NBER WORKING PAPER SERIES

THE POWER OF PROXIMITY TO COWORKERS

Natalia Emanuel
Emma Harrington
Amanda Pallais

Working Paper 31880
http://www.nber.org/papers/w31880

NATIONAL BUREAU OF ECONOMIC RESEARCH
1050 Massachusetts Avenue
Cambridge, MA 02138
November 2023, Revised June 2026

This project would not have been possible without the curiosity and commitment to research of our colleagues at the firm who shared data: Lauren, Susan, and Ebube. We thank Alex Albright, Jenna Anders, Eric Anderson, Garrett Anstreicher, Christopher Campos, Raji Chakrabarti, Kirsten Clinton, Ed Glaeser, Ben Hyman, Bill Johnson, Ezra Karger, Lawrence Katz, Gizem Kosar, Sitian Liu, Daniel Mangrum, Ishan Nath, Dev Patel, Soum Shukla, Jason Somerville, Mattie Toma, Tianyi Wang, and Melanie Wasserman for thoughtful comments. We also thank numerous seminar participants. We thank Jared Fang for research assistance. The findings and conclusions expressed are solely those of the authors and do not reflect the opinions or policy of our employers, the organizations that supported this work, the Federal Reserve Bank of New York, the Federal Reserve System, or the National Bureau of Economic Research.

The Power of Proximity to Coworkers
Natalia Emanuel, Emma Harrington, and Amanda Pallais
NBER Working Paper No. 31880
November 2023, Revised June 2026
JEL No. J16, J24, M15, M53, M54, O33, R23

ABSTRACT

How does proximity to coworkers affect training and productivity? We study software engineers at a Fortune 500 firm from 2019 to 2024, leveraging two shocks to proximity: (i) the office closures in 2020 and (ii) the subsequent return-to-office mandates in 2022 and 2023. In both cases, co-located teams experienced bigger changes in proximity than distributed ones, facilitating difference-in-differences designs. We find that sitting near teammates increases coding feedback by 18.3% and improves code quality. Gains are concentrated among less-tenured and younger employees, who are building human capital. However, there is a tradeoff: experienced engineers write less code when sitting near teammates. In national US data, we find evidence that the rise of remote work has had scarring effects on young college graduates. In remotable jobs, young graduates' unemployment rate increased relative to older graduates' post-pandemic (2022 2024) compared to pre-pandemic (2017 2019), a pattern we do not observe in non-remotable jobs.

Natalia Emanuel
Federal Reserve Bank of New York
natalia@nataliaemanuel.com

Emma Harrington
University of Virginia
emma.k.harrington4@gmail.com

Amanda Pallais
Harvard University
Department of Economics
and NBER
apallais@fas.harvard.edu

A data appendix is available at http://www.nber.org/data-appendix/w31880

Coworkers are more distant than ever before. In 2024, workers lived an average of 27 miles from the office, nearly three times farther than in 2019 (Akan et al., 2024). Remote and hybrid arrangements have become commonplace: employees work from home one to two days per week on average (Flood et al., 2024) and even on days when they are in the office, many of their colleagues are still at home or in distant offices (Goldberg, 2021). The growing distance may make it harder to learn from colleagues, who traditionally have been the source of about a sixth of lifetime human capital (Herkenhoff et al., 2024). Yet as work increasingly takes place on screens (Belden, 2022), it is possible that workers can learn just as effectively from afar. In an increasingly digital world, how does proximity to coworkers impact workers' productivity and on-the-job training?

To answer this question, we study software engineers at a Fortune 500 online retailer from 2019 to 2024. We focus on engineers based in the headquarters campus whose teammates varied in their physical proximity. Before the pandemic, some teams were co-located in a single building; other teams were split across two headquarters buildings, roughly a ten-minute walk apart; and still others were completely distributed, with teammates working remotely or in satellite campuses. When the offices closed for the pandemic, these differences became immaterial. By comparing differences across team types before the offices closed — when they varied in proximity — to afterwards — when they did not — we can identify the causal effects of proximity on engineers' training and productivity. We can further investigate whether these differentials re-emerge after the firm's return-to-office (RTO) mandates in 2022 and 2023, which brought co-located teams — those assigned to a single building — back together, while leaving geographically-distributed teams apart.

We find that sitting alongside teammates increases the digital feedback engineers receive on their code. Before finalizing their code, engineers must have it reviewed by colleagues who flag problems and suggest improvements. When the offices were open, engineers on co-located teams received 23.9% more comments (1.92 more per program) than engineers on multi-building teams — those split across the firm's two headquarters buildings. Once the offices closed, this advantage largely disappeared, narrowing by 18.3% (1.47 comments per program, p-value = 0.0026). The decline stemmed entirely from reduced feedback from teammates, rather than from other colleagues. Using a machine-learning algorithm to classify comment quality, we find that the lost comments were disproportionately those predicted to be helpful, actionable, impactful, and well-reasoned.

Proximity makes it easier to ask for feedback. Engineers sitting near their teammates ask more follow-up questions, consistent with neuroscience studies showing that in-person conversation generates stronger feelings of closeness than video calls and instant messages (e.g., Schwartz et al., 2022; de Groot et al., 2015; Seltzer et al., 2012). When proximate to their teammates, engineers also tap a wider range of people for feedback. This result mirrors classic studies showing that in-person interaction facilitates, rather than replaces, phone and email communication (Gaspar and Glaeser, 1998; Allen and Henn, 2007; Agrawal and Goldfarb, 2008) and echoes recent research showing that remote work silos communication networks (Yang et al., 2022; Gibbs, Mengel and Siemroth, 2023). Even with modern technology, face-to-face interaction is a complement rather than a substitute for digital communication.

Proximity is fragile. Even a 10-minute walk between two buildings on the same campus reduces feedback as much as being multiple states away, suggesting that small frictions to face-to-face contact can have outsized effects. Additionally, a single distant colleague can generate negative externalities for the rest of the team: when a new hire turns a co-located team into a distributed one, feedback between the original teammates drops sharply — even though they still sit together. In contrast, adding a new co-located teammate has no such effect. Moving team meetings online to accommodate a distant colleague appears to weaken connections among those who remain together.

Proximity does not affect all engineers equally; instead, it most impacts those with much to learn from their colleagues. New hires and young engineers receive more feedback when co-located and lose more mentorship when remote. Female engineers are also more affected by proximity, partly because they ask fewer coworkers for feedback when they cannot do so in person.

Mentoring improves engineers' code quality in the short and long run. Around the

RTO mandates, engineers on co-located teams, who saw larger increases in face-to-face time with teammates, experienced greater improvements in code quality than those on geographically-distributed teams. After the three-days-per-week office mandate was introduced, engineers on co-located teams became 2.2 pp less likely to add files that were subsequently deleted (p-value = 0.041) and 1.4 pp less likely to introduce bugs (p-value = 0.022) than engineers on distributed teams. Less-tenured and younger engineers experienced roughly twice the gains of the average engineer. The benefits of co-location are persistent: engineers with more experience on co-located teams continued to write better code, even after everyone shifted to remote work and feedback levels equalized.

However, there is a cost to mentorship: providing feedback and improving others' code requires effort. Experienced engineers, who offer the most feedback, write less code when sitting near their colleagues. Pre-pandemic, senior engineers on co-located teams wrote less code than their counterparts on distributed teams. When the offices closed, this gap closed and when the offices reopened, it widened again. The firm faces a tradeoff between boosting junior engineers' skills and allowing senior engineers more time to program.

When mentorship falters and talent is harder to build in-house, firms can instead buy it by hiring more experienced workers. We find evidence of this response: the firm shifted towards hiring older workers when the offices closed and then back towards younger hires once they reopened. National US data suggest that this firm is not unusual: in remotable jobs, young workers have seen larger increases in unemployment between 2017−2019 and 2022−2024 than older workers, following the persistent rise in remote work. No comparable gap emerges in non-remotable jobs, a difference that emerged before the spread of generative AI and persists after accounting for occupational exposure to AI (Eloundou et al., 2024; Schubert, 2025) The labor market pressures on young people in remotable jobs help explain broader trends: our estimates suggest that remote work accounts for 64% of the total unemployment increase among young college graduates over this period.

We also see evidence of the desire to buy talent by looking at who is poached from the Fortune 500 firm we study. When the offices were closed, engineers trained on co-located teams were more likely to leave for better jobs, suggesting that investing more in workers made it more likely they would leave the firm. Because the firm does not fully capture the returns to these investments, it might under-incentivize coworkers' investments in each other's human capital, even if it could perfectly observe them. This classic problem in general human capital formation (Becker, 1964) makes coworkers' social bonds to one another — which are often forged face-to-face — critical in facilitating investment.

Consistent with having the most to gain from proximity, young and less-tenured engineers come into the office more during the RTO periods, as evidenced by the firm's badge data. They are particularly likely to go in when their teammates are based in the same office, suggesting that the prospect of being near colleagues is a key draw for those with the most to learn. This pattern holds nationally: in 2022−2024, younger workers were more likely to be back in the office, both within tech and among college-educated workers broadly, even when limiting to non-parents. Yet the office only facilitates mentorship when colleagues are present: if teammates remain remote, young workers cannot easily sidestep the scarring effects of distance on their human capital development.

We provide evidence that proximity to coworkers increases on-the-job training and helps young and less-tenured workers build human capital. This finding highlights several key dimensions of heterogeneity in the impacts of remote work. Remote work may harm future productivity while boosting current productivity; it may harm the quality of work while increasing its quantity; and it may disadvantage inexperienced workers while helping experienced workers thrive. These heterogeneous impacts can help reconcile the wide range of productivity estimates in the literature, which tend to find positive effects on the quantity of work in settings with experienced workers (Bloom et al., 2015; Angelici and Profeta, 2024; Choudhury, Foroughi and Larson, 2021; Bloom, Han and Liang, 2022; Choudhury et al., 2024; Fenizia and Kirchmaier, 2024), while finding less positive or even negative effects on work quality especially when workers are less experienced (Dutcher, 2012; Battiston, Blanes i Vidal and Kirchmaier, 2021; Gibbs, Mengel and Siemroth, 2023; Atkin, Chen and Popov, 2022; Emanuel and Harrington, 2024; Ho, Jalota and Karandikar, 2024; Jalota and Ho, 2024; Wang, Li and Yu, 2025). These dimensions of heterogeneity are not the only drivers of differences across studies — remote work, for instance, appears consistently more challenging in developing countries (Atkin, Chen and Popov, 2022; Ho, Jalota and Karandikar, 2024; Jalota and Ho, 2024). But accounting for worker learning is essential to understanding the full range of remote work's effects.

Our findings also help resolve the puzzle of remote work's rarity before the pandemic (Mas and Pallais, 2020), which seemed at odds with workers' strong preference for remote work (Mas and Pallais, 2017; Maestas et al., 2023; He, Neumark and Weng, 2021; Lewandowski, Lipowska and Smoter, 2023; Cullen, Pakzad-Hurson and Perez-Truglia, 2025) and its often positive short-run effects on output. One explanation is that the value of the office lies in training for tomorrow and improving the quality — not the quantity — of work today.

Our paper also speaks to the debate over how changes in the nature of work affect workers at different stages of their careers. We find that remote work depresses demand for junior versus senior talent, aligning with Schubert (2025)'s finding that remote work raises experience requirements in job postings. The resulting increase in intergenerational inequality may compound those of generative AI, which recent evidence also finds reduces demand for junior talent (Brynjolfsson, Chandar and Chen, 2025; Lichtinger and Hosseini Maasoum, 2025).

The rest of the paper is organized as follows. Section I provides context on software engineering. Section II introduces the data. Section III describes the empirical design. Sections IV, V, and VI discuss how proximity impacts feedback, code quality, and coding output, respectively. Sections VIII-IX consider implications for hiring and office attendance in the firm and nationally. Section X concludes.

I Background on Software Development

Our data comes from a Fortune 500 online retailer, whose software engineers perform tasks typical of the industry: building the website's front-end interface, maintaining the back-end product catalog, developing internal tools for other parts of the firm (e.g., the finance team), and working on AI-driven features like product recommendation services.

The engineers follow a standard process for developing business-critical code on GitHub. When making changes, an engineer creates a "branch" from the primary code-base, which separates their rough drafts from live code. Before their changes can be merged back into live code, they must be reviewed. This key quality-control step lets other engineers verify that the new code is well-tested, error-free, and understandable to future engineers (including the author's future self). Typically, two reviewers weigh in: one from the author's immediate team and one with relevant external experience, such as familiarity with that part of the codebase. Strong norms and managerial expectations encourage engineers to give feedback when asked, but there are no explicit incentives to comment extensively, since such incentives would be too easily gamed with superficial comments.

In addition to vetting the current code, reviewers can build the author's skills. One manager explained: "We ask senior, technical folks...to make their code reviews a learning opportunity by, for example, including the reasoning behind suggested changes." In line with this goal, the reviewer has more experience at the firm than the code's author in 70% of reviews.

Teams follow an Agile management system that includes daily "stand-up" meetings where everyone literally stands to keep meetings brief. In these ten-to-fifteen-minute meetings, engineers share progress updates and flag anything blocking their work. These meetings also give engineers a chance to ask others to review their code, which many prefer to do in person rather than digitally (e.g., on Slack or GitHub).

The format of these daily meetings depends on teammates' physical proximity. Most teams follow an implicit "one Zoom, all Zoom" norm: if one person cannot be physically present, everyone pulls out their laptops and hops on a video call. As one engineer noted, "[my team] would almost never book a room and held all of our meetings [online] since we had a remote team member." Even within the headquarters campus, a short walk between buildings was enough to push some meetings online: it was hard to justify a ten-minute walk each way for an equally short meeting. Beyond daily stand-ups and the chitchat before and after them, co-located engineers also had more chances for informal face-to-face interactions around their desks, due to the firm's open office plan. Our paper investigates the ramifications of this close proximity on engineers' training and productivity.

II Data

We collate data to (i) characterize workers' backgrounds, (ii) identify workers' physical proximity to their teammates, and (iii) measure different facets of workers' productivity and on-the-job training. Together, our different data sources span from mid-2019 to mid-2024, covering the office closures of 2020 and two return-to-office waves in 2022 and 2023.

II.A Workers' Backgrounds and Demographics

Many of the firm's engineers are young and new to the company. Before the closures, engineers were, on average, 29 years old with just 1.4 years at the company, suggesting many were still building both general and firm-specific human capital (row $1-2$ of Table I). As is common in software engineering, the workforce is male-dominated: men compose 81% of our sample and 75% of programmers nationally (Ruggles et al., 2022). Only 22% of engineers had children — and just 17% of female engineers — consistent with the workforce's relative youth.[1]

II.B Identifying Teammates' Physical Proximity

We use personnel data to distinguish between co-located and distributed teams. We define an engineer's teammates as all employees reporting to the same manager.[2] Using each teammate's assigned office, we then classify teams as (i) co-located in one building, (ii) split across multiple buildings on the firm's headquarters campus (a ten-minute walk apart), or (iii) geographically distributed, with some members working remotely or in satellite offices. Throughout, we focus on engineers in the headquarters campus, whose proximity to their teammates varies. Before the pandemic, 49% of headquarter-campus

[1]Starting in June of 2020, the firm began collecting caregiving information, including childcare responsibilities. These data cover 70% of software engineers in our sample.

[2]This strategy identifies the teammates for the plurality of workers. We exclude the 1.2% of engineers with high-level managers who oversee multiple teams, but the results are similar without this restriction.

engineers were on co-located teams, 34% were on teams split across buildings, and 17% were on geographically-distributed teams.

By the time the offices reopened, the firm had consolidated its headquarter operations into one building, while expanding its satellite offices and remote workforce. Consequently, only 25% of headquarter-campus engineers were on co-located teams, with the rest geographically distributed.

Badge Data around RTOs: From 2022 onward, we observe office attendance using badge data, which records each employee's daily entry into the firm's buildings. During the first RTO — which encouraged two in-office days per week — engineers came in an average of 0.6 days per week. During the second RTO — which called for three in-office days per week — office attendance rose to 1.9 days per week. Throughout both periods, the firm did not specify which days employees should come in, but most engineers came in on Tuesday through Thursday (Figure A.1).

II.C Measuring Productivity & Learning

We have two extracts of data from the firm's code development system covering different periods. Around the pandemic office closures, we observe the quantity of code engineers wrote in the firm's main code-base and the feedback they received on it. Around the RTOs, we observe code quantity across all code-bases as well as proxies for its quality.

Data around the Office Closures: Our first extract spans from August 2019 until December 2020 and tracks all changes to the main code-base along with the feedback that engineers received on this code. In our analysis sample, 1,055 engineers made 29,809 distinct changes to the code-base and received 174,014 comments from their coworkers.

For every distinct change, we see who made the change, when it occurred, how many lines of code were added, and how many files in the main code-base were changed. The average modification — or "program" — adds 346 lines across 7 files. Engineers typically write 2 programs in the main code-base each month (see Table A.1 for the distribution). Our preferred metric of immediate productivity is programs written, as engineers are encouraged to write shorter, more modular programs that are easier to test and debug.

The data also detail every comment exchanged during each code review, including its text, timestamp, and the identity of the commenter. These comments function much like feedback on a paper draft: the commenter highlights specific portions of the code and comments about its structure, functionality, or readability. Engineers receive an average of 6.5 comments on each program (Table A.1 shows the distribution). Each comment averages 16 words (100 characters). The feedback is usually substantive, not only telling the engineer how to change the code but also explaining the underlying reasoning (see Section IV.B). Thus, this dataset lets us measure investments in engineers' human capital.

Data Around the RTOs: The second extract measures code quantity and quality for all the firm's code-bases from December 2020 to mid-2024. As before, these data track how frequently engineers make changes and the extent of those changes, but now spanning all code-bases rather than only the main one.

This dataset also includes two code-quality metrics. Engineers often assess quality along two dimensions: churn — whether engineers are ineffectively spinning their wheels — and bugs — whether they introduce errors or vulnerabilities. We measure churn as the frequency with which engineers add files that are deleted within the subsequent six months, which occurs in 15% of programs. A file may be deleted because it was a tangled mess of logic (also known as "spaghetti code") or because it introduced a feature that the firm later abandoned. Either way, such disposable code is not a good sign about the quality or utility of an engineer's code. We measure bugs by identifying changes that are immediately and fully reverted, typically indicating that the engineer's changes precipitated an emergency requiring a rapid rollback to an earlier version of the code. These more serious problems occur in 3.5% of programs.

III Empirical Design

Our goal is to identify how proximity to teammates affects engineers' investments in each other's human capital, as well as the quantity and quality of their code. We exploit two key shocks to proximity: (i) the office closures of COVID-19 and (ii) the subsequent return-to-office mandates. In both cases, co-located teams saw bigger changes in proximity than distributed ones, facilitating difference-in-differences (DiD) designs.

Office Closure Design: When the offices closed, engineers on co-located teams saw larger losses in proximity to their teammates than engineers on distributed teams. We leverage this difference in the following DiD design:

$$Y_{it} = \beta \text{Post Closure}_t \cdot \text{Co-Located Team}_i + \mu_i + \mu_t + X_{it}'\Gamma + \epsilon_{it}, \tag{1}$$

where Y_{it} represents either the feedback that engineer i received on her programs in month t (in Section IV) or the number of programs she wrote (in Section VI), but not the quality of her code, data on which is only available after the closures. The specification includes fixed effects for engineer (μ_i) and month (μ_t). Since this design considers a single focal event, it does not run into the problems that can arise with staggered treatment timing (e.g., Goodman-Bacon, 2021). We cluster standard errors by team, the unit of treatment assignment. We define Co-Located Team$_i = 1$ if an engineer was consistently on a co-located team — with all members assigned to the same building — throughout the pre-period, ensuring a stable treatment definition. The results are similar when we define team type contemporaneously.

Our primary identification strategy focuses on engineers whose teammates were all on the headquarters campus, but whose team was either all in one building or split across two. Within this sample, co-location often came down to chance: when a new hire joined, was there an open seat near the rest of the team? These seating logistics were more challenging for engineers working on internal tools, who aimed to sit near both end-users and fellow engineers, resulting in lower co-location rates (35%). Engineers on co-located and multi-building teams are broadly similar in their observable characteristics (Column 7 of Table I). Accounting for engineering group, engineers on one- and multi-building teams look observably similar on all dimensions except tenure (Column 8 of Table I).[3]

[3] In supplementary analysis, we compare co-located and geographically-distributed teams, but

For β to reflect the causal effect of losing proximity, a parallel-trends assumption must hold: absent the closures, feedback to engineers on co-located and multi-building teams would have evolved similarly. Reassuringly, pre-closure trends in feedback are parallel (Figure I). Nevertheless, potential threats remain. Workers and managers may sort onto co-located teams, and tasks may differ across team types — differences that could interact with time-varying shocks, particularly those from the pandemic.

Our preferred specification aims to relax this identifying assumption by including month-specific controls for the engineer's group, tenure, and age. These controls absorb residual variation, address imbalances across co-located and multi-building teams, and absorb interactions between these factors and the pandemic. For instance, if the pandemic affected website engineers differently from those working on internal tools, that difference would be captured by month-by-group fixed effects. Likewise, if the pandemic posed particular challenges for engineers who were new to the firm, that difference would be absorbed by month-by-tenure fixed effects. When analyzing feedback, we also control for program scope (quartics in the number of lines added, lines deleted, and files changed), which may mediate the extent of feedback engineers receive. Our full set of controls also includes month-specific controls for team size, gender, race, home zipcode, job level, and initial building assignment. The robustness of our results to the inclusion of various controls lends confidence to the design.

As a placebo check, we examine feedback from non-teammates, who experienced no differential change in proximity across co-located and multi-building teams. The null result for this check helps assuage identification concerns and contradicts some alternative explanations. If engineers on co-located teams simply needed more feedback or undertook projects requiring more input before the closures but not afterward, we would expect the effects to show up in non-teammate feedback as well. Instead, the differential changes are confined to teammate feedback — where proximity changed differentially — and absent for non-teammate feedback, where proximity evolved similarly for all engineers.

geographically-distributed engineers are more selected: older, more tenured, and higher level (Table A.2).

Finally, as a complementary strategy, we leverage within-engineer switches in team type. Before the office closures, 3.7% of engineers switched between co-located and multi-building teams, and another 6.0% saw changes in teammate proximity due to colleagues joining, leaving, or switching teams. These switches are associated with changes in feedback when the offices are open, but not after they close. This pattern holds when including engineer and engineer-by-manager fixed effects, helping to address concerns that selection into team type — at either the worker or manager level — drives the results.

Office Reopening Design: During the RTO period, the firm's headquarters campus had a single building, so we compare engineers on co-located (same-city) teams to those on geographically-distributed teams, defined as having at least one member at a different campus or working permanently remotely.[4] When the offices reopened, engineers on co-located teams saw larger gains in proximity than those on distributed teams. We estimate:

$$Y_{it} = \sum_{\tau \in \{\text{Closed,1st RTO,2nd RTO}\}} \gamma_\tau \, \text{Co-Located Team}_{it} \, \mathbb{1}[t \in \tau] + \mu_i + \mu_t + X_{it}'\Theta + u_{it}, \quad (2)$$

where Y_{it} is either the number of programs written or a measure of program quality, but not feedback received, data on which is only available until December 2020. This specification allows the effect of being on a co-located team to vary across three distinct periods — the office closures, the first two-day-per-week RTO, and the second three-day-per-week RTO. We allow Co-Located Team_{it} to vary over time because, over this multi-year period, nearly half of engineers (48.6%) spent time on both co-located (same-city) teams and geographically-distributed teams. The resulting within-engineer variation — with hundreds of switches per period — allows us to identify differences across team types in every period even with individual fixed effects (μ_i). Throughout, we focus on engineers who worked on the headquarters campus but whose teammates' locations varied.

The identifying assumption is that switching across team types is uncorrelated with other shocks to an engineer's code quantity or quality. One concern is that assignment to co-

[4]Permanently remote workers were given exceptions typically because they lived far from any campus, averaging 123 miles from the closest office, compared to 25 miles for on-site engineers.

located versus geographically-distributed teams appears less random than assignment to co-located versus multi-building teams: engineers on geographically-distributed teams tend to be older, more tenured, and more likely to work on internal tools (Table A.3). Our preferred specification addresses these differences by including month-by-tenure, month-by-age, and month-by-engineering-group fixed effects. We also show robustness to the full suite of controls.

Our strongest test of the identifying assumption exploits the fact that switching team types should *only* affect coding when it changes physical proximity to teammates. We examine three distinct periods — the office closures, the first RTO (two days per week), and the second RTO (three days per week) — and generate testable predictions for each. We predict that $\gamma_{closed} \approx 0$ since being assigned to the same office as one's teammates is inconsequential when that office is closed. We further expect that $\gamma_{2nd\,RTO} > \gamma_{1st\,RTO}$, since the second RTO entailed more days in the office and more overlap in teammates' schedules, since Tuesday−Thursday all became heavily trafficked days (Figure A.1). If switches across team types were simply correlated with unrelated shocks to code quantity or quality, this specific pattern of temporal heterogeneity would be difficult to explain.

IV Proximity & Feedback

If physical proximity causally boosts online feedback, three patterns should emerge around the office closures: (i) co-located engineers should receive more feedback when the offices are open, (ii) feedback should decline for everyone once the offices close, and (iii) the gap in feedback between previously co-located and already-distributed teams should narrow. All three predictions are borne out in Figure Ia. While the offices were open (left of the vertical line), engineers on co-located teams received 23.9% (or 1.92) more comments on their programs than engineers on multi-building teams, even after controlling for program length, engineer tenure, age, and engineering group (p-value = 0.0005). Once all engineers were remote (right of the vertical line), this difference shrank to 7.1% (or 0.57 comments per program), with the sharpest decline coinciding with the closures.

Our difference-in-differences design — comparing the pre- and post-closure feedback gap across team types — indicates that the greater loss of proximity among co-located teams reduced the feedback they received by 16.8%. Including individual engineer fixed effects, the DiD coefficient is 18.3% (or 1.47 comments per program, p-value = 0.0026, Column 3 of Table II). This estimate remains similar across alternative specifications, ranging from no controls ($R^2 = 1.5\%$) to the full set ($R^2 = 61\%$, Table IIa).[5] Results are similar when measuring feedback as total characters or words (Table IIIa, Columns 1−2), and effects are even larger when focusing on substantive, technical feedback, as discussed in Section IV.B. Furthermore, when teammates are proximate, they give feedback more quickly (Table IIIa, Column 3).

Notably, the differences in feedback between engineers on co-located and multi-building teams are driven solely by feedback from teammates, with no detectable effect on feedback from engineers on different teams (Columns 6−7 of Table II). This null contradicts some alternative explanations. If engineers on co-located teams simply needed more feedback, we would expect this to show up in non-teammate feedback as well. Similarly, if the projects undertaken by co-located teams happened to necessitate more feedback before the closures but not afterwards, the resulting decline in feedback would have occurred across all sources, not just within their own small teams.

Turning to engineers who switch team types, we find that the same engineer receives more feedback when on a co-located team than when on a multi-building team — but only when the offices are open (Column 1 of Table A.4). Once offices close and physical assignments become irrelevant, the co-location advantage disappears. Suggestively, these patterns persist when we include worker-by-manager fixed effects, indicating that the same engineer working under the same manager receives more feedback when her team is co-located than when it becomes distributed (e.g., because a new hire cannot find a seat near the rest of the team, Columns 3−4 of Table A.4).[6]

[5] Our full set of controls also include team size, demographics, zipcode, job level, and initial building.

[6] We consider a number of additional robustness checks. First, while our main analysis focuses on headquarter-campus engineers (to ensure more apples-to-apples comparisons), our findings are similar if we instead analyze all engineers regardless of their location (Figure A.2b). Second, our results are also similar if we limit our analysis to internal-tool engineers, who are more likely to be distributed to work near the users of their tools (Figure A.2c). Finally, our results are comparable if we consider teams that were more

Face-to-face and digital communication appear to be complements rather than substitutes: instead of compensating for lost in-person interaction with more online feedback, engineers who lose physical proximity exchange less online feedback as well. We find the same pattern in references to other communication channels (e.g., Slack) within code reviews (Table IIIa, Column 4). To the extent that proximate teammates also communicate more in person, our estimate of proximity's effect on online mentorship represents a lower bound on its total effect.

Drivers of the Complementarity: Being face-to-face with teammates appears to make engineers more comfortable seeking online feedback — both from a wider range of colleagues (the extensive margin) and in greater depth (the intensive margin). On the extensive margin, proximity expands engineers' feedback networks (Table IIIb, Column 3). Engineers on co-located teams are less likely to return repeatedly to the same commenter, instead drawing on a wider pool of commenters when the offices are open.

On the intensive margin, losing proximity reduces the back-and-forth exchange about the code (Table IIIc, Column 1). Much of this effect operates through follow-up questions: engineers on co-located teams ask 48.4% (0.12) more follow-up questions when the offices are open, a differential that vanishes once they close (Table IIIc, Column 4, p-value = 0.0083). Roughly half of proximity's total effect on feedback comes through these follow-up exchanges, with the other half coming from initial reviewer feedback (Column 2).

IV.A What Features of Proximity Matter?

Small Frictions: Small barriers to face-to-face contact can undermine feedback as much as large ones. Figure Ib shows that engineers on teams separated by a ten-minute walk receive no more feedback than those on teams spread across the country — both of whom receive less feedback than engineers on co-located teams. These findings accord with research on academics showing that being in a different building, or even on a different floor, reduces coauthorship (Kraut, Egido and Galegher, 1988; Catalini, 2018; Salazar Miranda and Claudel, 2021). Our results show that these small frictions also have big effects

or less co-located than average rather than whether all teammates were co-located (Figure A.2d).

for teammates who share projects and meet daily.

Externalities from Distant Teammates: The firm's "one Zoom, all Zoom" norm means that a single distant teammate can shift the entire team to virtual interaction. Consistent with this norm, we find that having just one distant teammate reduces feedback even among teammates who sit together. While the offices are open, engineers in the same building exchange 14.5% less feedback when even one teammate is in another building, relative to pairs whose whole team is in the same building (p-value = 0.037, Figure A.3).

New hires provide further evidence of these externalities. When no desk is available near the rest of the team, a new hire can transform a co-located team into a multi-building one. We find the transition to a multi-building team is associated with a sharp decline in the feedback exchanged between the original teammates who continue to sit together (the solid line in Figure Ic). By contrast, adding a new hire who does not disrupt the team's co-location produces no such change (the dashed line in Figure Ic). Taking the difference in these differences implies that gaining a single distant teammate reduces feedback by 1.71 comments per program (p-value = 0.095).[7] This decline is large enough to meaningfully erode the benefits of proximity for those who remain co-located.

IV.B Substantive Feedback

During code reviews, engineers exchange substantive feedback about their code, which diminishes when engineers are no longer co-located. Figure IIa tracks how losing proximity to teammates affects the frequency of the hundred most common words in the comments, excluding stop words like pronouns and prepositions (Bird, Klein and Loper, 2009). When engineers lose proximity to their teammates, the frequency of almost all of these words declines. Notably, many of these words are directly tied to how the code

[7]Sixteen teams (with 46 engineers) switch from co-located to multi-building teams around a new hire, and 117 teams (with 369 engineers) hire someone new but do not change their co-location. Our DiD estimates

$$\frac{\text{Comments}}{\text{Review}}_{ijt} = \alpha \text{Post Hire}_t \times \text{From One to Multi-Building Team}_{ij} + \psi \text{Post Hire}_t + \mu_{ij} + \epsilon_{ijt} \quad (3)$$

where μ_{ij} represents programmer (i) by commenter (j) pair fixed effects, ψ captures the impact of any new hire, and α captures the additional effect of a new hire who makes the team distributed.

works — e.g., about its *functions*, *methods*, *models*, and *return* values — and what the programmer has done to *check* programs to ensure they do not *throw exceptions* or create other bugs. These results suggest that losing proximity does not merely trim the fat of nitpicky comments, but instead reduces substantive feedback that may be critical for improving code quality.

Similar patterns emerge when we use supervised machine learning to assess comment quality. We hired external software engineers to label a random sample of 5,377 comments as (i) helpful, (ii) explaining the underlying reasoning for suggested changes, (iii) actionable, and (iv) likely to change the code (see Appendix III.A for details). Evaluators tended to view the comments positively, judging 76% to be helpful, 51% as explaining their reasoning, 60% to be actionable, and 52% as likely to change the code. We then trained a gradient-boosted decision tree (Chen and Guestrin, 2016) on the text of these labeled comments to predict the ratings for all 174,014 comments in the data (see Appendix III.B for details), achieving 64−78% accuracy in holdout samples.

Figure IIb shows that losing proximity to teammates reduces comments that would likely be rated positively. Before the office closures, engineers on co-located teams received more comments predicted to be helpful, well-reasoned, actionable, and likely to change the code. Once the offices closed, these gaps all disappeared — and high-quality comments declined across the board. In percentage terms, the decline in high-quality comments of 21−23% (Figure IIb) exceeds the overall decline in comment volume of 18.3% (Figure Ia). Thus, the comments that remain are of lower predicted quality (Table A.5b): 2.9 pp fewer comments are helpful (p-value = 0.039); 1.7 pp fewer explain their reasoning (p-value = 0.094); 1.7 pp fewer are actionable (p-value = 0.17), and 1.9 pp fewer likely change the code (p-value = 0.072).

IV.C Heterogeneous Effects of Proximity on Feedback

Feedback can help build both firm-specific and general human capital. Less-tenured engineers stand to gain more from firm-specific knowledge — such as familiarity with proprietary tools — while younger engineers stand to gain most from general advice, such as how to structure code. We would consequently expect both groups to receive more feedback than their more experienced counterparts. This is exactly what we see on co-located teams. By contrast, when teams are distributed, inexperienced engineers receive little more feedback than their more experienced colleagues.

Figure IIIa illustrates the heterogeneity by firm tenure. When offices are open, less-tenured engineers receive more feedback, particularly on co-located teams. When the offices close, feedback declines and converges to a uniformly lower level regardless of tenure. The drop is sharpest for junior engineers on co-located teams, who lose 2.03 more comments per program than junior engineers on already-distributed teams (p-value = 0.001). When engineers are distributed, less-tenured engineers have less opportunity to learn about the firm.

Figure IIIb shows similar patterns by engineer age. When the offices close, younger engineers cease to receive more feedback than their older colleagues. Those on co-located teams are more acutely affected by the closures, losing 2.47 more comments per program than young engineers on already-distributed teams (p-value = 0.0001). The age heterogeneity persists after controlling for the interaction of tenure and proximity (Table A.7).

By Gender: Women receive more feedback on their code than men, but only when they are co-located with their whole team (Figure IIIc). When the offices are open, women on co-located teams receive more feedback than their male teammates, while women on multi-building teams receive less feedback than the men on their teams. When the offices close, female engineers on co-located teams lose 3.71 (38.9%) more comments per program than female engineers on already-distributed teams (p-value$<$ 0.0001). This gender interaction persists after controlling for both tenure and age (Columns 5−6 in Table A.7).

Men are not inured to the effects of losing proximity, however: they lose 1.01 comments per program (13.1%, p-value = 0.047). For men, the effects are concentrated among younger engineers (gray squares in Columns 1 and 3 of Figure IIId). For women, the effects extend to older engineers who are new to the firm (black triangle, Column 2). Among engineers who are neither young nor new to the firm, proximity has no detectable effect on feedback for either men or women (Column 4).

Women appear more reluctant to solicit feedback remotely. When engineers lose proximity to their teammates, women receive feedback from 14.7% (0.26) fewer people per program (p-value = 0.0078, see Figure A.7) compared to a negligible decline of 2.6% (0.05) for men (p-value of gender difference = 0.0056). When women lose proximity to teammates, they stop asking as many people for feedback, while men continue much as before.

One might worry that the additional feedback women receive when sitting near their teammates reflects "mansplaining" rather than mentorship. Several patterns argue against this interpretation. First, feedback from both male and female colleagues declines when proximity is lost (Figure A.8a). Second, women receive fewer rude comments when co-located with teammates (Figure A.8b).[8] Third, like younger and less-tenured engineers, women disproportionately lose high-quality comments — predicted to be helpful, well-reasoned, actionable, and impactful — when proximity is lost (Figure A.9). Thus, the additional feedback women receive in person appears to be a benefit, not a burden.

In sum, these patterns suggest that losing proximity makes it particularly hard for less-tenured, younger, and female engineers to learn.

V Proximity & Code Quality

Proximity increases substantive feedback, which we would expect to improve code quality, especially for engineers who have the most to learn. These improvements could come through two mechanisms. The first is immediate: more detailed reviews from nearby teammates could catch errors and improve the structure of the code. The second operates over time: repeatedly receiving more substantive feedback may help engineers learn and write better code even when they are working apart from their teammates.

Immediate Impacts. To analyze immediate impacts, we turn to the RTO design, since code quality metrics are unavailable around the office closures. Figure IVa shows that the RTOs increased proximity more for co-located teams than for distributed ones: during

[8] We evaluate comments' rudeness using the same process as for their substance (see Appendix C).

the second RTO, for example, co-located engineers worked near 27% of their teammates on a typical day versus 13% for geographically-distributed teams.[9] The contrast is starker when considering days with the whole team present: 58% of co-located teams had at least one fully in-person day per month versus just 1% of geographically-distributed teams. These fully in-person days may be especially valuable for cohesion, given the negative externalities from even a single distant teammate (Section IV.A).

Exploiting this differential increase in proximity, we find that co-location improves code quality for inexperienced engineers. Figure IVb examines "disposable" code — files that are eventually deleted, either because they were poorly written or misdirected. During the fully remote period, rates of disposable code are similar across co-located and geographically-distributed teams, consistent with office assignments being irrelevant when the offices are closed. The same holds during the first RTO, which generated only modest increases in proximity. During the second RTO, however, engineers on co-located teams added 2.2 pp fewer files that were later deleted than engineers on geographically-distributed teams (p-value = 0.041). This improvement in concentrated among engineers who have the most to learn, with a 5.5 pp improvement for less-tenured engineers (the dark gray triangle, p-value = 0.097) and a 4.6 pp improvement for young engineers (the dark gray square, p-value = 0.016). By contrast, older and more tenured engineers show minimal differences across team types (the light gray square and triangle, respectively).

Similar patterns emerge for bugs, defined as programs that are immediately and fully reverted after introducing a problem (Figure IVc). During the second RTO, engineers on co-located teams were 1.4 pp less likely to introduce bugs than those on geographically-distributed teams (p-value = 0.022), a gap significantly larger than in either the first RTO or closure period (Columns 1−2 of Table A.10c). The effect was nearly twice as large for engineers who were new to the firm at 2.7 pp (p-value = 0.019).

We probe the robustness of these code-quality results to alternative controls. The results

[9] In the second RTO, engineers on co-located teams went to the office 41% of weekdays. On those days, 64% of their teammates were in, meaning they were with 27% of their teammates on average (= 100%×0.41× 0.64). When distributed engineers went to the office on 39% of weekdays, only 33% of their teammates were in, so they were with 13% of their teammates on average (= 100%×0.39× 0.33).

on disposable code are nearly identical with the full suite of controls (Figure A.10b). For the rarer outcome of introducing bugs, adding the full suite of controls does not appreciably change the point estimates but reduces their precision (Figure A.10c).

Long-Run Effects. To assess the long-run impacts of proximity on code quality, we examine whether pre-closure exposure to co-located teams predicts code quality once everyone is working remotely and receiving comparable feedback. These specifications all control for engineer age, tenure, and engineering group. Engineers who had been on co-located teams were 2.37 pp less likely to write disposable code (p-value = 0.013, Figure A.4a) and 3.09 pp less likely to introduce bugs during the closures (p-value = 0.0012, Figure A.4b). Much of these gaps persist when we include current team fixed effects: when two engineers work on the same team, the one who had been on a co-located team before the closures tends to write higher-quality code (Table A.6). Consistent with human capital accumulation, code quality improves monotonically with the number of pre-closure months spent on co-located teams (see Figure A.5).

Together, these results suggest that engineers who sit with their teammates receive more feedback, which builds their skills and enables them to write higher-quality code in both the short- and long-run.

VI Proximity & Code Quantity

Feedback can build engineers' skills, but at what cost? And who bears this burden? Feedback disproportionately comes from experienced engineers, and it is their comments that decline when proximity is lost (Figure Va). Giving such feedback is time-intensive: it requires reading and understanding the code, diagnosing potential issues, suggesting changes, and explaining the reasoning behind them. Losing proximity might therefore free senior engineers to write more code themselves.

The data support the hypothesis that proximity to teammates has an opportunity cost for senior engineers. Figure Vb shows that while the offices were open, senior engineers on co-located teams wrote 0.76 fewer programs per month for the firm's main code-base, consistent with mentorship consuming a meaningful share of their time (p-value = 0.0005). Once the offices closed, this difference disappeared — and senior engineers who lost proximity to their teammates saw a relative increase in output of 0.58 programs per month (p-value = 0.0014). Junior engineers showed no such pattern: they wrote as many programs on co-located teams as distributed ones before the offices closed. Once the offices closed, however, junior engineers previously on co-located teams wrote more programs, suggesting that prior feedback increased their human capital and sustained their long-run productivity.

For all engineers, our DiD design indicates that losing proximity to teammates increases immediate output by 0.48 programs per month (p-value = 0.0002, Column 1 of Table A.8). We see similar effects for other metrics of output, including total lines of code written and total number of files changed, both overall and by seniority (Columns 3−6). Results are similar for alternative specifications in Table A.9. The relative changes in productivity are more gradual than for feedback but are still unusual compared to other changes over this period.

We can further analyze whether differences in code quantity re-emerged after the offices reopen. Figure VI shows that during the fully remote period and the first RTO, productivity was similar regardless of team type. In the second RTO, this dynamic shifted: as co-located engineers started spending more time with their teammates, they wrote 0.91 (11.7%) fewer programs per month than those on geographically-distributed teams (p-value = 0.044). This pattern is robust to using the full suite of controls (Figure A.10a) and analyzing alternative measures of output like lines added (Table A.11).

This decline in programming quantity around the RTOs is driven by experienced engineers, in stark contrast to the code-quality improvements concentrated among juniors. Among engineers with more than sixteen months of tenure (the light gray triangles), those on co-located teams wrote fewer programs during the second RTO, while no significant difference emerged for less-tenured engineers (the dark gray triangles). Analogously, for engineers who were at least 29 years old (the light gray squares), those on co-located teams wrote 1.4 fewer programs per month during the second RTO (p-value = 0.0064, Columns 4−5 of Table A.10a), while no detectable difference emerged for younger engineers (the dark gray squares). These patterns suggest that co-location imposes an opportunity cost on experienced engineers, so when not sitting near junior colleagues, they get more done.

Together, these results point to a central tradeoff: proximity accelerates junior engineers' human capital development but costs senior engineers time. These findings resonated with engineers at the firm. When we presented our findings internally, one senior engineer described them as "a punch in the gut," explaining that she felt more productive at home but had worried that it was because she was mentoring less. To her — and the other nodding heads in the Zoom room — our results confirmed this concern.

VII Who Comes into the Office?

If young engineers have the most to gain from proximity, we would expect them to come into the office more — and they do. At our partner firm, engineers under the age of 29 were 8.8 pp (37.6%) more likely to come into the office during the two RTOs than older engineers when on co-located teams (the solid line in Figure VIIa).[10] This difference is roughly halved on geographically-distributed teams, where coming in cannot deliver proximity to the whole team (p-value of difference = 0.0085). Young engineers come in both for their teammates and their managers: co-located managers raise attendance by 2.6 pp, but the pull of co-located teammates is even stronger at 5.1 pp (Table A.13). Similar patterns hold for new hires (Table A.14) and persist after adjusting for commute time and parental status (Table A.15). These results suggest that those with the most to learn have a particular demand for face-to-face mentorship, although other factors — such as demand for socialization — may also play a role.

Age differences in office attendance extend throughout the tech sector. Figure VIIb draws on survey data from more than twenty thousand software engineers collected by Stack

[10] The oldest engineers — who are likely doing the most mentoring — also show elevated attendance on co-located teams, producing a U-shaped pattern by age. For engineers over forty, the gap between co-located and distributed teams is 5.7 pp (p-value = 0.0004).

Overflow, the leading Q&A site for software developers (Stack Overflow, 2023). Nearly half of engineers under age 25 are in the office each day, compared to a quarter of older engineers (p-value < 0.00001). Within age groups, engineers with less than five years of coding experience are 5.8 pp more likely to be on-site than their same-age peers with more experience (Figure VIIb, p-value < 0.00001), consistent with those who have the most to learn having the strongest incentive to show up.

The pattern extends beyond tech. Census Household Pulse data show higher office attendance among young college-educated workers (U.S. Census Bureau, 2023) (Figure VIIc). This pattern persists when limiting to non-parents, indicating that it is not simply driven by parents being more likely to work from home (Figure A.15). While many forces may contribute, the results are consistent with a few complementary mechanisms: young workers are more likely to choose to go into the office to build their skills, firms are more adamant that young workers come into the office, and firms are more willing to hire young workers into roles where they can be mentored in person.

VIII Who is Hired?

Our results indicate that without proximity to coworkers, it is harder to develop talent. A natural response to reduced proximity would be to buy talent built elsewhere, by hiring more experienced engineers. Figure VIIIa shows the firm does exactly this. When the offices were closed (in black), the firm hired older engineers, effectively buying talent built at other firms. By contrast, when the offices were open — both pre-closure (in light gray) and post-RTO (in dark gray) — the firm hired younger engineers, who may have needed to build their human capital at the firm. Indeed, Figure VIIIb shows that the whole age distribution of hires shifted: over half of hires were under the age of 29 both before closure and after reopening, compared to less than a third during the closure.

Geographic variation in the firm's hiring tells a similar story. For headquarter-campus jobs, most new hires' teammates are local, making proximity feasible when offices are open. For other jobs — which are fully remote or in satellite campuses — teammates are almost always distant.[11] When the offices are open, the firm's headquarters campus hires are 7-10 years younger than those hired into distributed roles (Figure VIIIc). By contrast, during the office closures — when everyone was far from their teammates — this age gap narrowed substantially, as the firm hired older workers everywhere.[12] This pattern extends beyond tech to the firm's broader corporate population (Figure A.12).

While supply-side factors could contribute to these patterns, these results are consistent with proximity shifting the relative demand for younger versus older engineers. When the firm cannot ensure physical proximity — because of either the timing or location of the job — it hires older workers who have already acquired skills at other firms.

Poaching. We can also investigate other firms' hiring decisions by looking at who gets poached from our firm. We define engineers as being poached if they voluntarily leave and say that they are going to a better job.[13] During the office closures, engineers who had accumulated more human capital were more attractive to other firms: 1.2% of co-located engineers were poached each month, compared to 0.9% of multi-building engineers of similar tenure, age, and engineering group (p-value of difference = 0.044, Figure IXa). These differences compounded over time: by the end of the closure, almost a quarter of co-located engineers had been poached, versus a sixth of multi-building engineers. We see no corresponding differences in firings, layoffs, or other types of quits (Figure A.13). Consistent with the differences reflecting accumulated human capital, we estimate a significant dose response: engineers who spent more of their pre-closure time on co-located teams were more likely to be poached after the offices closed (Table A.12).

The poaching differences are concentrated among engineers who are building more general human capital. Younger engineers on co-located teams receive feedback that likely builds transferable skills and are disproportionately poached (Figure IXb). Female engineers on co-located teams similarly receive substantially more feedback and also dispro-

[11] Indeed, 98% of engineers hired outside the headquarters campus have teammates assigned to other offices.

[12] The patterns are qualitatively similar when excluding remote workers and only focusing on those hired in the main versus satellite campuses (Figure A.11).

[13] Using data from Glassdoor, we see that poached engineers do typically go to positions that pay more.

portionately land better jobs elsewhere. Tenure, by contrast, does not moderate the effect, consistent with less-tenured engineers' feedback being more firm-specific.

One reason the firm may struggle to retain these engineers is that its compensation system rewards relative performance within teams rather than absolute skill — and so may fail to price in the human capital advantage that co-location confers. Consistent with this, total compensation is similar for engineers on co-located and multi-building teams both before and after the office closures (Figure A.14).

Poaching erodes the firm's returns on human capital investments. Thus, even if the firm could perfectly observe coworkers' investments in each other's human capital, it would still under-incentivize them (Becker, 1964). Social bonds between proximate coworkers may help to fill this void and mitigate underinvestment in general human capital.

IX Who Can't Find a Job?

This section uses large-scale survey data to show suggestive evidence that our firm's hiring practices are not unique: the rise of remote work has made it relatively harder for young people to find jobs. We compare unemployment trends in remotable jobs — like software engineering or marketing — to non-remotable jobs — like mechanical engineering or nursing — using Dingel and Neiman (2020)'s occupational classification.[14] We focus on college-educated workers who are more similar to the software engineers whom we study (Appendix D shows results for non-college graduates).

Young college graduates in remotable occupations have found it more difficult to find jobs since the pandemic, while older workers in the same occupations have fared better. Between 2017−2019 and 2022−2024, young graduates (under 29) in remotable occupations experienced a 0.88 pp increase in unemployment (p-value < 0.00001), while older graduates saw a marginal decline of 0.11 pp (p-value = 0.053).[15] The resulting widening

[14] Hansen et al. (2023) show that some remotable occupations in Dingel and Neiman (2020)'s classification are rarely remote in practice, such as teachers and managers in in-person sectors. Our results are robust to dropping these potentially misclassified occupations (Figure A.16).

[15] We focus on the age threshold of 29 years old because that is what we have used throughout the paper. However, results are similar with alternative definitions. Figure A.17a shows that unemployment rates

of the age gap in unemployment coincided with the pandemic and has remained elevated alongside rates of remote work (Figure Xa).[16] In non-remotable occupations, by contrast, young graduates' relative unemployment rate spiked up briefly in 2020 before returning to baseline.

Figure Xb presents results from a triple-difference regression comparing the change in the age gap in unemployment across remotable and non-remotable occupations.[17] Young college graduates in remotable occupations experienced a 0.65 pp greater increase in unemployment than would have been expected based on trends among either older workers in remotable occupations or young workers in non-remotable ones (p-value = 0.029). This differential is robust to controlling for occupations' exposure to generative AI and its potentially heterogeneous effects across age groups, using the occupational index in Eisfeldt, Schubert and Zhang (2023), which aggregates up from Eloundou et al. (2024)'s task-level exposures (the coefficients in triangles in Figure Xb). This figure also shows that the differential increase is driven by involuntary and persistent forms of unemployment, not voluntary job leaving nor temporary layoffs.

A back-of-the-envelope calculation indicates that, between 2017−2019 and 2022−2024, remote work can explain 64% of the increase in unemployment among *all* young college graduates, many of whom enter remotable jobs. For this calculation, we scale our triple difference estimate (0.65 pp) by the share of young graduates in remotable jobs (61%): this would predict a 0.4 pp increase in all young college graduates' unemployment, which is 64% of the realized increase of 0.63 pp.

systematically increased for young workers in remotable jobs, while declining for older workers.

[16] While our partner firm shifted back to hiring younger workers after its RTOs in 2022−2023, many employers of remotable workers have continued to offer more remote work flexibility. Additionally, some RTOs do not always meaningfully increase proximity because teams are geographically distributed (Goldberg, 2021). Indeed, for such distributed positions, our partner firm hires older workers.

[17] The triple-difference coefficient is β in:

$$\begin{aligned}\text{Unemployed}_{it} = {} & \beta 1[\text{Age}_{it} < 29] \times \text{Remotable}_i \times \text{Post}_t + \delta 1[\text{Age}_{it} < 29] \times \text{Post}_t + \sigma \text{Remotable}_i \times \text{Post}_t \\ & + \psi 1[\text{Age}_{it} < 29] \times \text{Remotable}_i + \mu_a + \mu_o + \mu_t + v_{it},\end{aligned} \tag{4}$$

where μ_a denotes age fixed effects; μ_o, occupation fixed effects; and μ_t, year fixed effects. We exclude the height of the pandemic in 2020−2021 and instead compare 2022−2024 to 2017−2019.

While many have attributed the recent labor market challenges of young college graduates to generative AI (e.g., Roose, 2025), the uptick in their unemployment rate predates the rapid diffusion of AI and is concentrated more among those in remotable occupations than those whose tasks are most amenable to automation. Of course, generative AI may play a more defining role as it diffuses further, and other factors may also be contributing to young graduates' labor market struggles. As a result, the unemployment patterns should be interpreted with caution. Still, the evidence suggests that the rise of remote work meaningfully contributes to the recent challenges of young graduates.

X Conclusion

This paper shows that physical proximity between coworkers meaningfully increases mentorship and skill development — even among software engineers, the quintessential digital natives with an array of virtual communication tools. Face-to-face interaction is particularly valuable for younger and less-experienced workers, who receive more feedback from senior colleagues when co-located and ultimately write higher-quality code. Thus, an important side effect of remote work's rise may be its scarring effects on young workers, who struggle to learn from colleagues in an increasingly distributed world.

But mentorship is not free: it impedes the output of experienced workers who do the mentoring. Firms thus face a tradeoff: remote work enhances productivity today, as experienced workers focus on their own output, while sacrificing productivity tomorrow, as junior workers receive less mentorship and develop fewer skills.

Remote work stands to shift who is hired, with adverse effects for young workers. Our retailer hired fewer young workers when employees were distant from their teammates — both during the office closures and when hiring outside the headquarters campus. This mirrors hiring patterns nationally, where young workers' unemployment rates rose alongside the rise of remote work, particularly in remotable occupations.

Our results indicate that remote work may have nuanced impacts on gender equity. While it may enable working mothers to remain in the workforce (Harrington and Kahn, 2023;

Ho, Jalota and Karandikar, 2024; Jalota and Ho, 2024), it may be costly for young women, whose professional development appears to be especially sensitive to physical proximity to colleagues.

One worker cannot achieve these gains by going into the office alone — proximity depends on teammates doing the same. Moreover, even one distant teammate can degrade connections among those who are physically together, and short distances can have outsized impacts on coworker interactions. Realizing the full returns to in-person work thus requires not just individual presence, but collective, coordinated attendance.

THE FEDERAL RESERVE BANK OF NEW YORK, UNITED STATES
UNIVERSITY OF VIRGINIA, UNITED STATES
HARVARD UNIVERSITY AND THE NATIONAL BUREAU OF ECONOMIC RESEARCH, UNITED STATES

References

Agrawal, Ajay and Avi Goldfarb, "Restructuring Research: Communication Costs and the Democratization of University Innovation," *American Economic Review*, 2008, *98* (4), 1578–90.

Akan, Mert, Jose Maria Barrero, Nicholas Bloom, Tom Bowen, Shelby Buckman, Steven J Davis, Luke Pardue, and Liz Wilke, "Americans Now Live Farther From Their Employers," Daily Report, Hoover Institution February 2024.

Allen, Thomas and Gunter Henn, *The organization and architecture of innovation*, Routledge, 2007.

Angelici, Marta and Paola Profeta, "Smart working: work flexibility without constraints," *Management Science*, 2024, *70* (3), 1680–1705.

Atkin, David, M. Keith Chen, and Anton Popov, "The Returns to Face-to-Face Interactions: Knowledge Spillovers in Silicon Valley," Working Paper 30147, National Bureau of Economic Research June 2022.

Battiston, Diego, Jordi Blanes i Vidal, and Tom Kirchmaier, "Face-to-face Communication in Organizations," *The Review of Economic Studies*, 2021, *88* (2), 574–609.

Becker, Gary S, *Human capital: A Theoretical and Empirical Analysis, with Special References to Education*, The University of Chicago Press, 1964.

Belden, Martha, "Has WFH changed the way we see?," 2022. All About Vision, https://www.allaboutvision.com/conditions/computer-vision-syndrome/have-screens-changed-vision/ (visited April 10, 2025).

Bird, Steven, Ewan Klein, and Edward Loper, *Natural Language Processing With Python: Analyzing Text With the Natural Language Toolkit*, O'Reilly Media, Inc., 2009.

Bloom, Nicholas, James Liang, John Roberts, and Zhichun Jenny Ying, "Does Working from Home Work? Evidence from a Chinese Experiment," *The Quarterly Journal of Economics*, 2015, *130* (1), 165–218.

_, Ruobing Han, and James Liang, "How hybrid working from home works out," Technical Report, National Bureau of economic research 2022.

Brynjolfsson, Erik, Bharat Chandar, and Ruyu Chen, "Canaries in the Coal Mine?: Six Facts about the Recent Employment Effects of Artificial Intelligence," *Working Paper*, 2025.

Catalini, Christian, "Microgeography and the Direction of Inventive Activity," *Management Science*, 2018, *64* (9), 4348–4364.

Chen, Tianqi and Carlos Guestrin, "XGBoost: A Scalable Tree Boosting System," in "Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining" 2016, pp. 785–794.

Choudhury, Prithwiraj, Cirrus Foroughi, and Barbara Larson, "Work-from-anywhere: The Productivity Effects of Geographic Flexibility," *Strategic Management Journal*, 2021, *42* (4), 655–683.

_ , Tarun Khanna, Christos A Makridis, and Kyle Schirmann, "Is hybrid work the best of both worlds? Evidence from a field experiment," *Review of Economics and Statistics*, 2024, pp. 1–24.

Cullen, Zoe B, Bobak Pakzad-Hurson, and Ricardo Perez-Truglia, "Home Sweet Home: How Much Do Employees Value Remote Work?," Working Paper 33383, National Bureau of Economic Research January 2025.

de Groot, Jasper HB, Monique AM Smeets, Matt J Rowson, Patricia J Bulsing, Cor G Blonk, Joy E Wilkinson, and Gün R Semin, "A sniff of happiness," *Psychological science*, 2015, *26* (6), 684–700.

Dingel, Jonathan I and Brent Neiman, "How Many Jobs Can Be Done at Home?," *Journal of Public Economics*, 2020, *189*, 104235.

Dutcher, E Glenn, "The Effects of Telecommuting on Productivity: An Experimental Examination. The Role of Dull and Creative Tasks," *Journal of Economic Behavior & Organization*, 2012, *84* (1), 355–363.

Eisfeldt, Andrea L, Gregor Schubert, and Miao Ben Zhang, "Generative AI and Firm Values," Technical Report, National Bureau of Economic Research 2023.

Eloundou, Tyna, Sam Manning, Pamela Mishkin, and Daniel Rock, "GPTs are GPTs: Labor market impact potential of LLMs," *Science*, 2024, *384* (6702), 1306–1308.

Emanuel, Natalia and Emma Harrington, "Working Remotely? Selection, Treatment, and the Market for Remote Work," *American Economic Journal: Applied Economics*, 2024, *16* (4), 528–559.

Fenizia, Alessandra and Thomas Kirchmaier, "Not Incentivized Yet Efficient: Working From Home in the Public Sector," Discussion Paper 2036, Centre for Economic Performance, London School of Economics and Political Science 2024.

Flood, Sarah, Miriam King, Renae Rodgers, Steven Ruggles, Warren Robert J., Daniel Backman, Annie Chen, Grace Cooper, Stephanie Richards, Megan Schouweiller, and Michael Westberry, "IPUMS CPS: Version 12.0 [dataset]," 2024. Data retrieved from IPUMS CPS, `https://cps.ipums.org/cps/` (visited March 20, 2024).

Gaspar, Jess and Edward L Glaeser, "Information Technology and the Future of Cities," *Journal of Urban Economics*, 1998, *43* (1), 136–156.

Gibbs, Michael, Friederike Mengel, and Christoph Siemroth, "Work from Home and Productivity: Evidence from Personnel and Analytics Data on Information Technology Professionals," *Journal of Political Economy Microeconomics*, 2023, *1* (1), 7–41.

Goldberg, Emma, "The Worst of Both Worlds: Zooming from the Office," November 2021.

Goodman-Bacon, Andrew, "Difference-in-Differences with Variation in Treatment Timing," *Journal of Econometrics*, 2021, *225* (2), 254–277.

Hansen, Stephen, Peter John Lambert, Nicholas Bloom, Steven J Davis, Raffaella Sadun, and Bledi Taska, "Remote Work across Jobs, Companies, and Space," Technical Report, National Bureau of Economic Research 2023.

Harrington, Emma and Matthew E Kahn, "Has the Rise of Work-From-Home Reduced the Motherhood Penalty in the Labor Market," Working Paper 2023.

He, Haoran, David Neumark, and Qian Weng, "Do Workers Value Flexible Jobs? A Field Experiment," *Journal of Labor Economics*, 2021, *39* (3), 709–738.

Herkenhoff, Kyle, Jeremy Lise, Guido Menzio, and Gordon M Phillips, "Production and Learning in Teams," *Econometrica*, 2024, *92* (2), 467–504.

Ho, Lisa, Suhani Jalota, and Anahita Karandikar, "Bringing Work Home: Flexible Arrangements as Gateway Jobs for Women in West Bengal," Working Paper, Structural Transformation and Economic Growth 2024.

Jalota, Suhani and Lisa Ho, "What Works for Her? How Work-from-Home Jobs Affect Female Labor Force Participation in Urban India," Working Paper, SSRN 2024.

Kraut, Robert, Carmen Egido, and Jolene Galegher, "Patterns of Contact and Communication in Scientific Research Collaboration," in "Proceedings of the 1988 ACM conference on Computer-supported cooperative work" 1988, pp. 1–12.

Lewandowski, Piotr, Katarzyna Lipowska, and Mateusz Smoter, "Mismatch in Preferences for Working From Home: Evidence From Discrete Choice Experiments With Workers and Employers," Discussion Paper 16041, IZA March 2023.

Lichtinger, Guy and Seyed Mahdi Hosseini Maasoum, "Generative AI as seniority-biased technological change: Evidence from US resume and job posting data," *Available at SSRN*, 2025.

Maestas, Nicole, Kathleen J Mullen, David Powell, Till Von Wachter, and Jeffrey B Wenger, "The value of working conditions in the United States and implications for the structure of wages," *American Economic Review*, 2023, *113* (7), 2007–2047.

Mas, Alexandre and Amanda Pallais, "Valuing Alternative Work Arrangements," *American Economic Review*, 2017, *107* (12), 3722–59.

_ **and** _ , "Alternative Work Arrangements," *Annual Review of Economics*, 2020, *12* (1), 631–658.

Miranda, Arianna Salazar and Matthew Claudel, "Spatial Proximity Matters: A Study on Collaboration," *PLOS One*, 2021, *16* (12), e0259965.

Roose, Kevin, "For Some Recent Graduates, the A.I. Job Apocalypse May Already Be Here," May 2025.

Ruggles, Steven, Sarah Flood, Matthew Sobek, Daniel Backman, Annie Chen, Grace Cooper, Stephanie Richards, Renae Rogers, and Megan Schouweiler, "IPUMS USA: Version 14.0 [dataset]," 2022. data retrieved from IPUMS USA, https://usa.ipums.org/usa/ (visited January 19, 2024).

Schubert, Gregor, "Organizational Technology Ladders: Remote Work and Generative AI Adoption," Working Paper, SSRN April 2025.

Schwartz, Linoy, Jonathan Levy, Yaara Endevelt-Shapira, Amir Djalovski, Olga Hayut, Guillaume Dumas, and Ruth Feldman, "Technologically-assisted communication attenuates inter-brain synchrony," *Neuroimage*, 2022, *264*, 119677.

Seltzer, Leslie J, Ashley R Prososki, Toni E Ziegler, and Seth D Pollak, "Instant messages vs. speech: hormones and why we still need to hear each other," *Evolution and Human Behavior*, 2012, *33* (1), 42–45.

Stack Overflow, "Stack Overflow Developer Survey 2017-2023," 2023. Data retrieved from https://survey.stackoverflow.co/ (visited June 3, 2024).

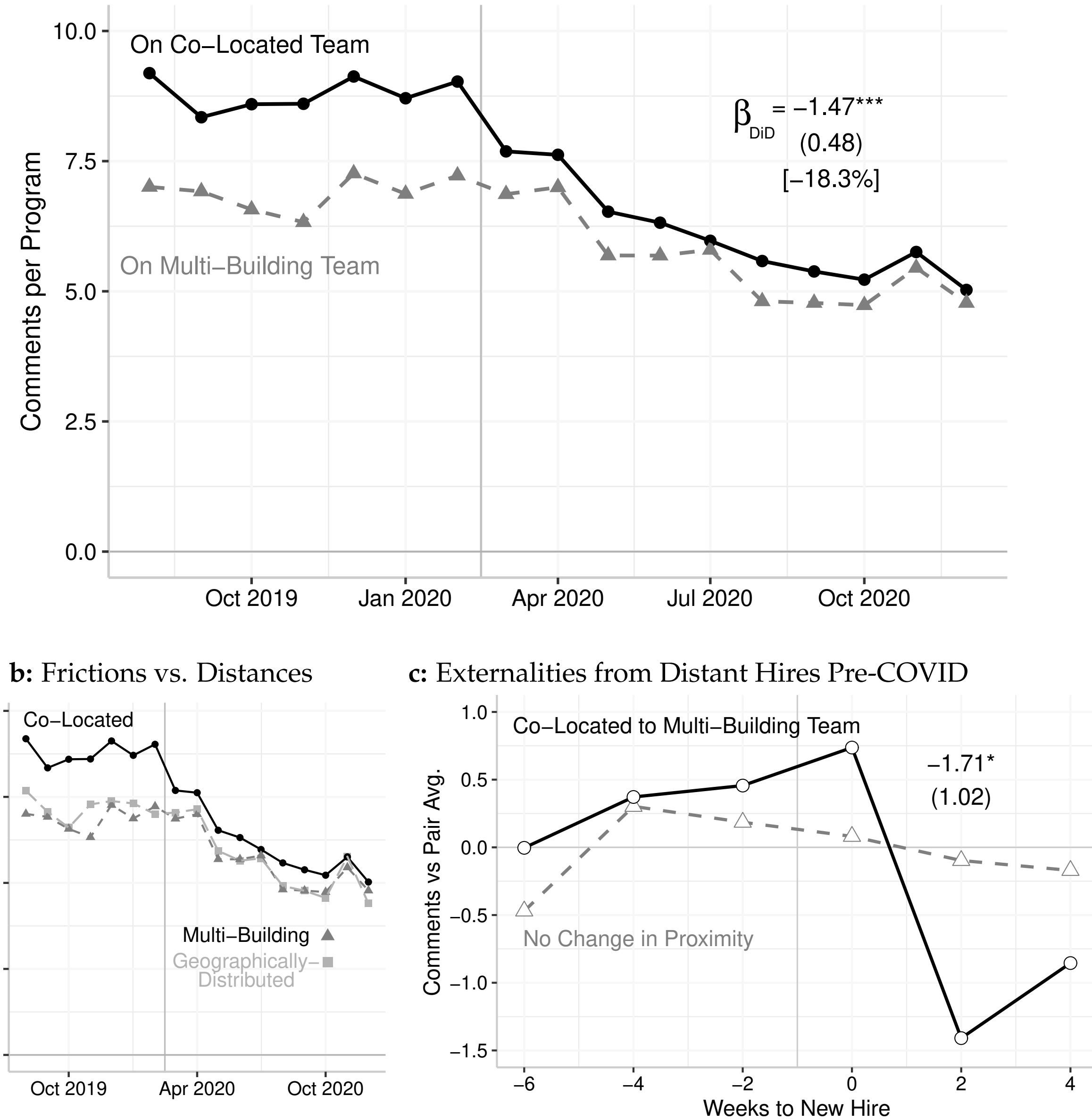
U.S. Census Bureau, "Household Pulse Survey," 2023. Data retrieved from Cenus, https://www.census.gov/programs-surveys/household-pulse-survey/datasets.html (visited January 19, 2023).

Wang, Baolian, Peixin Li, and Jiawei Yu, "The Impact of Return-to-Office Mandates on Equity Analysts," *Available at SSRN 5206058*, 2025.

Yang, Longqi, David Holtz, Sonia Jaffe, Siddharth Suri, Shilpi Sinha, Jeffrey Weston, Connor Joyce, Neha Shah, Kevin Sherman, Brent Hecht et al., "The Effects of Remote Work on Collaboration Among Information Workers," *Nature: Human Behaviour*, 2022, *6* (1), 43–54.

Figure I: Proximity to Teammates and Online Feedback

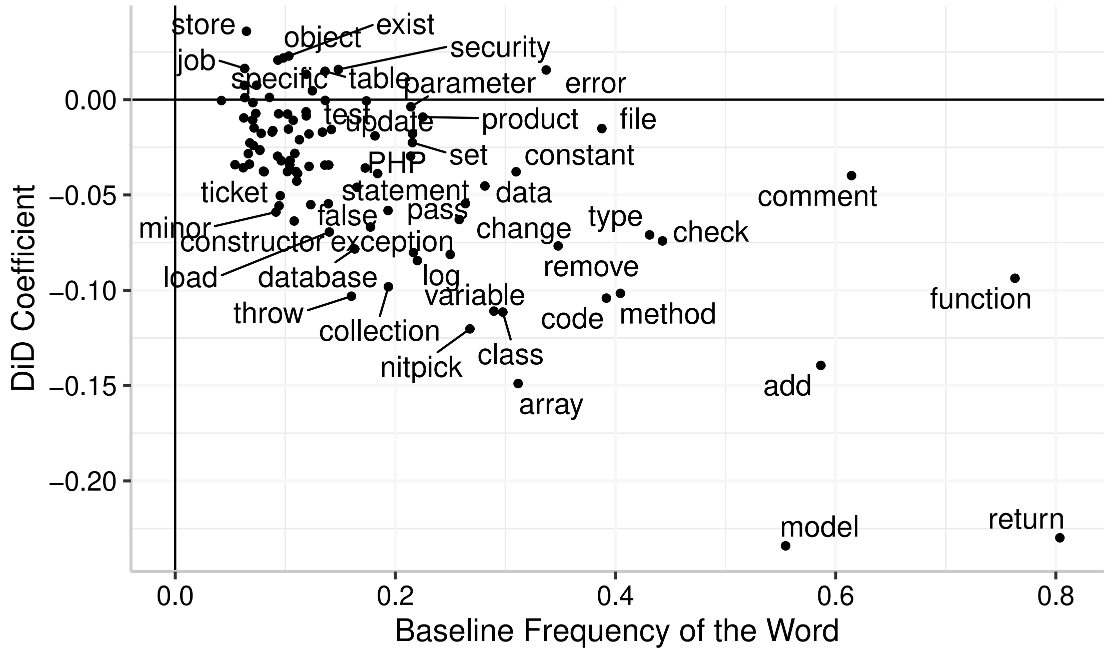
a: Difference-in-Differences around the Office Closures



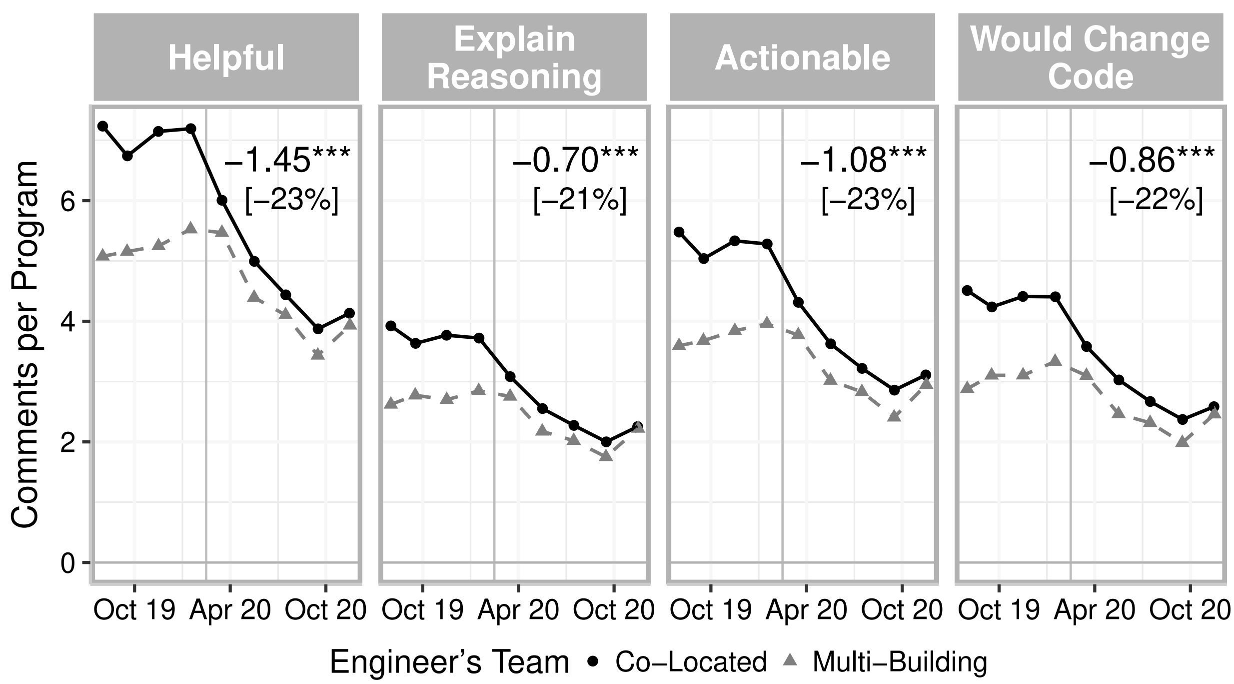
Notes: This figure analyzes whether engineers who are physically proximate to their teammates receive more feedback online. Panel a illustrates our difference-in-differences (DiD) design that compares engineers on co-located teams (N=637) to engineers on multi-building teams (N=418) around the COVID-19 office closures (the gray line). The plot residualizes by our preferred controls for program scope, engineer age, tenure, and engineering group. The reported coefficient is the pooled DiD coefficient with engineer fixed effects (Table II, Column 6), with the percent change in brackets. Panel b extends the analysis to include teams geographically distributed across campuses. Panel c shows a complementary, pre-COVID design, which compares new hires that convert teams from co-located to multi-building teams (N=16 teams) versus new hires that do not change the co-location of their team (N=119 teams). This analysis is limited to comments between pairs of co-located teammates who both pre-date the 6-week pre-period. The plot shows comments per program relative to the average in the coder-commenter pair, with the annotated DiD coefficient from Equation 3. Standard errors are clustered by team. $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$.

Figure II: Effects of Losing Proximity on Substantive Feedback

a: Common Words in Feedback



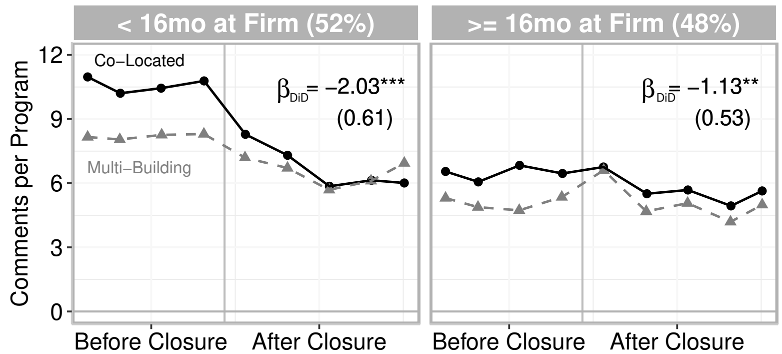
b: Predicted Comment Ratings



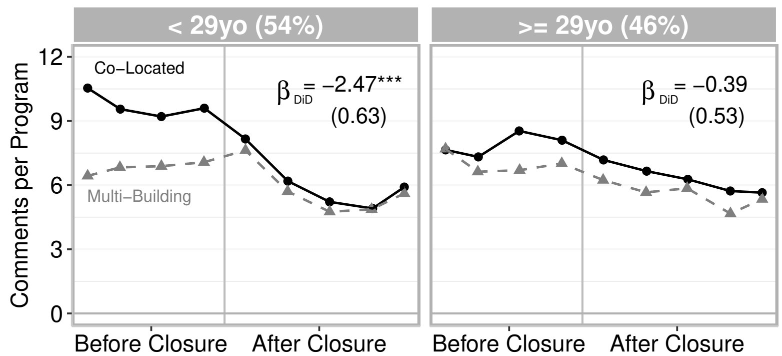
Notes: This figure examines feedback content. Panel a illustrates how losing proximity to teammates affects individual word usage, with each word's pre-closure frequency on the x-axis and the difference-in-differences (DiD) coefficient on the y-axis. Panel b shows the impact of losing proximity on predicted comment quality. Predictions come from a supervised machine learning algorithm applied to the human ratings of external software engineers (see Appendix C). The annotated coefficient reflects the DiD estimate, with percentage effects in brackets. All DiD specifications include engineer fixed effects and our preferred month-specific controls for engineer age, tenure, and engineering group. Standard errors are clustered by team. *p<0.1; **p<0.05; ***p<0.01.

Figure III: Heterogeneous Effects of Proximity

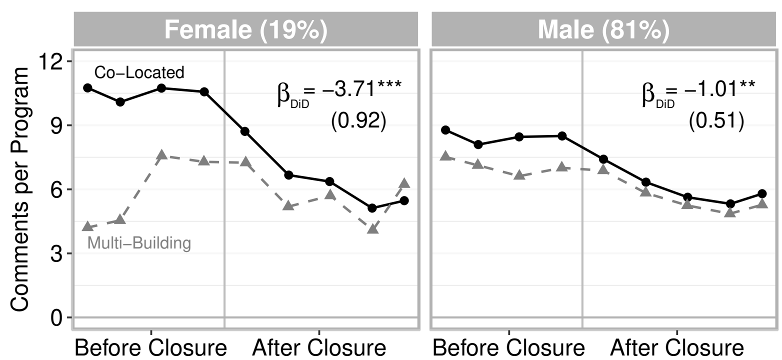
a: By Tenure



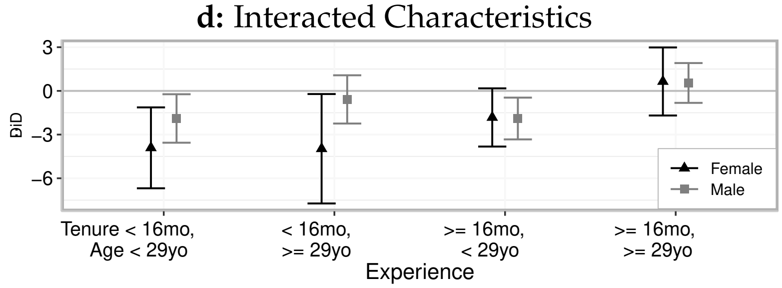
b: By Age



c: By Gender



d: Interacted Characteristics



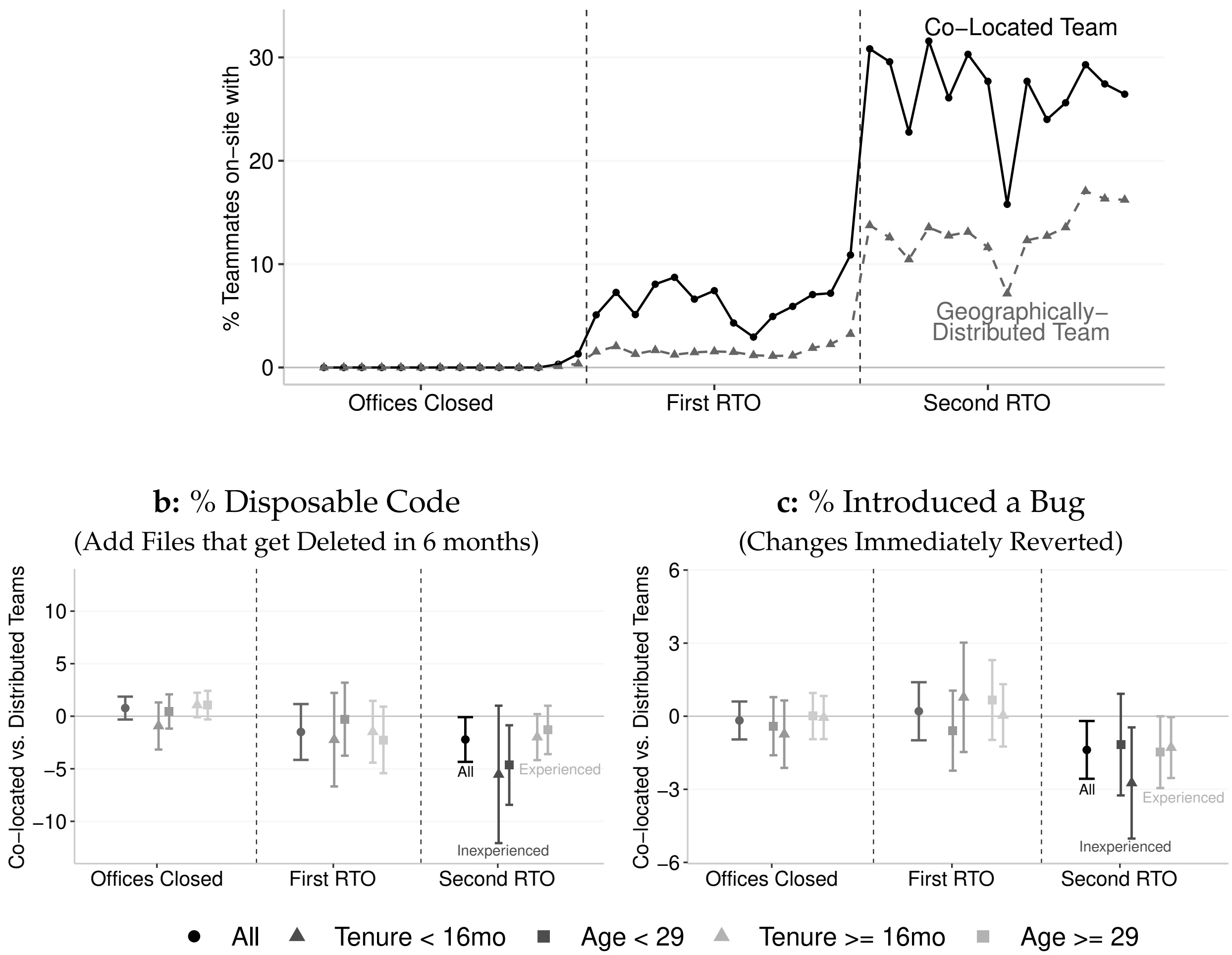
Notes: This figure replicates Figure Ia by (a) tenure (relative to the mean of 16 months), (b) age (relative to the mean of 29), and (c) gender. Tenure and age are measured at the beginning of each month. Panel d shows the DiD coefficients from a regression with all the interactions. All specifications use our preferred controls and cluster by engineering team. *p<0.1; **p<0.05; ***p<0.01.

Figure IV: Return to Office (RTO) and Code Quality

a: Differential Changes in Proximity

b: % Disposable Code (Add Files that get Deleted in 6 months)

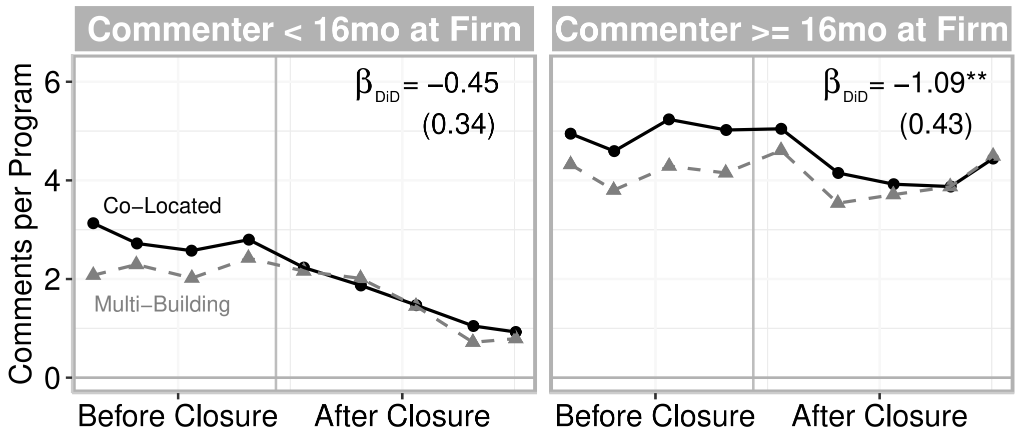
c: % Introduced a Bug (Changes Immediately Reverted)



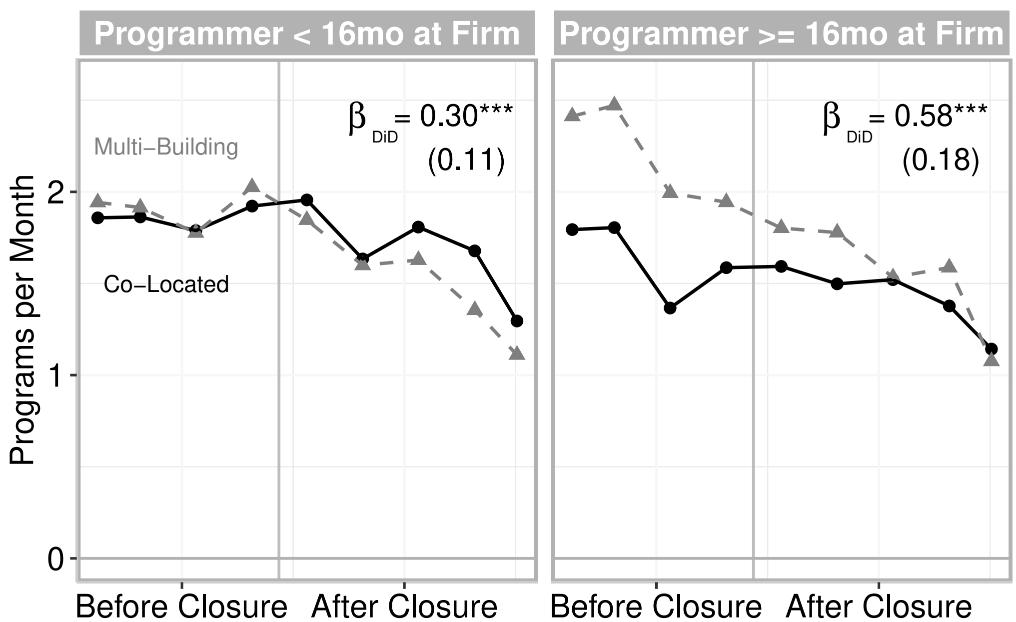
Notes: This figure illustrates changes around the firm's return-to-office (RTO) phases, comparing engineers in the headquarters (HQ) who are on teams that are all HQ-based versus those that are distributed. Panel a plots face-to-face time with teammates: for each weekday, the measure is zero if the worker is out of the office and otherwise is the share of her teammates also in the office. Panels b-c show code quality differences between engineers on HQ-based and geographically-distributed teams with controls for engineer age, tenure, engineering group, and engineer fixed effects. Experience is measured both by tenure at the firm (using a 16-month cutoff) and age (using a 29 year-old cutoff), measured for each engineer at the beginning of the month. Error bars represent 95% confidence intervals, with standard errors clustered by team.

Figure V: Tradeoffs of Proximity

a: Comments Given by Tenure

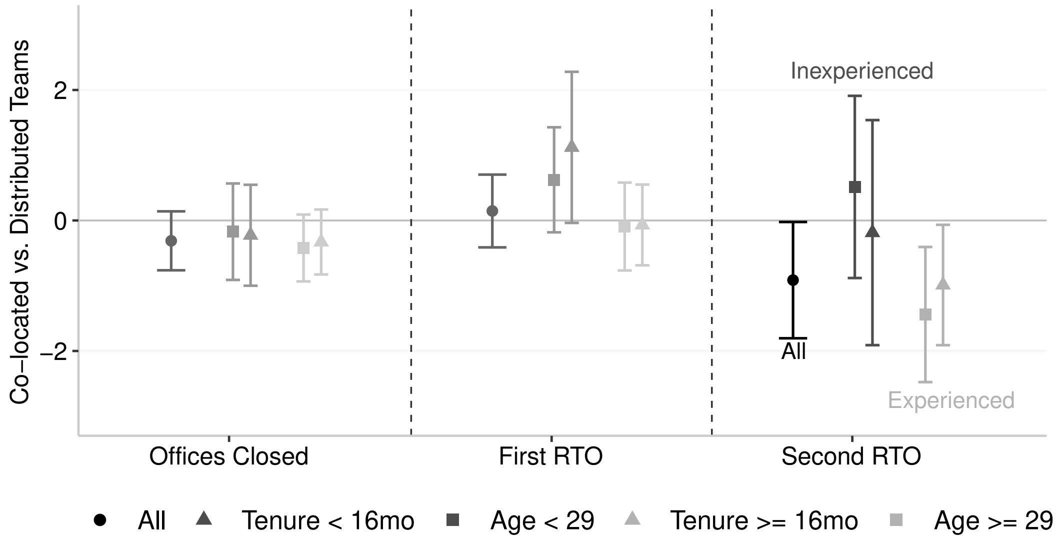


b: Programs Written by Tenure



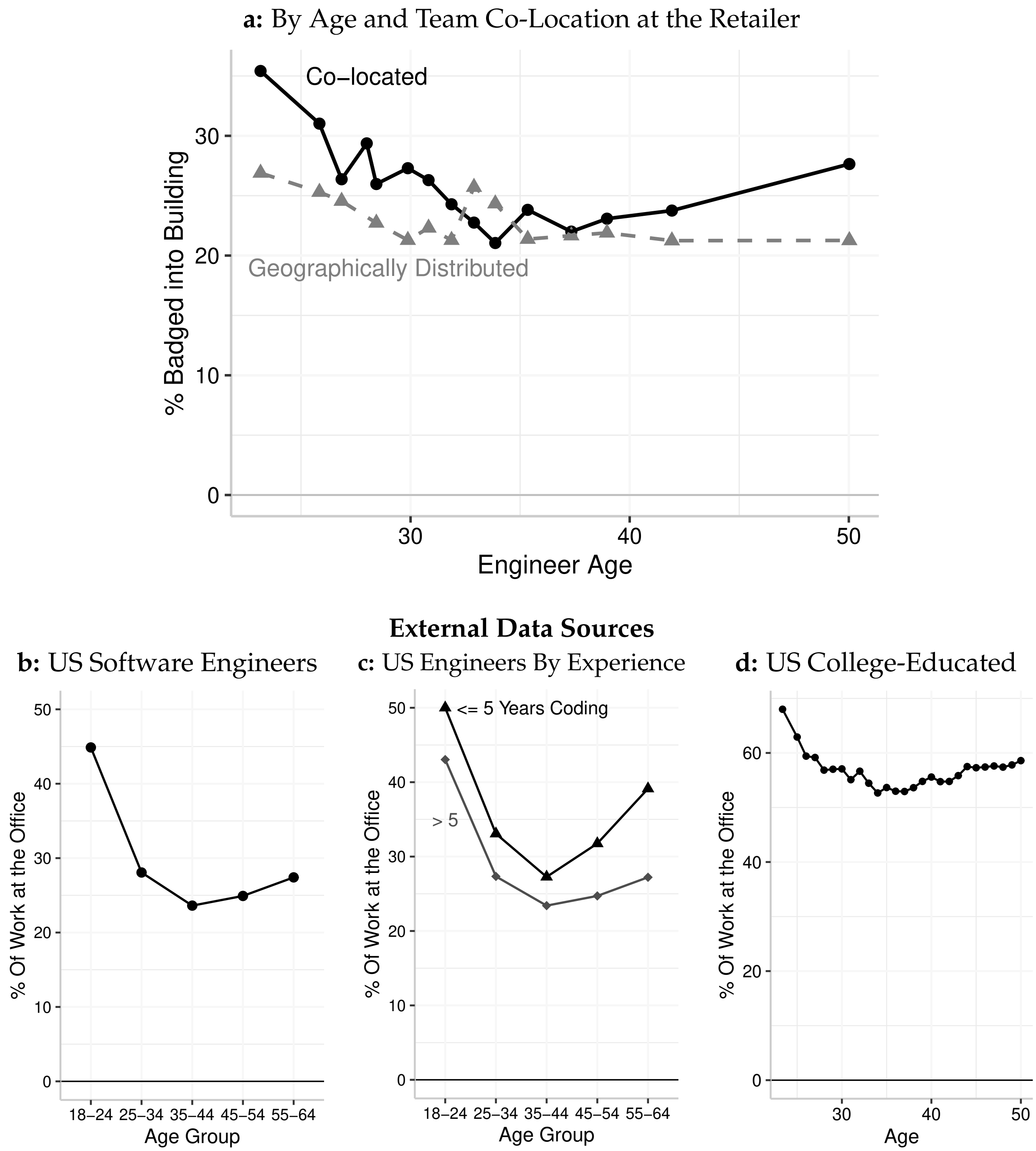
Notes: This figure illustrates the tradeoffs of proximity for engineers of different tenures. Panel a shows comments per program broken down by the seniority of the commenter rather than the program writer. Panel b shows programs written per month. Both plots residualize by our preferred controls for engineer age, tenure, and engineering group, as well as program scope (quartics in the number of lines added, lines deleted, and files changed) in Panel a. In each plot, tenure is measured at the beginning of each month. The annotated coefficients report the pooled DiD estimate (Equation 1) for the respective subsample, with standard errors clustered by team. $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$.

Figure VI: Return to Office (RTO) and Code Quantity



Notes: This figure illustrates changes around the firm's return-to-office (RTO) phases, comparing engineers in the headquarters (HQ) who were on teams that are all HQ-based versus those that are distributed. The shows differences in programs written per month between engineers on HQ-based and geographically-distributed teams with controls for engineer age, tenure, engineering group, and engineer fixed effects. Experience is measured both by tenure at the firm (using a 16-month cutoff) and age (using a 29 year-old cutoff), measured at the beginning of the month for each engineer. Error bars represent 95% confidence intervals, with clustering by team.

Figure VII: Who Returns to the Office

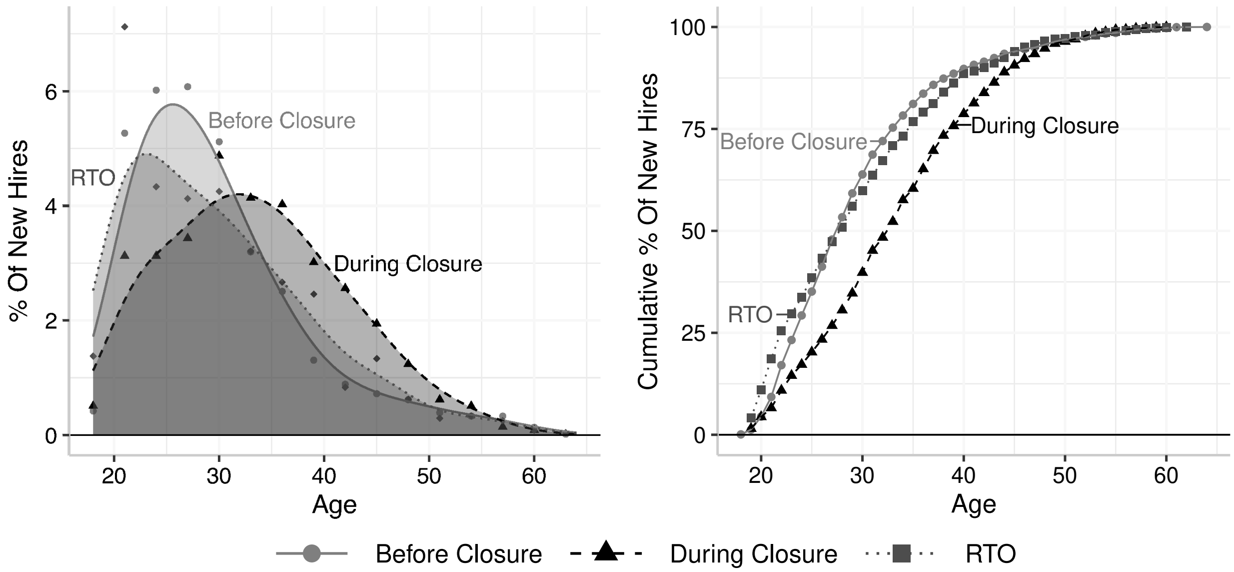


Notes: This figure illustrates office attendance patterns by age. Panel a plots office attendance by engineer age at the firm. Data comes from badge swipes in the headquarter's (HQ's) security system, and the sample focuses on engineers based in the HQ during both RTOs from spring of 2022 through spring of 2024. The y-axis represents the percent of weekdays people badge in. The x-axis plots age, split into fifteen quantiles. The black series includes engineers whose teammates were all HQ-based; the gray series includes those with some teammates who were not based in the HQ. Panels b and c use data from a 2022 and 2023 survey conducted by Stack Overflow, the top online Q&A site for software engineers (N = 22,233) (Stack Overflow, 2023). Panel d plots data from the Census's Household Pulse Surveys from 2022 and 2023 (U.S. Census Bureau, 2023), limiting to college-educated workers. The y-axis plots the share of reported in-office time.

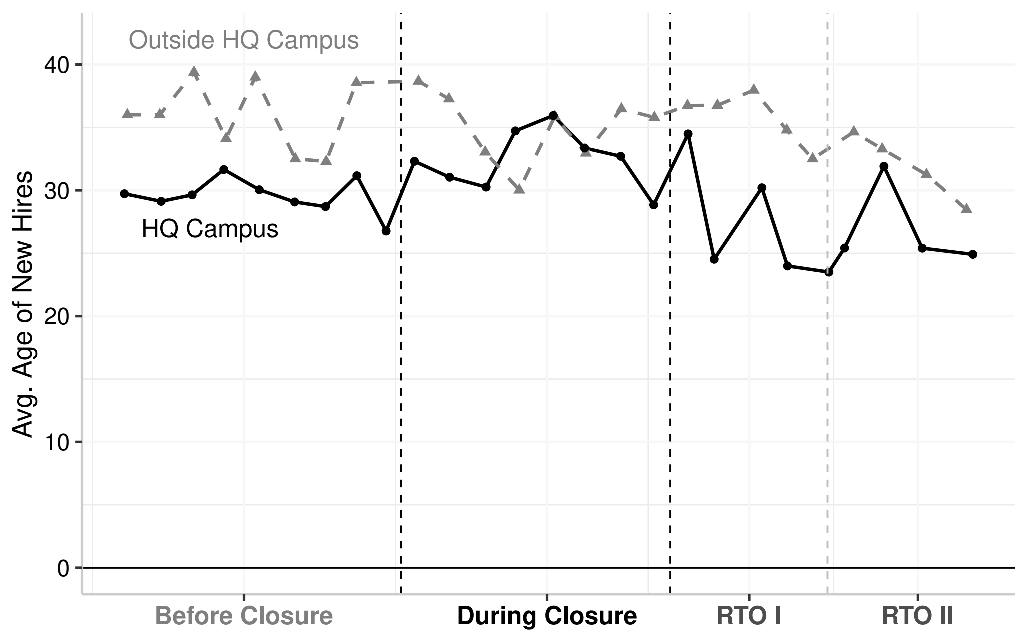
Figure VIII: Younger Hires when Proximity is Feasible

a: Age Distribution over Time

b: Age CDFs



c: Heterogeneity by Hire Location



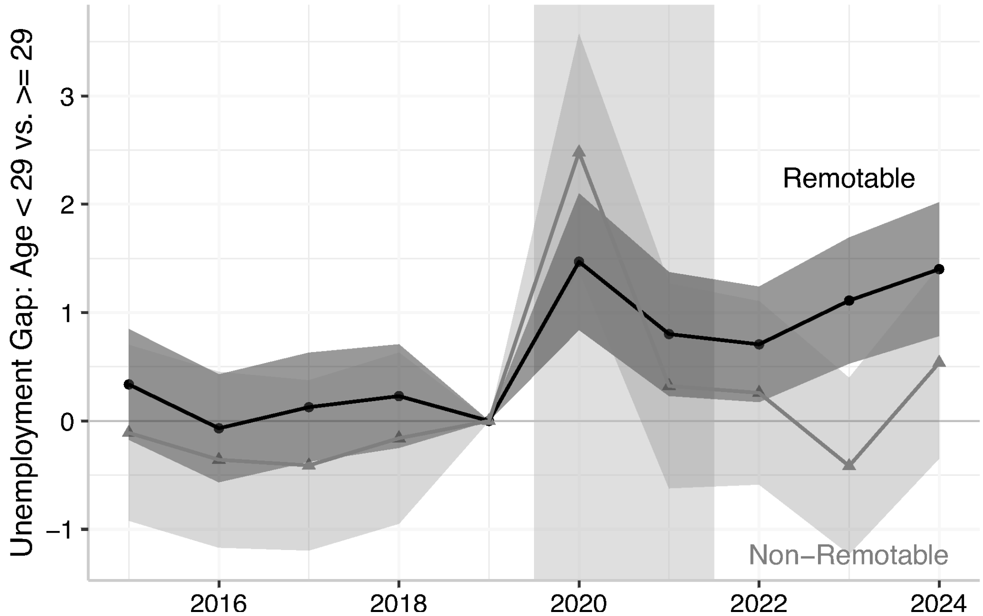
Notes: This figure illustrates the shifting age distribution of new hires among the tech workers at the firm. Panel a shows the distribution of new hires, the kernel densities are Gaussian densities with bandwidths of three years and the points show average densities in three-year age-groups. Panel b shows the cumulative distribution functions. Both plots differentiate between the pre-closure period (in light gray), the closure period (in black), and the reopening and return-to-office (RTO) period (in dark gray). Panel c focuses on heterogeneous trends by the location of the new hire. The solid black line shows the pattern for those hired in the headquarters (HQ) campus, who would likely sit with their teammates when the offices were open. The dashed gray line shows the pattern for those hired outside the HQ campus, who would likely be distant from at least some teammates regardless of whether the offices were open (98% of these engineers have teammates assigned to other offices).

Figure IX: Poaching from the Firm during the Office Closures

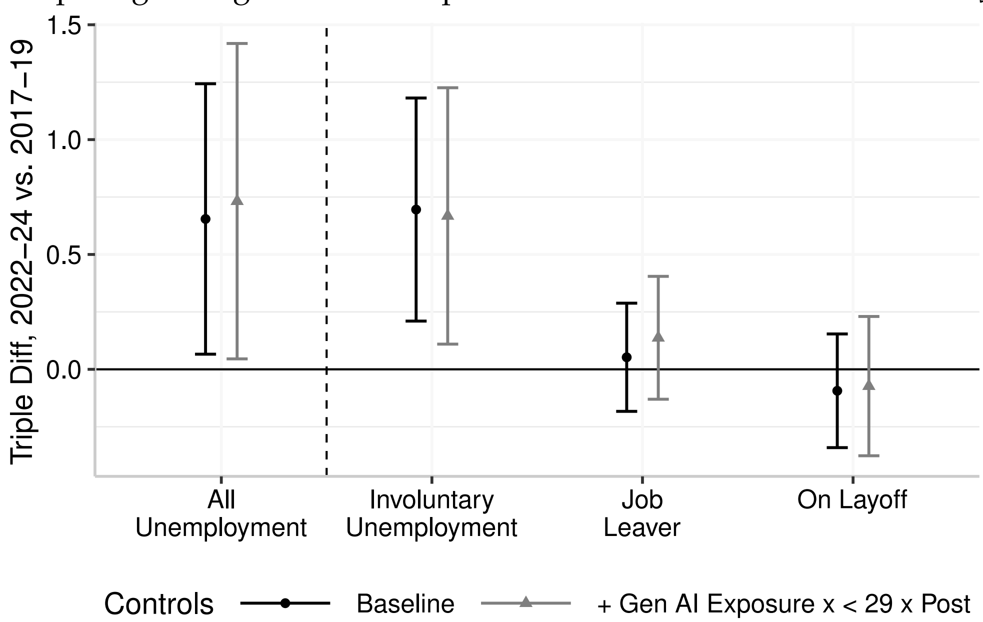
Notes: This figure investigates whether engineers who were on co-located teams and so received more human capital investments were more likely to be poached by other firms. The analysis focuses on when offices were closed, proximity was impossible, and so other firms may have been more inclined to hire those with more accumulated human capital. Panel a shows the differences between engineers on co-located versus multi-building teams. Engineers on co-located teams were consistently co-located in the pre-closure period. Panel b investigates the differences between co-located and multi-building teams for different subgroups of engineers — of different age, gender, and tenure. The annotation indicates the significance of the difference across the two groups (e.g., between younger and older engineers). All specifications include our preferred time-varying controls for engineer group, age, and tenure, as well as gender (where applicable). Standard errors are clustered by team. $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$

Figure X: The Rise of Remote Work and Young People's Unemployment

a: Changes in Youth Gap in Unemployment in Remotable and Non-Remotable Jobs



b: Comparing Change in Youth Gap in Remotable vs. in Non-Remotable Jobs



Notes: This figure analyzes changes in young people's relative unemployment rates around the rise of remote work. Panel a shows dynamic changes in the youth gap in unemployment (between those <29 and ≥ 29) relative to 2019, with occupation and age fixed effects. These changes are shown separately for those in remotable jobs (in black circles) and non-remotable jobs (in gray triangles). Figure A.17b shows a continuous version of this analysis. Panel b shows triple difference coefficients comparing the change in the youth gap in unemployment in remotable versus non-remotable jobs, excluding the peak pandemic years of 2020−2021 (Equation 4). Baseline controls include age, occupation, and year fixed effects; additional controls fully interact occupational exposure to generative AI (Eisfeldt, Schubert and Zhang, 2023) with indicators for being <29 and the post period (2022−2024). Data is from the Current Population Survey (Flood et al., 2024), limited to college graduates between ages 22 and 64. Standard errors are clustered by survey respondent.

Table I: Summary Statistics: Co-Located and Multi-Building Teams

	Mean	P10	P50	P90	Mean Co- Located	Mean Multi- Building	Difference	
Age (Years)	28.76	23.00	28.00	35.00	28.51	29.09	-0.58	0.36
							(0.42)	(0.56)
Firm Tenure (Years)	1.36	0.17	0.92	3.50	1.21	1.56	-0.34***	-0.42***
							(0.11)	(0.12)
% Female	18.58	0.00	0.00	100.00	19.53	17.33	2.19	-2.75
							(2.78)	(3.26)
Job Traits								
Job Level	1.71	1.00	2.00	3.00	1.62	1.82	-0.20***	-0.06
							(0.06)	(0.07)
Salary + Stocks	121,262	100,000	115,000	150,050	119,075	124,161	-5,086***	-894
							(1,810)	(2,210)
Team Traits								
# Teammates	6.09	3.00	6.00	10.00	5.72	6.57	-0.85**	-0.42
							(0.42)	(0.47)
Manager Tenure	2.87	0.78	2.33	5.90	2.84	2.92	-0.08	-0.41
							(0.32)	(0.36)
Manager Job Level	3.30	2.00	3.00	4.00	3.21	3.42	-0.21**	-0.08
							(0.09)	(0.10)
Engineer Group								
Back-End	12.73	0.00	0.00	100.00	21.23	1.48	19.75***	–
							(3.46)	
Front-End	19.83	0.00	0.00	100.00	30.47	5.76	24.71***	–
							(4.61)	
Internal Tools	60.00	0.00	100.00	100.00	37.23	90.13	-52.90***	–
							(5.26)	
AI Features	7.44	0.00	0.00	0.00	11.07	2.63	8.44***	–
							(3.13)	
Engineer Group Controls								✓
# Software Engineers	1,055	1,055	1,055	1,055	637	418		
# Teams	304	304	304	304	206	121		

Notes: This table shows traits of the engineers, their job, and their team before the offices closed for COVID-19. The sample includes engineers whose teams are all in the headquarters campus. P10 refers to the 10th percentile, P50 to the median, and P90 to the ninetieth percentile. The last two columns present the pre-closure difference between engineers on co-located and multi-building teams. The last column (Column 8) includes engineering group controls: indicators for whether the engineer works on front-end website design, back-end catalog management, internal tools for others in the company, or AI features. "Job level" refers to the engineer's position within the firm's hierarchy from zero (an intern) to six (senior staff). Standard errors in parentheses are clustered by engineering team. *p<0.1; **p<0.05; ***p<0.01.

Table II: Proximity to Teammates and Online Feedback

	Comments per Program						
						Teammate	Non-Teammate
	(1)	(2)	(3)	(4)	(5)	(6)	(7)
Post x Co-Located	-1.29^{***}	-1.35^{***}	-1.47^{***}	-1.73^{***}	-1.25^{**}	-1.33^{***}	0.12
	(0.48)	(0.49)	(0.48)	(0.49)	(0.49)	(0.38)	(0.31)
Co-Located Team	1.16^{**}	1.92^{***}					
	(0.52)	(0.55)					
Post	-1.22^{***}						
	(0.36)						
Pre-Mean, Co-Located	8.04	8.04	8.04	8.04	8.04	4.28	3.73
Percentage Effects							
Post x Co-Located	-16.1%	-16.8%	-18.3%	-21.5%	-15.6%	-31%	3.3%
Co-Located	14.5%	23.9%					
Group x Month FE		✓	✓	✓	✓	✓	✓
Program Scope Quartics		✓	✓	✓	✓	✓	✓
Tenure x Month FE		✓	✓	✓	✓	✓	✓
Age x Month FE		✓	✓	✓	✓	✓	✓
Engineer FE			✓	✓	✓	✓	✓
Other Traits x Month FE				✓	✓	✓	✓
Building x Month FE				✓	✓	✓	✓
% Co-Located Team	58.3	58.3	58.3	58.3	58.3	58.3	58.3
# Teams	304	304	304	304	304	304	304
# Engineers	1,055	1,055	1,055	1,055	1,055	1,055	1,055
# Engineer-Months	9,304	9,304	9,304	9,304	9,304	9,304	9,304
R^2	0.02	0.43	0.56	0.61	0.61	0.54	0.63

Notes: This table examines the relationship between physical proximity to teammates and the online feedback that engineers receive on their code. Each column estimates the difference-in-differences model in Equation 1, which is based on the differential loss of proximity among co-located versus multi-building team engineers around the office closures. The last two columns presents a placebo check: proximity to teammates should impact feedback from teammates but not from non-teammates. Each observation is an engineer-month, with the dependent variable being the average number of comments per program received. Program scope controls include quartics for the number of lines added, lines deleted, and files changed. Engineer traits include gender, being a person of color, home zipcode, and job level. Building-by-month fixed effects allow for differential changes in feedback for programmers who sat in the main and auxiliary buildings. The sample includes engineers who submit programs to the firm's main code-base and whose teams are all in the firm's headquarters campus. Standard errors are clustered by team. $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$.

Table III: Proximity and Other Dimensions of Online Feedback

a: Extensiveness and Timeliness of Online Conversations

	Total Characters	Total Words	Hours to Comment	% Other Online Convo
	(1)	(2)	(3)	(4)
Post x Co-Located Team	-164.10^{**}	-26.87^{**}	1.12^{*}	-1.58^{*}
	(67.78)	(11.09)	(0.64)	(0.95)
Pre-Mean, Co-Located Team	833.24	136.92	16.02	4.06
Post x Co-Located Team as %	-19.7%	-19.6%	7%	-38.9%

b: Intensive and Extensive Margins

	Intensive	Extensive	
	Comments per Commenter	Commenters per Program	Distinct Commenters per Program
	(1)	(2)	(3)
Post x Co-Located Team	-0.63^{***}	-0.05	-0.12^{**}
	(0.23)	(0.06)	(0.06)
Pre-Mean, Co-Located	4.36	1.77	1.37
Post x Co-Located as %	-14.3%	-2.6%	-9%

c: Back-and-Forth Conversations

		Commenter's	Program Writer's		Commenter's
	Back and Forths	Initial Comments	Replies	Questions	Follow-up Comments
Post x Co-Located Team	-0.39^{***}	-0.71^{**}	-0.72^{**}	-0.15^{***}	-0.76^{**}
	(0.13)	(0.28)	(0.29)	(0.06)	(0.37)
Pre-Mean, Co-Located	1.95	4.91	2.14	0.24	3.13
Post x Co-Located as %	-19.9%	-14.4%	-33.7%	-62.7%	-24.3%

Notes: This table considers alternative metrics of online feedback: (a) the extent and timeliness of feedback, (b) the extensive and intensive margins of feedback, and (c) the back-and-forth conversation between the commenter and program writer. In panel a, the frequency of other online conversations is measured as the percent of reviews with references to Slack discussions. In panel b, if a programmer wrote five programs in a month and had the same person comment on all of them, she would have one commenter per program but only 0.2 distinct commenters per program. Distinct commenters per program aims to approximate the size of the engineer's network. In panel c, the total number of back and forths in Column 1 measures each time there is a author switch in the comments: for example, a commenter giving a set of comments, an author asking a set of questions, and then a commenter clarifying their initial feedback would count as three. Each specification replicates Column 3 of Table II. $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$.

Online Supplementary Materials for

The Power of Proximity to Coworkers

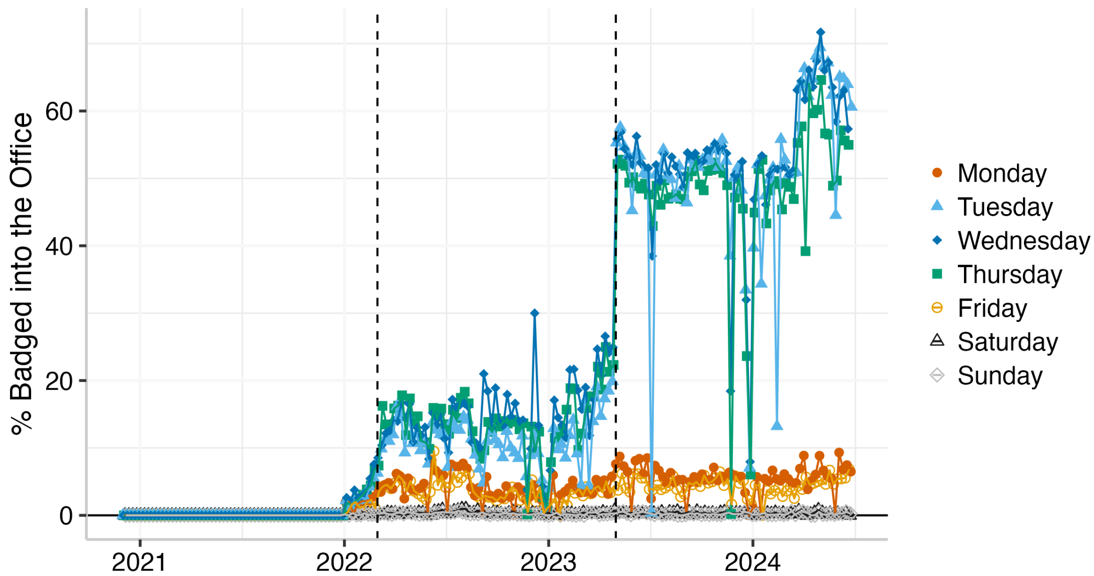
Natalia Emanuel · Emma Harrington · Amanda Pallais

Table of Contents

B Tables

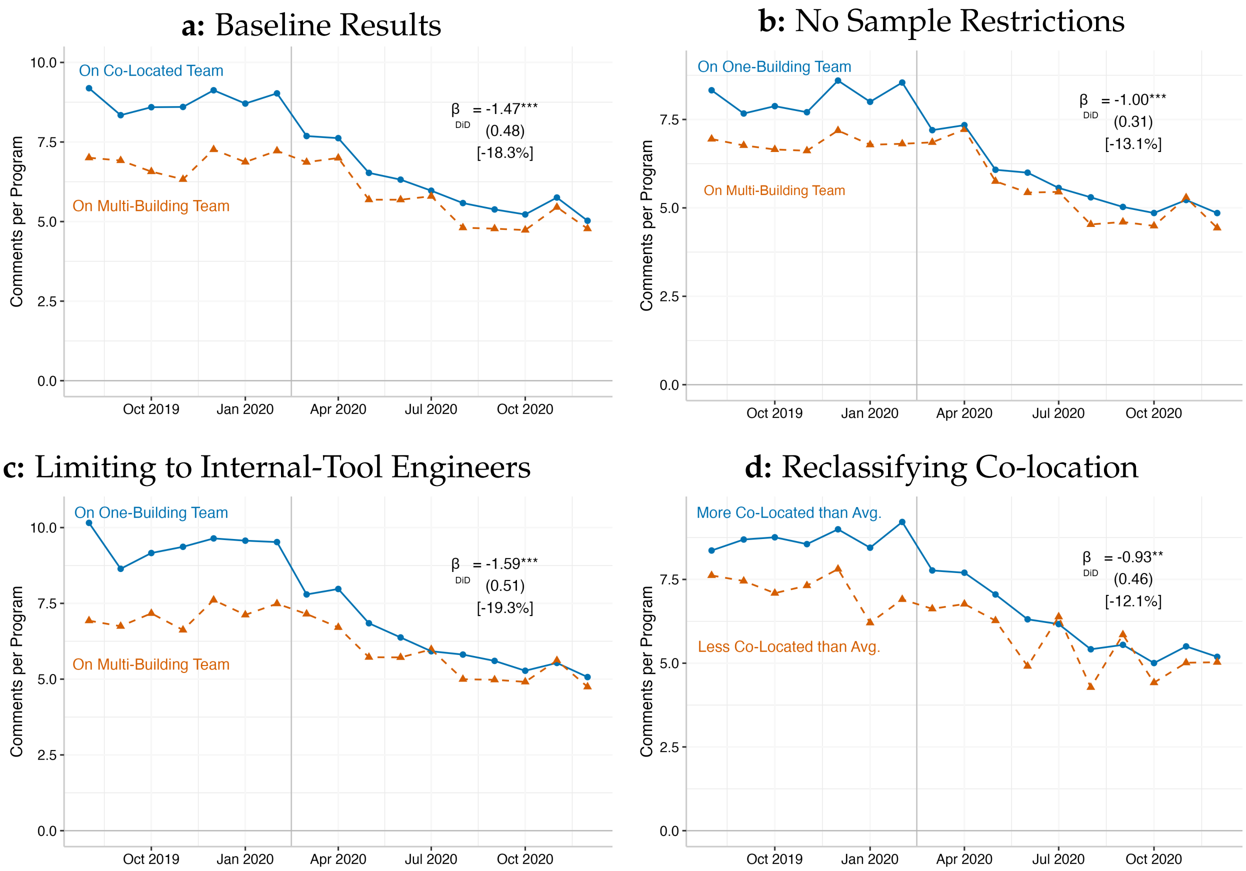
A Figures

Figure A.1: Badge Data of Engineers by Day



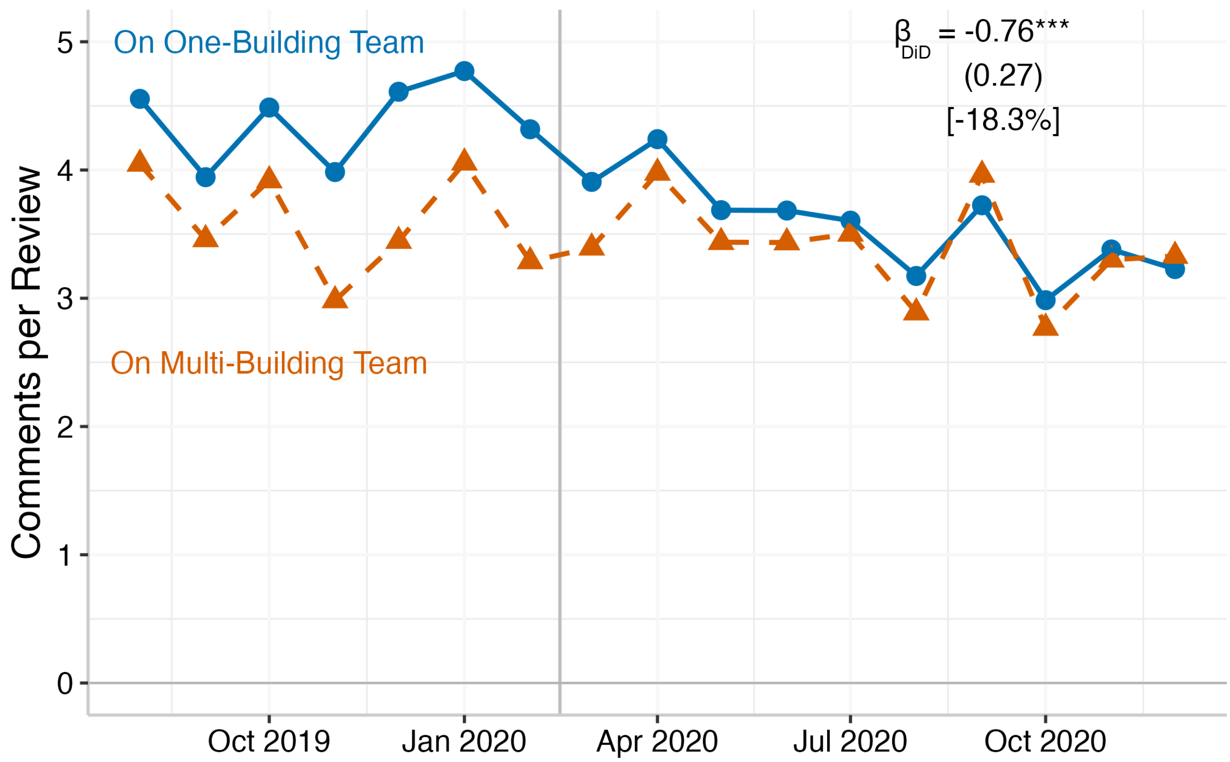
Notes: This figure illustrates the share of engineers who badged into the office each day for those working in the headquarters campus. The vertical lines highlight the firm's two return-to-office mandates: the first required two days per week, while the second required three.

Figure A.2: Robustness of Proximity to Teammates and Online Feedback



Notes: This figure probes the robustness of Figure Ia to alternative ways of implementing the design. Panel a repeats the baseline design for reference. Panel b does not apply any sample restrictions: this panel includes engineers who are outside the headquarters campus and engineers under high-level managers, who may manage multiple teams. Panel c limits to internal-tool engineers who are more likely to be on multi-building teams to sit near those who use their tools. Panel d differentiates teams that are more versus less co-located than average (87%) rather than co-located or not. Each plot residualizes by our preferred month-specific controls for engineer tenure, age, and engineering group, as well as quartics for program length. Standard errors are clustered by team. *p<0.1; **p<0.05; ***p<0.01.

Figure A.3: Externalities from Distant Teammates on Interactions between Co-located Teammates

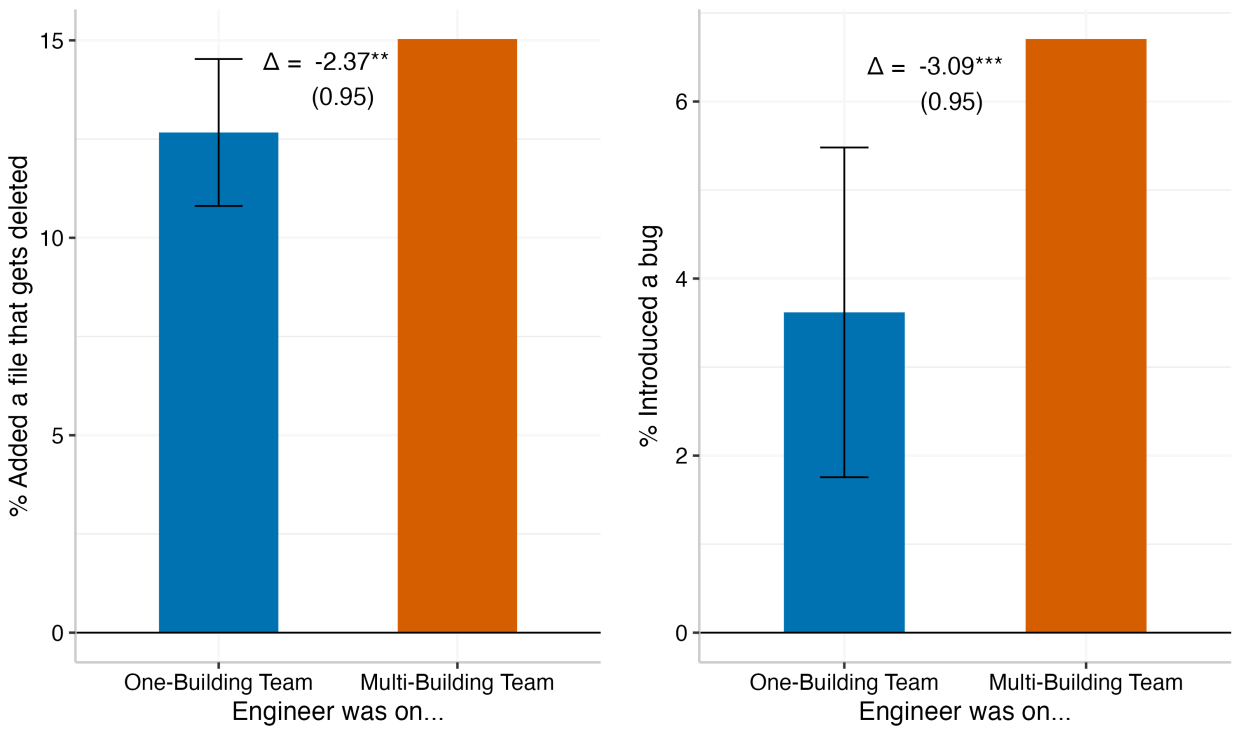


Notes: This figure investigates how the interactions between co-located teammates differ depending on whether the rest of the teammates are also co-located. The figure replicates the analysis in Figure Ia, but the dependent variable is the average number of comments exchanged when co-located teammates review each other's work. The annotated coefficient shows the difference-in-differences coefficient with the percentage effect in brackets. Standard errors are clustered by team. *p<0.1; **p<0.05; ***p<0.01.

Figure A.4: Downstream Differences in Code Quality

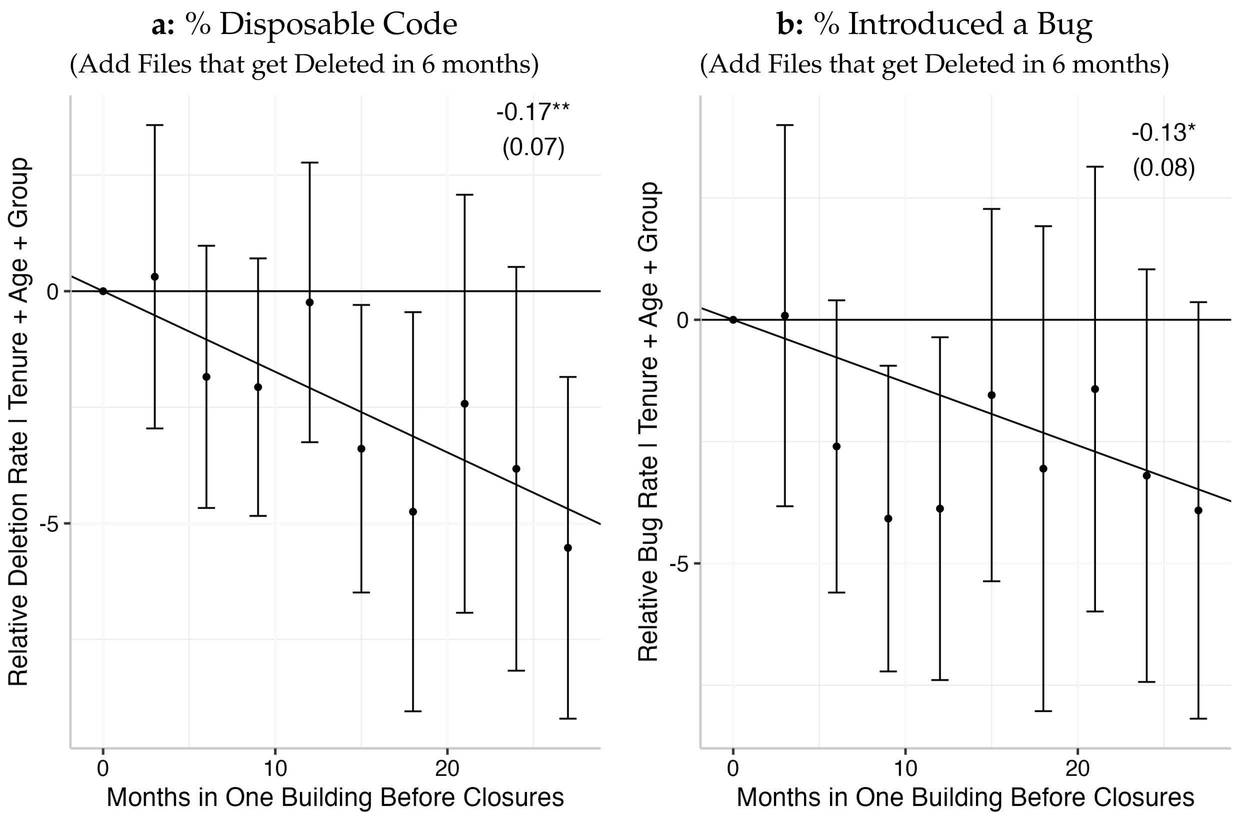
a: % Disposable Code
(Add Files that get Deleted in 6 months)

b: % Introduced a Bug
(Changes Immediately Reverted)



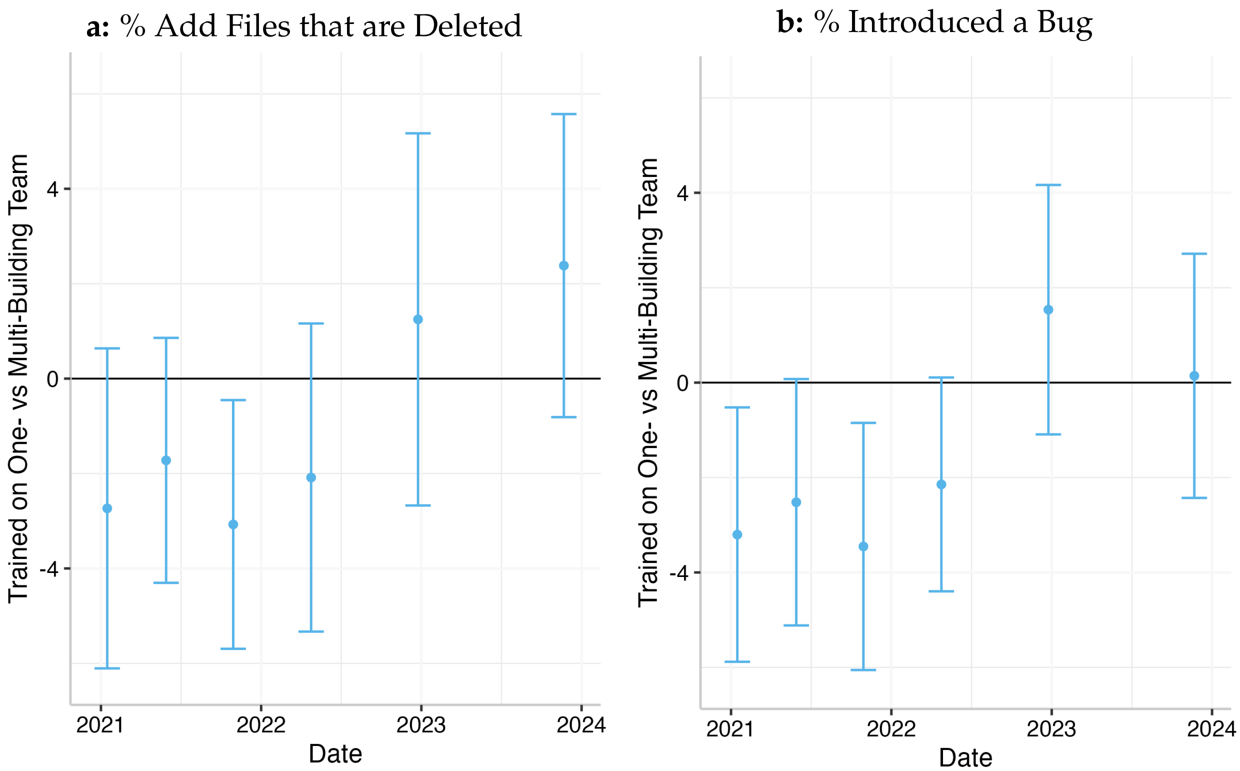
Notes: This figure examines long-run differences in code quality, using two different quality metrics. The period covers December 2020 (when these metrics started to be recorded) until the office reopening in 2022 (see Figure A.6 for the full time series). Panel a shows the percent of programs where the engineer added a file that later got deleted. Files may be deleted because the code was fully rewritten or because the firm decided the feature was a dead-end. Panel b focuses on introducing a bug, as defined by all the changes that the engineer makes getting reverted by a subsequent program. The annotated coefficient compares the two sets of engineers, with controls for engineer age, tenure, and engineering group. An engineer is defined as being on a co-located team if they were consistently co-located in the pre-closure period (fall of 2019 and winter of 2020), and as being on a multi-building team otherwise. Standard errors are clustered by engineering team. *p<0.1; **p<0.05; ***p<0.01.

Figure A.5: Code Quality by Exposure to One- Versus Multi-Building Teams



Notes: This figure complements Figure A.4 by showing the dose response of code quality to time spent on co-located teams. The outcome period starts in December 2020 when quality metrics started to be recorded and goes to when the offices first reopened in March 2022. The x-axis is the number of months that the engineer was on a co-located team before the offices closed, measured back to when our HR data starts in January 2018. The points each represent four months grouped together. The coefficients compare the two sets of engineers, with controls for engineer age, tenure, and engineering group. The error bars represent 95% confidence intervals. The annotated coefficient represents the linear fit illustrated on the graph. Standard errors are clustered by engineer since the team can change over time. *p<0.1; **p<0.05; ***p<0.01.

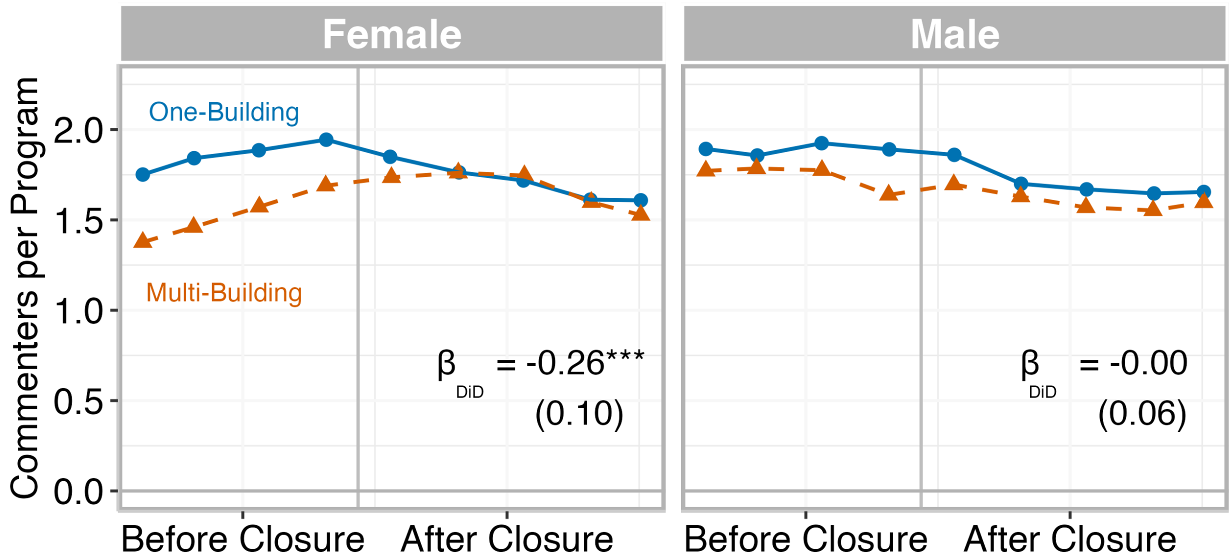
Figure A.6: Dynamics of Code Quality for those Trained on One- Versus Multi-Building Teams



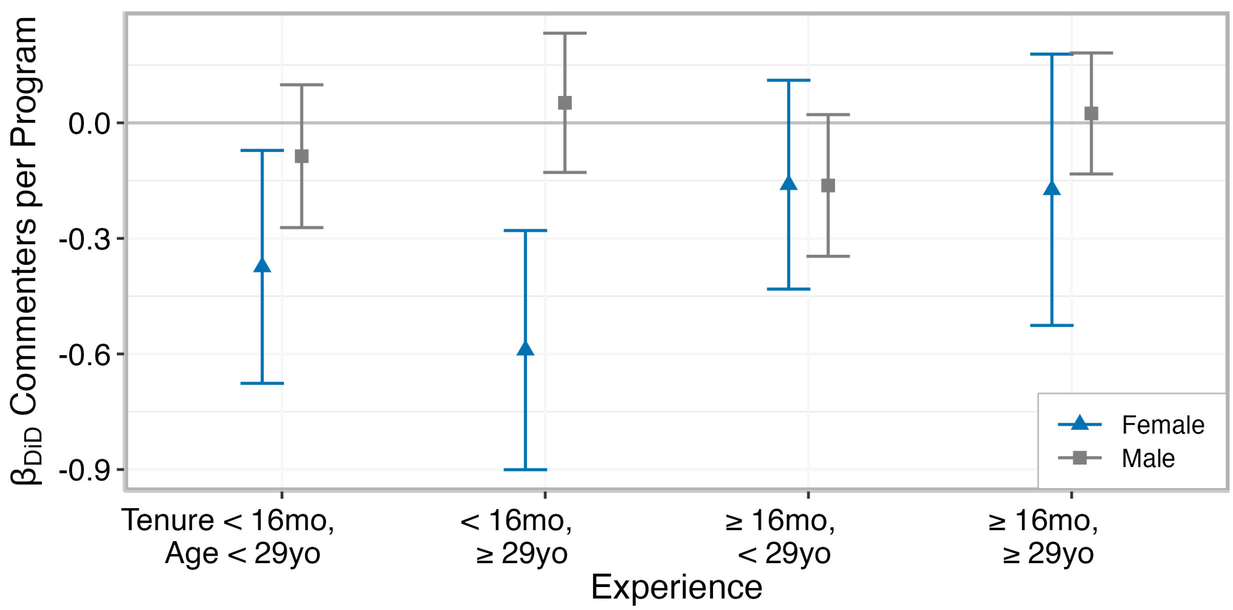
Notes: This figure complements Figure A.4 by showing the full time-series of the code quality differences between engineers trained on co-located teams versus those trained on multi-building ones. The period starts in December 2020 when quality metrics started to be recorded. The x-axis is grouped into six quantiles, which are imbalanced because there is attrition out of the sample. Panel a investigates the percent of programs where the engineer adds a file that later gets deleted. Files may be deleted because the code was fully rewritten or because the firm decided the feature was a dead-end. Panel b investigates introducing a bug, as defined by all the changes that the engineer made getting reverted by a subsequent program. The coefficients compare the two sets of engineers, with controls for engineer age, tenure, and engineering group. Standard errors are clustered by team.

Figure A.7: Gendered Effects of Proximity on Number of Commenters

Panel a: By Gender



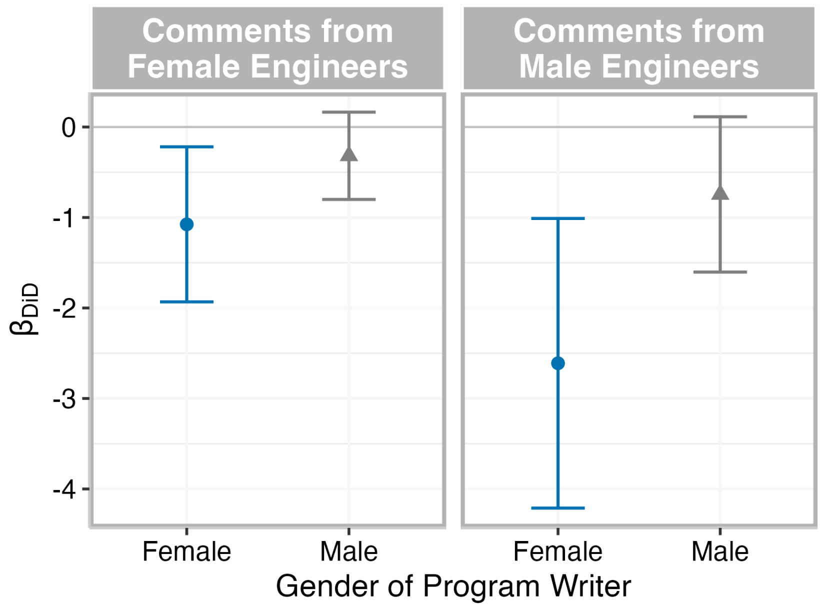
Panel b: Interacted Coefficients



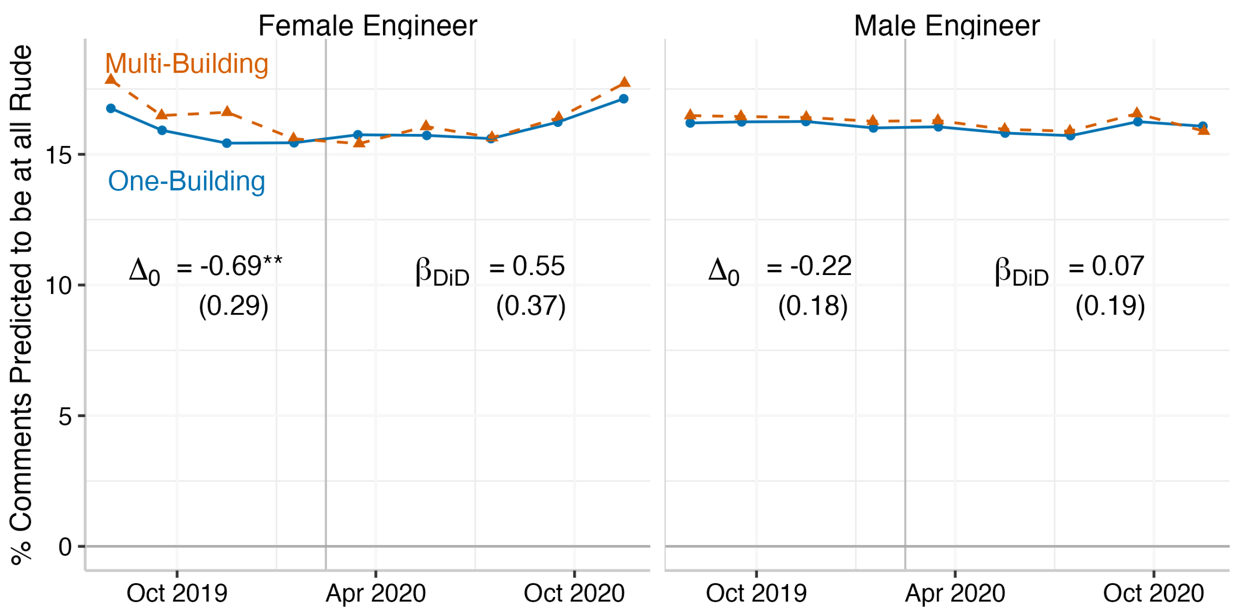
Notes: This figure explores gendered effects of proximity on the total number of people who comment on programmers' work. Panel a replicates Figure IIIc with the outcome of number of commenters rather than comments per program. Panel b is the parallel analogue of IIId. All DiD specifications use our preferred controls for month-specific effects of engineering group, engineer age, and engineer tenure. Standard errors are clustered by engineering team. *p<0.1; **p<0.05; ***p<0.01.

Figure A.8: Mansplaining?

Panel a: By Commenter Gender

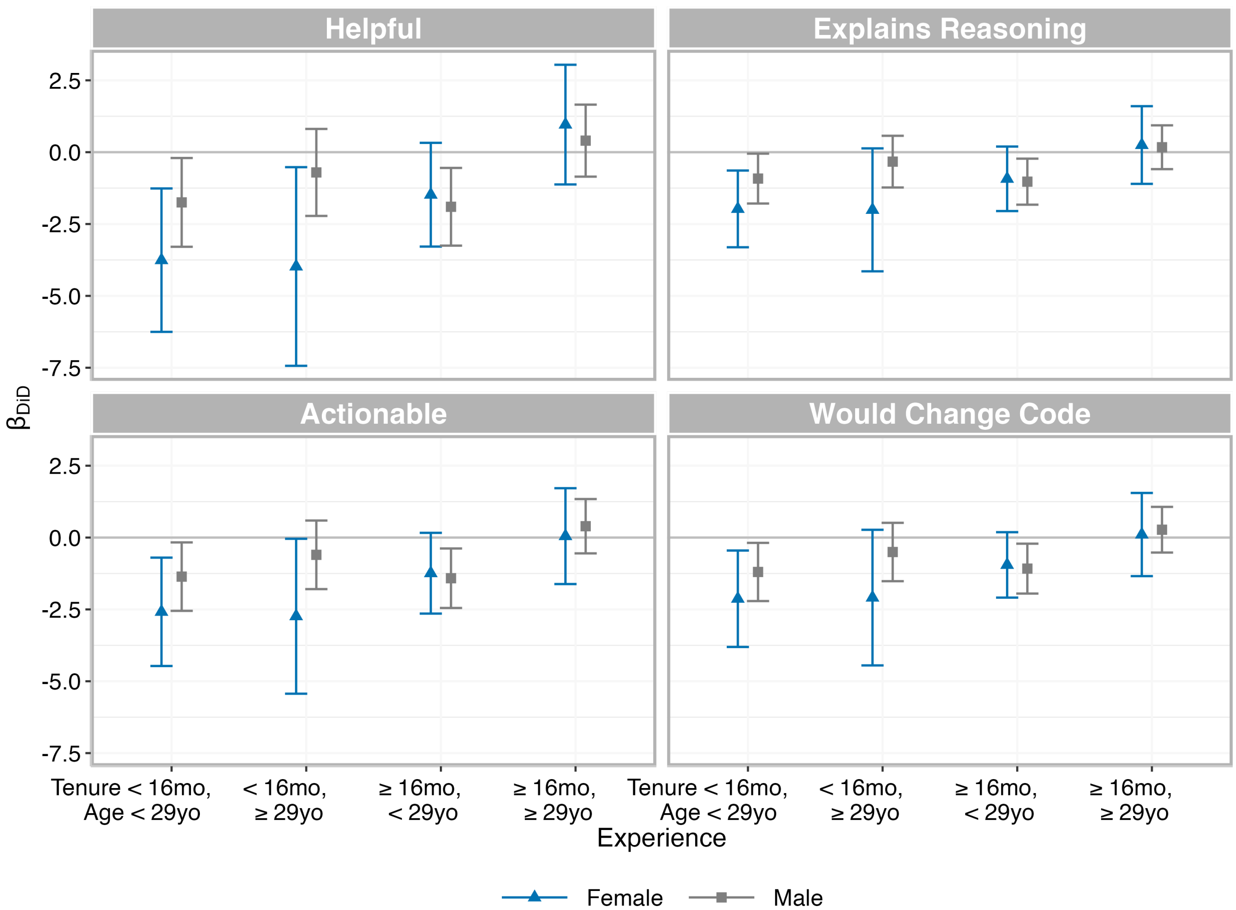


Panel b: Rudeness of Feedback



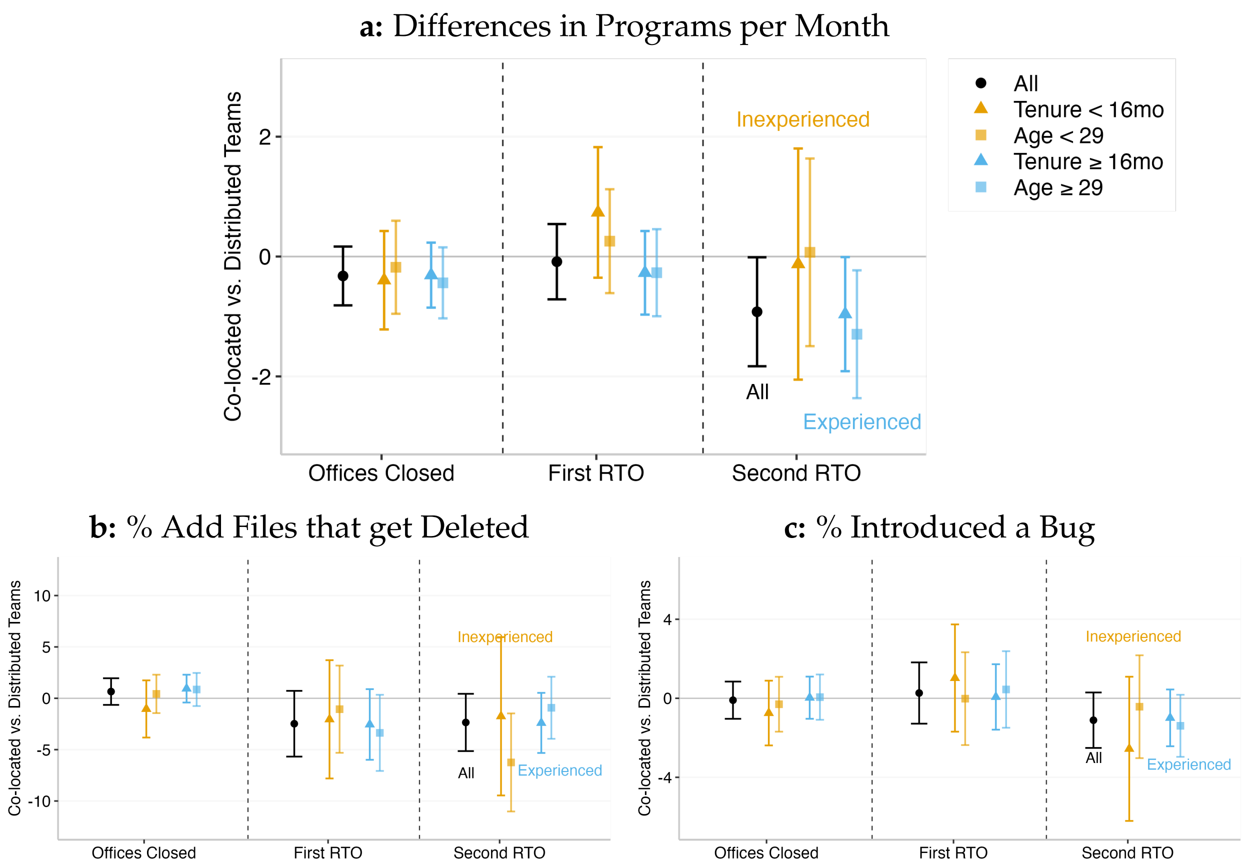
Notes: This figure investigates whether the gendered effect of proximity could be due to "mansplaining." Panel a shows our DiD estimates for comments from both female commenters (on the left) and male commenters (on the right). Panel b investigates comments that are predicted to be very, moderately, or a little bit rude. To form these predictions, we use the ratings of independent evaluators to train a supervised machine learning algorithm (see Appendix C for details). In these plots, the annotated coefficient on the left focuses on the pre-period difference and those on the right report the DiD estimates. Throughout, the specifications control for program scope and month-specific effects of engineer gender, age, tenure, and engineering group. Standard errors are clustered by team. $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$.

Figure A.9: Heterogeneity in Proximity's Effects on High-Quality Comments



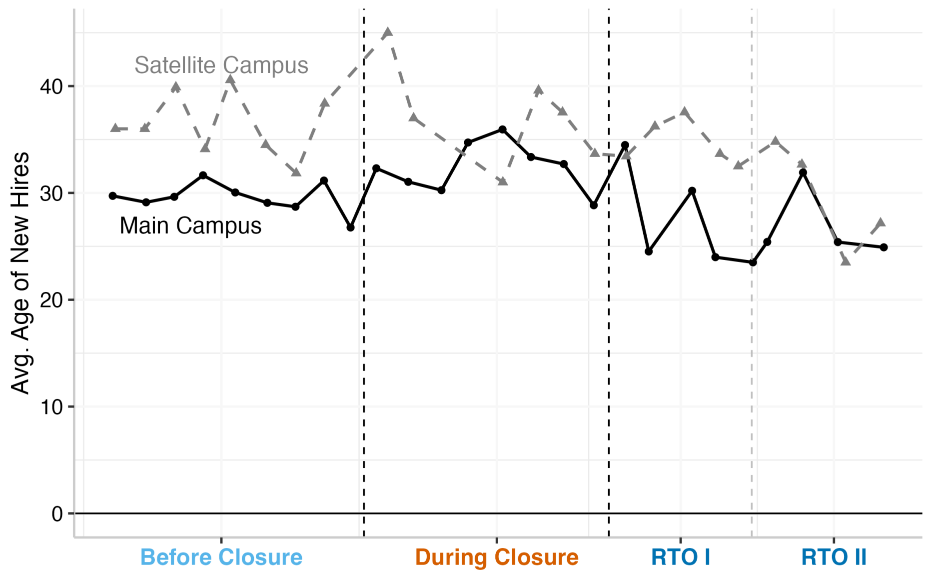
Notes: This figure investigates the heterogeneous effects of proximity on the total number of comments per program that are predicted to be helpful, well-reasoned, actionable, and impactful for the code. To generate these predictions, we employed independent raters to evaluate a random sample of comments. We then used a supervised machine learning algorithm to scale up to the whole dataset (see Appendix C for details). For each outcome, the coefficients come from a single difference-in-differences (DiD) design where the estimated effect of losing proximity is allowed to depend on engineer gender, age, and tenure. This is an interacted version of Equation 1, with controls for program scope and month-specific effects of engineer gender, age, tenure, and engineering group. Standard errors are clustered by team. *p<0.1; **p<0.05; ***p<0.01.

Figure A.10: RTOs with the Full Set of Controls



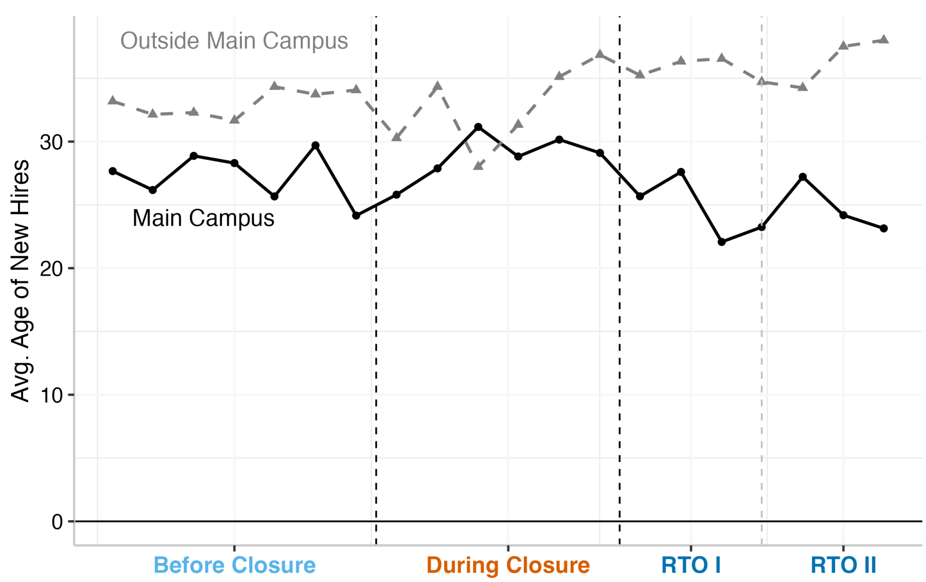
Notes: This figure replicates Figure IV but includes the full set of controls. In addition to our preferred month-specific controls for engineer age, tenure, and engineering group, this adds month-specific controls for gender, race, home zipcode, and job level. Experience is measured both by tenure at the firm (using a 16-month cutoff) and age (using a 29 year-old cutoff). Error bars represent 95% confidence intervals, with clustering by team.

Figure A.11: Hiring Patterns in Main vs. Satellite Campuses



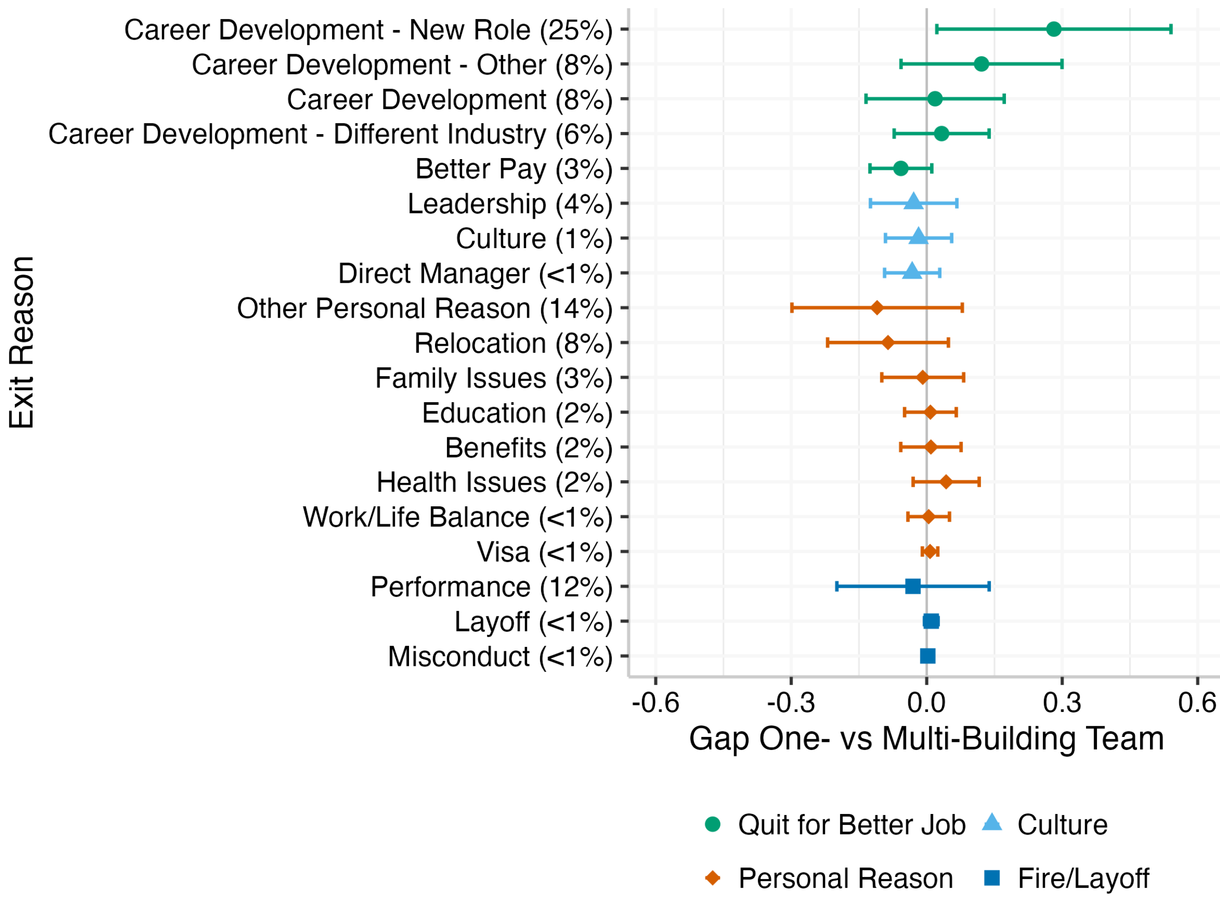
Notes: This figure replicates Figure VIIIc but excludes fully remote hires. Satellite-campus engineers were much more likely to be on geographically-distributed teams with some members who were in the headquarters. At the end of the period, the firm was trying to change by actively engineering "atomic" teams that could operate independently in each of the campuses.

Figure A.12: Hiring Patterns in the Rest of the Firm's Corporate Roles



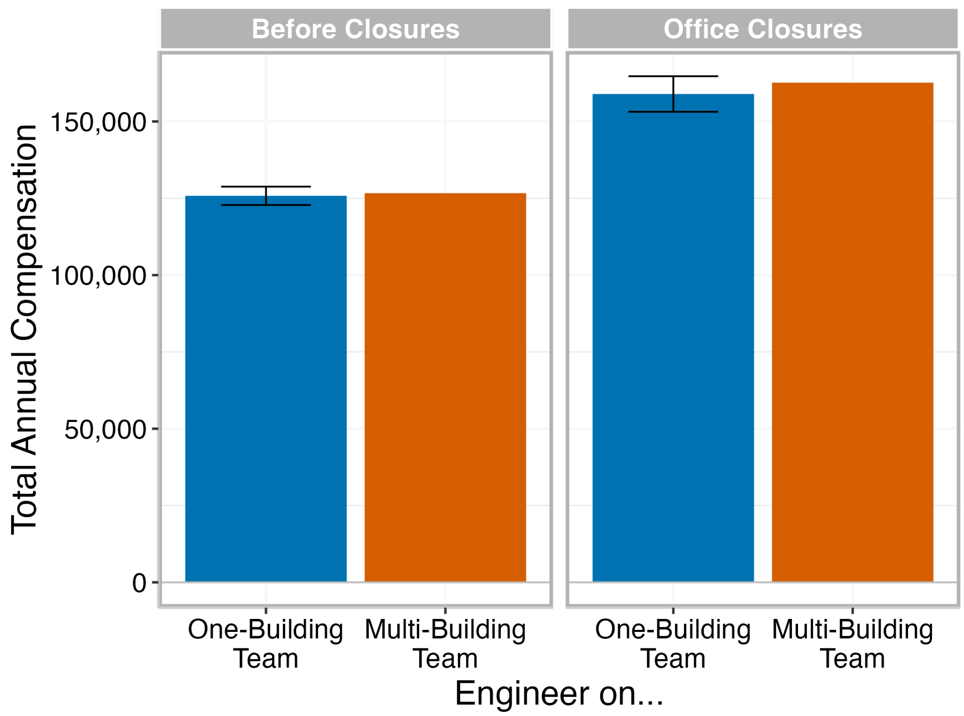
Notes: This figure replicates Figure VIIIc but focuses on the rest of the firm's corporate population excluding software engineering roles.

Figure A.13: Types of Exits for One- versus Multi-Building Teams during the Office Closures



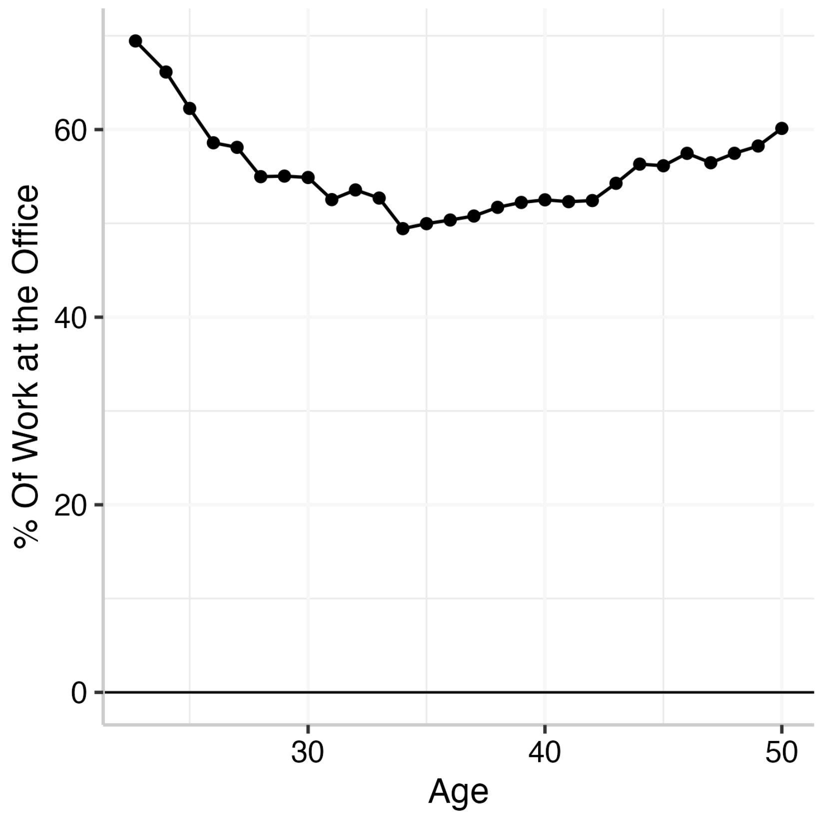
Notes: This figure unpacks Figure IX by differentiating between all the possible exit reasons. The y-axis plots specific categories for leaving that people give in exit interviews. The firm groups the many reasons that people give into these categories. The label reports the firm-defined category followed by the percentage of exits that fall into that category. The x-axis plots the differences in that type of exit between engineers on co-located and multi-building teams during the office closures. All specifications include our preferred time-varying controls for engineer group, age, and tenure. Standard errors are clustered by team and error bars show 95% confidence intervals.

Figure A.14: Proximity and Pay Around Office Closures



Notes: This figure illustrates differences in total compensation between one- and multi-building teams around the office closures. Each panel show differences between engineers on one- and multi-building teams with controls for engineer age, tenure, engineering group, and engineer fixed effects. Error bars represent 95% confidence intervals, with clustering by team.

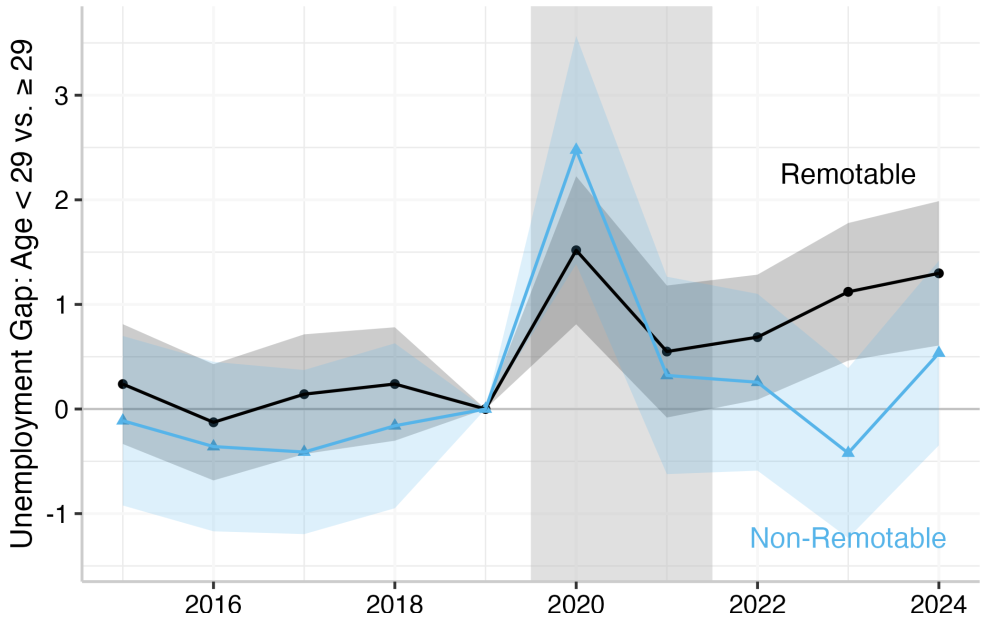
Figure A.15: Return to the Office by Age Excluding Parents



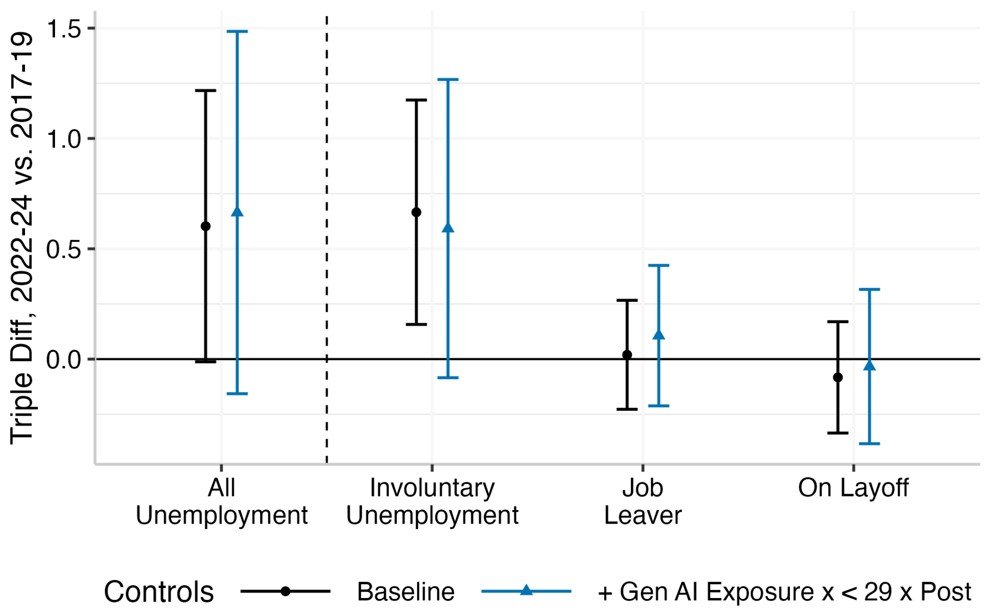
Notes: This figure replicates Figure VIId but excludes parents from the sample.

Figure A.16: The Rise of Remote Work and Young People's Unemployment: Alternative Definition of Remotable

a: Changes in Youth Gap in Unemployment in Remotable and Non-Remotable Jobs



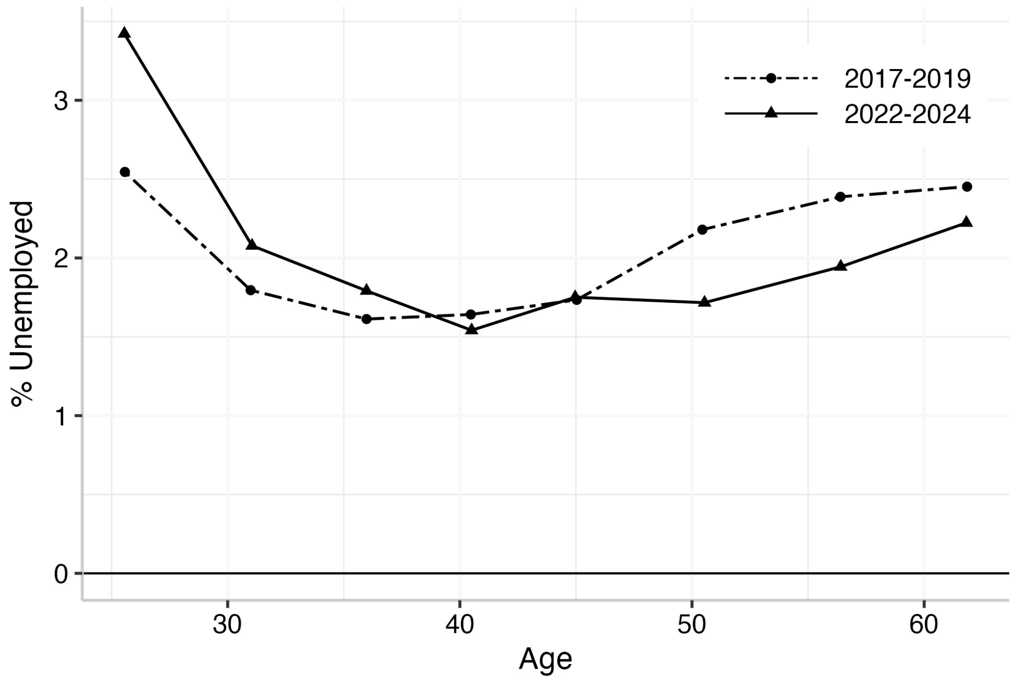
b: Comparing Change in Youth Gap in Remotable vs. in Non-Remotable Jobs



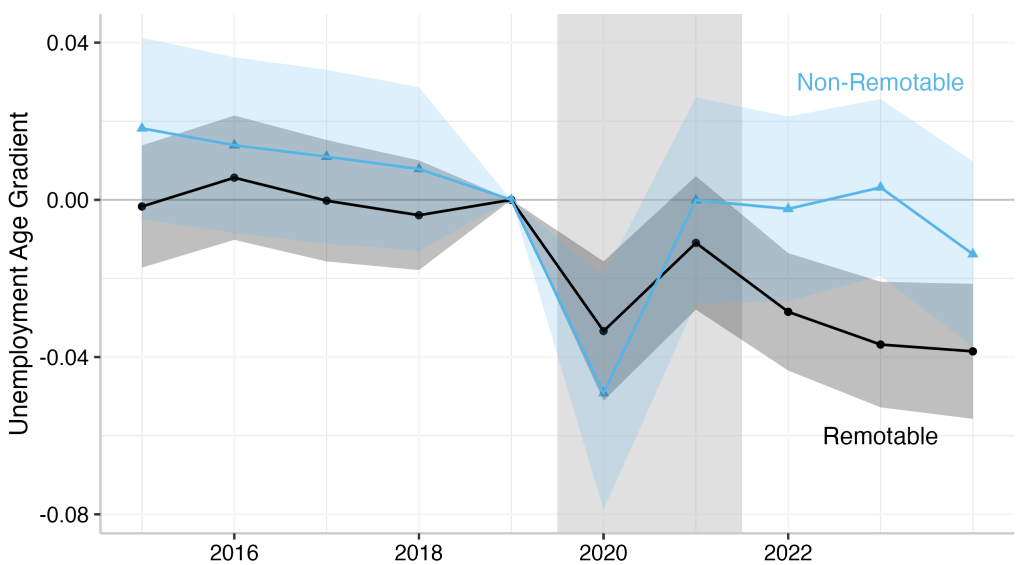
Notes: This figure replicates Figure X but excludes occupations that are classified as remotable by the Dingel-Neiman index but which Hansen et al. (2023)'s analysis of job posts suggests are in-person jobs: these are specifically teachers (other than professors) and management positions in in-person sectors (like forestry or food services). Data is from the Current Population Survey (Flood et al., 2024), limited to college graduates between ages 22 and 64. Standard errors are clustered by survey respondent.

Figure A.17: Age Gradient in Unemployment and the Rise of Remote Work

Panel (a): Unemployment by Age Over Time in Remotable Jobs



Panel (b): Changing Age Gradient in Unemployment



Notes: This figure shows the changes in unemployment over time for people in remotable and non-remotable occupations for people of different ages. Panel (a) focuses on those in remotable jobs and compares unemployment rates by age in 2022−2024 to those in 2017−2019. Age is grouped into octiles. Panel (b) replicates Figure Xa using continuous age rather than a binary indicator for being under the age of 29.

B Tables

Table A.1: Summary Statistics Distributions of Outcomes

Panel (a): Engineers Before the Office Closures

	Mean	SD	Q10	Q50	Q90
Programs in Main Code-Base per Month	1.8	2.2	0.0	1.0	5.0
Comments per Program	7.5	10.2	1.0	4.0	18.0
# Software Engineers	1,055	1,055	1,055	1,055	1,055
# Teams	304	304	304	304	304

Panel (b): Engineers Around the Office reopenings

	Mean	SD	Q10	Q50	Q90
Programs per Month	7.8	9.1	0.0	5.0	19.0
% Introduce Bugs	3.5	18.3	0.0	0.0	0.0
% Add Files that get Deleted	13.2	21.6	0.0	0.0	40.0
# Software Engineers	2,431	2,431	2,431	2,431	2,431
# Teams	995	995	995	995	995

Notes: This table shows the distribution of outcomes. The sample includes engineers whose teams are all in the headquarters campus. Panel (a) focuses on the period before the offices closed where we have data on the main code-base. Panel (b) focuses on the office reopenings where we have data on all code-bases.

Table A.2: Summary Statistics: Co-Located versus Geographically-Distributed Teams Before the Office Closure

	Mean	P10	P50	P90	Mean One-Building	Mean Fully-Distributed	Difference	
Age (Years)	28.81	23.00	28.00	36.00	28.54	29.32	-0.77*	-0.77*
							(0.42)	(0.40)
Firm Tenure (Years)	1.50	0.17	1.08	3.58	1.21	2.02	-0.81***	-0.79***
							(0.12)	(0.13)
% Female	19.55	0.00	0.00	100.00	19.53	19.58	-0.06	-0.14
							(3.08)	(3.33)
Job Traits								
Job Level	1.76	1.00	2.00	3.00	1.62	2.00	-0.38***	-0.38***
							(0.08)	(0.07)
Salary + Stocks	121,947	100,000	115,000	150,010	119,075	128,429	-9,354***	-9,688***
							(2,151)	(2,147)
Team Traits								
# Teammates	6.53	3.00	6.00	10.00	5.72	8.05	-2.33***	-2.17***
							(0.45)	(0.41)
Manager Tenure	3.10	0.80	3.45	5.88	2.84	3.65	-0.81**	-0.79**
							(0.32)	(0.31)
Manager Job Level	3.29	3.00	3.00	4.00	3.21	3.42	-0.21**	-0.21**
							(0.09)	(0.09)
Engineer Group								
Back-End	15.88	0.00	0.00	100.00	21.23	5.94	15.29***	–
							(4.37)	
Front-End	39.93	0.00	0.00	100.00	30.47	57.52	-27.04***	–
							(7.79)	
Internal Tools	34.43	0.00	0.00	100.00	37.23	29.22	8.01	–
							(7.34)	
AI Features	9.76	0.00	0.00	0.00	11.07	7.32	3.75	–
							(3.25)	
Engineer Group Controls								✓
# Software Engineers	931	931	931	931	637	294		
# Teams	292	292	292	292	206	108		

Notes: This table shows traits of the engineers, their job, and their team before the offices closed. The sample includes engineers in the headquarters campus whose teammates were either all in co-located or distributed across campuses. P10 refers to the 10th percentile, P50 to the median, and P90 to the ninetieth percentile. The last two columns compare the attributes of co-located and geographically-distributed teams. Column 7 does not include controls, while Column 8 includes engineer group fixed effects (for back-end, front-end, internal tools, and AI). "Job level" refers to the engineer's position within the firm's hierarchy from zero (an intern) to six (senior staff). Standard errors in parentheses are clustered by engineering team. $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$.

Table A.3: Summary Statistics: Co-Located versus Geographically-Distributed Teams around the Office Reopenings

					Mean			
	Mean	P10	P50	P90	One-Building	Fully-Distributed	Difference	
Age (Years)	31.49	25.00	30.00	39.00	30.13	31.82	-1.69***	-1.50***
							(0.23)	(0.25)
Firm Tenure (Years)	3.37	0.92	3.08	6.08	2.69	3.53	-0.84***	-0.60***
							(0.09)	(0.08)
% Female	20.10	0.00	0.00	100.00	22.54	19.51	3.03*	2.42
							(1.72)	(1.77)
Job Traits								
Job Level	2.56	1.00	3.00	4.00	2.27	2.63	-0.36***	-0.30***
							(0.04)	(0.04)
Salary + Stocks	181,466	110,303	174,950	246,680	166,102	185,181	-19,079***	-16,408***
							(2,343)	(2,376)
Team Traits								
# Teammates	5.33	2.00	5.00	9.00	3.72	5.72	-2.00***	-1.94***
							(0.20)	(0.23)
Manager Tenure	3.73	0.92	3.32	7.02	3.86	3.70	0.17	0.40***
							(0.15)	(0.15)
Manager Job Level	3.95	3.00	4.00	5.00	3.71	4.00	-0.29***	-0.27***
							(0.04)	(0.04)
Engineer Group								
Back-End	18.44	0.00	0.00	100.00	27.69	16.21	11.48***	–
							(3.11)	
Front-End	19.13	0.00	0.00	100.00	20.59	18.78	1.82	–
							(2.79)	
Internal Tools	34.66	0.00	0.00	100.00	22.15	37.69	-15.53***	–
							(3.28)	
AI Features	22.14	0.00	0.00	100.00	24.10	21.66	2.44	–
							(2.83)	
Engineer Group Controls								✓
# Software Engineers	2,380	2,380	2,380	2,380	1,298	2,105		
# Teams	915	915	915	915	414	802		

Notes: This table shows traits of the engineers, their job, and their team around the return-to-office mandates. The sample includes engineers in the headquarters campus. P10 refers to the 10th percentile, P50 to the median, and P90 to the ninetieth percentile. The last two columns compare the attributes of co-located and geographically-distributed teams. Column 7 does not include controls, while Column 8 includes engineer group fixed effects (for back-end, front-end, internal tools, and AI). "Job level" refers to the engineer's position within the firm's hierarchy from zero (an intern) to six (senior staff). Standard errors in parentheses are clustered by engineering team. *p<0.1; **p<0.05; ***p<0.01.

Table A.4: Proximity & Feedback Identified from Team Type Switchers

	Comments per Program			
	(1)	(2)	(3)	(4)
One-Building$_{it}$ Before Closure	0.96**		0.85	
	(0.45)		(0.54)	
One-Building$_{it}$ After Closure	0.09		-0.29	
	(0.36)		(0.52)	
One-Building$_{it}$ x Post		-0.87^{*}		-1.14^{**}
		(0.51)		(0.56)
Pre-Mean, One-Building	8.27	8.27	8.27	8.27
Percentage Effects:				
One-Building$_{it}$ Before Closure	11.6%	11.6%	10.2%	10.2%
One-Building$_{it}$ After Closure	1.1%		-3.5%	
One-Building$_{it}$ x Post		-10.5%		-13.8%
Preferred Controls	✓	✓	✓	✓
Engineer FE	✓	✓	✓	✓
Engineer x Manager FE			✓	✓
# Pre-Closure Switcher Engineers	102	102	102	102
# All Engineers	1,055	1,055	1,055	1,055
# Pre-Closure Switcher Teams	51	51	51	51
# All Teams	304	304	304	304
# Engineer-Months	9,304	9,304	9,304	9,304

Notes: This table uses a complementary design to evaluate how proximity to teammates affects feedback, using engineers who switch between team types. Each column includes engineer fixed effects, as well as our preferred set of time-varying controls for engineer group, tenure, and age and program scope quartics (as in Column 3 of Table II). In this table, we define Co-Located Team$_{it}$ at the monthly level for each engineer, which allows us to identify the relationship between being on a co-located team and the feedback the engineer receives with engineer fixed effects, both before and after the office closures (in Column 1) as well as the difference in these coefficients (in Column 2). Columns $3-4$ include engineer by manager fixed effects to compare the same engineer under the same manager as a function of whether the team is all co-located. Standard errors are clustered by team. *p<0.1; **p<0.05; ***p<0.01.

Table A.5: Proximity and Predicted High-Quality Comments

Panel a: Total Substantive Feedback

	Comments per Program Predicted to be...			
	Helpful	Explain Reasoning	Actionable	Would Change Code
	(1)	(2)	(3)	(4)
One-Building x Post	-1.45^{***}	-0.70^{***}	-1.08^{***}	-0.86^{***}
	(0.46)	(0.27)	(0.36)	(0.31)
Pre-Mean, One-Building	6.37	3.35	4.67	3.91
Post x One-Building as %	-22.8%	-21%	-23.1%	-22%

Panel b: Percent of Substantive Feedback

	% Comments Predicted to be...			
	Helpful	Explain Reasoning	Actionable	Would Change Code
	(1)	(2)	(3)	(4)
One-Building x Post	-2.86^{**}	-1.75^{*}	-1.66	-1.86^{*}
	(1.38)	(1.04)	(1.21)	(1.03)
Pre-Mean, One-Building	50.79	33.12	39.29	33.94
Post x One-Building as %	-5.6%	-5.3%	-4.2%	-5.5%
# Teams	304	304	304	304
# Engineers	1,055	1,055	1,055	1,055
# Engineer-Months	9,304	9,304	9,304	9,304

Notes: This table evaluates how proximity to teammates relates to the predicted quality of comments. Each specification replicates Column 3 of Table II. Panel a focuses on the total number of comments per program predicted to be (1) helpful, (2) well-reasoned, (3) actionable, and (4) likely to cause the programmer to change the code. These patterns are also illustrated in Figure IIb. Panel b focuses on the share of comments that fall into these categories. To form these predictions, we employ external software engineers to rates 5,337 comments on each of these dimensions. We then use a supervised machine learning algorithm to scale up from this sample to generate predictions for the entire dataset. Appendix C provides more details. Standard errors are clustered by team. *p<0.1; **p<0.05; ***p<0.01.

Table A.6: Downstream Differences in Code Quality

	% Added a File that gets Deleted		% Introduced a Bug	
Was on a One Building Team	-2.37^{**}	-2.90^{**}	-3.09^{***}	-1.83
	(0.95)	(1.37)	(0.95)	(1.20)
Dependent Mean	5.32	5.32	14.45	14.45
Percentage Effects				
Team in One Building	-16.4%	-20.1%	-58%	-34.4%
Current Team FE		✓		✓
# Teams	285	285	285	285
# Engineers	828	828	828	828
# Engineer-Months	10,258	10,258	10,258	10,258

Notes: This table examines long-run differences in code quality, using two different quality metrics. The period covers December 2020 (when these metrics started to be recorded) until the office reopening in 2022. The first two columns focus on the percent of programs where the engineer adds a file that later gets deleted. Files may be deleted because the code was fully rewritten or because the firm decided the feature was a dead-end. The last two columns focus on introducing a bug, as defined by all the changes that an engineer made getting reverted by a subsequent program. Every specification includes our preferred controls for engineer age, tenure, and engineering group, with each interacted with the month. The even columns also include fixed effects for the engineer's current team. Standard errors are clustered by team. *p<0.1; **p<0.05; ***p<0.01.

Table A.7: Robustness of Heterogeneity by Age & Gender

	Comments per Program					
	(1)	(2)	(3)	(4)	(5)	(6)
Young (<29yo) x Post x One-Building	-2.09^{***}	-1.84^{**}	-1.72^{**}			
	(0.69)	(0.71)	(0.71)			
Female x Post x One-Building			-1.41^{*}	-2.70^{***}	-2.41^{**}	-2.33^{**}
			(0.72)	(0.94)	(0.94)	(0.96)
Pre-Mean, One-Building	8.04	8.04	8.04	8.04	8.04	8.04
One Building x Pre-Post Closure x...						
Months at Firm Indicators		✓	✓		✓	✓
Age in Years Indicators						✓
% One-Building Team	58.3	58.3	58.3	58.3	58.3	58.3
# Teams	304	304	304	304	304	304
# Engineers	1,055	1,055	1,055	1,055	1,055	1,055
# Engineer-Months	9,304	9,304	9,304	9,304	9,304	9,304

Notes: This table probes the robustness of heterogeneity in proximity's effects on feedback by age and gender to the inclusion of other interaction terms. Columns 1 and 4 show the baseline heterogeneity by (1) age and (4) gender for reference. These designs mirror those in Panels b and c of Figure III. Columns 2 and 5 include detailed interactions of engineer tenure (with different indicators for every month of tenure) with being on a co-located team and the pre- and post-closure period. These specifications investigate whether there remains significant differentials by engineer age or gender when accounting for tenure. Column 6 adds analogous interactions for engineer age (with different indicators for every year). All specifications include our preferred controls for program scope, engineer fixed effects, and month-specific controls for engineer age, tenure, and engineering group. Standard errors are clustered by engineering team. *p<0.1; **p<0.05; ***p<0.01.

Table A.8: Proximity to Teammates and Engineer Output

	Monthly Contributions to the Main Codebase					
	Programs		Lines Added		Files Changed	
	(1)	(2)	(3)	(4)	(5)	(6)
Post x One-Building	0.39***		101.00***		1.75*	
	(0.10)		(36.76)		(0.94)	
Senior ($\geq$ 16mo) x Post x One-Building		0.58***		133.90**		3.50**
		(0.18)		(58.74)		(1.53)
Junior ($<$ 16mo) x Post x One-Building		0.30***		84.64**		0.88
		(0.11)		(41.95)		(1.10)
Pre-Mean, One-Building Team	1.71	1.71	320.52	320.52	9.64	9.64
# Engineers	1,055	1,055	1,055	1,055	1,055	1,055
# Engineer-Months	16,058	16,058	16,058	16,058	16,058	16,058

Notes: This table investigates the relationship between sitting near teammates and monthly output of (Columns 1−2) programs submitted to the main code-base, (Columns 3−4) lines of code added, and (Columns 5−6) files changed. The odd columns estimate the aggregate effects, while the even columns differentiate by engineer tenure (split by the median tenure). Each specification estimates Equation 1, with our preferred controls of engineer fixed effects and month-specific controls for engineer age, tenure, and engineering group. The sample includes engineers who ever submitted a program to the firm's main code-base and whose teammates are all in the firm's headquarters campus. *p<0.1; **p<0.05; ***p<0.01.

Table A.9: Robustness of Proximity to Teammates and Engineer Output

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
	Panel (a): Programs per Month						
Post x One-Building Team	0.47***	0.45***	0.48***	0.48***	0.40***	0.37***	0.32***
	(0.11)	(0.13)	(0.13)	(0.13)	(0.11)	(0.11)	(0.12)
One-Building Team	−0.48***	−0.33**	−0.37**	−0.36**			
	(0.16)	(0.16)	(0.16)	(0.16)			
Pre-Mean, One-Building Team	1.71	1.71	1.71	1.71	1.71	1.71	1.71
Post x One-Building Team as %	27.5%	26.3%	27.7%	27.9%	23.4%	21.4%	18.9%
One-Building Team as %	-28.1%	-19.1%	-21.8%	-21.1%			
R^2	0.01	0.07	0.15	0.18	0.52	0.55	0.55
	Panel (b): Lines Added per Month						
Post x One-Building Team	105***	92**	108***	106**	102***	102***	123***
	(36)	(39)	(42)	(41)	(37)	(36)	(39)
One-Building Team	−193***	−158***	−186***	−181***			
	(43)	(44)	(46)	(45)			
Pre-Mean, One-Building Team	321	321	321	321	321	321	321
Post x One-Building Team as %	32.9%	28.6%	33.7%	33.2%	31.7%	31.8%	38.5%
One-Building Team as %	-60.2%	-49.3%	-58%	-56.6%			
R^2	0.02	0.04	0.12	0.14	0.41	0.44	0.44
	Panel (c): Files Changed per Month						
Post x One-Building Team	1.93**	1.66	2.06*	2.01*	1.80*	1.79*	2.12**
	(0.97)	(1.06)	(1.11)	(1.09)	(0.94)	(0.97)	(1.08)
One-Building Team	−3.97***	−3.62***	−4.24***	−4.12***			
	(1.12)	(1.15)	(1.18)	(1.16)			
Pre-Mean, One-Building Team	9.64	9.64	9.64	9.64	9.64	9.64	9.64
Post x One-Building Team as %	20%	17.2%	21.3%	20.9%	18.7%	18.6%	21.9%
One-Building Team as %	-41.2%	-37.6%	-44%	-42.7%			
R^2	0.01	0.03	0.10	0.13	0.38	0.41	0.41
Engineer Group x Month FE		✓	✓	✓	✓	✓	✓
Months at Firm x Month FE			✓	✓	✓	✓	✓
Age x Month FE				✓	✓	✓	✓
Engineer FE					✓	✓	✓
Engineer Traits x Month FE						✓	✓
Main Building x Month FE							✓
# Teams	304	304	304	304	304	304	304
# Engineers	1,055	1,055	1,055	1,055	1,055	1,055	1,055
# Engineer-Months	9,304	9,304	9,304	9,304	9,304	9,304	9,304

Notes: This table investigates the relationship between sitting near teammates and monthly output of (a) programs submitted to the main code-base, (b) lines of code added, and (c) files changed. Each specification estimates Equation 1, with controls defined in Table II. The sample includes engineers who ever submitted a program to the firm's main code-base and whose teammates are all in the firm's headquarters campus. *p<0.1; **p<0.05; ***p<0.01.

Table A.10: RTO Difference-in-Differences

	Panel (a): Programs per month					
2nd RTO x Co-Located Team	−0.60	−1.06**				
	(0.52)	(0.51)				
Tenure < 16mo: 2nd RTO x Co-Located Team			0.04	−1.28		
			(0.95)	(1.02)		
Tenure ≥ 16mo: 2nd RTO x Co-Located Team			−0.62	−0.88		
			(0.55)	(0.54)		
Age < 29: 2nd RTO x Co-Located Team					0.69	−0.11
					(0.80)	(0.77)
Age ≥ 29: 2nd RTO x Co-Located Team					−1.02*	−1.35**
					(0.60)	(0.63)
Dependent Mean	7.83	7.83	7.83	7.83	7.83	7.83
	Panel (b): % Add file that gets deleted					
2nd RTO x Co-Located Team	−2.99**	−0.72				
	(1.23)	(1.70)				
Tenure < 16mo: 2nd RTO x Co-Located Team			−4.61	−3.30		
			(3.53)	(3.72)		
Tenure ≥ 16mo: 2nd RTO x Co-Located Team			−3.06**	−0.51		
			(1.28)	(1.84)		
Age < 29: 2nd RTO x Co-Located Team					−5.10**	−4.36*
					(2.15)	(2.45)
Age ≥ 29: 2nd RTO x Co-Located Team					−2.36*	0.95
					(1.33)	(1.97)
Dependent Mean	13.17	13.17	13.17	13.17	13.17	13.17
	Panel (c): % Introduce a Bug					
2nd RTO x Co-Located Team	−1.21*	−1.59*				
	(0.73)	(0.81)				
Tenure < 16mo: 2nd RTO x Co-Located Team			−2.00	−3.52**		
			(1.30)	(1.38)		
Tenure ≥ 16mo: 2nd RTO x Co-Located Team			−1.23	−1.32		
			(0.78)	(0.90)		
Age < 29: 2nd RTO x Co-Located Team					−0.76	−0.57
					(1.27)	(1.20)
Age ≥ 29: 2nd RTO x Co-Located Team					−1.48	−2.14*
					(0.91)	(1.12)
Dependent Mean	3.46	3.46	3.46	3.46	3.46	3.46
DiD versus Office Closure	✓		✓		✓	
DiD versus 1st RTO		✓		✓		✓
# Teams	995	995	995	995	995	995

Notes: This table investigates the difference-in-differences between co-located teams and geographically-distributed teams in the second RTO versus in either the office closure period (odd columns) or in the first RTO (even columns). This table tests whether the differences in Figure IV are significantly different from each other. Each specification uses the controls defined in Figure IV. The sample includes engineers who are in the firm's headquarters campus. *p<0.1; **p<0.05; ***p<0.01.

Table A.11: RTO and Alternative Measures of Coding Quantity

	Monthly Programming Output			
	Programs	Added Lines	Changed Files	Main Code-Base Programs
Second RTO x Co-located Team	-0.91^{**}	-133.60	-2.58	-0.17
	(0.45)	(101.10)	(2.44)	(0.14)
First RTO x Co-located Team	0.15	17.72	0.26	-0.04
	(0.28)	(70.69)	(1.56)	(0.06)
Fully Remote x Co-located Team	-0.31	51.60	1.59	-0.03
	(0.23)	(35.08)	(0.99)	(0.06)
Dependent Mean	7.83	7.83	711.41	711.41
Percentage Effect				
Second RTO x Co-located Team	-11.67%	-18.78%	-10.99%	-25.16%
	(5.81)	(14.21)	(10.38)	(20.85)
# Teams	995	995	995	995
Observations	47,806	47,806	47,806	47,806
R^2	0.56	0.43	0.54	0.48

Notes: This table investigates the relationship between sitting near teammates and different dimensions of programming output. Column 1 show number of programs pr month (as in Figure IVb). Column 2 shows lines of code added; Column 3, files changed; and Column 7, programs per month in the main code-base. Each specification estimates Equation 2, with engineer fixed effects and our preferred month-specific controls for engineer age, tenure, and engineering group. The sample includes engineers who ever submitted a program and who are themselves in the firm's headquarters campus. $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$.

Table A.12: Quits and Time Spent on Co-Located Teams

Panel (a): During Office Closures

	% Quit for Better Job			
	(1)	(2)	(3)	(4)
Share of Pre-Closure Months on One-Building Team	0.51**	−0.12	0.29	−0.78
	(0.25)	(0.54)	(0.28)	(0.58)
Share One-Building Team x Tenure < 16mo		0.71		0.78
		(0.53)		(0.53)
Share One-Building Team x Age < 29yo				0.76**
				(0.37)
Share One-Building Team x Female			0.98**	0.88**
			(0.42)	(0.42)
Dependent Mean	1.01	1.01	1.01	1.01
# Teams	299	299	299	299
# Engineers	995	995	995	995
R^2	0.005	0.005	0.005	0.01

Panel (b): Before Office Closures

	% Quit for Better Job			
	(1)	(2)	(3)	(4)
Share of Pre-Closure Months on One-Building Team	−0.01	0.09	0.06	0.13
	(0.18)	(0.25)	(0.19)	(0.26)
Share One-Building Team x Tenure < 16mo		−0.11		−0.11
		(0.30)		(0.30)
Share One-Building Team x Age < 29yo				0.06
				(0.28)
Share One-Building Team x Female			−0.38	−0.39
			(0.29)	(0.30)
Dependent Mean	0.26	0.26	0.26	0.26
# Teams	304	304	304	304
# Engineers	1,055	1,055	1,055	1,055
R^2	0.03	0.04	0.04	0.04

Notes: This table investigates the relationship between quits for better jobs during the closures and the time that the engineer had spent on a co-located team beforehand. This estimates a continuous version of the analysis shown in Figure IX. All specifications include our preferred time-varying controls for engineer group, age, and tenure, as well as gender (where applicable). Standard errors are clustered by team. $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$.

Table A.13: Office Attendance by Age Interacted with Team and Manager Proximity

	Badged into Building			
	(1)	(2)	(3)	(4)
Young (<29) x Co-Located Team (Everyone in HQ)	5.44***	3.98***		
	(1.39)	(1.51)		
Young (<29) x All Teammates in HQ			5.13***	3.65***
			(1.14)	(1.27)
Young (<29) x Manager in HQ			2.63***	2.28**
			(0.99)	(1.12)
Older (≥29) x Co-Located Team (Everyone in HQ)	1.47			
	(0.90)			
Older (≥29) x All Teammates in HQ			1.48**	
			(0.74)	
Older (≥29) x Manager in HQ			0.35	
			(0.70)	
Co-Located Team (Everyone in HQ)		1.47		
		(0.90)		
All Teammates in HQ				1.48**
				(0.74)
Manager in HQ				0.35
				(0.70)
Young (<29)	4.84***	4.84***	3.29***	3.29***
	(0.59)	(0.59)	(0.93)	(0.93)
Dependent Mean	23.45	23.45	23.45	23.45
# Teams	782	782	782	782
Observations	519,016	519,016	519,016	519,016
R^2	0.32	0.32	0.32	0.32

Notes: This table analyzes young people's revealed preference to be with their coworkers and their managers. The sample is limited to engineers in the headquarters campus and includes weekdays after the first return-to-office mandate in 2022. The dependent variable is whether a worker badged into the office on that day. The first two columns focus on whether the whole team is co-located in the headquarters as in Figure VIIa. The second two columns differentiate between (i) all the teammates being in the headquarters and (ii) the manager being in the headquarters with the engineer. Both of these things must hold for the team to be co-located. Standard errors are clustered by team. *p<0.1; **p<0.05; ***p<0.01.

Table A.14: Office Attendance by Tenure Interacted with Team and Manager Proximity

	Badged into Building			
	(1)	(2)	(3)	(4)
Low Tenure (<16 mo) x Co-Located Team	5.50***	3.97**		
	(1.61)	(1.69)		
Low Tenure (<16 mo) x All Teammates in HQ			4.92***	3.50**
			(1.45)	(1.55)
Low Tenure (<16 mo) x manager in HQ			3.16**	2.53*
			(1.38)	(1.44)
High Tenure (≥16 mo) x Co-Located Team	1.53*			
	(0.83)			
High Tenure (≥16 mo) x All Teammates in HQ			1.42**	
			(0.69)	
High Tenure (≥16 mo) x manager in HQ			0.63	
			(0.62)	
Co-Located Team		1.53*		
		(0.83)		
All Teammates in HQ				1.42**
				(0.69)
manager in HQ				0.63
				(0.62)
Low Tenure (<16 mo)	8.45***	8.45***	6.60***	6.60***
	(0.83)	(0.83)	(1.08)	(1.08)
Dependent Mean	23.57	23.57	23.57	23.57
# Teams	791	791	791	791
Observations	528,346	528,346	528,346	528,346
R^2	0.32	0.32	0.32	0.32

Notes: This table analyzes new hires' revealed preference to be with their coworkers and their managers. The sample is limited to engineers in the headquarters campus and includes weekdays after the first return-to-office mandate in 2022. The dependent variable is whether a worker badged into the office on that day. The first two columns focus on whether the whole team is co-located as in Figure VIIa. The second two columns differentiate between the manager being in the headquarters with the engineer versus the rest of the team being in the headquarters. Standard errors are clustered by team. *p<0.1; **p<0.05; ***p<0.01.

Table A.15: Office Attendance by Age with Controls

	Badged into Building		
Young (<29) x Co-Located Team	4.09***	4.87***	4.43**
	(1.51)	(1.51)	(1.81)
Young (<29)	4.71***	3.72***	2.09***
	(0.59)	(0.61)	(0.78)
Commute Time (in Hours) x Co-Located Team		3.79	6.71**
		(2.55)	(3.40)
Commute Time		−8.98***	−9.78***
		(1.15)	(1.28)
Father x Co-Located Team			−3.16
			(2.05)
Father			−1.04
			(0.69)
Mother x Co-Located Team			−1.43
			(2.26)
Mother			−3.24***
			(1.09)
Co-Located Team	1.33	−0.71	−1.57
	(0.89)	(1.47)	(1.66)
Date FE	✓	✓	✓
Dependent Mean	23.52	23.51	23.68
# Teams	790	790	683
Observations	527,186	526,665	369,731
R^2	0.32	0.32	0.33

Notes: This table analyzes the age differences in going into the office and how those differences interact with whether the engineer's team is co-located or distributed. The sample is limited to engineers in the headquarters campus and includes weekdays after the first return-to-office mandate in 2022. The dependent variable is whether a worker badged into the office on that day. Commute time comes from using the Google API to calculate travel time from the engineer's home address (collected for nearly all engineers) to the main headquarters. Parenthood information comes from firm-conducted survey, where non-response reduces the sample size more appreciably. Standard errors are clustered by team. *p<0.1; **p<0.05; ***p<0.01.

C Predicting Comment Quality: Crowdsourced Comment Evaluation & Supervised Machine Learning Predictions

We predict the quality of each comment along multiple dimensions. To do this, we first crowdsource labels of about five thousand comments by employing a set of external evaluators to rate a random subset of comments. We then use a supervised machine learning algorithm to scale this approach and generate predicted labels for the nearly two hundred thousand comments in our dataset.

III.A Crowdsourcing Comment Evaluation

We asked external evaluators to rate the quality of a random subset of comments along several dimensions. We recruited the evaluators through Upwork, selecting workers whom Upwork flagged as being top in the programming languages used by the firm. All the evaluators worked as software engineers, knew the programming languages used by the firm (e.g., PHP or Java), and had both written and received code reviews. For each comment, the engineers were asked to imagine that they had received the comment on a piece of code that they had written. They were then asked to respond to the following questions:

- Would you find this comment helpful?
- Do you think you would change your code because of this comment?
- Does this comment suggest actionable steps to change your code?
- Does this comment explain the reason for changing your code?
- Is the tone of this comment rude?

For the first four questions, the crowd-sourced engineers could answer "yes," "no," or "not enough information." For the question about tone, they could answer "No," "A little bit," "Moderately," "Very," or "Not enough information."

A total of 5,377 comments were evaluated by 22 software engineers. Comments were selected at random, stratifying by pre-post period, one- versus multi-building teams, and engineer gender. Each comment was stripped of any firm-specific content (e.g., the name of the firm) or code that may contain sensitive information.

For any particular dimension, engineers said they did not have enough information to rate between 4 to 26 percent of the comments. Of the comments that could be evaluated without additional information, 87 percent were considered helpful, 68 percent were deemed to be actionable, 70 percent were seen as likely to result in changing code, 58 percent gave a reasoning for the change, and 85 percent were considered to not be even a little bit rude.

The crowdsourced evaluations were provided by experienced engineers. Sixty-eight percent worked as software engineers for 5 or more years. All of them had some college experience, and 86 percent had a college degree. These engineers had all written and received code reviews in the past, having received approximately 600 reviews and written approximately 560 reviews on average. Additionally, to verify that the engineers were sufficiently competent to provide meaningful evaluations of the comments, we conditioned their participation upon successfully answering the following technical questions.

- What is the time complexity of the following Python function that finds the maximum element in a list?

  ```
  def find_max_element(lst):
      max_element = lst[0]
      for element in lst:
          if element > max_element:
              max_element = element

      return max_element
  ```

  - O(1)
  - O(n)
  - O(log n)
  - O(n^2)

- Suppose you have an array of integers in ascending order. You need to find a target element in the array and return its index. If the target element is not present in the array, you should return -1. Which of the following algorithms would be most appropriate for this task?

  - Linear Search
  - Binary Search
  - Depth-First Search (DFS)
  - Breadth-First Search (BFS)

- Which of the following data structures is typically used to implement a Last-In-First-Out (LIFO) behavior?

  - Linked-List
  - Queue
  - Hash Table
  - Stack

We included five overlapping comments to calculate measures of inter-rater reliability.

III.B Supervised Machine Learning Prediction

To scale up from the labeled comments, we use a supervised machine learning algorithm, specifically a gradient-boosted decision-tree algorithm (Chen and Guestrin, 2016). In our setting, the predictors are the total character length of the comment, its number of words, and a vector of all the distinct words that appear in the comment, after dropping words like prepositions and pronouns (aka stop words) and those that appear in less than 1% of the training comments. These omissions help to reduce dimensionality.

Gradient-boosted decision-trees start with a simple decision tree and then iteratively refine it. For example, this approach could start with a simple decision tree that says that a comment with the word "nitpick" is unlikely to be helpful. The algorithm will then iteratively build on itself, taking the residuals from the original tree as the new object of the prediction. This iteration could note that, for example, when "nitpick" occurs with a substantive word like "model" or "data," it often is helpful. In this way, gradient boosting can iteratively arrive at relatively strong predictors from simple building blocks. We specifically limit the initial trees to a depth of three and iterate the model a hundred times.

We evaluate the accuracy of the model, using a hold-out sample of 20% of the labeled comments. Table A.16 summarizes the results of these validation exercises.

Table A.16: Summarizing Prediction Accuracy

Attribute	Accuracy	Uninformed Benchmark	Inter-rater Reliability
Helpful	77.7%	76.1%	70.1%
	[75.1%, 80.1%]		
Explains reasoning	68.2%	51.1%	59.9%
	[65.4%, 71%]		
Actionable	69.4%	60.3%	60.5%
	[66.6%, 72.1%]		
Likely to change code	63.9%	51.9%	51.7%
	[60.9%, 66.7%]		

Notes: This table examines the accuracy of our prediction models, using the 20% of the labeled data that is held out as a test sample. To generate the labeled data, we employed software engineers outside our firm to evaluate a random sample of 5,377 comments on multiple dimensions (see Appendix III.A for details on recruitment and sample validation). These raters assessed each comment along multiple dimensions, including whether the comment (i) was helpful, (ii) explained the underlying reasoning, (iii) was actionable, and (iv) was likely to change the code. To scale this approach, we used a supervised machine learning algorithm to generate predictions on the likely label of all 174,014 comments in our main sample. Specifically, we used the XGBoost algorithm (Chen and Guestrin, 2016). To evaluate the accuracy of this approach, this table compares the accuracy of the predictions in the test sample to an uninformative benchmark that always predicts the most common label and an alternative benchmark based on the inter-rater reliability of the raters themselves.

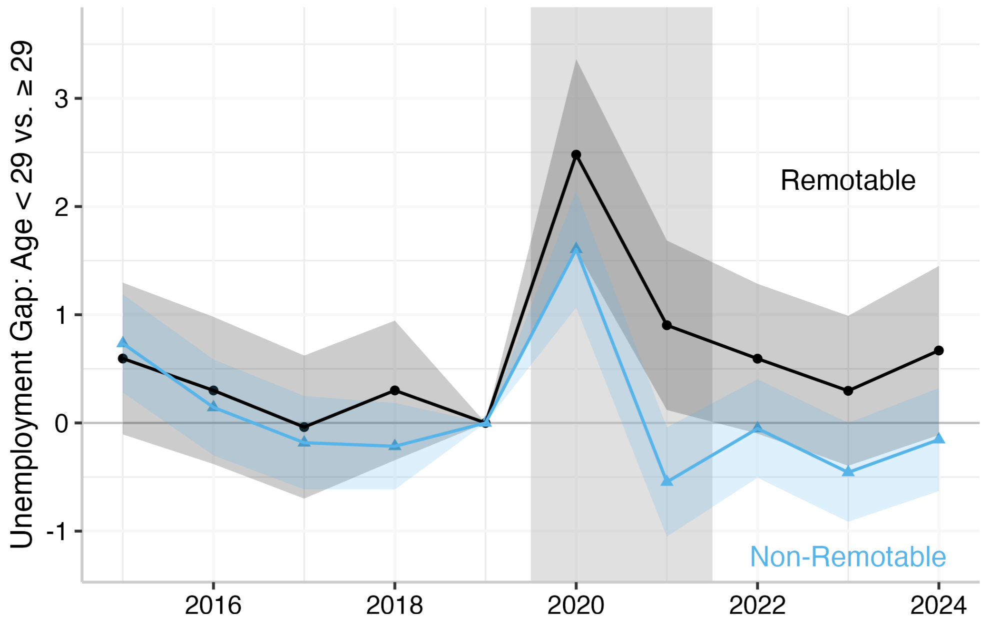
D Unemployment Trends for Non-College Graduates

Section IX focuses on the recent labor market challenges of young college graduates — and how much they can be explained by the rise of remote work. This section instead focuses on non-college graduates.

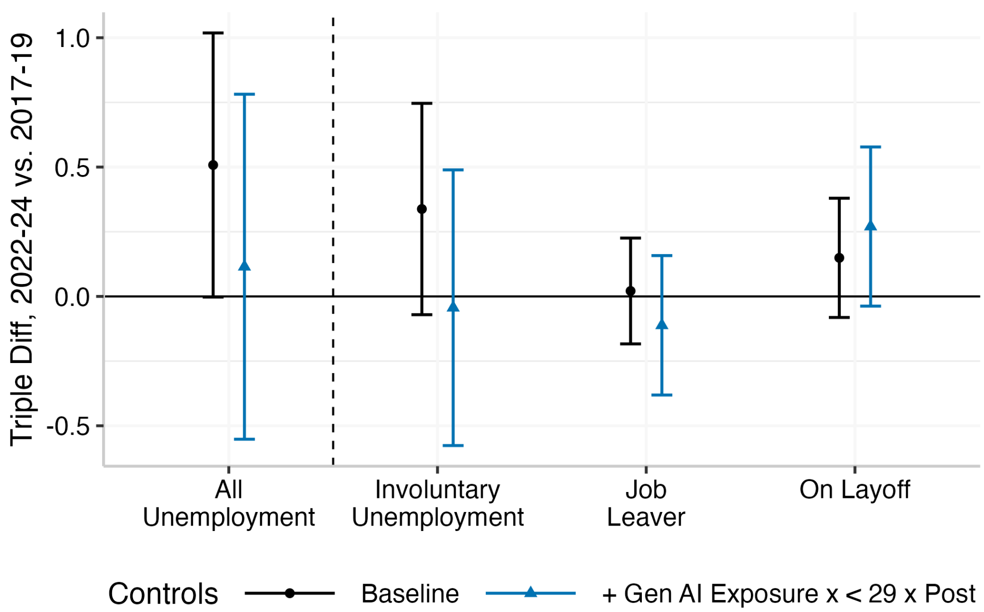
For non-college graduates, young workers in remotable jobs have also experienced increases in unemployment compared to other non-college workers. This difference is less robust to controlling for generative AI exposure, potentially because some more routine jobs, like those of customer service representatives, have been the first to be automated by generative AI (Figure A.18). In the aggregate, young non-college graduates have been much less impacted by the rise of remote work than their college-educated peers because they are much less likely to be in remotable jobs — 22% versus 61%, respectively. Thus, the rise of remote work can also help explain the divergence in unemployment rates between young college graduates and non-college graduates (Column 1−2 of Table A.17a). Controlling for differential changes in young people's unemployment rates across remotable and non-remotable jobs meaningfully narrows the divergence in unemployment rates between college grads and non-college grads (Column 3 of Table A.17a).

Figure A.18: The Rise of Remote Work and Young People's Unemployment Among Non-College Graduates

a: Youth Gap in Unemployment by Remote-Work Potential Over Time



b: Comparing Change in Youth Gap in Remotable vs. in Non-Remotable Jobs



Notes: This figure replicates Figure X but focuses on non-college graduates. Data is from the Current Population Survey (Flood et al., 2024), limited to non-college graduates between 18 and 64. Standard errors are clustered by survey respondent.

Table A.17: College Gap in Unemployment and Remote Work

(a): Changes among Young People (< 29 years old)

	% Unemployed			
	(1)	(2)	(3)	(4)
College Educated x Post	0.82***	0.78***	0.53***	0.54***
	(0.20)	(0.18)	(0.20)	(0.20)
Remotable x Post			0.61***	0.41*
			(0.19)	(0.23)
Gen AI Exposure (Z) x Post				0.16
				(0.11)
Post	−0.20	−0.13	−0.25**	−0.20
	(0.12)	(0.11)	(0.12)	(0.13)
Dependent Mean	6.34	5.44	5.44	5.44
# Workers	219,732	214,360	214,360	214,334

(b): Changes among Older People (≥ 29 years old)

	% Unemployed			
	(1)	(2)	(3)	(4)
College Educated x Post	0.0000	0.01	−0.02	−0.01
	(0.07)	(0.07)	(0.07)	(0.07)
Remotable x Post			0.07	0.01
			(0.07)	(0.09)
Gen AI Exposure (Z) x Post				0.04
				(0.04)
Post	−0.10**	−0.11**	−0.11*	−0.09
	(0.05)	(0.05)	(0.06)	(0.06)
Dependent Mean	2.91	2.87	2.87	2.87
# Workers	686,046	681,486	681,486	681,423
R^2	0.003	0.003	0.003	0.003

Notes: This table analyzes differential changes in unemployment across college-educated and non-college-educated workers and the contribution of remote work to these differences. Panel (a) focuses on workers under 29 and Panel (b) on workers at least 29 years old. All specifications controls for baseline differences. Column 1 includes all college and non-college people in the sample. Column 2 limits to those who have been recently employed, so their prior occupation is known. Column 3 includes information on remotable work, using Dingel and Neiman (2020)'s classification. Column 4 includes information on occupations' generative AI exposure from Eisfeldt, Schubert and Zhang (2023), converted to z-scores for interpretability. Data is from the Current Population Survey (Flood et al., 2024), limited to those under the age of 64 and at least 18 for non-college workers and at least 22 college-degree holders. Standard errors are clustered by survey respondent. *p<0.1; **p<0.05; ***p<0.01.