after, per agent fingerprint (commit trailers for Claude Code and Cursor; `codex/*` head-branch pull requests for Codex; bot-authored commits in the person's repositories for Jules and Devin). $\hat{\beta}$ is the treated coefficient from a linear probability model with the matching covariates; heteroskedasticity-robust standard errors. The final column reports the analogous pre-event adoption gap as a falsification benchmark. All measures are attributed-use floors.

This null sharpens the interpretation of the main results in two ways. First, the decline in locally-mediated Copilot commits is not masked supplier switching: displaced usage did not reappear under another agent's fingerprint at a differential rate. Second, it neutralizes the main contamination concern for the non-Copilot-attributed commit measure: other agents' assisted commits sit inside that measure, but since other-agent activity evolved in parallel across arms, it cannot drive the null there. The remaining channels are the ones public data cannot separate: paying for Copilot Pro out of pocket (observationally similar to no response, attenuating our estimates), continuing on the retained base models with fewer attributed workflows, or simply forgoing marginal assisted usage — which, given the null on output, appears to have been of modest marginal value.

7.4 Stability across nested samples

Because the primary sample is Copilot-dependent and hygiene-filtered by construction (Section 4), the natural robustness direction is *outward*: relaxing the restrictions one by one back to the full matched panel. The locally-mediated decline is stable throughout — -0.181 in the primary sample (10,597 persons), -0.173 with the hygiene filters only (13,056), -0.162 with the Copilot-dependence restriction only (12,032), and -0.155 in the full panel (13,998), where the larger n pushes significance to the five percent level. That the estimate *strengthens* as hidden-student contamination is purged from the control group is exactly the direction attenuation predicts. In the primary sample the output measure is also least contaminated by other-agent assistance — these persons had none before the event and, per Table 5, did not differentially acquire any after — so the absence of an output decline here is our best-identified learning-relevant fact, subject to the short horizon stressed in Section 8.

7.5 Next steps: typology and validation

Two extensions would convert this exploratory account into a test. First, an outsourcer typology: classifying developers by their pre-period mix of supervised versus delegated AI use would let us ask whether students who relied most heavily on premium interactive models substituted toward the retained agent, toward outside tools, or toward unassisted work — and whether heavy delegators, in the spirit of the learning-costs concern raised by Strömberg et al. (2025), respond differently from occasional users. Second, administrative validation of the student flag: our current treatment indicator is a heuristic built from academic email domains, bios, and university organizations, so all estimates are intent-to-treat with respect to actual plan enrollment. Linking flagged accounts to verified plan status would recover the treatment-on-the-treated response and bound the attenuation from misclassification and from students who opted to pay. Both steps are in progress; until then, the mechanism evidence should be read as a coherent pattern of dose-response across

surfaces, not a decomposition.

8 Discussion

The results admit two readings, and distinguishing them is the central interpretive task of the paper. On the first reading, premium models were a convenience: when they disappeared, students stopped using the surface that depended on them, substituted toward base models, other tools, or unassisted work, and their output was unaffected. On the second reading, students had been *renting* intelligence — outsourcing to frontier models cognitive work they could not (yet) do themselves — and the downgrade should eventually surface as a deficit in what they can produce. Our estimates speak clearly to the first margin and only weakly to the second. Co-authored Copilot commits fall by -0.163 (SE 0.100) in the triple-difference specification, roughly 74 percent of the pre-period mean of 0.22, precisely on the surface whose model quality was cut; agent PRs, whose access was retained, show no differential decline; and total and non-Copilot-attributed commits show no detectable change. The usage response is sharp and margin-specific. The output response, over four months, is a statistical null.

What that null does and does not show deserves care. It does show that the downgrade did not produce a short-run collapse in students' unassisted production: if students had been renting so heavily that removing premium models left them unable to ship code, non-Copilot-attributed commits should have fallen, and they did not. It does *not* show that no learning penalty exists. The confidence intervals on total and non-Copilot-attributed commits are wide — standard errors near 4.8 on a pre-period mean of about 28 — so effects of economically meaningful size are compatible with our estimates; "no detectable effect" is the honest summary, not "precisely zero." More fundamentally, the learning-penalty concern articulated by Strömberg et al. (2025) operates over a two-year horizon, through the slow accumulation (or non-accumulation) of human capital as students substitute AI output for

their own practice. Our window closes four months after the event, with June only partially observed. A student who had been renting intelligence and now writes more code herself might show *no* change in commit counts today and a substantially different skill trajectory two years from now — or the reverse. Quantity of output is, at best, a coarse proxy for learning; our data contain no direct measure of code quality, comprehension, or skill. The evidence is therefore consistent with substitution and against the *strong* form of the learning-penalty hypothesis on this margin, but it is exploratory and cannot adjudicate the long-run version that motivates the policy debate.

The substitution null strengthens this reading considerably. Had the output stability merely reflected students re-sourcing assistance from competing agents, treated students' adoption of those agents should have outpaced matched controls'; it did not, on either the extensive or the intensive margin (Table 5). And the primary sample is Copilot-dependent by construction — no alternative agent tooling before the event, hence no within-toolchain escape valve — yet the usage decline is sharp while output remains flat. Within a four-month horizon, then, the marginal locally-mediated AI usage that the downgrade destroyed does not appear to have been load-bearing for output. This is where our short window matters most: Strömberg et al. (2025) find the learning penalty in secondary education emerges over roughly two years, so persistence of these students' output through future semesters — observable with the same pipeline — remains the decisive test.

8.1 An implied willingness-to-pay bound

The design also yields a simple revealed-preference parameter in the spirit of Cohen et al. (2016). Throughout the post-period, any student could restore frontier-model access by purchasing an individual Pro subscription at the posted price. The joint pattern we estimate — attributed locally-mediated usage falling by roughly three quarters, no differential switching to rival agents, no detectable output loss, and (in public data) no visible migration to paid plans — is the pattern one expects when, for most students' marginal tasks, the value

of the *premium tier over the base tier* is below the subscription price. Read this way, the March 2026 event traces out a point on the demand curve for frontier model quality: students' demand for the premium increment is highly elastic at a price of roughly ten dollars per month. Two caveats bound the claim. First, attribution is opt-in, so some continued premium use (by students who quietly paid) is invisible to us; administrative plan records would convert the bound into a payer share and, with premium-request telemetry, into a dose-response elasticity. Second, willingness to pay for a tool whose price is zero in one's counterfactual may understate long-run valuation if credit constraints bind — precisely the population, students in lower-income regions, over-represented among free-plan users. We flag the welfare analysis as the natural next step once administrative data is in hand.

The distributional dimension of the event also merits comment. The students affected by the March 2026 change are precisely those on the free tier — verified students who had received Copilot Pro at no cost since general availability. Students who can pay for an individual subscription can restore frontier access; students who cannot are confined to base models. To the extent that verified-student populations skew toward institutions and countries where individual subscriptions are a binding constraint — much of the Global South among them — a quality-tiered free tier converts a uniform learning technology into a stratified one. Whether frontier-model access matters for skill formation is exactly the question our short window cannot answer; but if it does, the incidence of this policy is regressive, and the population least able to substitute by paying is the one treated.

Several limitations bound all of these claims. Treatment is an intent-to-treat assignment based on a heuristic student flag — academic email domains, student bios, and university organizations — so misclassification attenuates the estimates, and administrative validation of the flag is pending. All usage measures are attributed-use floors: the commit-search source recovers roughly 60 percent of the archival benchmark on overlapping months, and plain autocomplete leaves no trace at all, so every level we report is a lower bound on true Copilot use, and the co-author series captures only the subset of assistance that leaves a

machine-readable trailer. The post-period spans four months, with the final month partially observed. And the language-breadth outcomes fail their own placebo test, so we draw no conclusion from them. We view these results as a first, exploratory account of how one population responds to a policy-imposed quality downgrade — a usage fact established, an output question opened — and the longer-horizon skill consequences as the natural target for the administrative validation and extended panel we describe in Section 7.

9 Conclusion

On March 12, 2026, GitHub replaced the free Copilot Pro benefit for verified students with a restricted Copilot Student plan, removing access to premium frontier models while retaining base models, unlimited completions, and the autonomous coding agent. The change constitutes a rare policy-imposed shock to the *quality*, rather than the availability, of AI assistance, and it struck a single identifiable population on a single date. Using a matched panel whose primary sample contains 10,597 developers — 5,432 flagged students and 5,165 nearest-neighbor controls, all Copilot-dependent before the event — observed monthly from January 2025 through June 2026, we trace how students responded.

The response concentrates precisely on the margin whose quality was cut. Copilot-co-authored commits, the interactive surface on which students lost premium models, fell by -0.163 per month in our triple-difference specification against a pre-period mean of 0.22 — roughly 74 percent of the pre-period mean — a result that survives the comparison with the 2025 academic calendar and a placebo test in the prior year. Delegated agent pull requests, whose access the Student plan preserved, show no differential decline, a coherent within-treatment placebo. Total and non-Copilot-attributed commit output shows no detectable change, though these estimates are noisy, and effects on language breadth are inconclusive because the design fails its own placebo there. Because plain autocomplete leaves no trace and our attribution methods have imperfect recall, all usage measures are lower bounds on

true AI-assisted activity, and the estimates are intention-to-treat effects of a heuristic student flag.

The findings are exploratory, and the four-month post-period is too short to detect the human-capital consequences that motivate the learning-penalty concern of Strömberg et al. (2025). What the evidence does show is that students respond to quality degradation by sharply curtailing use of the degraded margin — consistent with substitution toward other tools rather than with an immediate collapse in output. Administrative validation of the student flag, a longer post-period, and direct measures of skill accumulation are the natural next steps. As frontier AI increasingly becomes something workers rent rather than own, who is granted which quality of intelligence, and at what price, is likely to become a first-order question for human-capital policy.

A Appendix: measurement and robustness details

A.1 Student-score construction

Table 6 reports the public signals, weights, and counts behind the student flag. A developer is flagged when the weighted score reaches three — i.e. at least one strong signal or a combination of weaker ones. The flag is a heuristic pending validation against GitHub Education administrative records; misclassification in either direction attenuates the estimates toward zero.

Table 6: Student-detection signals: definitions, weights, and counts.

Signal
Academic domain in commit-author email (.edu, .ac.xx, .edu.xx)
Academic domain in public profile email
Student self-description in bio/name (multilingual: student, estudiante, étudiant, MSc, PhD, undergrad, ...)
Member of university-named GitHub organization
Weak terms in bio (university, college, campus, ...)
University-shaped company field
Flag rule: weighted score ≥ 3

Notes. "All enriched" counts are over the 429,320 developers appearing in any Copilot fingerprint; "among flagged" over mutually exclusive.

A.2 Outlier-rule robustness

Table 7 re-estimates the two headline outcomes under alternative outlier rules on the purge-and-Copilot-dependent base sample, varying only the treatment of extreme committers. The locally-mediated decline is negative and significant under every rule; total commits are null under every rule. The winsorized specification compresses the count outcome mechanically and is reported for completeness.

Table 7: Outlier-rule ladder: headline estimates under alternative rules.

	Co-author commits		Total commits		
Outlier rule	DiD	(SE)	DiD	(SE)	N
No outlier rule	-0.172*	(0.090)	-0.446	(0.947)	11,741
Winsorize outcomes at p99	-0.014**	(0.006)	0.093	(0.467)	11,741
Trim pre-max > p99 (primary rule)	-0.186**	(0.091)	-0.539	(0.776)	11,623
Trim pre-max > p95	-0.096***	(0.031)	-0.294	(0.643)	11,157
Hard cap 1,000/month	-0.178**	(0.090)	-0.493	(0.811)	11,724

Notes. Base sample: control purge + Copilot-dependence, pair-consistent, before any outlier rule. "Pre-max" refers to a developer's maximum monthly commit count in October 2025–February 2026; rules trim the developer or winsorize the outcome as indicated. Developer and month fixed effects; SEs clustered by developer.

A.3 Matching and identification diagnostics

Table 8 collects the diagnostics referenced in Sections 5 and 6: distributional balance (variance ratios and Kolmogorov–Smirnov tests), the omnibus predictability-of-treatment check, joint pre-trend tests over three windows, and permutation inference for the headline estimate. The variance ratio for raw follower counts reflects a small number of extremely followed accounts among controls; the matched covariate is the log transform, whose ratio is 0.98. Table 9 compares primary-sample students with all other flagged students, characterizing the population the estimand covers.

Table 8: Matching and identification diagnostics, primary sample.

A. Distributional balance			
Covariate	Variance ratio	KS statistic	KS p
log(1+pre-event events)	1.01	0.005	1.000
Account age (years)	1.00	0.011	0.872
log(1+followers)	0.98	0.006	1.000
Delegation share	1.02	0.005	1.000
Pre-event events (raw)	1.27	0.005	1.000
Followers (raw)	0.08	0.006	1.000
B. Omnibus balance (logit of treatment on all covariates)			
Pseudo-R^2	0.00004		
Likelihood-ratio p	0.957		
AUC	0.505 (0.5 = arms indistinguishable)		
Propensity range	treated $[0.498, 0.525]$, control $[0.498, 0.524]$		
C. Joint pre-trend tests, co-author commits (Wald, cluster-robust)			
All 13 pre-event months	$\chi^2 = 22.24$, $p = 0.052$		
Six months nearest event	$\chi^2 = 5.93$, $p = 0.431$		
Four months nearest event	$\chi^2 = 1.86$, $p = 0.762$		
D. Permutation inference, co-author DiD			
Observed estimate	-0.181		
Permutation p (two-sided, 200 label shuffles)	< 0.005 (0 draws as extreme; null s.d. 0.096)		

Notes. Panel A: variance ratio is s_T^2/s_C^2 (benchmark 0.5–2). Panel B: logit of treatment on all matching covariates. Panel C: cluster-robust Wald tests that the treated × month event-study coefficients before February 2026 are jointly zero. Panel D: treatment labels permuted across persons, model re-estimated per draw.

Table 9: Representativeness: primary-sample students versus all other flagged students.

	Median		Mean	
	Primary	Other flagged	Primary	Other flagged
Account age (years)	3.2	2.8	4.17	3.82
Followers	4.0	2.0	18.17	19.49
Public repositories	18.0	16.0	28.2	26.08
Students	5,432		15,703	

Notes. "Other flagged" are the flagged students outside the primary sample (inactive on Copilot pre-event, rival-agent users, filtered, or unmeasurable). Profile fields as of enrichment (July 2026).

References

Daron Acemoglu and Pascual Restrepo. Tasks, automation, and the rise in US wage inequality. *Econometrica*, 90(5):1973–2016, 2022.

Ajay Agrawal, Joshua Gans, and Avi Goldfarb. *Prediction Machines: The Simple Economics of Artificial Intelligence*. Harvard Business Review Press, 2018.

Anthropic. Claude 3.7 Sonnet and Claude Code. Anthropic Announcement, February 24, 2025.

David Autor, Caroline Chin, Anna Salomons, and Bryan Seegmiller. New frontiers: The origins and content of new work, 1940–2018. *Quarterly Journal of Economics*, 139(3): 1399–1465, 2024.

Erik Brynjolfsson, Danielle Li, and Lindsey R. Raymond. Generative AI at work. *Quarterly Journal of Economics*, 139(4):1919–1971, 2024. NBER Working Paper 31161, 2023.

Brantly Callaway and Pedro H. C. Sant'Anna. Difference-in-differences with multiple time periods. *Journal of Econometrics*, 225(2):200–230, 2021.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Cognition. Introducing Devin, the first AI software engineer. Cognition Blog, March 12, 2024.

Peter Cohen, Robert Hahn, Jonathan Hall, Steven Levitt, and Robert Metcalfe. Using big data to estimate consumer surplus: The case of uber. Technical report, NBER Working Paper 22627, 2016.

Kevin Cui, Mert Demirer, Sonia Jaffe, Leon Musolff, Sida Peng, and Tobias Salz. The effects of generative AI on high-skilled work: Evidence from three field experiments with software developers. *Management Science*, 2025. Forthcoming.

Fabrizio Dell'Acqua, Edward McFowland III, Ethan R. Mollick, Hila Lifshitz-Assaf, Katherine Kellogg, Saran Rajendran, Lisa Krayer, François Candelon, and Karim R. Lakhani. Navigating the jagged technological frontier: Field experimental evidence of the effects of AI on knowledge worker productivity and quality. *Harvard Business School Working Paper*, (24-013), 2023.

Thomas Dohmke. GitHub Copilot is generally available to all developers. GitHub Blog, June 21, 2022.

Tyna Eloundou, Sam Manning, Pamela Mishkin, and Daniel Rock. GPTs are GPTs: Labor market impact potential of large language models. *Science*, 384(6702):eadh2586, 2024.

Nat Friedman. Introducing GitHub Copilot: Your AI pair programmer. GitHub Blog, June 29, 2021.

GitHub. GitHub Copilot: The agent awakens. GitHub Blog, February 6, 2025.

Andrew Goodman-Bacon. Difference-in-differences with variation in treatment timing. *Journal of Econometrics*, 225(2):254–277, 2021.

Manuel Hoffmann, Sam Boysel, Frank Nagle, Sida Peng, and Kevin Xu. Generative ai and the nature of work. Technical report, Harvard Business School Working Paper 25-021, 2024.

Eirini Kalliamvakou, Georgios Gousios, Kelly Blincoe, Leif Singer, Daniel M. German, and Daniela Damian. The promises and perils of mining github. In *Proceedings of the 11th Working Conference on Mining Software Repositories (MSR)*, 2014.

METR. Measuring the impact of early-2025 AI on experienced open-source developer productivity. Technical report, METR, 2025.

Shakked Noy and Whitney Zhang. Experimental evidence on the productivity effects of generative artificial intelligence. *Science*, 381(6654):187–192, 2023.

Sida Peng, Eirini Kalliamvakou, Peter Cihon, and Mert Demirer. The impact of AI on developer productivity: Evidence from GitHub Copilot. *arXiv preprint arXiv:2302.06590*, 2023.

Alexander Quispe. Agentic delegation and the language frontier of software developers: A model and evidence from claude code on github. Technical report, Working paper, California Institute of Technology, 2026.

David Strömberg, Victor Lei, and Yanhui Wu. The generative ai learning penalty: Evidence from chinese secondary education. Technical report, CEPR Discussion Paper No. 21577, 2025.