The Power of Proximity to Coworkers

Training for Tomorrow or Productivity Today?[1]

Natalia Emanuel[2] · Emma Harrington[3] · Amanda Pallais[4]

November 10, 2023

Abstract

Amidst the rise of remote work, we ask: what are the effects of proximity to coworkers? We find being near coworkers has tradeoffs: proximity increases long-run human capital development at the expense of short-term output. We study software engineers at a Fortune 500 firm, whose main campus has two buildings several blocks apart. When offices were open, engineers working in the same building as all their teammates received 22 percent more online feedback than engineers with distant teammates. After offices closed for COVID-19, this advantage largely disappears. Yet sitting together reduces engineers' programming output, particularly for senior engineers. The tradeoffs from proximity are more acute for women, who both do more mentoring and receive more mentorship when near their coworkers. Proximity impacts career trajectories, dampening short-run pay raises but boosting them in the long run. These results can help to explain national trends: workers in their twenties who often need mentorship and workers over forty who often provide mentorship are more likely to return to the office. However, even if most mentors and mentees go into the office, remote work may reduce interaction: pre-COVID, having just one distant teammate reduced feedback among co-located workers.

Keywords: Remote work, on-the-job training, peer effects, telecommunication, gender, inclusion, worker retention
JEL: J24, M15, M53, M54, J16, O33, R23

---

[1]This project would not have been possible without the curiosity and commitment to research of our colleagues at the firm who shared data: Lauren and Susan. 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 seminar participants at Boston University Questrom, the Federal Reserve Bank of Minneapolis, Federal Reserve Bank of Boston, Federal Reserve Board of Governors, Google Diversity Group, Harvard, Microsoft Research, NABE Conference, NASA Langley Research Center, NBER Summer Institute, SOLE Conference, Stanford Remote Work Conference, University of Chicago, University of Illinois at Urbana-Champaign, University of Iowa, University of Maryland, University of Michigan, University of Southern California, University of Virginia, Wharton, and the Yale Cowles Labor/Public Conference. 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 or the Federal Reserve System.

[2]The Federal Reserve Bank of New York (natalia@nataliaemanuel.com)
[3]University of Virginia (emma.k.harrington4@gmail.com)
[4]Harvard University and NBER (apallais@fas.harvard.edu)

Until recently, "office work" was primarily done in the office. Even after the advent of new communication technologies — such as Skype (in 2003) and Zoom (in 2013) — 94 percent of Americans worked in the office on most work days (U.S. Census Bureau, 2019). The COVID-19 pandemic severed many workers' ties to the office, and many have yet to return (Barrero et al., 2023; Hansen et al., 2023). But it is still unclear why the office was so central for so long and how the seismic shift in workers' relationship with the office will affect work and workers.

In the wake of COVID-19, firms were sharply divided about the value of the office. In 2020, Mark Zuckerberg, CEO of Meta, reflected that "a lot of people are actually saying that they're more productive" working from home (Newton, 2020). On the other hand, James Gorman, CEO of Morgan Stanley, found the office central: "The office is where we teach, where our interns learn. That's how we develop people" (Kelly, 2021). Could both CEOs have been right? Could working in the office facilitate investments in workers' skills for tomorrow that diminish productivity today?

We study the impact of sitting together in the office for software engineers at a Fortune 500 online retailer.[1] This firm gave us access to the online feedback that engineers write about each other's computer code as well as metrics of engineers' programming output. We find that sitting near coworkers increases the online feedback that engineers receive on their computer code. Engineers ask more follow-up questions online when sitting together, and so, proximity can not only increase in-person but also digital communication. Proximity is particularly integral to the online feedback received by young and less tenured engineers. Yet mentorship of junior engineers is not free. Engineers — particularly those with more experience — write more programs when *not* sitting near their junior colleagues. Both of these impacts on mentors and mentees are more pronounced for female engineers. The

[1] Software engineers compose an important segment of the labor market, accounting for 5 percent of labor income in 2020. Software engineering is also highly remotable, with 47 percent of engineers working remotely in 2020. Among those working remotely full-time in 2020, software engineers accounted for 11 percent of labor income (and 8 percent of employment).

intertemporal tradeoff from proximity is reflected in workers' pay path: sitting near coworkers leads to fewer early pay raises but increases pay raises in the long run as workers build more human capital. The returns to training partially accrue outside the firm: engineers trained near teammates are more likely to quit for higher-paying jobs elsewhere.

Our results suggest that working from home (WFH) has divergent effects over different time spans, as short-term gains come partly at the cost of workers' long-run development. Consistent with this, Mark Zuckerberg became less optimistic about WFH's consequences. Looking at longer-term performance data from Meta, he came to believe that the costs of WFH exceeded its benefits for junior workers (Zuckerberg, 2023). Beyond software engineering, the tradeoffs from remote work may be more extreme with fewer established systems for digital mentorship.[2]

To understand the dynamic tradeoffs from WFH, we build a simple two-period model of mentorship where each worker is first junior and then senior. Each junior engineer is paired with a senior engineer, who sits either nearby or at a distance. The junior engineer can ask their senior counterpart for mentorship, which is hard for the firm to observe and therefore reward. Mentorship takes time for both the senior engineer — who makes additional suggestions — and the junior engineer — who figures out how to incorporate the suggested changes. As a result, mentorship comes at the cost of short-term output. Senior engineers find mentoring taxing but refusing to mentor even more costly, and so they provide mentorship when asked. Junior engineers value mentorship but find it costly to ask for it, especially when sitting apart. Thus, junior engineers ask for and receive less feedback when remote.

In our model, sitting near a colleague increases mentorship but reduces output to-

[2]Software engineering may be well-suited to remote work for several reasons. First, software engineers produce digital output, unlike, for example, physicians or mechanical engineers. Second, it is industry-standard for software engineers to use the Agile meeting system, which mandates daily, real-time meetings even when remote. Third, it is also industry-standard to mandate feedback through established, online systems (like Github). Other occupations that do not meet regularly or give feedback online may see larger declines in mentorship when remote.

day. However, *having* sat with a colleague in a prior period increases engineers' human capital, which impacts their career outcomes. When the offices are open, junior engineers who sit near their teammates will have lower pay. But, once the offices closed and there was no more differential investment, engineers who had sat on one building teams will see higher pay.

Empirically, we assess the effects of sitting together for software engineers. This setting features data on both mentorship and output. We observe measures of mentorship, since it is the industry standard for software engineers to write online peer-reviews of each other's code before it is deployed. Reviewers aim to ensure computer code is free of bugs and other glitches while teaching engineers to write better code in the future.[3]

At the firm, engineers varied in their proximity to one another before COVID-19. The firm has two buildings on its main engineering campus, several blocks apart. Prior to COVID-19, some teams were assigned desks all in one building, while others spanned the buildings. Engineers who worked on software for internal processes were particularly likely to be spread across buildings since it can be helpful to sit near teammates as well as non-engineers who used the tools they built (e.g., in sales or finance). Among engineers working on similar software, whether or not their team was all in one building was due in part to desk availability when their teammates were hired. Desk positions altered team dynamics. When the offices were open, the 637 engineers on one-building teams met in-person daily, while the 418 engineers on multi-building teams usually held short, daily meetings online.[4] As a result, multi-building teams operated more like remote teams even when the offices were open.

To identify the causal effects of proximity, we first evaluate the differences between

[3] One manager told us "We ask senior, technical folks...to make their code reviews a learning opportunity by, for example, including the reasoning behind suggested changes."

[4] 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."

one- and multi-building teams when the offices were open. Second, we conduct a difference-in-differences analysis, utilizing the fact that engineers on one-building teams saw bigger changes in proximity when the offices closed than engineers on multi-building teams, who were already functioning more like remote teams before COVID-19. This design relies on a parallel-trends assumption: namely that engineers on one- and multi-building teams were similarly shocked by the pandemic itself. Reassuringly, among those working on similar software, engineers had broadly similar characteristics regardless of their proximity to their teammates. Further, while COVID-19 did cause a surge in online retail, engineers at the firm reported that the uptick in demand did not change the nature or intensity of their work. Our results are robust to allowing for differential effects of the pandemic for engineers working on different software and with different observable characteristics.

We find that proximity increases mentorship. While offices were open, engineers on one-building teams received 22 percent more comments on their code than engineers on multi-building teams (p-value = 0.0003). Once the offices closed and everyone was fully distributed, the gap largely disappeared.[5] Sitting near teammates primarily affects feedback *received* by junior engineers and *given* by senior engineers with more experience at the firm. While we cannot directly observe how feedback affects code quality, external engineers rated most comments as helpful, actionable, and likely to lead to changes in the code. Proximity did not significantly change these ratings, suggesting that proximity leads to additional substantive feedback from senior engineers to junior ones.

We find that engineers sitting near their teammates receive more feedback, partially because they ask more follow-up questions during code reviews. This greater comfort with asking for additional clarifications highlights how face-to-face interaction complements — rather than solely substitutes for — online communication. If sitting together also facilitates face-to-face conversation about code, our analysis of

[5] We find similar effects for the total length of code reviews and mentions of other online conversations (e.g., on Slack).

online feedback provides a lower bound on proximity's total effect on mentorship.

In a placebo check, we find that engineers who sit near all their teammates do not receive significantly more feedback from engineers outside their teams. This suggests that co-located engineers do not simply need more feedback which would affect comments from all sources. Furthermore, two complementary designs (with different identifying assumptions) show similar effects of proximity. First, we find that proximity to non-teammates increases feedback from non-teammates. Second, we find that engineers with teammates outside the main campus receive less feedback when the offices were open, but this gap closes with the offices.[6]

We find that distant teammates impose negative externalities on the mentorship of teammates sitting together. These externalities can explain about a third of proximity's impact. Furthermore, before COVID-19, adding a new hire in another building reduces feedback among proximate teammates (who predate the new hire), while adding a new hire in the same building has no such impact. Teams' attempts to accommodate distant teammates by, for example, moving in-person meetings online, have substantial negative externalities.[7]

Our findings indicate that additional mentorship has an opportunity cost: engineers who sit near all their teammates write fewer programs. Our difference-in-differences estimate suggests that proximity reduces programs written per month by 23 percent (p-value $= 0.008$), with similar effects on total lines of code and total files changed. The effects on output are present for both junior and senior engineers but are particularly pronounced for senior engineers, who do most of the mentoring.

We find that the tradeoffs from proximity are more acute for women. Before the of-

[6]Our main results are limited to workers whose teammates all work on the main campus pre-pandemic, but other engineers did work in other campuses or from home.

[7]This complements existing research that finds that introducing remote work increased absenteeism among coworkers who stayed in the office (Linos, 2018). These spillovers may arise not only because remote work changes how remote workers interact with their on-site colleagues, but also because it changes how on-site colleagues interact with one another.

fices closed, female engineers who were in the same building as all their teammates received 40 percent more feedback than female engineers with distant teammates — twice the difference as for male engineers. When the offices shut down for COVID-19, lost proximity mattered more for women: the triple difference indicates a differential decline of 21 percent. These effects are largely driven by follow-up questions and clarifications, suggesting that women feel more comfortable asking for additional feedback in-person.[8] At the same time, senior female engineers do much more mentoring when they are seated near their colleagues, leading to a larger negative effect of proximity on the programs they write.

Proximity affects workers' career outcomes. When offices are open, junior workers on one-building teams are 5 percentage points less likely to receive a pay raise, consistent with their lower output (p-value = 0.032). However, once the offices close — and mentorship equalizes — these engineers benefit from the mentorship that they have received and are 7 percentage points more likely to receive a pay raise (p-value = 0.066). Both of these differences are suggestively larger for female engineers.

Quits also reflect the impact of proximity. Before COVID-19, quits were relatively rare. However, with the rise of remote work, quits increased as it became easier to move to higher-paying tech firms in Silicon Valley without relocating from this firm's east-coast city. Notably, workers who had been trained on one-building teams saw a 1.2 percentage point greater increase in quits, about twice that of engineers trained on multi-building teams (p-value of difference = 0.01). Moreover, engineers on one-building teams were more likely to move to roles at other firms with higher salaries (according to Glassdoor). These results are consistent with the greater training on one-building teams giving engineers the skills they need to secure higher-paying jobs elsewhere. As with pay raises, the effects are larger for women. We do

[8]Proximity's larger impact for women does not seem to be driven by men being more likely to over-explain feedback when nearby. The effect comes from both male and female commenters. Furthermore, external engineers were more likely to rate comments received by female engineers as helpful, actionable, and likely to lead to code changes on both one- and multi-building teams. External raters rarely viewed comments received by either male or female engineers as rude.

not see the same impacts on firings, which while insignificant, suggest that workers on one-building teams were less likely to be fired once the offices close.

Finally, we examine who works at the office versus from home. Before COVID-19, decisions about work locations were consistent with the firm believing that the long-run benefits of proximity outweighed the short-run costs. The most junior workers — who receive the most training – and the most senior workers and managers — who provide the most training — were more likely to be office-based. This pattern aligns with national trends in 2022−2023, where both workers in their twenties and workers over forty are the most likely to have returned to the office with those in their thirties more likely to stay at home even if they do not have children (U.S. Census Bureau, 2023). Additionally, once the offices close, we see that the firm is less likely to hire very junior engineers and, instead, opts to hire workers with more prior training. While this change could be influenced by many factors, it is consistent with the idea that when the firm faces challenges in training workers, it decides to "buy" talent instead of "building" it.

Our study contributes to the remote work literature, potentially resolving a puzzle as to why remote work was rare before the pandemic (Mas and Pallais, 2020) despite workers' high willingness to pay for remote work (Mas and Pallais, 2017; Maestas et al., 2018; He et al., 2021) and remote work's positive impacts on productivity in some settings (Bloom et al., 2015; Choudhury et al., 2020) and modestly negative impacts in others (Dutcher, 2012; Gibbs et al., 2023; Emanuel and Harrington, 2023).

Our paper is particularly related to studies that look at the impact of coworker proximity on collaboration. Battiston et al. (2021) finds that when the worker who processes incoming 911 calls is co-located with the dispatcher who sends police to the scene, the two communicate more and the police arrive sooner. Yang et al. (2022) show that remote work reduced the breadth of workers' communication networks at Microsoft, while DeFilippis et al. (2020) find that the COVID-19 closures reduced the depth of communication, as people had fewer long meetings with few participants. In a lab experiment, Brucks and Levav (2022) find videoconferencing inhibits idea generation.[9] Finally, Atkin et al. (2019) find there are large negative impacts on productivity when data-entry workers in India are randomized to work from home, which grow over time, suggesting remote work leads to less learning.

Our paper also adds to the growing literature which quantifies the importance of coworkers in on-the-job learning. Patenters and teachers who work with more productive peers perform better later in their careers (Akcigit et al., 2018; Jackson and Bruegmann, 2009), as do sales workers who seek advice from their coworkers (Sandvik et al., 2020).[10] More generally, working in a firm with coworkers with higher wages or more education is strongly correlated with higher subsequent wage growth (Herkenhoff et al., 2018; Jarosch et al., 2021; Nix, 2020). Yet it is unclear whether physical proximity *per se* is necessary for these spillovers or instead whether being in the same firm, school, or intellectual community would suffice even at a distance. In academia, Azoulay et al. (2010) and Waldinger (2012) find that physical distance is less important than intellectual distance in determining spillovers, while Boudreau et al. (2017) and Catalini (2018) find that sitting in the same building significantly increases the likelihood of coauthorship. Our paper finds a large role for physical proximity among coworkers.

Our paper also contributes to the literature assessing the career consequences of face-time. Bloom et al. (2015) find working in the office substantially increases promotion rates conditional on productivity. Cullen and Perez-Truglia (2023) find face-time with managers substantially boosts promotions. Such findings raise concerns about "proximity bias" of managers (Tsipursky, 2022). We provide evidence that

[9] In another lab experiment, Dutcher and Saral (2022) find that individuals were more pessimistic about the productivity of remote workers, which undermined team production.

[10] A related literature studies the impacts of contemporaneous peer effects on productivity. While grocery store clerks (Mas and Moretti, 2009), envelope stuffers (Falk and Ichino, 2006), and fruit pickers (Bandiera et al., 2010) are all significantly more productive if they work near faster peers, Cornelissen et al. (2017, 2023) estimate small contemporaneous impacts of coworkers on workers' wages in the economy overall. Having highly-qualified peers need not always have positive effects on contemporaneous career outcomes if, for example, competition with highly qualified peers reduces current wages (Johnsen et al., 2023).

in-person work may accelerate career progression by helping workers build skills as well as make connections.

Finally, the paper contributes to the urban literature that has investigated whether information and communication technologies will complement or substitute for proximity. With the rise of the internet, many predicted the death of distance (Cairncross, 2001; Friedman, 2005). Yet urban economists have long noted the possibility that online technologies would complement rather than substitute for physical proximity (Gaspar and Glaeser, 1998). Agrawal and Goldfarb (2008) find internet connectivity tended to increase collaboration between researchers at physically proximate universities, while Chen et al. (2022) show that the costs of having a distributed research team have fallen over time with the rise of better communication technologies. We find proximity and digital communication are complements even after recent technological advancements.

The next section presents our model. Section II describes our data and setting, and Section III details our empirical strategy. We show that physical proximity increases mentorship in Section IV and that proximity reduces contemporaneous output in Section V. We show that proximity's tradeoff is more pronounced for women in Section VI and that it translates into career outcomes in Section VII. Finally, we analyze work-from-home decisions pre-pandemic in Section VIII. Section IX concludes.

I Model

In our overlapping generations model of mentorship, workers live for two periods: a junior period and a senior period. Each junior worker (j) is paired with a senior worker (s); the pair can be seated nearby (n) or apart (a). Senior workers are required to give junior workers feedback, but each period, junior workers choose whether to ask for additional feedback, which we call mentorship. If asked, the senior worker chooses whether to provide mentorship, which is unobservable to the firm.[11]

[11]While firms can observe whether senior workers provide any feedback to junior colleagues, they cannot fully observe the time spent or quality of mentoring, just as universities may be able to ob-

Each period t, workers produce output with quantity y_{it} (which we observe in the data) and quality q_{it} (which we do not observe).[12] The value of their output is $y_{it}q_{it}$. Without receiving or providing mentorship, workers produce output with quantity $\tilde{y}_{it} = y + \epsilon_{it}$ and quality $\tilde{q}_{it} = q + v_{it}$ where $\epsilon_{it} \in [\epsilon_l, \epsilon_h]$, and $v_{it} \in [v_l, v_h]$. Workers know ϵ_{it} and v_{it} for both periods when making their choices; these shocks are unobservable to the firm.[13]

Giving and receiving mentorship both take time and so reduce current-period output. Providing mentorship reduces seniors' quantity produced by m_s. Receiving mentorship reduces juniors' quantity produced by m_j as they, for example, respond to comments and internalize feedback. Receiving mentorship increases the quality of workers' future output, increasing next-period quality by b.

Asking for mentorship is costly for junior workers and costlier if workers are not seated together.[14] Asking for additional feedback costs c_{jn} when seated near the senior worker and c_{ja} when seated apart, where

$$c_{jn} < b(y + \epsilon_l) - m_j(q + v_h) < b(y + \epsilon_h) - m_j(q + v_l) < c_{ja}. \tag{1}$$

That is, the net benefits of receiving feedback for juniors exceed the cost of asking for feedback only when juniors are seated near their colleagues.

Senior workers have no returns to providing additional feedback. But, if asked, they face a utility cost of rejecting the request: $c_s > m_s(q + v_h)$, stemming from social desirability and office norms.

serve how many students faculty advise but not the time spent or quality of advising.

[12]For example, quality includes how quickly the code runs, how it handles edge cases, and whether it has bugs.

[13]The noise in output prevents the firm from perfectly inferring and compensating mentorship. Since allowing the firm to imperfectly reward mentorship does not affect our conclusions, we abstract away from this. Workers' decisions are unchanged if they are also uncertain about the shocks, but their foresight simplifies the exposition.

[14]Asking for mentorship may be less costly in person because, for example, a junior worker can better time requests so that they are convenient for the senior worker. Alternatively, asking in person may seem more natural and less demanding.

At the end of the period, y_{it} and q_{it} are observed. Each worker is paid her marginal product $y_{it}q_{it}$. There is no discounting.

The resulting equilibrium is straightforward. Junior workers will ask for additional feedback only when seated nearby (given Equation 1). When asked, senior workers will agree (since $c_s > m_s(q + v_h)$).

The model yields several predictions.

1. Sitting together increases mentorship that is asked for and then received.
2. Sitting together decreases the quantity of current-period programming output for junior and senior workers.
3. Sitting together decreases the pay of junior workers.
4. Senior workers who previously sat with their colleagues should have higher pay conditional on current seating location.

Mentorship is valuable for recipients who request it, so proximity increases the well-being of potential mentees. However, mentorship is costly for potential mentors. Whether mentorship is positive on-net depends on whether mentorship's benefits for juniors exceed its costs for seniors. The model does not take a stand on this but we return to this question in Section VIII.

Gender. We extend the model to allow for gender differences. A fraction $\lambda > 0$ of men face neither a cost of asking for mentorship nor a cost of rejecting mentorship; all others face the costs defined above.

Rejection costs are observable, so senior men without rejection costs do not receive mentorship requests, regardless of location. Senior women and men with rejection costs mentor more when sitting nearby than when seated far from their coworker because they are more likely to be asked. Junior men without asking costs always ask for mentorship, regardless of location. Junior women and men with asking costs ask for mentorship only if they are seated near their coworker.

This implies that, for women, proximity leads to

5. larger increases in receiving and asking for mentorship,
6. larger increases in providing mentorship, and
7. larger decreases in output.

Quits. We next extend the model by giving workers a choice to quit between their junior and senior periods. There is a group of superstar tech firms (e.g., Google) at which productivity is more sensitive to skill and training. Specifically, worker i's marginal product at a superstar firm is $(1+\sigma)(y_{it}q_{it}) - \bar{f}$ where $\sigma, \bar{f} > 0$. The term $\bar{f}$ represents super-star firms' greater investment in IT that augments productivity but has fixed costs per engineer. As a result, only highly-productive workers produce more on net at the superstar firms. This structure assumes that productivity ($y_{it}q_{it}$) translates across firms and mentorship provides general skills training.[15]

We assume that superstar firms can observe the worker's senior-period y_{it} and q_{it} through technical interviews. They then decide whether to make offers to workers and, if so, at what wage. Workers compare their outside and inside wages and decide whether to quit. Because there are several superstar firms, firms will offer workers their marginal product.

Workers face a utility cost of leaving the current firm: l. Before COVID-19, the cost of switching firms (l_0) was high because it often involved a cross-country move. After tech offices closed for COVID-19, l plummeted to ($l_1 < l_0$) since a physical move was no longer necessary in the short- to medium-term. We assume $l_0 > \sigma y_{it}q_{it} - \bar{f}$ for all i, but l_1 is sufficiently low that $l_1 < \sigma y_{it}q_{it} - \bar{f}$ for some i.[16]

[15]While, in practice, some of the training is likely to be firm-specific (e.g., the way the firm names its variables), much of it offers general lessons about coding technique that would be valuable across firms (e.g., effective ways to test code). As long as σ is sufficiently large, our predictions about quits hold even if some of the training is firm-specific.

[16]For simplicity, we assume that workers' expectations about how much mentoring they will have to do are the same across firms.

Given the high moving cost before COVID (l_0), no one quits for a better job at a superstar firm. But workers for whom the wage gain of moving to a better firm exceeds the moving cost — $\sigma y_{it} q_{it} - \bar{f} > l_1$ — will move after the pandemic. Since the benefit of moving is increasing in productivity, workers who have been mentored are more likely to move. This implies that

8. Once the pandemic starts, junior workers who previously sat near their coworker are more likely to quit.
9. This difference in quits will be more pronounced for women than men.

II Data and Setting

Our data include peer code reviews of software engineers at a Fortune 500 firm between August 2019 and December 2020. Personnel data identifies each engineer's office building and teammates.[17] We first characterize our sample of engineers and then detail how we measure online feedback in code reviews and proximity to teammates in personnel records.

II.A Characterizing the Sample of Software Engineers

Personnel records from the firm's human resources department provide information on each engineer's job title, hire date, termination date (if applicable), pay rate, age, gender, and parental status (from a June 2020 firm-wide survey).[18]

Software engineering is a predominantly male occupation, both at this firm and more broadly: 81 percent of the engineers in our sample (and 75 percent of programmers nationally) are male.[19] Engineers at the firm tend to be young, with an

[17] We are able to match 99 percent of engineers across the peer-review and personnel datasets.

[18] A third of engineers participated in the June 2020 parenthood survey, with a comparable 30 percent in one-building teams and 35 percent in multi-building teams.

[19] For the firm statistic, gender information comes from human resources data; for the national statistic, data comes from US Census data, weighted to account for sampling (U.S. Census Bureau, 2019). Both sources use respondents' self-reported gender. In the Census, we define software engineers as (1) Computer Scientists and Systems Analysts, Network systems Analysts, and Web Developers (occupation 2010 code = 1000), (2) Computer Programmers (1010), or (3) Software Developers, Applications and Systems Software (1020).

average age of 29 compared to 40 nationally. Consistent with their youth, only 17 percent of the firm's engineers are parents.

The engineers we study handle typical tasks for software engineers at online retail firms. Some teams maintain the front-end interface for the website, while others maintain the website's back-end database that determines the products displayed in search results. Finally, a third group develops internal tools for the firm's supply chain and sales/service teams. The supply-chain tools help ensure that products can be efficiently located in warehouses and shipped to customers' homes. The sales/service tools help call-center agents track purchases and resolve issues like damaged or delayed deliveries.

II.B Programming and Code-Review Data

Our data includes coworkers' comments on the code that runs the firm's front-end website, back-end databases, and internal processes. We only have data from the firm's primary code-base, not other, smaller code-bases that handle more specialized tasks (e.g., applications for the firm's retail stores).[20] To maintain code quality, every piece of code is reviewed by at least one other engineer before it is committed to the code-base. This is standard practice in software engineering.

Our data describes the initial piece of code — including its author, its time-stamp, how many files it changed, and how many lines were added/deleted — and every peer comment — including its author, text, and time-stamp. The 1,055 engineers in our main analysis wrote 29,959 pieces of code and received 174,424 peer comments. Unfortunately, our data does not include measures of code quality, such as whether a program introduced a bug or had to be rewritten.

Programming Output. We can use this dataset to measure an engineer's monthly contributions to the main code-base. On average, engineers submitted two pro-

[20]We study only the engineers who work on the primary code-base, not those who work on more specialized tasks. Our sample includes approximately half of the engineers at the firm; we do not report the precise share to protect the firm's anonymity.

grams per month to this code-base, each of which changes nearly 500 lines of code and affects seven different files. The typical engineer also submits code to other more-specialized code-bases which are outside the scope of our data.

Our preferred measure of programming output is the monthly number of programs submitted to the main code-base. We also use alternative measures such as the total lines of code written and the number of files changed. To reduce the influence of outliers, we winsorize programming-output outcomes at the 95th percentiles.[21]

Peer-Review Process. Before each program is committed to the code-base, it is peer reviewed. Engineers typically receive feedback from one commenter but sometimes receive feedback from multiple commenters, who have different expertise (e.g., on the programming language versus the part of the code-base). Typically the commenter is a more experienced engineer than the program writer, either in terms of age or tenure at the firm (Figure A.1). It usually takes nearly a day (sixteen hours) to receive the first comment on the engineer's program.

Reviewers' comments often aim to improve a program's reliability or clarity and give engineers general advice to improve future coding. Reviewers average six comments per program, each of which averages eighty characters. The number of comments per program is our main measure of mentorship. While the firm can observe the number of comments, it does not reward writing more comments since such an incentive system would generate pointless comments. The metric is consequently useful as an undistorted lens into feedback to other programmers.

We use the text of the comments to draw out common themes. Principal component analysis identifies grouping of words that frequently appear together in comments (Appendix Section I.A details the approach). Two of the top components are about verifying that programs are working as expected (see Figure A.2(a) for word clouds

[21] This is useful because some changes, for example, changing a commonly-used variable's name, can lead to many lines and files being changed without representing a substantive change in the code-base.

and Table A.1 for illustrative examples of the top components). One component identifies comments that are about function output, which often concern edge cases such as empty values. Another component identifies comments about how to test code, which often requires the programmer to clearly articulate the code's expected behavior and rewrite the code to have separate, testable components.

Comments are generally helpful and actionable. Software engineers outside our firm evaluated a random subset of comments along several dimension (Appendix I.B details our methodology). For comments that the external engineers had enough information to evaluate, 87 percent of comments were considered helpful, 70 percent were rated as likely to cause the engineer to change the code, and 58 percent explained their reasoning (see Table A.2 for a random subset of comments along with their evaluations). Moreover, only 15 percent were considered rude.[22] Comments received by female engineers and engineers with less experience in the firn were especially likely to be rated as helpful and consequential for the code, while being marginally less likely to be rated as rude (Tables A.3-A.4).

Peer reviews often involve a back-and-forth conversation between the commenter and the engineer. For the typical review, a commenter gives an initial set of comments on the code. The engineer can then respond to these comments to ask for further clarification or check whether the changes that she made were sufficient. The commenter often then replies with a clarification, additional feedback, or acknowledgement of the changes. Seventy-one percent of reviews have a back-and-forth between the commenter and coder. Below is a relatively non-technical example of such an interchange where the back-and-forth helps the coder clarify what was missing in her program and what could be improved in subsequent code:

> *Commenter at 3:14pm:* Can you please add testing details to this program?

[22] We considered a comment rude if it was rated as "Very," "Moderately," or "A little bit" rude.

> *Coder at 3:32pm:* What do you mean by testing details? I added more information on the description if that helps. Let me know if you need further information.
>
> *Commenter at 3:40pm:* [I meant] what you did to validate that your changes are working as expected. Here is an example testing doc I made for a ticket in the past: {link to example}.
>
> *Coder at 4:18pm:* [I added a] document in description: {link to document}.

Without this iteration on the code, the coder may not have identified and rectified the omission and learned how to write better code going forward.

As in this example, most comments are given during standard work hours (between 8 AM and 6 PM on the weekdays). When the offices were open, 96 percent of comments were given during standard work hours, which declined by only 0.8 percentage points when the offices closed (Figure A.3).[23]

Requesting Feedback. Engineers are responsible for asking coworkers to review their code. While engineers can request feedback in the code-review system, they typically ask first outside the system, either in person or in a direct message (e.g., on Slack). Engineers can ask for feedback from teammates or from engineers outside of their own small five- to six-person teams. Engineers might ask for feedback from a non-teammate with relevant expertise in the part of the code-base, the programming language, and/or the type of problem being solved. About two-thirds of engineers' feedback comes from teammates.

Engineers often ask teammates for feedback before or after daily team meetings. Teams have daily fifteen-minute "stand-up" meetings and longer one- to two-hour meetings each week. The nature and frequency of meetings both follow a set routine

[23]Our results are comparable when limiting to interactions in standard business hours (Figure A.4).

under Agile management, which is common in the industry.[24] Teams use this meeting schedule before and after the office closures and regardless of their proximity to one another. Thus, teammates' proximity does not typically affect the frequency of meetings but often does affect their medium (i.e., Zoom versus in-person).

There are no explicit incentives to give peer reviews, but there are strong norms (and managerial expectations) to give feedback when asked.

II.C Personnel Records

Personnel records detail each engineer's office building and manager.

Identifying Teams. Two features of the firm's organizational structure mean that an engineer's manager allows us to identify her teammates. First, workers at the firm always directly report to a single manager rather than to multiple managers according to one of the firm's human resources directors. Second, managers can only oversee multiple teams once they reach a certain level in the company. We limit the main analysis sample to engineers under mid-level managers. We also limit to teams where all engineers sit in the main campus – either in the main building or the auxiliary building, several blocks away.[25] We exclude the small number of engineers hired after the offices closed in March 2020.

Proximity to Teammates. Because of limited desk availability, some engineers sit in the same building as all of their teammates, while others have at least one teammate in another building. Once one engineer is in a different building, daily meetings are often held online since a ten- to fifteen-minute meeting does not justify a twenty-

[24]Engineers organize their work in two-week sprints in an Agile workflow. The team first meets to plan the work. In daily "scrum" meetings, engineers discuss their progress and what others could do to help, including reviewing code. Each sprint includes a backlog meeting to review outstanding tasks and a retrospective to debrief. Teams also regularly meet to discuss the products being built.

[25]Eighty-five percent of the firm's engineers worked in the main campus. We drop the 7 percent of engineers whose managers and teammates we cannot identify and the 14 percent of engineers who are not managed by mid-level managers at the firm. We limit to the 1,055 engineers whose teammates all worked in the firm's main campus. We separately consider the 215 engineers whose teammates worked remotely or in satellite campuses in Section IV.

minute round-trip walk. As a result, engineers on one-building teams may more easily discuss their work face-to-face than engineers on multi-building teams before, during, and after meetings.

All teams face desk constraints that can make it difficult to place everyone together. This constraint is more likely to bind for teams who work on developing internal tools since it can be advantageous for these teams to sit near the stakeholders who use their tools. Co-locating such a team alongside the team that it serves is numerically more challenging and thus results in more multi-building teams among engineers working on internal tools than in the other engineering groups. When analyzing the effects of sitting near teammates, we account for differences across these engineering groups. We also show the robustness of results to limiting to engineers who work on internal tools.

Before the pandemic, 637 engineers were on teams where all of the members worked together in one building, while the remaining 418 engineers were on teams that spanned the two office buildings. For engineers on multi-building teams, 30 percent of their team — or one to two teammates — were in the other building. We define engineers as being in one-building teams if we always observe them in the same building as all of their teammates during the pre-period from August 2019 through February 2020. During the pre-period, only 14 percent of engineers switched teams, and only 2 percent of engineers switched buildings.[26]

COVID-19 Closures. The office closures due to COVID-19 eliminated differences in coworkers' proximity. On Friday March 6th, most engineers went home from the office expecting to return the following Monday. Almost no engineers came into the office come Monday, though the firm did not officially close the campus

[26]By construction, everyone who is categorized as being on a one-building team was always categorized as being on a one-building team in every month in the pre-period; 75 percent of engineers who we categorize as being on multi-building teams were categorized as being on multi-building teams in every month of the pre-period and 25 percent were categorized as being on a one-building team in at least one month. Our results are similar when we use whether engineers were in one- versus multi-building team in any given month instead.

immediately. Engineers could collect any belongings that they may have left in the office. After the closures, engineers continued to work on the same laptops, VPN into the same systems, and work on the same code-bases as they had before the pandemic. Engineers continued to work from home during the entire post-period in our sample: the return to the office is beyond the scope of our data. Thus, during the entire post-period, all engineers were physically separated from their coworkers.

Compensation. Finally, personnel records detail engineers' salaries as well as their other forms of compensation. Our analyses of compensation focus on engineers' salaries, since salaries constitute over three quarters of total compensation and are most consistently recorded and shared with us by the firm. In addition to base salaries, an average of 6 percent of compensation comes from end-of-year bonuses and another 15 percent comes from stocks. We have a few snapshots that give information on bonuses and equity at several points in time. We find that they are fairly formulaically determined: 76 percent of the variation in bonuses and 90 percent of the variation in equity can be explained by tenure at the firm and initial job level.

There is no explicit performance pay at the firm for programs written or mentorship given. Instead, these dimensions of performance are evaluated in tri-annual performance reviews and then used to determine salary changes. Anecdotally, performance reviews focus more on output than mentorship, which has been a source of frustration for some engineers with whom we spoke.

Quits & Fires. Our personnel records further show workers' quits and firings. For about two thirds of quits, we can see the destination firm and position. We can use these data to infer pay at these destination positions using Glassdoor data.

III EMPIRICAL DESIGN

To identify proximity's impact on mentorship and programming output, we compare engineers on one- and multi-building teams who work on similar software. Because building assignment depended on what desks were free at the time that engineers started, much of the pre-pandemic difference in online feedback is likely due to the causal effect of proximity. Yet some of the differences might reflect unobservable differences between engineers on one- and multi-building teams. To net out unobservable differences, we utilize the building closures of COVID-19, which forced all teammates to work separately. In a difference-in-differences design, we assess how the greater loss of proximity for engineers on one-building teams translates into the change in online feedback received and programs written. We estimate:

$$Y_{it} = \beta \text{Post}_t \cdot \text{One-Building Team}_i + \alpha \text{Post}_t + \psi \text{One-Building Team}_i + X_{it}'\psi + \epsilon_{it}, \quad (2)$$

where each observation represents a given programmer i in month t. We cluster standard errors at the team level since that is the unit of treatment assignment. This design considers a single focal event — the pandemic-related office closures in March 2020 — so does not run into the problems that can arise when treatment is staggered over time (e.g., Goodman-Bacon, 2021).

We also estimate a dynamic version of Equation 2 that allows the difference between engineers on one- and multi-building teams to vary flexibly by month m:

$$Y_{it} = \sum_{m \neq \text{Feb '20}} \alpha_m \text{One-Building}_i \cdot \mathbb{1}[t = m] + \sigma \text{One-Building}_i + \mu_t + X_{it}'\gamma + u_{it}, \quad (3)$$

where the month before the office closures, February 2020, is held out as the reference month and μ_t denotes month fixed effects.

Our difference-in-differences design relies on the parallel-trends assumption — namely, that engineers who were initially proximate to all of their teammates faced similar pandemic shocks as those who were distant from some teammates. We probe the robustness of this parallel-trends assumption in a few ways. First, we test for imbalances in baseline characteristics and assess robustness to adding controls in X_{it}, which condition the parallel-trends assumption on covariates. Second, we assess placebo checks, using the source of feedback and the timing of treatment. Third, we test for differential pre-trends between engineers on one- and multi-building teams.

III.A Balance in Engineer Characteristics

Table 1 describes the sample, comparing engineers whose teams are all in one building with those whose teams span the two buildings. Engineers' baseline characteristics are largely well-balanced after accounting for engineering group (Column 5). Engineers on one- and multi-building teams have similar demographics (Rows 3−6), similar job level and pay (Rows 8−9), and managers with similar tenure, level, and pay (Rows 10−12). The one notable difference is firm tenure (Row 7): engineers who had been at the company longer were more likely to end up physically separated from at least one teammate so averaged an additional 5 months at the firm.

III.B Controls

Preferred controls: Our preferred controls account for engineering group (i.e., front-end website, back-end databases, or internal tools). We allow the controls for engineering groups to have different effects before versus after the COVID-19 office closures to account for any differential shocks to the demand for these tasks. We further include indicators for the number of months that the engineer has been at the firm and allow the effects of firm experience to differ before and after the offices closed. For online feedback, we also control for program scope — quartics in the number of files changed, the number of lines added, and the number of lines deleted — which might mediate the feedback that an engineer receives.

Full set of controls: Our full set of controls also includes team size, as well as indicators for the engineer's age (in years), gender, being Black, indigenous, or a person of color (BIPOC), home zipcode, job-level, and initial building.[27] We allow all these coefficients to differ before and after the COVID-19 closures to allow different types of engineers to face different pandemic shocks. We further include engineer fixed

[27]For home zipcode, we include the 32 zipcodes with at least 10 engineers. For engineers in less populous zipcodes, we include an indicator for being in the firm's primary state or an adjoining state.

effects to handle any changes in the composition of engineers.

III.C Testing Pre-trends

There was no significant differential trend in peer comments in the pre-period across engineers in one- and multi-building teams (p-value = 0.37 for the raw and p-value = 0.61 for our full set of controls). Indeed, in the months leading up to the office closures, peer commenting did not systematically change for either group.[28] Further, a Wald test does not reject the null that the differences between one- and multi-building teams were the same in every pre-period month (p-value = 0.92 for the raw and p-value = 0.98 for our full set of controls).

We see similar parallel trends in programs written in the pre-period: there is no significant differential trend across engineers in one- and multi-building teams (p-value = 0.96 for the raw and p-value = 0.95 for our full set of controls), and indeed, programs written per month did not substantially change for either group in the pre-period.[29] A Wald test also does not reject the null that the differences between one- and multi-building teams were the same in every pre-period month (p-value = 0.93 for the raw and p-value = 0.92 for our full set of controls).

IV PROXIMITY'S IMPACT ON ONLINE MENTORSHIP

Consistent with the model's first prediction, we find that engineers who are seated near their teammates ask for and receive more online feedback than those seated farther away from their teammates. Since physical proximity can lead to in-person advice too, our estimates of proximity's impact on online mentorship are likely to be lower bounds of proximity's total effect on mentorship.

Engineers on one-building teams received more feedback than engineers on multi-building teams only when the offices were open. Figure 1(a) shows this, plotting the

[28]Peer comments insignificantly increased by 2.7 percent per month (or 0.20 comments) for engineers in one-building teams and by 0.51 percent for engineers in multi-building teams.

[29]Programs written insignificantly decreased by about 1.5 percent per month for both groups.

average number of comments received per program without controls. Initially, engineers on one-building teams received more feedback on their code than engineers on multi-building teams. But this gap disappears when the offices close, suggesting that physical proximity to teammates explained the initial gap. When the offices were open, engineers on one-building teams received 22.4 percent more comments per program (p-value = 0.0003) than did engineers on multi-building teams, when controlling for hire month, program length, and engineering group (our preferred specification in Column 4 of Table 2).[30] This gap narrowed to only 7.9 percent after the offices closed. Thus, the difference-in-differences design indicates that the greater loss of proximity for engineers on one-building teams reduced feedback by 14.4 percent (p-value = 0.007).

The differential decline in feedback for engineers on one-building teams versus multi-building teams is closely tied to the timing of the office closures.[31] The event study in Figure 1(b) illustrates this, plotting the coefficients from Equation 3 conditional on our preferred set of controls. As the figure shows, other untreated months do not feature similar changes in the feedback received by one- versus multi-building teams (also shown in the placebo check in Figure A.6).

While the offices remain closed, engineers who had been near all their teammates never regain the greater level of interaction that they had in the office. The persistence suggests our effects are not a fleeting byproduct of transitioning to new technologies for engineers accustomed to in-person interactions with their teammates.[32]

[30]The slight uptick in comments received just before offices closed is largely due to a small uptick in hiring of new college graduates, particularly on one-building teams. Our preferred controls account for the relationship between tenure and comments received. Allowing tenure effects to vary by engineer age entirely removes the differential blip (see Figure A.5) and if anything makes our estimates a bit larger. We do not adopt this as our preferred specification since it reduces precision.

[31]One might have expected that those who sat together in the office and had a habit of increased online collaboration might have collaborated more after the offices closed at least in the short-term. But we see little evidence for a lasting effect of proximity.

[32]Since all engineers were familiar with Github, the online software used for giving comments on code, it is not surprising that there was not much technological adaptation. While engineers on one-building teams might have been less comfortable with Zoom and other online tools, our persistent effects suggest that the transitory costs of learning to use these tools likely did not drive our findings.

Our results are robust to adding a variety of controls (Table 2), a stability that is notable given the increase in the R^2 from 2 percent to 50 percent. Our results are also robust to including local-linear time-trends for engineers on one- and multi-building teams (Table A.5) and to limiting to alternative bandwidths around the office closures (Figure A.7). We also find similar results when limiting to engineering teams who work on internal tools for others in the firm and so are more likely to end up spread across buildings (Figure A.8 and Table A.6).[33]

We find that the differential declines in feedback are driven by comments given during standard work hours (8AM $-$ 6PM, Monday through Friday), when teammates on one-building teams would have been proximate to one another before the offices closed but not afterwards (Figure A.4).

Substantive Feedback. We find similar results using different measures of feedback in Table 3(a). Proximity enhances not only the number of comments (Column 1) but also the total number of characters (Column 2). It leads to timelier feedback, reducing delays between program submission and the receipt of the first comment (Column 3). Proximity also appears to *increase* references to other online conversations, such as email, Slack, or Zoom (Column 4), suggesting that collaboration does not simply migrate to another means of digital communication when engineers are distant. Instead, proximity increases communication on several digital channels.

We also find that various different types of comments are impacted by proximity, including those about testing code and verifying that functions are producing the right outputs (Figure A.2(b)). Both of these themes capture substantive feedback that is likely to be time-consuming to give, broadly applicable to other programs, and important for the code's reliability. In addition, since the vast majority of comments proffered are helpful (see Appendix I.B), the dropoff in comments is not just "trim-

[33]Since engineers who worked on internal tools found it useful to sit near those who used their tools, their teams were more likely to face desk constraints and end up in multiple building. The fact that we find similar results when limiting to these groups suggests that our results are not driven by differential shocks to those working on internal tools versus the website.

ming the fat" of unhelpful comments. There is no statistically significant change in the nature of the comments around office closures (Table A.7).

Asking for Feedback. Our results indicate that engineers ask for more feedback when seated near their colleagues and consequently receive more online mentorship when physically proximate to coworkers.

While we do not directly observe initial requests for feedback (which are typically made in person or over Slack), we do observe the number of commenters per program. Since commenters have to be asked to review, we view this as a proxy for the number of people asked. Engineers on one-building teams have 11.5 percent more commenters before the closures and this gap is more than halved when the offices close (Column 2 of Table 3(b)).

Through the code-review system, we can also directly see the number of follow-up questions that the program writer asks the commenter. We find that engineers on one-building teams ask 47.5 percent more follow-up questions than engineers on multi-building teams when the offices are open, and this gap disappears once the offices close (Column 3 of Table 3(c)). More follow-up questions lead to more clarification and additional feedback: indeed, commenters' additional follow-up comments account for more than half of the effect of proximity on total feedback received (comparing Column 4 of Table 3(c) to Column 1 of Table 3(a)).

Placebo Check & Complementary Designs. As a placebo check, Figure 1(c) shows that losing proximity to teammates only affects feedback from teammates not feedback from other engineers. The null effect for non-teammate comments suggests engineers' need for feedback was not the main factor: if engineers on one-building teams simply needed more help before the offices closed but not afterwards, this change would impact comments from both teammates and non-teammates.

A similar design shows that proximity to *non*-teammates only impacts feedback from non-teammates. While the offices were open, engineers in the main building were near 71 percent of the main campus's engineers and the main campus's lunch room. When the offices closed, these engineers saw larger declines in feedback from non-teammates than did engineers who sat in the auxiliary building (Figure A.9(a)). Engineers in the main building did not see larger declines in teammate feedback, conditional on their type of team (Figure A.9(b)). This finding suggests that serendipitous watercooler chats facilitate online interaction across teams.

We also find similar impacts when we compare one-building teams to teams that are spread across campuses (Figure A.10). Engineers whose teammates either worked on another campus or worked from home received the fewest comments on their programs before the pandemic, but these gaps also closed when the offices closed. Engineers on multi-campus teams do not receive substantially fewer comments than engineers whose teammates are just a few blocks away. The comparable results for multi-building and multi-campus teams suggest that small frictions to face-to-face contact can have out-sized effects on feedback. Further, this similarity suggests that our main results do not simply reflect some managers agitating for their teams to be unified in one building on the main campus.

IV.A Mentorship of Juniors by Seniors

Our results are driven by feedback received by less-experienced engineers and feedback given by more-experienced engineers.

As illustrated in Figure 2(a), junior engineers (with less than the median tenure of 16 months) receive more feedback generally, and the feedback that they receive is also more sensitive to teammate proximity.[34] When the offices were open, junior engineers on one-building teams received 27 percent more feedback than those on multi-building teams (Column 2 of Table A.8(a)), which quickly narrowed when

[34] As discussed in footnote 30, the pre-closure uptick in comments received by juniors and given by seniors on one-building teams is due to a small increase in hiring of new college graduates. Allowing tenure effects to vary by engineer age removes the differential blip (see Figure A.5(b)-(c)).

the offices closed. On the other hand, proximity did not impact feedback for more-experienced engineers either before or after the closures (Column 5 of Table A.8(a)).

When we consider the seniority of the comment writers in Figure 2(b), we see the opposite pattern.

These findings are consistent with the model in which juniors receive extra mentoring from senior colleagues when sitting together in the office.

Mentorship is particularly important for engineers who are both young and new to the firm. Young, junior engineers receive the most feedback, and the feedback that they received was most sensitive to their proximity to their teammates (Figure A.11(a)). Young engineers who had considerable experience at the firm also received more feedback when sitting near their teammates (Figure A.11(b)). These patterns are consistent with young engineers having more to learn from their more experienced colleagues and proximity facilitating these knowledge flows.

IV.B Externalities from Distant Teammates

In the model, engineers are paired with one mentor, but, at this firm, engineers work in teams. Distant teammates have externalities on the other members of their teams' interactions, decreasing the feedback workers get from their teammates who sit nearby. These externalities likely arise because once one worker is distant, teams often hold online meetings instead of in-person ones.[35]

We measure the impact of a distant teammate in two ways. First, Figure 3(a) uses our main empirical design that compares one- and multi-building teams around the office closures but limits the analysis to teammates who are in the *same building*. Before the offices closed, an engineer with distant teammates received 17 percent fewer comments per review from a *proximate* teammate than did an engineer whose teammates were all in her building. This gap largely closed once the offices shut

[35]The small-talk before in-person meetings, which is often hard to replicate on Zoom, may help ease requests for additional help.

down for the pandemic (Column 2 of Table A.9). These externalities explain about 30 percent of the feedback gap between one- and multi-building teams (Table A.9).

Second, we examine team dynamics around a new hire. We compare teams where the new hire converted the team from being a one-building team into a multi-building team to teams where the new hire did not affect whether the team was co-located. We estimate

$$\text{Comments}_{ijt} = \gamma \text{Post Hire}_{it} \cdot \text{One- to Multi-building}_{i} + \sigma \text{Post Hire}_{it} + \mu_{ij} + v_{ijt} \quad (4)$$

where i indexes the coder, j indexes the commenter, t indexes the time. The dependent variable, Comments_{ijt}, is the number of comments given by commenter j to programmer i on a review. We only consider coders and commenters who are in the same building and were hired before the 6-week window around the new hire.

Figure 3(b) shows that one-building teams with a new hire in another building see a sharp decline in online feedback between same-building teammates. Teams that were always in one building or multiple buildings do not. The estimated decrease is 1.7 comments per review for teams becoming multi-building team relative to other teams with new hires, which is similar conditional on program scope (Table A.10).

Our two designs both find that having even one teammate in another location diminishes feedback from same-building colleagues.[36] These results suggest that even as workers come back to the office after the pandemic, their interactions will be affected by coworkers who continue to work remotely.

V PROXIMITY'S IMPACT ON PROGRAMMER OUTPUT

When junior engineers sit near their coworkers, they receive more mentorship, but this has an opportunity cost. Both junior and senior engineers write fewer programs when seated near their teammates. The effects are particularly large for senior en-

[36]For both designs, the effect is driven by engineers who are new to the firm (Figure A.12).

gineers. Thus, proximity creates a trade-off between long-run human capital development and short-run output.

Figure 4(a) illustrates the consequences of proximity for workers' programming output. When the offices were open, engineers who sat near all their teammates submitted 19 percent fewer programs per month (or 0.32 fewer programs) than similar engineers whose teammates were distributed across the main campus's two buildings (p-value = 0.04, Column 3 of Table 4(a)). When the offices closed for COVID-19, this gap narrowed. The programming output of all engineers fell — likely due to the pandemic's many stressors — but the decline was less precipitous for engineers who had been proximate to all of their colleagues in the office. Thus, in relative terms, engineers who had sat near all their teammates in the office caught up to the programming output of engineers on already-distributed teams once they too were distributed (shown in Figure 4(b)).

Our difference-in-differences estimate indicates that proximity to teammates reduces programming output by 24 percent (0.41 programs per month, p-value = 0.0001, Column 3 of Table 4(a)). Table 4 shows similar results for different specifications. This table also shows comparable patterns for lines of code written, ruling out the explanation that engineers on one-building teams simply submitted more frequent, smaller chunks of code once they were no longer sitting near their colleagues. We also see similar patterns when we replicate this analysis for the subset of engineering teams who build internal tools for others in the firm and so are more likely to end up spread across the buildings (Figure A.13 and Table A.11).

The effect of proximity on programming output is more pronounced for senior engineers, as illustrated in Figure 4(c). While the offices were open, senior engineers on one-building teams wrote 39 percent fewer programs than senior engineers on distributed teams (Column 1 of Table A.12). After the offices closed, this gap quickly closed. The difference-in-differences indicates that proximity reduces senior engineers' output by 30 percent. The effects are even more pronounced for lines of code and files changed (Column 3−6 of Table A.12).

Junior engineers on one-building teams also submit slightly fewer programs than junior engineers on distributed teams before the offices closed but not afterwards (Figure 4(c)). This negative effect of proximity on junior engineers' programming output is driven by the most junior engineers who also see the largest declines in feedback when proximity is lost (Figure A.14). This pattern is consistent with the most inexperienced engineers being able to submit more programs when they spend less time responding to feedback and refining their programs.

The results on programming output are consistent with the model's third prediction that mentorship has opportunity costs for both mentors and mentees. Our results suggest that additional mentoring has larger opportunity costs for the mentors than the mentees.

VI GENDER

In our model, we allowed for the possibility that some men are comfortable asking for feedback, even when remote. This implies that proximity has a larger impact on the feedback female engineers ask for and receive (Prediction 5).

Consistent with this prediction, we find that female engineers' feedback is more sensitive to their proximity to their teammates. Figure 5(a) shows this, plotting the raw time-series of the comments received per program on one-and multi-building teams, separately for female and male program writers. Before the offices closed, female engineers on one-building teams received 40 percent more comments than their multi-building counterparts, relative to a gap of only 18 percent for male engineers (Column 1 of Table A.13(a)). After offices closed, the additional feedback received by engineers on one-building teams shrank by much more for female engineers (30 percent, p-value = 0.0005) than for male engineers. The triple difference indicates that losing proximity decreased feedback by 21 percent more for female engineers than for male engineers (p-value = 0.01, Column 1 of Table A.13(b)). We find similar effects for auxiliary measures of feedback (Table A.14).

The patterns are not consistent with female engineers simply needing more feedback on their code. Instead, female programmers on multi-building teams received *fewer* comments than their male counterparts. With our preferred controls, this gap represents an 19 percent deficit in feedback for female engineers on distributed teams while the offices were open (p-value = 0.01). Differences in tenure and age by engineer gender also do not explain the differential impact of proximity. While female engineers do tend to be about two years younger and have one fewer months at the firm, we still find gender differences after allowing for the effects of proximity to vary by engineers' age and tenure (Table A.15).

The data suggest that these gendered effects on feedback stem from differences in engineers' willingness to ask for feedback when remote. We first consider the number of commenters per program, a proxy for the number of people whom the engineer asked to review the code (Figure 5(b)). When the buildings were open, female engineers had more commenters on their code on one-building teams than on multi-building teams — a gap that was larger than the male programmers' gap. When the buildings closed, female engineers also experienced more of a drop off in commenters on one-building teams. On average, female engineers saw a decline in the number of commenters 13 percent larger than that of male engineers (p-value = 0.003, Column 2 of Table A.13(b)).

Within the code-review process, we find that proximity matters more for female engineers' willingness to ask follow-up questions. Figure 5(c) illustrates this result. When the offices were open, female engineers were more likely to ask follow-up questions on one-building teams than multi-building teams — a differential that was largely absent for male engineers. Female engineers also see a suggestively larger decline in follow-up questions when the offices closed. This differential in follow-up questions is reflected in commenters' engagement in the review. The gendered effect of proximity on comments received is almost entirely driven by follow-up responses from commenters, not in their initial feedback on the code (Figure A.15 and Columns 3−5 of Table A.13).

One might worry that female engineers receive more feedback when sitting near teammates solely because their male colleagues over-explain to them. Instead, we find that the gendered effect of proximity on the feedback that engineers receive is present for comments originating from both male and female commenters (Figure A.16). This pattern suggests that the gender match between the programmer and commenter is not the primary moderator of the effect of proximity on mentorship.

Relatedly, one might worry that female engineers receive more unhelpful or nitpicky comments, particularly when sitting near their colleagues. Instead, we find that the comments received by female engineers are more likely to be rated as helpful and consequential for the code (Table A.3(a)).[37] These gender differences in the substance of feedback are broadly similar on one-building teams before and after the offices closed (Table A.3(b)). Thus, proximity seems to increase the volume of feedback received by female engineers without reducing its substance, which tends to be especially high for female engineers.

Our model also predicts that proximity has larger impacts on the feedback that female engineers give (Prediction 6) because they find it more costly to reject requests. We see this borne out in Figure 5(d): female commenters give much more feedback on one-building teams before the pandemic and see a much bigger decrease in the feedback they give on one-building teams when the offices close. Using our preferred controls, the difference-in-differences design indicates that losing proximity reduces the extensiveness of female engineers' code reviews by 27 percent more than those of male engineers (p-value = 0.07, Column 1 of Table A.16(b)).

These gaps translate into output (Prediction 7). Women face a larger output cost of

[37]Comments received by female engineers are also marginally more likely to be rated as giving actionable advice and explaining the reasoning behind suggested changes (Columns 5−8). They are also marginally less likely to be rated as rude (Columns 9−10).

sitting near their teammates than do men (Figure 5(e)). As with the other outcomes, this is true both when looking at the difference between one- and multi-building teams before COVID-19 and the differential change once the offices close. With our preferred controls, the estimated cost of proximity for programming output is suggestively 5 percent larger for women but not significant at traditional levels (p-value = 0.29, Column 2 of Table A.16(b)). As expected, the output cost of proximity is particularly large for senior women, who do a lot of the mentoring (Figure A.17). Similar effects show up in other output measures (Columns 3−4 of Table A.16).

VII Career Outcomes

We next analyze how the tradeoffs created by proximity impact engineers' career trajectories. Throughout this section, we consider workers who were hired before our data starts, so they have spent enough time at the firm to gain skills from their coworkers before the offices close. This restriction does not affect our analysis of the short-run costs of proximity, but it does allow us to more cleanly estimate the long-run returns to having sat with coworkers.

VII.A Pay

The model predicts that junior engineers would earn less when proximate to coworkers because they write fewer programs. Consistent with this prediction, junior workers on one-building teams were substantially less likely to get a pay raise than those on distributed teams while the offices were open (Column 2 of Table 5).[38]

The model's prediction about wages for senior workers on one-building teams is less clear. On the one hand, senior engineers write less code when proximate to teammates, which would translate into lower wages. On the other hand, although team type can vary over workers' careers, they were more likely to be trained on

[38]We focus on pay raises rather than pay levels because pay levels may reflect prior experience that we cannot observe since it occurs either outside the firm or prior to our time period. Instead, pay raises more accurately reflect recent performance. Three times per year, the firm reviews workers' performance and decides whether to increase a worker's pay.

one-building teams and so accumulated more human capital. Empirically, senior engineers on one-building teams are similarly likely to receive pay raises as seniors on multi-building teams (Column 2).

Female engineers on one-building teams suggestively take a larger wage hit, which may reflect greater time spent being mentored and offering mentorship.

Once the offices close, there is no difference in mentoring between team types. Thus, the model predicts that workers who were trained on one-building teams should have more human capital and earn more than their counterparts. This is borne out in the data (Columns 4−6 of Table 5). Workers who had been on one-building teams were (insignificantly) 5.5 percentage points more likely to receive a raise after the offices closed (Column 4). This is particularly pronounced for junior engineers (whom we know were trained on one-building teams), who were 7.2 percentage points more likely to receive a raise after the closures (Column 5, p-value = 0.066). This differential is also suggestively more pronounced among female engineers, for whom mentorship was particularly sensitive to proximity (Column 6).[39] These patterns are similar when we limit the sample to engineers who work on internal tools (Table A.17).[40] The patterns look qualitatively similar when we consider salary changes in levels or the inverse hyperbolic sine of these salary changes in Table A.18.

Consistent with junior engineers who were trained on one-building teams being more productive once the offices closed, they were also suggestively, although insignificantly, less likely to be fired (Table A.19).

So far, we have been agnostic about whether training's long-term benefits exceed its short-term costs. Ideally, we could compare the pre-pandemic wages of senior workers who had always been on one-building teams — and so experienced both

[39]Figure A.18 illustrates these reversals in relative raises graphically.

[40]We see the same reversal in rates of pay raises overall (Columns 1 vs. 4), with more pronounced effects for female engineers. The patterns look more similar for junior and senior engineers, however, rather than looking more pronounced for junior engineers.

training's costs and benefits — with the wages of senior workers who were always on multi-building teams. Unfortunately, we cannot implement this test because we do not know whether senior workers were trained on one- or multi-building teams, due to the short duration of the data, the changing nature of teams, and the movement of engineers across teams (particularly as senior workers get promoted).

VII.B Quits

Consistent with the model, workers trained on one-building teams are more likely to quit once tech jobs go remote and so it's easier to move to superstar tech firms. Figure 6(a) shows that before the pandemic, quit rates are low for workers on both one- and multi-building teams. After the start of the pandemic, quit rates increase for both groups, and workers trained on one-building teams are more likely to quit.

Figure 6(b)-(c) show that the increase in quits is driven by the groups that likely gained the most human capital from being nearby. Panel (b) shows that the effect is particularly large for junior engineers, who were trained on one-building teams. Indeed, junior engineers who had sat with their coworkers in the office saw a 1.2 percentage point greater increase in quits, about twice that of engineers who were trained on multi-building teams (p-value of difference = 0.01). Panel (c) show that the impact is larger for women, who we have seen gain much more mentorship from proximity. We also see higher quit rates for younger engineers (Figure A.19), who also have bigger boosts in feedback from sitting near their teammates.[41]

A natural question is whether quits represent steps up in engineers' careers. For two-thirds of the engineers who quit, we observe their new job title and company and can find the average compensation in that new position using data from Glassdoor. We categorize a worker as quitting for higher pay if they quit for a posi-

[41] We find similar quit patterns overall and by gender when limiting to engineers who work on internal tools. Engineers on one-building teams are significantly more likely to quit after the offices closed, which is especially true among women (Figure A.20). However, we do not see the same heterogeneity by engineers' tenure, potentially due to the more limited sample.

tion with higher average compensation than that in their old position at the firm.[42] Among workers who quit, 78 percent leave for higher-paying jobs.[43] These advantageous departures drive our results: engineers who had been on one-building teams are more likely to quit for higher-paying positions once the offices close but not to quit for lower-paying positions (Table A.20 and Figure A.21).[44] Junior engineers and women who were on one-building teams see particularly large upticks in departures for higher-paying jobs after the closures. These patterns suggest the returns to mentorship partially accrue outside the firm.

VIII Who Works in the Office?

We assume that the firm cannot incentivize mentorship directly, but the firm can affect mentorship through its remote-work policy.[45] If mentorship's benefits exceed its costs, we expect to see that:

1. When proximity is possible, both mentees and mentors will be on-site.

2. When proximity is difficult, the firm will shift away from requiring mentorship. They will "buy" talent by hiring more experienced workers rather than "building" skills internally.

Both predictions are borne out in our firm's remote work and hiring decisions.

Before the pandemic, when proximity was possible, mentees — with the most junior job levels — and mentors — with the most senior job levels or management roles — were required to be in the office (Figure 7(a)). These policies created a U-shaped

[42] We use Glassdoor data for average salary at our firm as well as at the destination firm to compare apples to apples. The Glassdoor data aligns well with the firm's administrative data: the correlation between an engineer's actual base pay and the average base pay in that role on Glassdoor is 0.73.

[43] This rate is slightly higher after the offices closed (78.9 percent) than beforehand (76.2 percent). What changed most, however, was the tripling in the rate of voluntary departures.

[44] When asked why they quit, most workers say they quit for a better job. While we take workers' responses to this with a grain of salt, results are also similar if our dependent variable is quits that workers report are to better jobs.

[45] Junior and senior workers pay for the training through lower wages, but if the firm has to compete over workers with other firms, it has an incentive to provide the efficient level of mentorship.

pattern in office work across worker age: the youngest and oldest engineers were more likely to be in the office (Figure A.22). These patterns of on-site work are reflected in national trends from the Household Pulse Survey in which the youngest workers and relatively older college-graduates are more likely to return to the office in 2022−2023, when limiting to those workers without children (Figure 7(b)). In 2023, 74 percent of workers aged 21−23 were working on-site, relative to just 60 percent of 30−32 year old workers.

Consistent with the second prediction, our firm shifted its hiring practices around the office closures. Before the offices closed, 55 percent of hires were the lowest level of engineer; afterward, that number sank to 37 percent (p-value of change = 0.0072), leading to a marked decline in the share of workers who need mentoring. By contrast, the share of hires who are unlikely to need as much training (levels 3 and above) increased from 23 percent to 40 percent (p-value = 0.0072). Of course, it's possible that hiring practices shifted for other reasons, such as senior engineers being more willing to change jobs when they did not need to move cities.

To the extent that mentorship increases junior workers' skills, barriers to mentorship may have scarring effects on less experienced workers. First, even if junior workers go to the office, potential mentors may not. Further, if firms are more likely to "buy" talent rather than "build" it when in-person work is challenging, then job opportunities may be harder to come by for junior workers.

IX CONCLUSION

Remote work leads to a tradeoff. It increases output today, particularly from more senior workers. But remote work decreases training of more junior workers, which has future costs. Work arrangements seem to respond to this tradeoff with junior workers and potential mentors less likely to work remotely.

We find the tradeoffs of proximity are particularly pronounced for women. When not sitting near their colleagues, junior women ask for and receive less mentorship.

Yet when apart, senior women provide less guidance to their junior colleagues and can therefore focus more on their own productivity.

We find that one worker's choice to work remotely impacts their peers. When more experienced workers choose to work from home, junior workers may learn less. The rise of remote work could consequently have scarring effects on less experienced workers, who learn less from their coworkers and have less mobility to higher-paying jobs at other firms.

We further find that a single remote worker can have an outsized impact, depressing collaboration even between co-located coworkers. This finding suggests policies coordinating workers' locational choices may yield benefits. For example, it may be more efficient to have firms or teams sort into being fully in-person or fully remote than to have hybrid teams where a few remote workers affect their in-person colleagues. Moreover, coordinating which days teams spend in the office may lead to more fully in-person meetings and more mentorship. This raises the question of how much of the mentorship benefits of in-person work can be achieved by only a few days per week in the office.

Finally, if there is a permanent increase in remote work post-pandemic, can alternative management practices encourage more training of junior workers? While mentorship is hard to fully observe, it will be interesting to see whether firms start collecting more mentorship metrics, putting more weight on training in worker evaluations, and formalizing training that was previously based on informal interaction.

REFERENCES

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

Akcigit, Ufuk, Santiago Caicedo, Ernest Miguelez, Stefanie Stantcheva, and Valerio Sterzi, "Dancing with the stars: Innovation through interactions," Technical Report, National Bureau of Economic Research 2018.

Atkin, David, Keith Chen, and Anton Popov, "The Returns to Face-to-face Interactions: Knowledge Spillovers in Silicon Valley," 2019.

Azoulay, Pierre, Joshua S Graff Zivin, and Jialan Wang, "Superstar Extinction," *The Quarterly Journal of Economics*, 2010, *125* (2), 549–589.

Bandiera, Oriana, Iwan Barankay, and Imran Rasul, "Social Incentives in the Workplace," *The Review of Economic Studies*, 2010, *77* (2), 417–458.

Barrero, Jose Maria, Nicholas Bloom, and Steven J Davis, "Why Working from Home Will Stick," Technical Report, National Bureau of Economic Research 2023.

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.

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.

Boudreau, Kevin J, Tom Brady, Ina Ganguli, Patrick Gaule, Eva Guinan, Anthony Hollenberg, and Karim R Lakhani, "A Field Experiment on Search Costs and the Formation of Scientific Collaborations," *Review of Economics and Statistics*, 2017, *99* (4), 565–576.

Brucks, Melanie S and Jonathan Levav, "Virtual Communication Curbs Creative Idea Generation," *Nature*, 2022, pp. 1–5.

Cairncross, F., *The Death of Distance: How the Communications Revolution is Changing Our Lives* The Death of Distance: How the Communications Revolution is Changing Our Lives, Harvard Business School Press, 2001.

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

Chen, Chinchih, Carl Benedikt Frey, and Giorgio Presidente, "Disrupting Science," 2022.

Choudhury, Prithwiraj, Cirrus Foroughi, and Barbara Zepp Larson, "Work-from-Anywhere: The Productivity Effects of Geographic Flexibility," in "Academy of Management Proceedings" number 1. In '2020.' Academy of Management Briarcliff Manor, NY 10510 2020, p. 21199.

Cornelissen, Thomas, Christian Dustmann, and Uta Schönberg, "Peer Effects in the Workplace," *American Economic Review*, 2017, *107* (2), 425–56.

_ , _ , and _ , "Knowledge Spillovers, Competition, and Individual Careers," *Working Paper*, 2023.

Cullen, Zoe and Ricardo Perez-Truglia, "The old boys' club: Schmoozing and the gender gap," *American Economic Review*, 2023, *113* (7), 1703–1740.

DeFilippis, Evan, Stephen Michael Impink, Madison Singell, Jeffrey T Polzer, and Raffaella Sadun, "Collaborating During Coronavirus: The Impact of COVID-19 on the Nature of Work," *NBER Working Paper*, 2020, (w27612).

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.

Dutcher, Glenn and Krista Saral, "Remote Work and Team Productivity," 2022.

Emanuel, Natalia and Emma Harrington, "Working Remotely? Selection, Treatment, and the Market for Remote Work," *Fed Staff Reports Working Paper*, 2023.

Falk, Armin and Andrea Ichino, "Clean Evidence on Peer Effects," *Journal of Labor Economics*, 2006, *24* (1), 39–57.

Friedman, Thomas L, *The World is Flat: A Brief History of the Twenty-first Century*, Macmillan, 2005.

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.

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.

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," Technical Report, National Bureau of Economic Research 2018.

Jackson, C Kirabo and Elias Bruegmann, "Teaching Students and Teaching Each Other: The Importance of Peer Learning for Teachers," *American Economic Journal: Applied Economics*, 2009, *1* (4), 85–108.

Jarosch, Gregor, Ezra Oberfield, and Esteban Rossi-Hansberg, "Learning from Coworkers," *Econometrica*, 2021, *89* (2), 647–676.

Johnsen, Julian, Hyejin Ku, and Kjell G Salvanes, "Competition and Career Advancement," 2023.

Kelly, Jack, "Morgan Stanley CEO James Gorman On His Return-To-Work Plan: 'If You Can Go To A Restaurant In New York City, You Can Come Into The Office'," *Forbes*, June 2021.

Linos, Elizabeth, "Does Teleworking Work for Organizations?," Technical Report, Harvard University Working Paper 2018.

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," Technical Report, National Bureau of Economic Research 2018.

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.

— **and Enrico Moretti**, "Peers at Work," *American Economic Review*, 2009, *99* (1), 112–45.

Newton, Casey, "Mark Zuckerberg on Taking His Massive Workforce Remote," *The Verge*, May 2020.

Nix, Emily, "Learning Spillovers in the Firm," Technical Report, Working Paper 2020.

Sandvik, Jason J, Richard E Saouma, Nathan T Seegert, and Christopher T Stanton, "Workplace Knowledge Flows," *The Quarterly Journal of Economics*, 2020.

Tsipursky, Gleb, "What Is Proximity Bias and How Can Managers Prevent It?," *Harvard Business Review*, 2022.

U.S. Census Bureau, "2019 American Community Survey Public Use Microdata Samples," 2019. data retrieved from IPUMS USA, https://usa.ipums.org/usa/ (visited January 19, 2023).

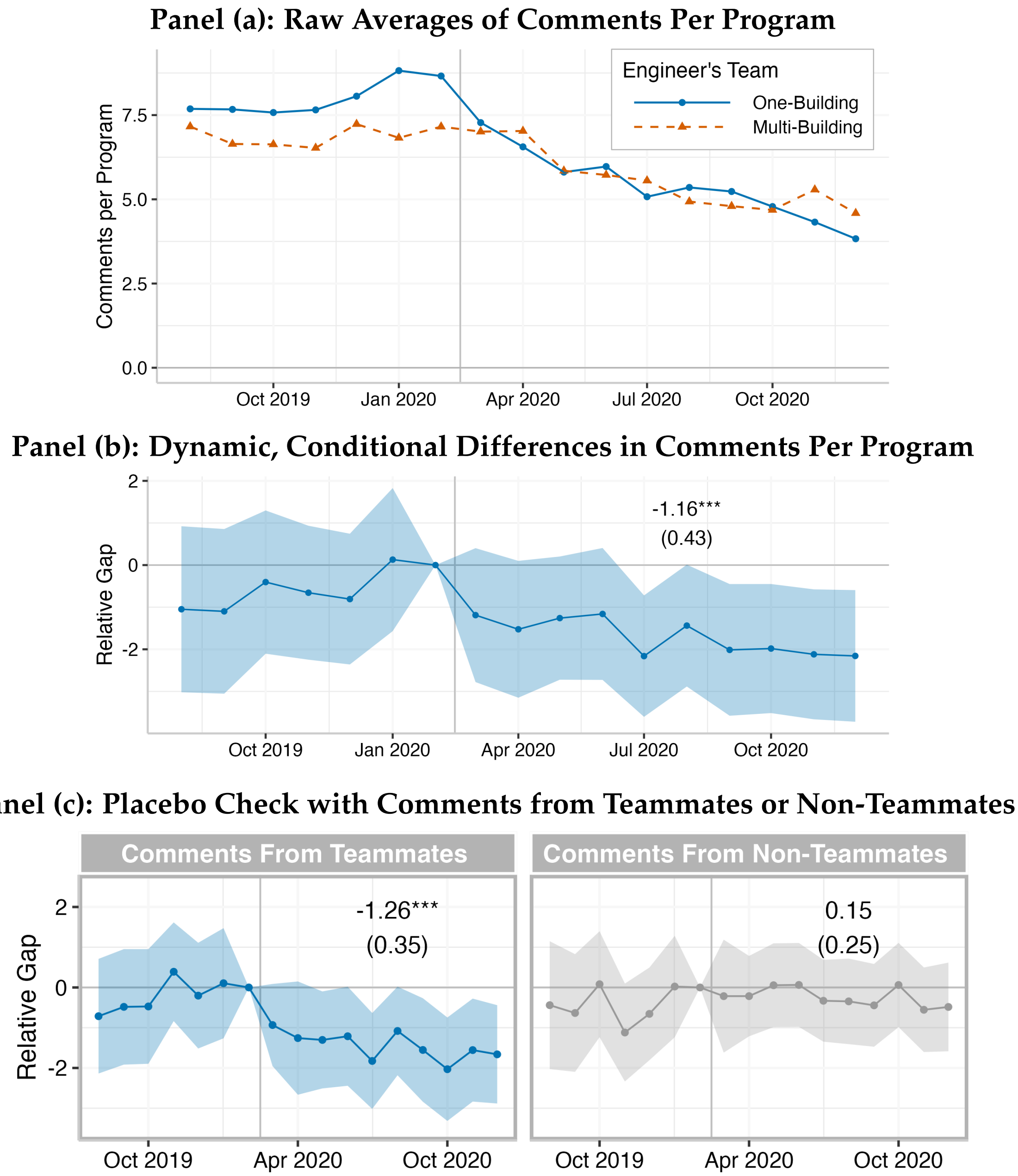
—, "Household Pulse Survey," 2023. data retrieved from Cenus, https://www.census.gov/programs-surveys/household-pulse-survey/datasets.html (visited January 19, 2023).

Waldinger, Fabian, "Peer Effects in Science: Evidence from the Dismissal of Scientists in Nazi Germany," *The Review of Economic Studies*, 2012, *79* (2), 838–861.

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.

Zuckerberg, Mark, "Update on Meta's Year of Efficiency," *Meta Newsroom*, March 2023.

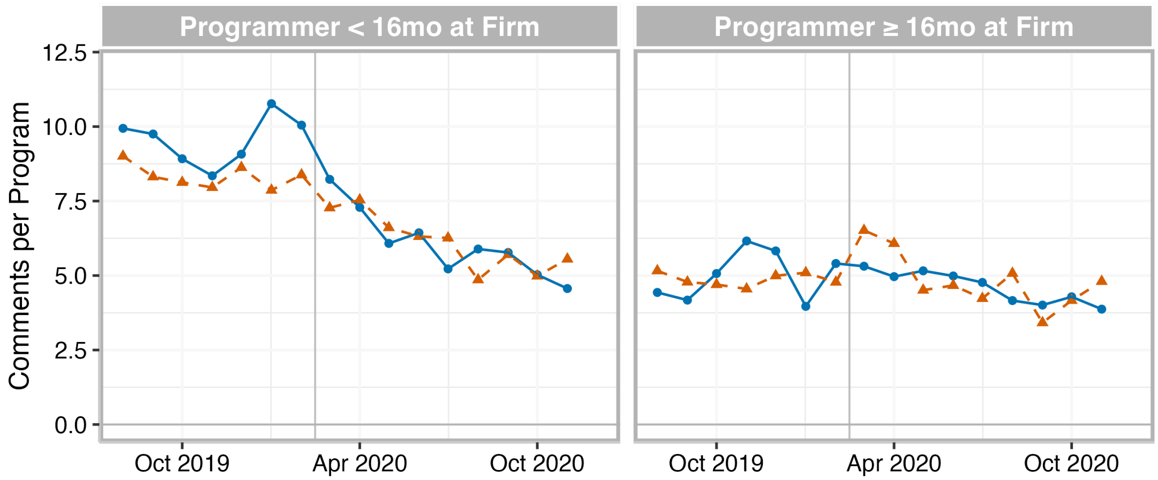
Figure 1: Proximity to Teammates and Online Feedback



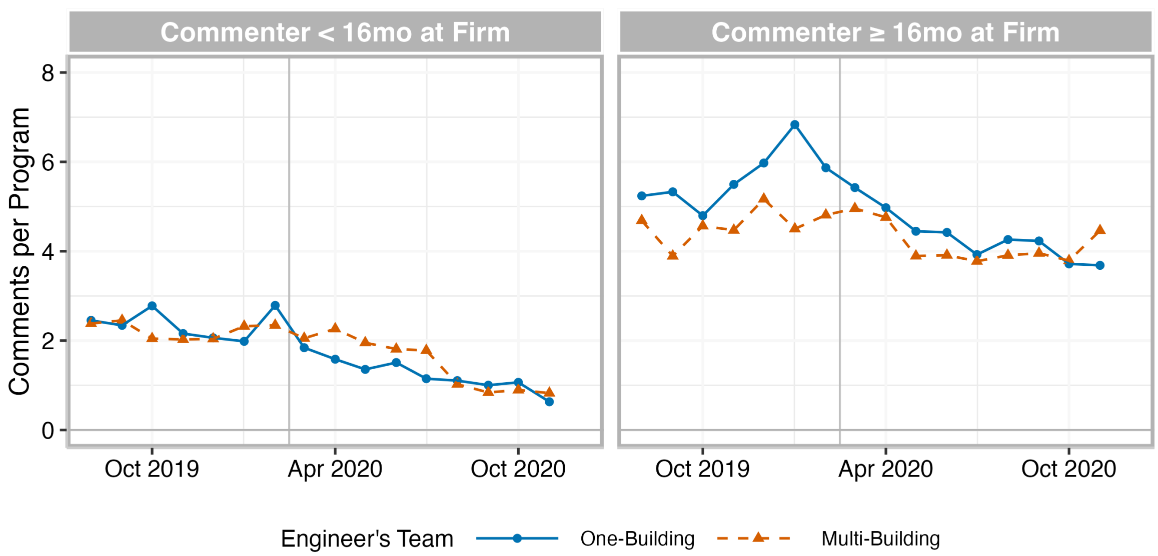
Notes: This figure illustrates the online feedback received by engineers on one-building teams (N=637) and on multi-building teams (N=418) around the COVID-19 office closures (the grey vertical lines). Panel (a) plots the raw averages, while Panel (b) plots the differences from Equation 3, conditional on our preferred controls (as in Column 4 of Table 2). Panel (c) uses the same specification as in (b) to illustrate a placebo check: the left panel show comments from teammates which should be affected, and the right panel shows comments from non-teammates which should not be. The ribbons in Panels (b) and (c) show 95% confidence intervals with clustering by team. The annotated coefficients are the difference-in-differences estimate from Equation 2. The sample limits to engineers whose teammates all worked in the main campus. *p<0.1; **p<0.05; ***p<0.01.

Figure 2: Proximity and Mentorship

Panel (a): Comments per Program by Program Writer's Tenure



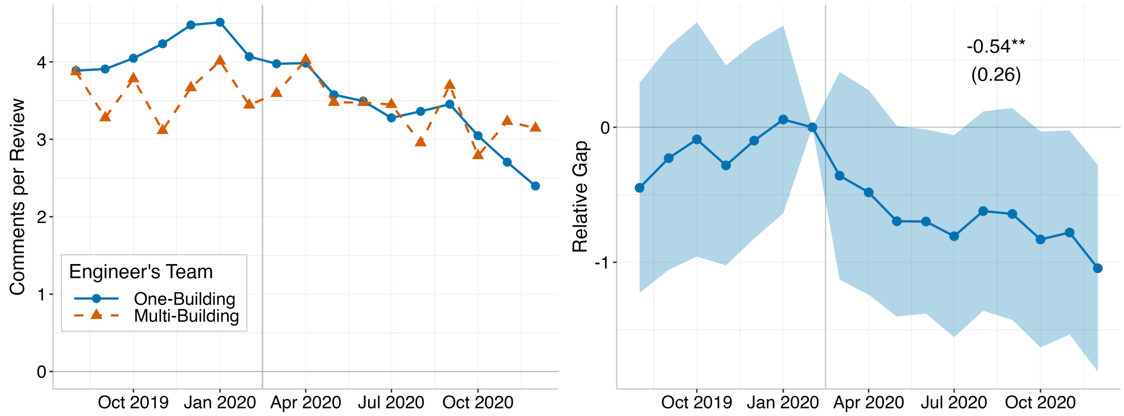
Panel (b): Comments per Program by Commenter's Tenure



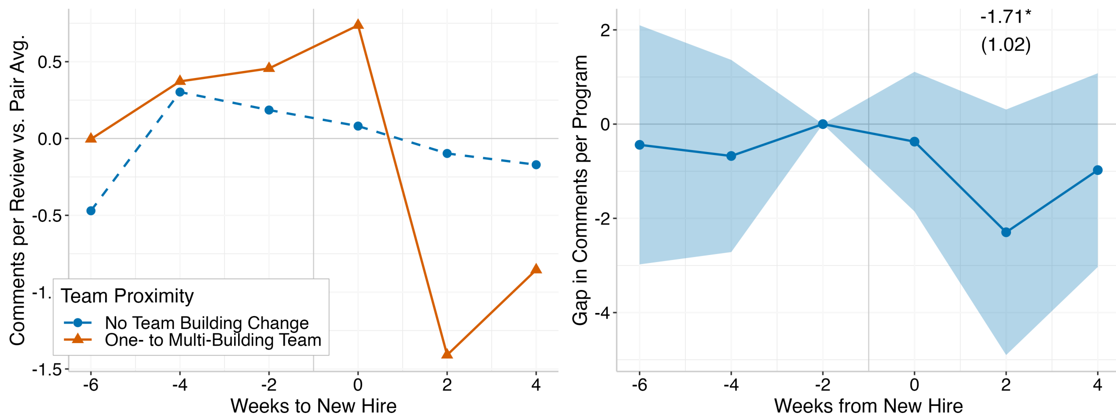
Notes: Panel (a) shows the effects of proximity on online feedback received by engineers of different tenures. It shows the raw monthly averages of comments received per program for engineers on one- and multi-building teams, separately by those below the median tenure of 16 months. Panel (b) shows a comparable plot of raw averages of comments per program broken down by the seniority of the commenter rather than the program writer.

Figure 3: Externalities from Distant Teammates

Panel (a): Diff-in-Diff Design for Reviews from Same-Building Teammates



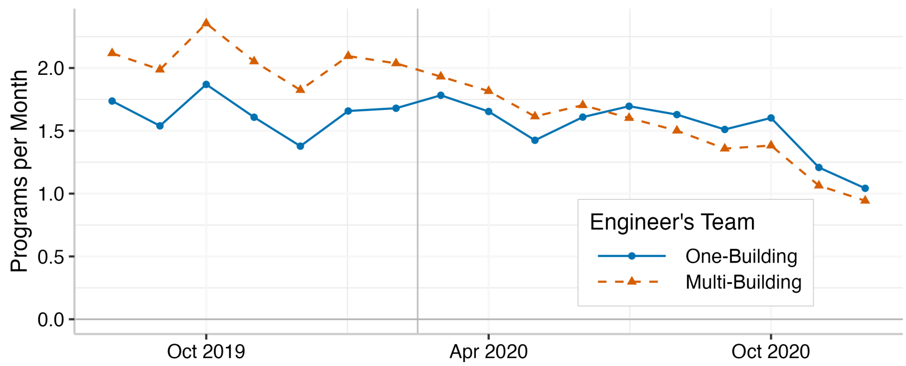
Panel (b): Pre-COVID Hire in Another Building



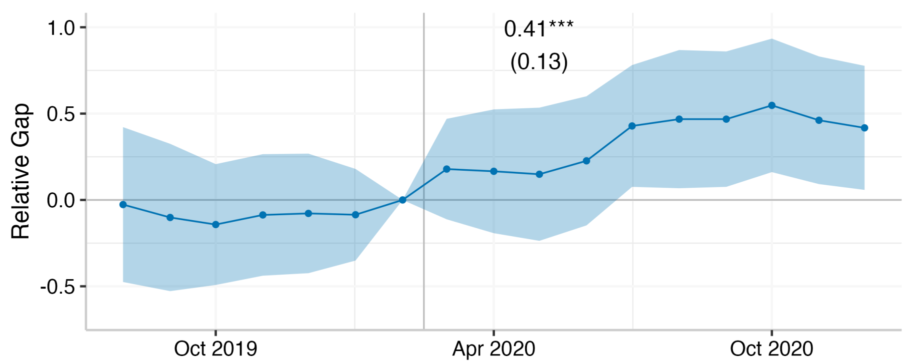
Notes: This figure investigates the externalities from having a distant teammate on the feedback exchanged between teammates who sit in the same building. Panel (a) replicates Figure 1 but focuses on comments received per review from a same-building teammate. Panel (b) considers the impact of a new hire that converts the team from a one-building team to a multi-building team versus a new hire that does not change the distribution of the team, as in Equation 4. The plots focus on pre-existing relationships between teammates in the same building. The left plot shows comments received on each program relative to the average in the coder-commenter pair. The right plot shows the conditional difference in feedback between these two groups, with fixed effects for engineer pairs. The sample is limited to engineers and commenters in the same building on the main campus and hired before the 6-week pre-period. Ribbons reflect 95% confidence intervals with standard errors clustered by team. *p<0.1; **p<0.05; ***p<0.01.

Figure 4: Proximity to Teammates and Engineer Output

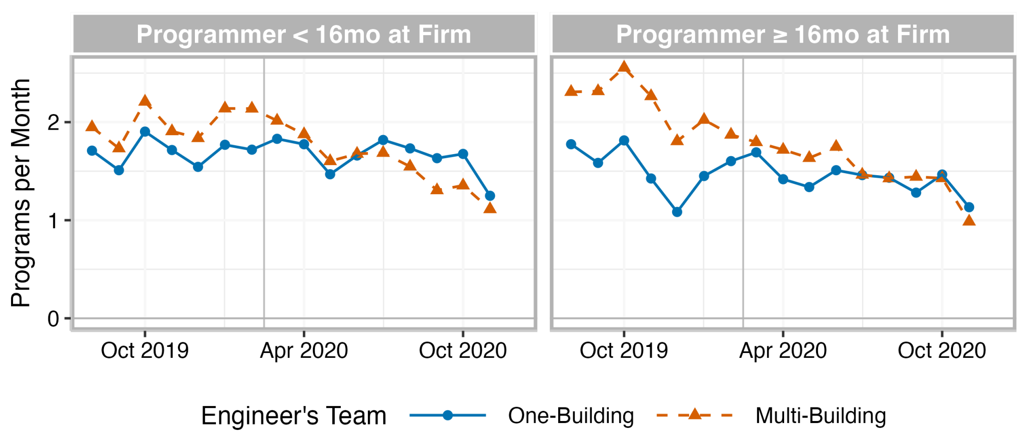
Panel (a): Raw Averages of Programs Written Per Month



Panel (b): Dynamic, Conditional Differences in Programs Per Month

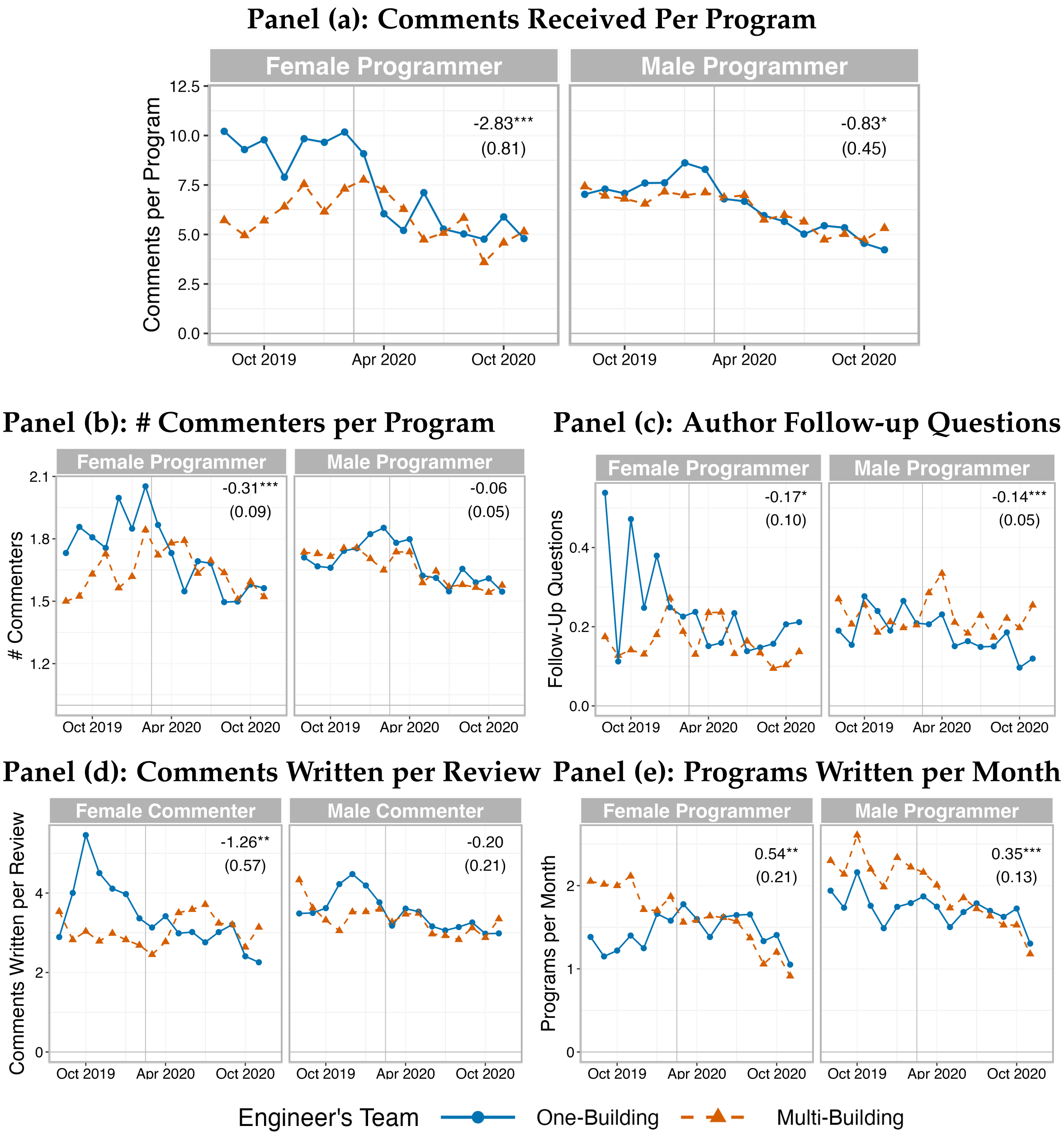


Panel (c): Programs Per Month by Program Writer's Tenure



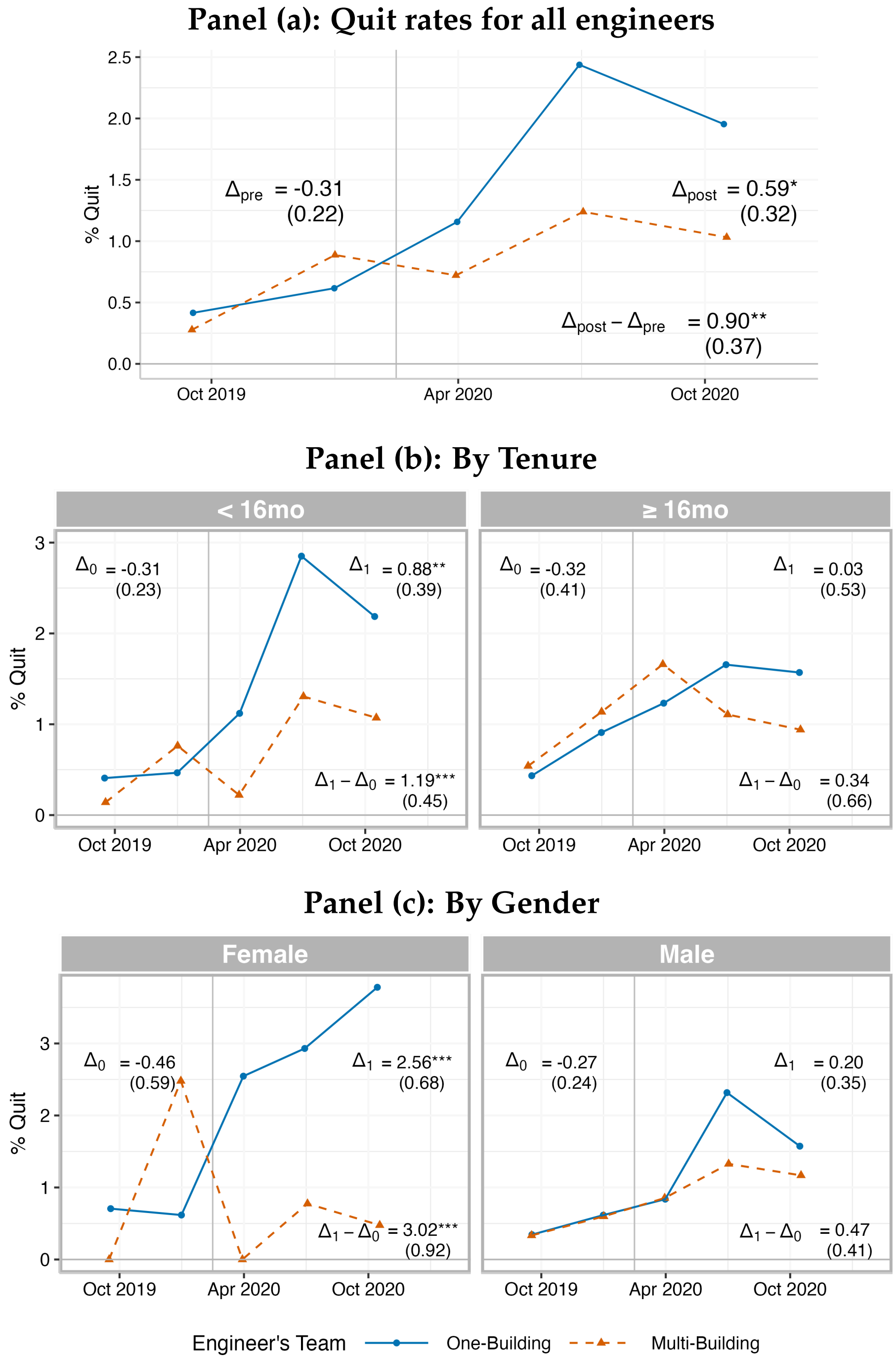
Notes: This figure illustrates the impact of proximity on short-term output. Panel (a) shows the raw monthly averages of the number of programs submitted per month for engineers on one-building and multi-building teams. Panel (b) shows the difference-in-differences estimates with our preferred set of controls, with standard errors clustered by engineering team. Panel (c) plots raw averages separately by those who have been at the firm for longer or shorter than the median tenure (16 months) before the offices closed. *p<0.1; **p<0.05; ***p<0.01.

Figure 5: Gendered Impacts of Proximity to Teammates



Notes: This figure illustrates the gendered impact of proximity on mentorship given and received. Each plot shows the raw monthly averages of comments received per program for engineers on one- and multi-building teams, separately for female and male engineers. Panel (a) shows the average number of comments that engineers receive on their programs. Panel (b) shows the average number of commenters that comment on the engineer's program, which is a proxy for how many programmers the engineer asked to look at their code. Panel (c) shows the average number of follow-up questions that authors ask commenters in the code-review process. Panel (d) turns to giving rather than receiving mentorship, plotting the average number of comments written per code review. Panel (e) shows the average number of programs that engineers submit to the main code-base each month. The sample limits to engineers whose teammates all worked in the main campus. The annotated coefficients represent difference-in-differences specifications for each gender, using our preferred controls with standard errors clustered by engineering team. *p<0.1; **p<0.05; ***p<0.01.

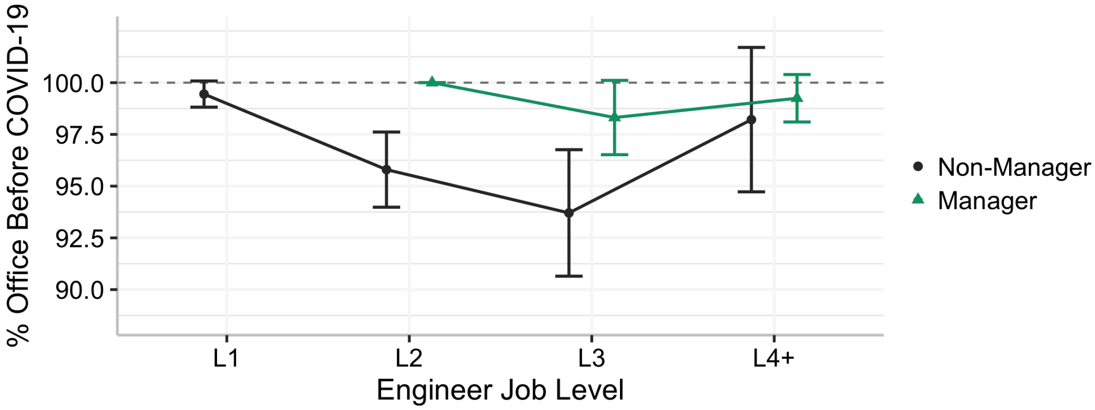
Figure 6: Impacts of Proximity on Quits



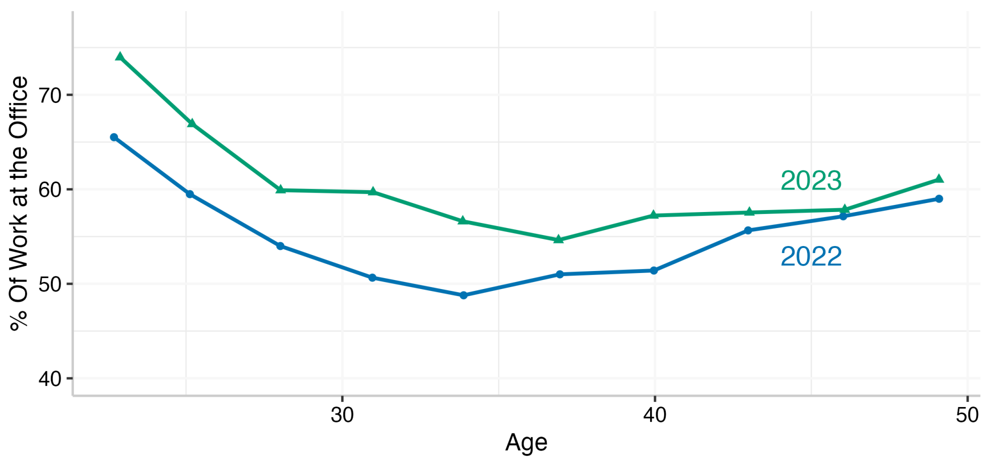
Notes: This figure illustrates the effects of proximity on quits (a) overall, (b) by pre-COVID tenure, and (c) by self-identified gender. Each plot shows the raw quit rates for engineers on one-building and multi-building teams. The annotated coefficients use our preferred set of controls for engineering group and engineer tenure, with standard errors clustered by engineering team. The sample includes engineers who worked on the main code-base, whose teammates were all in the main campus, and who were hired before the start of our sample. Results are similar when focusing on quits to better-paying jobs in Figure A.21. *p<0.1; **p<0.05; ***p<0.01.

Figure 7: Who Works in the Office

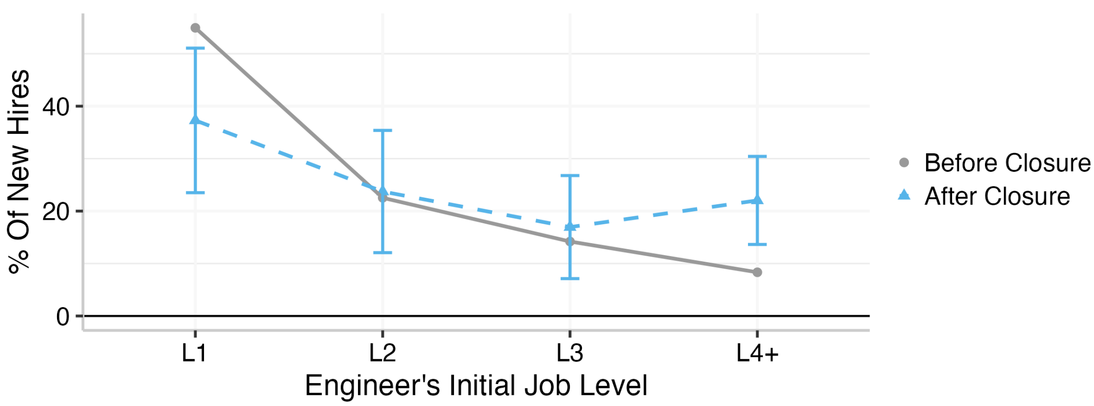
Panel (a): Percent in the Office by Job Level and Manager Status



Panel (b): Percent College-Educated, Non-Parents in the Office Nationally



Panel (c): Workers Hired Before and After the Pandemic by Job Level



Notes: Panel (a) shows the share of engineers of a given job level at the firm who worked on-site rather than remotely before the pandemic's office closures. L1 is the most junior while L4 and above are the most senior engineers. Panel (b) uses data from the Household Pulse Survey, limiting to college-educated workers without children to show the percent of work occurring on-site. Panel (c) shows the share of new hires at each job level before and after the office closures. Error bars reflect 95% confidence intervals.

Table 1: Summary Statistics: One- and Multi-Building Teams

	Mean	One-Building	Multi-Building	Δ_0	
% Work on Internal Tools	60.00	37.23	90.13	-52.90***	–
				(5.26)	
# Teammates	6.09	5.72	6.57	-0.85**	-0.42
				(0.42)	(0.47)
Engineer Traits					
% Female	18.58	19.53	17.33	2.19	-2.75
				(2.78)	(3.26)
% BIPOC	23.71	21.85	26.17	-4.32	-2.12
				(3.07)	(3.71)
% Parent	16.82	17.00	16.54	0.46	8.30
				(4.91)	(6.32)
Age (Years)	28.80	28.54	29.13	-0.59	0.37
				(0.42)	(0.56)
Job Traits					
Firm Tenure (Years)	1.36	1.21	1.56	-0.34***	-0.42***
				(0.11)	(0.12)
Job Level	1.71	1.62	1.82	-0.20***	-0.06
				(0.06)	(0.07)
Salary (in $1,000s)	113.42	111.73	115.67	-3.94***	-0.36
				(1.44)	(1.77)
Manager Traits					
Manager Tenure	2.87	2.84	2.92	-0.08	-0.41
				(0.32)	(0.36)
Manager Job Level	3.30	3.21	3.42	-0.21**	-0.08
				(0.09)	(0.10)
Manager Salary (in $1,000s)	146.10	143.87	149.08	-5.22**	-1.28
				(2.65)	(3.09)
Engineer Group Controls					✓
# Software Engineers	1,055	637	418		
# Teams	304	206	121		

Notes: This table shows traits of the engineers, their job, and their managers in the pre-period before the offices closed for COVID-19. The sample includes engineers whose teams are all in the main campus. Parenting responsibilities come from a June 2020 survey conducted by the firm. Job level refers to the engineer's position within the firm's hierarchy from zero (an intern) to six (senior staff). Engineering group controls account for whether the team works on front-end website design, back-end databases, or internal tools (e.g., for the sales team or supply chain). Standard errors in parentheses are clustered by engineering team. *p<0.1; **p<0.05; ***p<0.01.

Table 2: Proximity to Teammates and Online Feedback

	Comments per Program							
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Post x One-Building Team	-1.29^{***}	-1.58^{***}	-1.52^{***}	-1.16^{***}	-1.15^{***}	-1.18^{***}	-1.51^{***}	-1.14^{**}
	(0.48)	(0.50)	(0.45)	(0.43)	(0.44)	(0.44)	(0.46)	(0.47)
One-Building Team	1.16^{**}	1.45^{**}	2.35^{***}	1.80^{***}	1.78^{***}			
	(0.52)	(0.61)	(0.53)	(0.49)	(0.50)			
Post	-1.22^{***}							
	(0.36)							
Pre-Mean, One-Building Team	8.04	8.04	8.04	8.04	8.04	8.04	8.04	8.04
Percentage Effects								
Post x One-Building Team	-16.1%	-19.7%	-18.9%	-14.4%	-14.3%	-14.6%	-18.7%	-14.1%
One-Building	14.5%	18.1%	29.2%	22.4%	22.2%			
% One-Building Team	58.3	58.3	58.3	58.3	58.3	58.3	58.3	58.3
Engineer Group x Post FE		✓	✓	✓	✓	✓	✓	✓
Program Scope Quartics			✓	✓	✓	✓	✓	✓
Months at Firm x Post FE				✓	✓	✓	✓	✓
Team Size x Post					✓	✓	✓	✓
Engineer FE						✓	✓	✓
Engineer Traits x Post FE							✓	✓
Main Building x Post FE								✓
# Teams	304	304	304	304	304	304	304	304
# Engineers	1,055	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	9,304
R^2	0.02	0.02	0.29	0.36	0.36	0.49	0.50	0.50

Notes: This table investigates the relationship between sitting near teammates and the online feedback that engineers receive on their computer code. Each observation is an engineer-month, and the dependent variable is the average number of comments that the engineer receives per program. Each column estimates the difference-in-differences in Equation 2, which compares the change in feedback for engineers who were in the same building as all of their teammates before COVID-19 to those on teams already distributed across multiple buildings around the office closures. Column 1 presents the raw estimates. Column 2 includes time-varying controls for engineering group (e.g., front-end website design). Column 3 adds controls for program scope (quartics for the number of lines added, number of lines deleted, and number of files changed). Column 4 allows for differential changes in feedback for engineers with different tenure (in months). Column 5 allows for differential changes in feedback depending on team size. Column 6 includes engineer fixed effects. Column 7 adds additional controls for engineer age (in years), gender, whether the engineer is a person of color, home zipcode, and job level. Columns 8 includes building-by-post fixed effects to allow programmers who sat in the main and auxiliary buildings to experience different changes in feedback around the office closures. The sample includes engineers who submit programs to the firm's main code-base in the month and whose teams are all in the firm's main campus. Standard errors are clustered by team. *p<0.1; **p<0.05; ***p<0.01.

Table 3: Proximity and Dimensions of Online Feedback

Panel (a): Feedback Length, Delay, and Mentions of Other Online Conversations

	Comments per Program (1)	Total Characters (2)	Hours to Comment (3)	% Other Online Convo (4)
Post x One-Building Team	-1.16^{***}	-135.70^{**}	1.01^{*}	-1.23
	(0.43)	(57.93)	(0.60)	(0.82)
One-Building Team	1.80^{***}	201.60^{***}	-1.43^{**}	2.39^{***}
	(0.49)	(62.44)	(0.59)	(0.88)
Pre-Mean, One-Building Team	8.04	833.24	16.02	4.06
Post x One-Building Team as %	-14.4%	-16.3%	6.3%	-30.5%
One-Building Team as %	22.4%	24.2%	-8.9%	58.8%

Panel (b): Intensive and Extensive Margins

	Comments per Commenter (1)	Commenters per Program (2)
Post x One-Building Team	-0.35^{*}	-0.10^{**}
	(0.20)	(0.05)
One-Building Team	0.51^{**}	0.20^{***}
	(0.21)	(0.06)
Pre-Mean, One-Building Team	4.36	1.77
Post x One-Building Team as %	-8%	-5.7%
One-Building Team as %	11.7%	11.5%

Panel (c): Back-and-Forth Conversations

	Commenter's	Program Writer		Commenter's
	Initial Comments	Replies	Questions	Follow-up Comments
	(1)	(2)	(3)	(4)
Post x One-Building Team	-0.48^{*}	-0.63^{**}	-0.14^{***}	-0.68^{**}
	(0.26)	(0.26)	(0.05)	(0.34)
One-Building Team	0.76^{***}	0.68^{**}	0.11^{**}	1.04^{***}
	(0.26)	(0.29)	(0.05)	(0.39)
Pre-Mean, One-Building Team	4.91	2.14	0.24	3.13
Post x One-Building Team as %	-9.8%	-29.4%	-58.3%	-21.8%
One-Building Team as %	15.5%	31.6%	47.5%	33.2%

Notes: This table considers alternative metrics of online feedback: (a) the extent and timeliness of feedback, (b) the intensive and extensive margins of feedback, and (c) the back-and-forth conversation between the commenter and program writer. Each specification replicates Column 4 of Table 2, reported in Column 1 of Panel (a) for reference. $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$.

Table 4: 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***	0.41***	0.40***	0.35***	0.33***	0.32**
	(0.11)	(0.13)	(0.13)	(0.13)	(0.11)	(0.12)	(0.13)
One-Building Team	−0.48***	−0.33**	−0.32**	−0.33**			
	(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%	24.1%	23.6%	20.2%	19%	18.9%
One-Building Team as %	-28.1%	-19.1%	-19%	-19.5%			
R^2	0.01	0.05	0.08	0.08	0.46	0.46	0.46
	Panel (b): Lines Added per Month						
Post x One-Building Team	105***	92**	93**	94**	92**	83**	117***
	(36)	(39)	(40)	(40)	(38)	(39)	(42)
One-Building Team	−193***	−158***	−169***	−173***			
	(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%	29%	29.2%	28.8%	26%	36.5%
One-Building Team as %	-60.2%	-49.2%	-52.7%	-54%			
R^2	0.02	0.03	0.04	0.04	0.34	0.35	0.35
	Panel (c): Files Changed per Month						
Post x One-Building Team	1.93**	1.66	1.47	1.46	1.36	1.13	1.69
	(0.97)	(1.05)	(1.06)	(1.07)	(1.00)	(1.02)	(1.12)
One-Building Team	−3.97***	−3.62***	−3.73***	−3.85***			
	(1.12)	(1.15)	(1.17)	(1.15)			
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%	15.2%	15.2%	14.1%	11.8%	17.5%
One-Building Team as %	-41.2%	-37.5%	-38.7%	-39.9%			
R^2	0.01	0.02	0.04	0.04	0.31	0.32	0.32
Engineer Group x Post FE		✓	✓	✓	✓	✓	✓
Months at Firm x Post FE			✓	✓	✓	✓	✓
Team Size x Post				✓	✓	✓	✓
Engineer FE					✓	✓	✓
Engineer Traits x Post FE						✓	✓
Main Building x Post 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 2, with controls defined in Table 2. 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 main campus. *p<0.1; **p<0.05; ***p<0.01.

Table 5: Effect of Proximity to Teammates on Pay Raises

	% Pay Raise					
	Offices Open			Offices Closed		
	(1)	(2)	(3)	(4)	(5)	(6)
One-Building Team	-3.38^{*}			5.48		
	(2.05)			(3.53)		
Junior (< 16mo) x One-Building Team		-4.86^{**}			7.17^{*}	
		(2.26)			(3.89)	
Senior ($\geq$ 16mo) x One-Building Team		-0.07			2.06	
		(3.89)			(4.96)	
Female x One-Building Team			-6.20			11.75^{*}
			(4.61)			(7.12)
Male x One-Building Team			-2.79			4.26
			(2.13)			(3.64)
Dependent Mean	19.23	19.23	19.23	40.94	40.94	40.94
Junior (< 16mo) Mean	17.16	17.16	17.16	17.16	17.16	17.16
Senior ($\geq$ 16mo) Mean	23.46	23.46	23.46	42.41	42.41	42.41
Female Mean	20.63	20.63	20.63	41.46	41.46	41.46
Male Mean	18.92	18.92	18.92	40.83	40.83	40.83
Percentage Effect						
One-Building Team	-17.6%			13.4%		
Junior (< 16mo) x One-Building Team		-28.3%			17.9%	
Senior ($\geq$ 16mo) x One-Building Team		-0.3%			4.9%	
Female x One-Building Team			-30.1%			28.3%
Male x One-Building Team			-14.8%			10.4%
Preferred Controls	✓	✓	✓	✓	✓	✓
# Teams	262	262	262	256	256	256
# Engineers	801	801	801	720	720	720
# Engineer-Review	1,988	1,988	1,988	1,851	1,851	1,851

Notes: This table investigates how the likelihood of a pay raise differs for engineers on one-building teams while the offices were open (Columns 1−3) and after the offices closed (Columns 4−6). Each column includes our preferred, time-varying controls for engineering type and firm tenure. Each observation is an engineer for a given tri-annual review-period that ends in October, March, or July. The March 2020 review period is included in the pre-period since it is based on pre-closure performance. The sample includes engineers who worked on the main code-base, whose teammates were all in the main campus, and who were hired before the start of our sample. Standard errors are clustered by team. *p<0.1; **p<0.05; ***p<0.01.

A APPENDIX

I.A Principal Component Analysis

To identify common themes in the data, we use principal component analysis (PCA). We first transform each comment into a vector of words. We then strip the comments of "stop words," such as "the", "a", "we", and "she" and use stemming to join together words like "test," "tests," and "testing." We then limit to words that appear in at least one hundred distinct comments. We then count the number of times each of these words appears in each comment. Even this more parsimonious representation of the code-review data is high-dimensional, with over a thousand variables. To interpret the data, we reduce its dimensionality with PCA. PCA transforms the data into a new coordinate system, in which most of the variation can be described in fewer dimensions. Table A.1 shows illustrative examples of the primary principal components. Figure A.2 show how these components differ across one- and multi-building teams before and after the offices closed.

I.B Crowdsourced 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. Table A.2 shows a random subsample of comments along with their evaluations.

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 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)

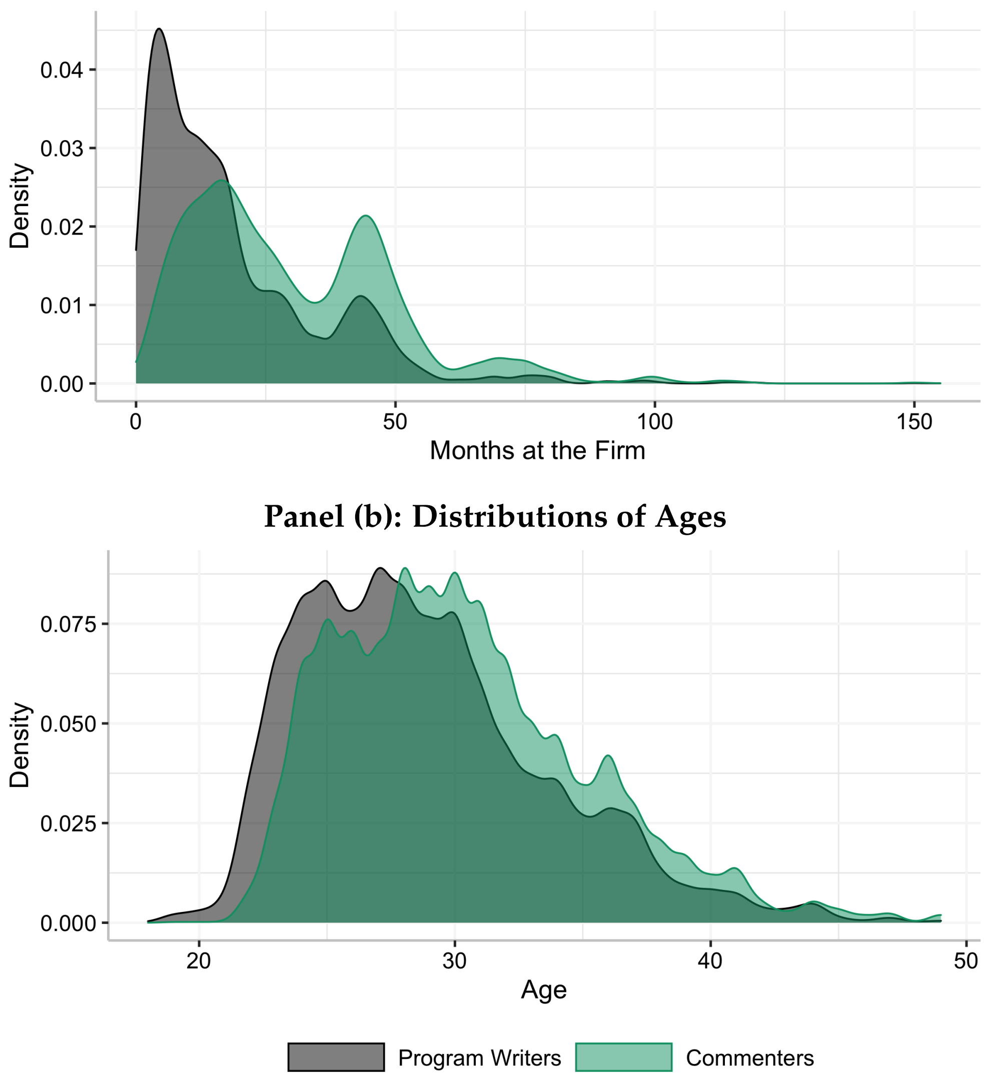
- Which of the following data structures is typically used to implement a Last-In-First-Out (LIFO) behavior?
  - Linked-List
  - Queue
  - Hash Table
  - Stack

We included five overlapping comments to calculate measures of inter-rater reliability. The engineers' responses adhered with the median response more than seventy percent of the time for each of the rated dimensions. The engineers have Fleiss' Kappa measure of inter-rater reliability of $\kappa = 0.09$ for comment helpfulness, $\kappa = 0.23$ for actionability, $\kappa = 0.28$ for implementability, $\kappa = 0.16$ for including rationale, and $\kappa \approx 0$ for the tone of the comment, which they deemed to be almost never rude.

I.C Figures & Tables

Figure A.1: Program Writer and Commenter Age and Experience

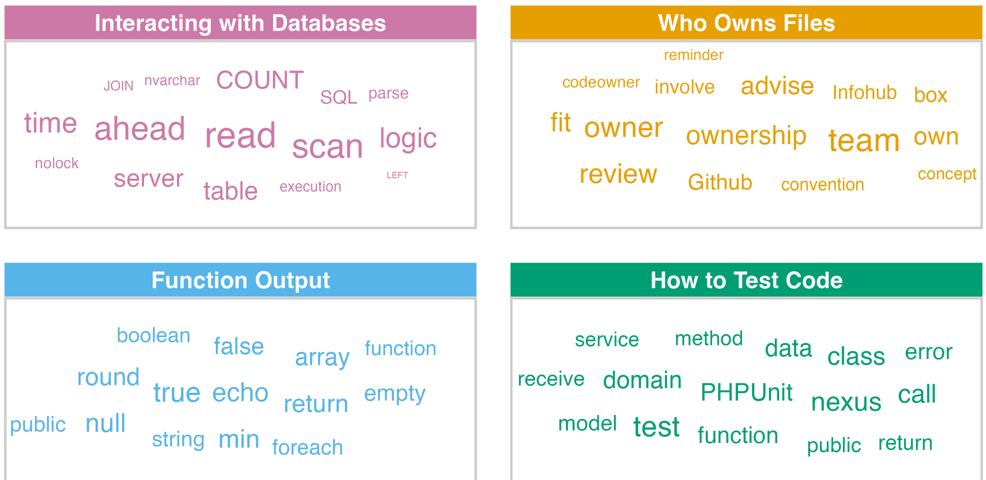
Panel (a): Distributions of Firm Tenure



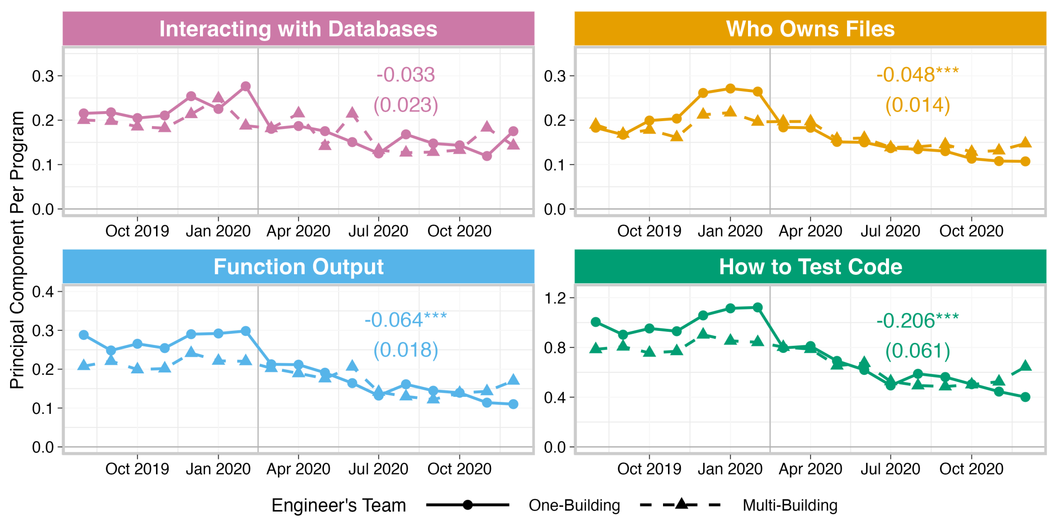
Notes: These figures contrast the experience of program writers and comments (a) at the firm and (b) in their careers. The grey histogram shows the densities for engineers who write programs, weighted by the number of programs that they write. The green distributions show the densities for engineers who write comments on code, again weighted by the number of programs they comment upon.

Figure A.2: Themes in Programmers' Online Peer Feedback

Panel (a): Word Loadings in Principal Components

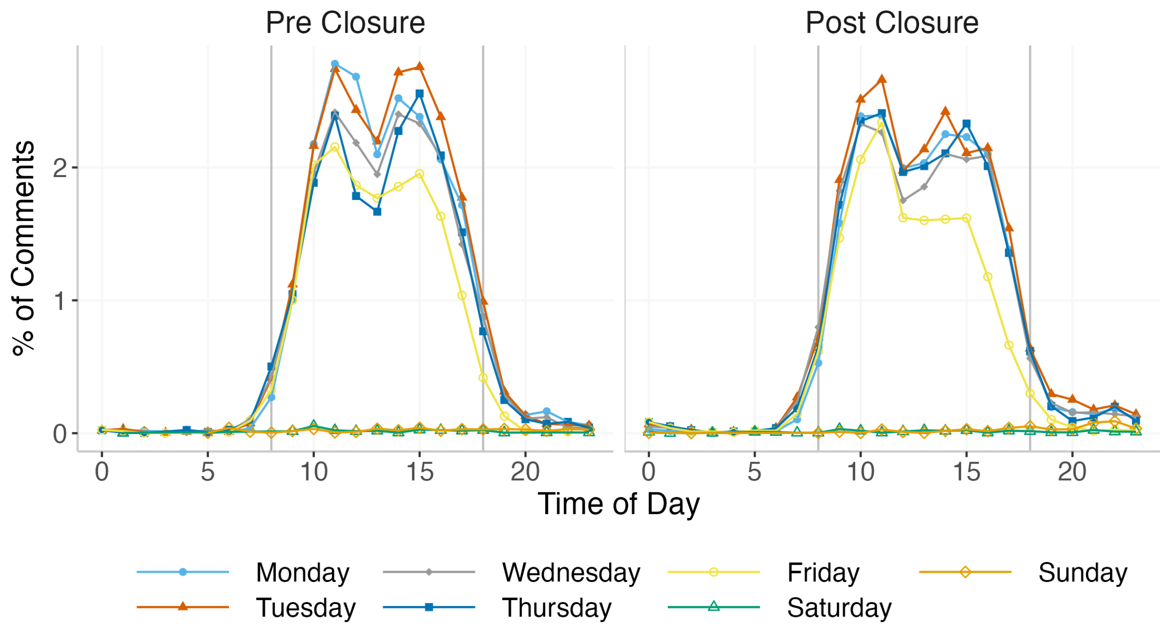


Panel (b): Proximity and Themes in Feedback



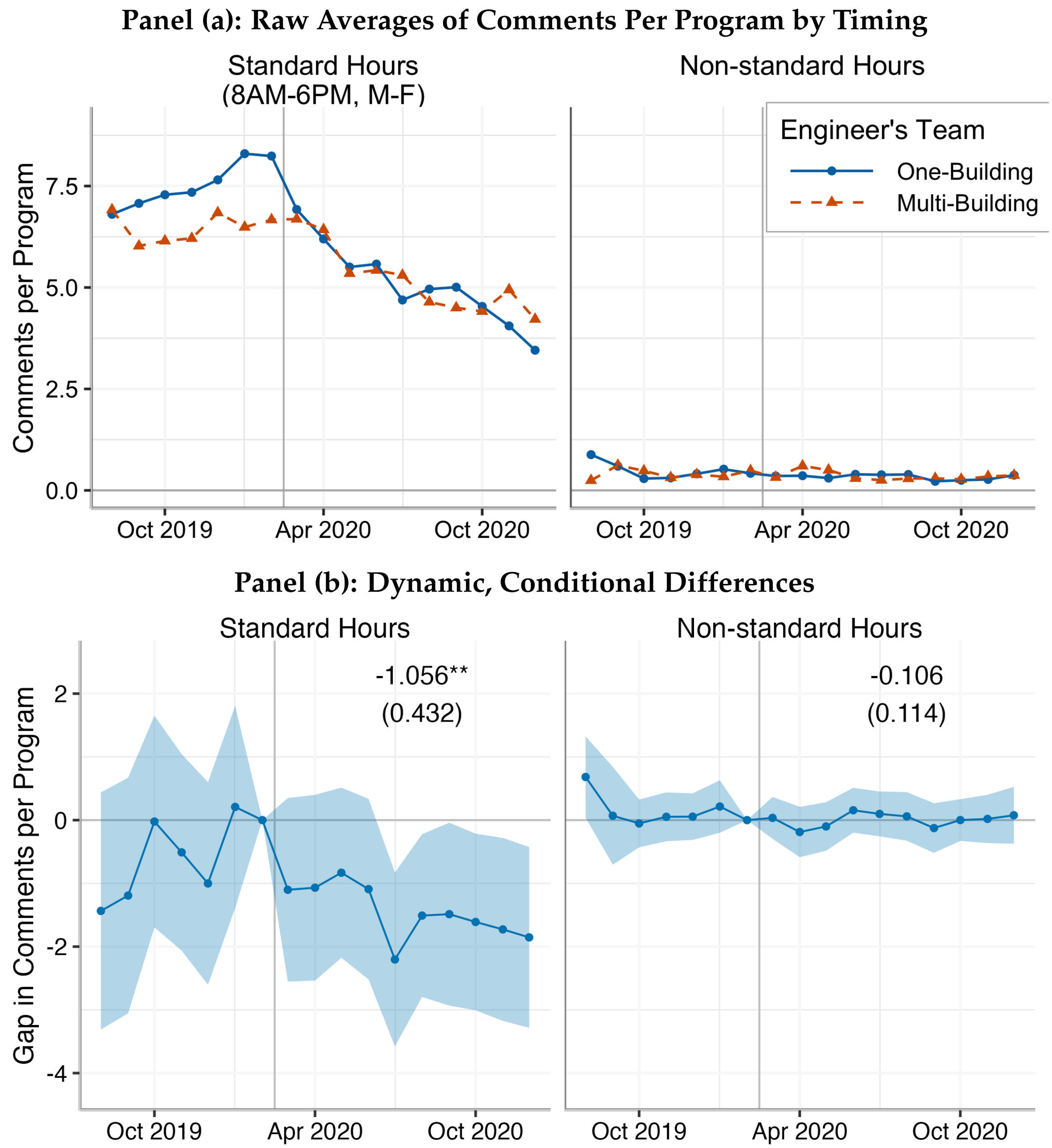
Notes: This figure illustrates the common themes in programmers' online feedback to one another, and how proximity impacts these themes. These themes are identified using principal component analysis (see Section I.A for details and Table A.1 for emblematic example comments). Panel (a) presents the words with the highest loading for each component, with the size reflecting the loading. The first component identifies comments that are about how to *read* data from databases often in the structured query language, *SQL*. The second component identifies two, typically non-overlapping groups of comments. One is about asking *owners* of code on *Github* to *review* suggested changes. The other is about what *functions return*, with special attention to edge cases, like *null* values and *empty arrays*. The final component shown here is about *test*ing code, often using the testing suite *PHPUnit*. Panel (b) replicates the analysis in Figure 1 for these components in the comments on each program. The annotated coefficient is the difference-in-differences estimate conditional on our preferred controls. Standard errors are clustered by engineering team. *p<0.1; **p<0.05; ***p<0.01.

Figure A.3: Timing of Comments Over the Course of the Day



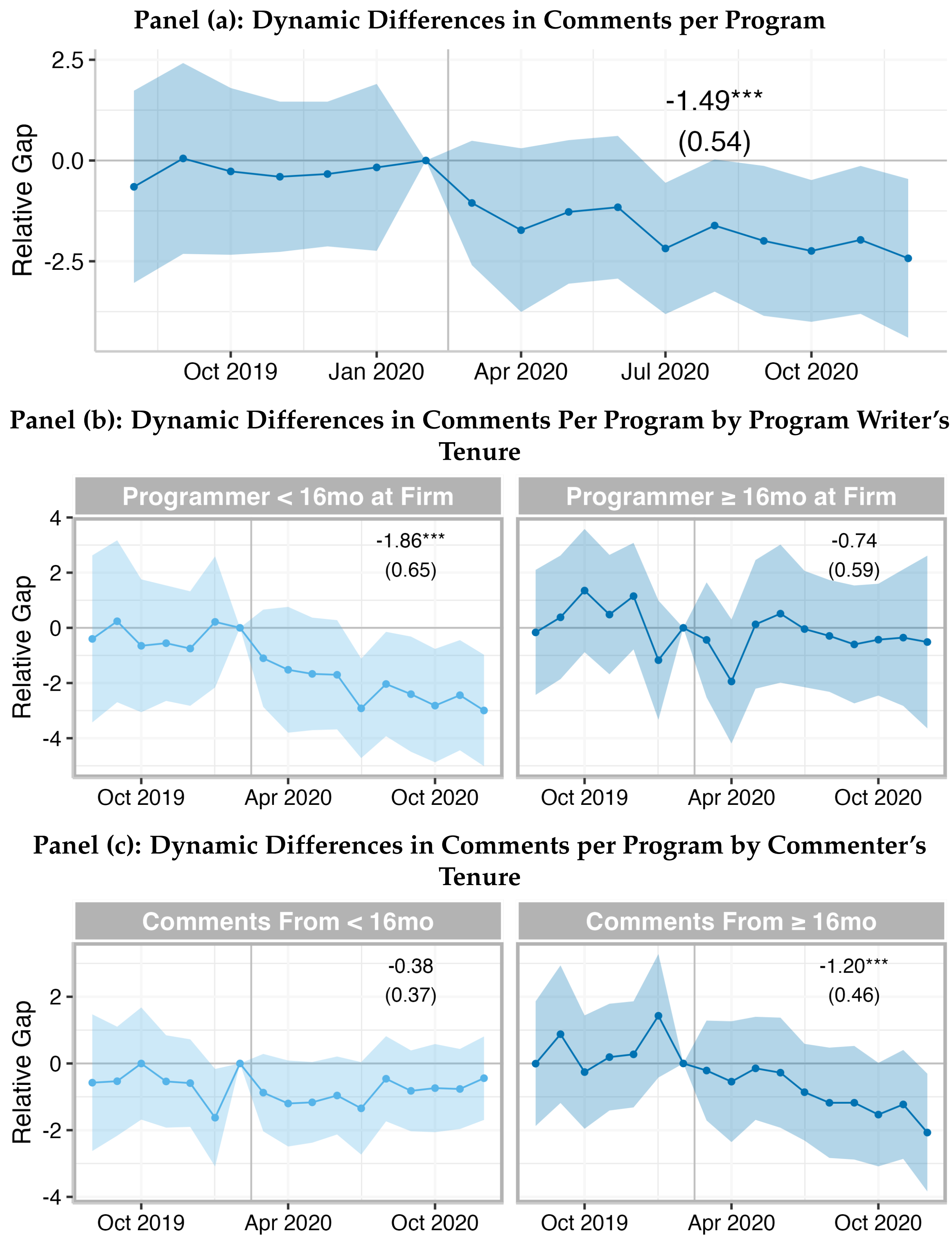
Notes: This figure plots the timing of comments over the course of the day. The x-axis plots the time of day in military time. The y-axis is the percent of comments that occur in that particular time of day on that particular day of week. The left plot is the period before the office closures of COVID-19. The right plot is the period after the office closures of COVID-19. The vertical lines highlight typical office hours from 8am to 6pm.

Figure A.4: Proximity to Teammates and Online Feedback Inside and Outside of Standard Work Hours



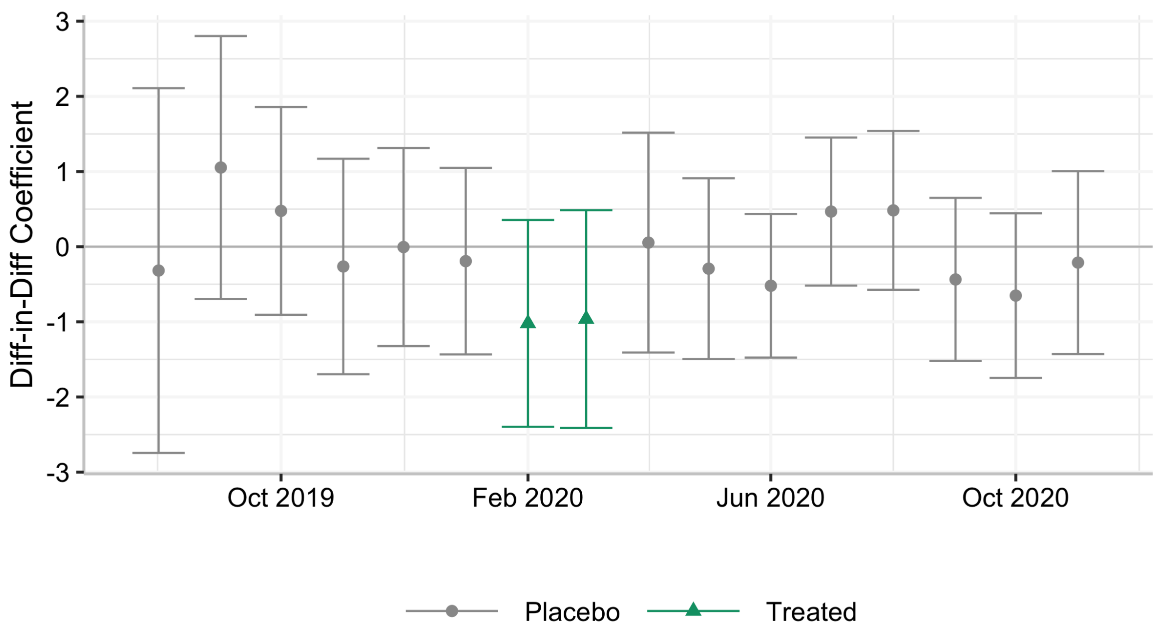
Notes: This figure illustrates the online feedback received by engineers in one-building teams (N=637) and engineers on multi-building teams (N=418) before and after the offices closed for COVID-19 (the grey vertical lines). The left plots consider comments given in standard work hours (8AM to 6PM, Monday through Friday); the right plots consider comments given in other times. Panel (a) plots the raw averages, while Panel (b) plots the differences, conditional on our preferred controls for program scope, engineering type, and tenure. The ribbon is a 95% confidence interval with clustering by engineering team. The annotated coefficient is the difference-in-differences estimate from Equation 2. Only engineers whose teammates all worked in in the main campus are included. *p<0.1; **p<0.05; ***p<0.01.

Figure A.5: Proximity to Teammates' Impact, Conditional on Engineer Tenure by Age



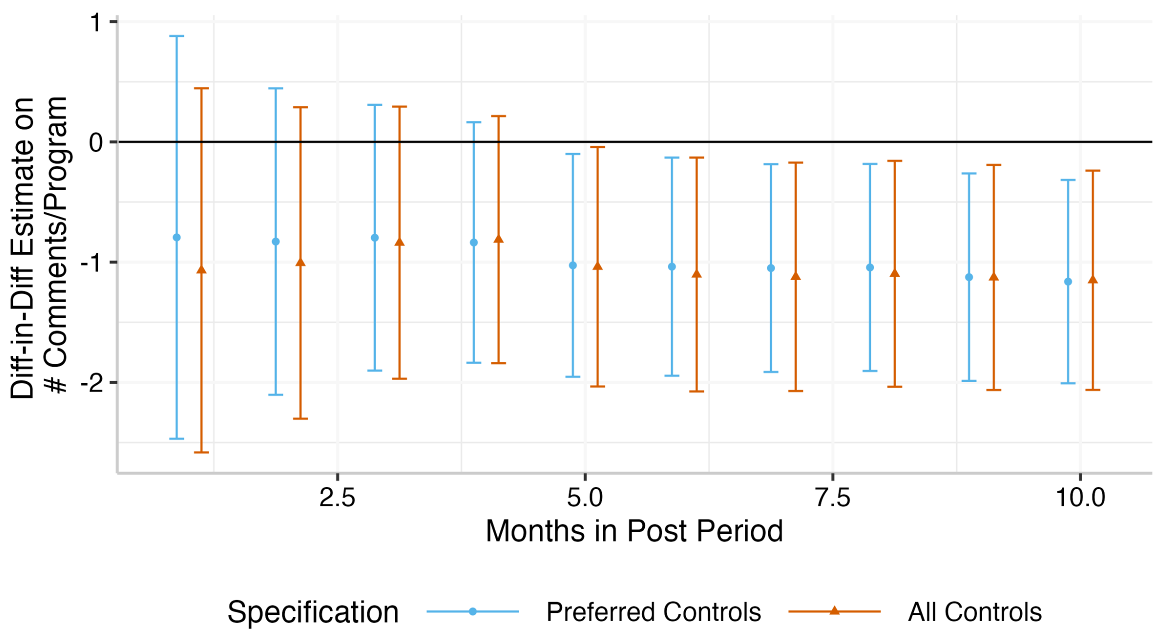
Notes: Panel (a) replicates the analyses in Figure 1(b), including additional time-varying controls for tenure and age. This allows us to remove the effect of a hiring spell that occurred in the pre-period. Panels (b) and (c) replicate Figure 2 with the same set of controls.

Figure A.6: Placebo Treatment Dates' Effects of Proximity on Online Feedback from Coworkers



Notes: This figure illustrates difference-in-differences estimates that compare the change in comments for engineers on one- and multi-building teams in two-month bandwidths. The grey circles show periods that do not include the treated window; the green triangles include the treated window. All regressions include our preferred controls for engineering type, engineer tenure, and program scope (in column four of Table 2). The error bars are 95% confidence intervals with standard errors clustered by engineering team.

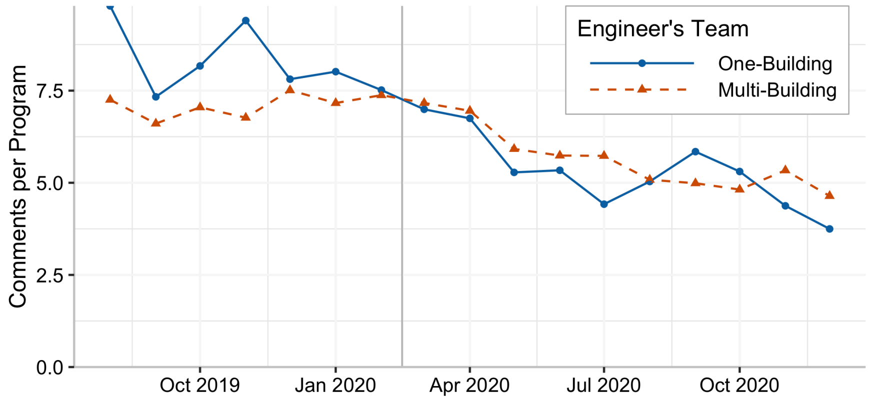
Figure A.7: Robustness of the Effect of Proximity on Online Feedback from Coworkers to Alternative Post-Periods



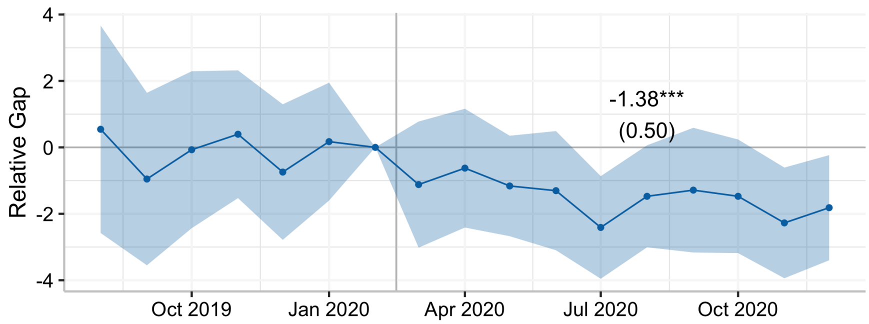
Notes: This figure illustrates how the difference-in-differences estimate from Equation 2 — that compares engineers on one- and multi-building teams, before and after the office closures — varies with the number of months in the post period. The blue circles are the coefficients using our preferred controls for engineering type, engineer tenure, and program scope (in column four of Table 2); the red triangles are the coefficients using the full set of controls (in column six of Table 2). The error bars are 95% confidence intervals with standard errors clustered by engineering team.

Figure A.8: Proximity to Teammates and Online Feedback for Engineers Working on Internal Tools

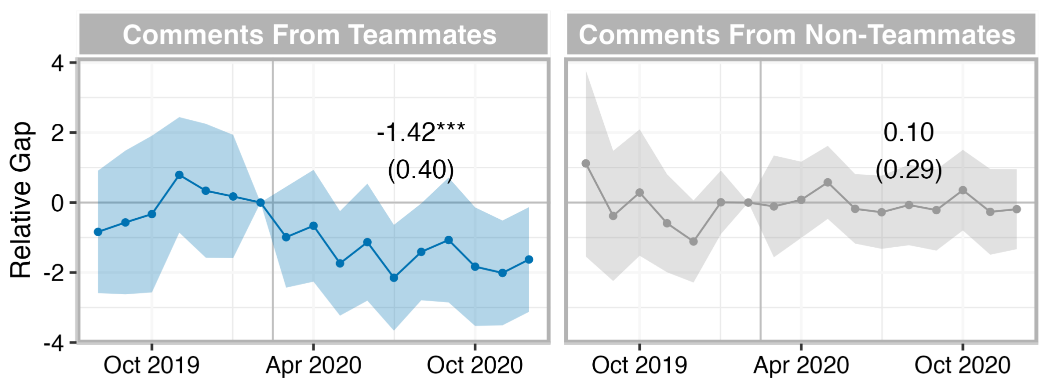
Panel (a): Raw Averages of Comments Per Program



Panel (b): Dynamic, Conditional Differences in Comments Per Program

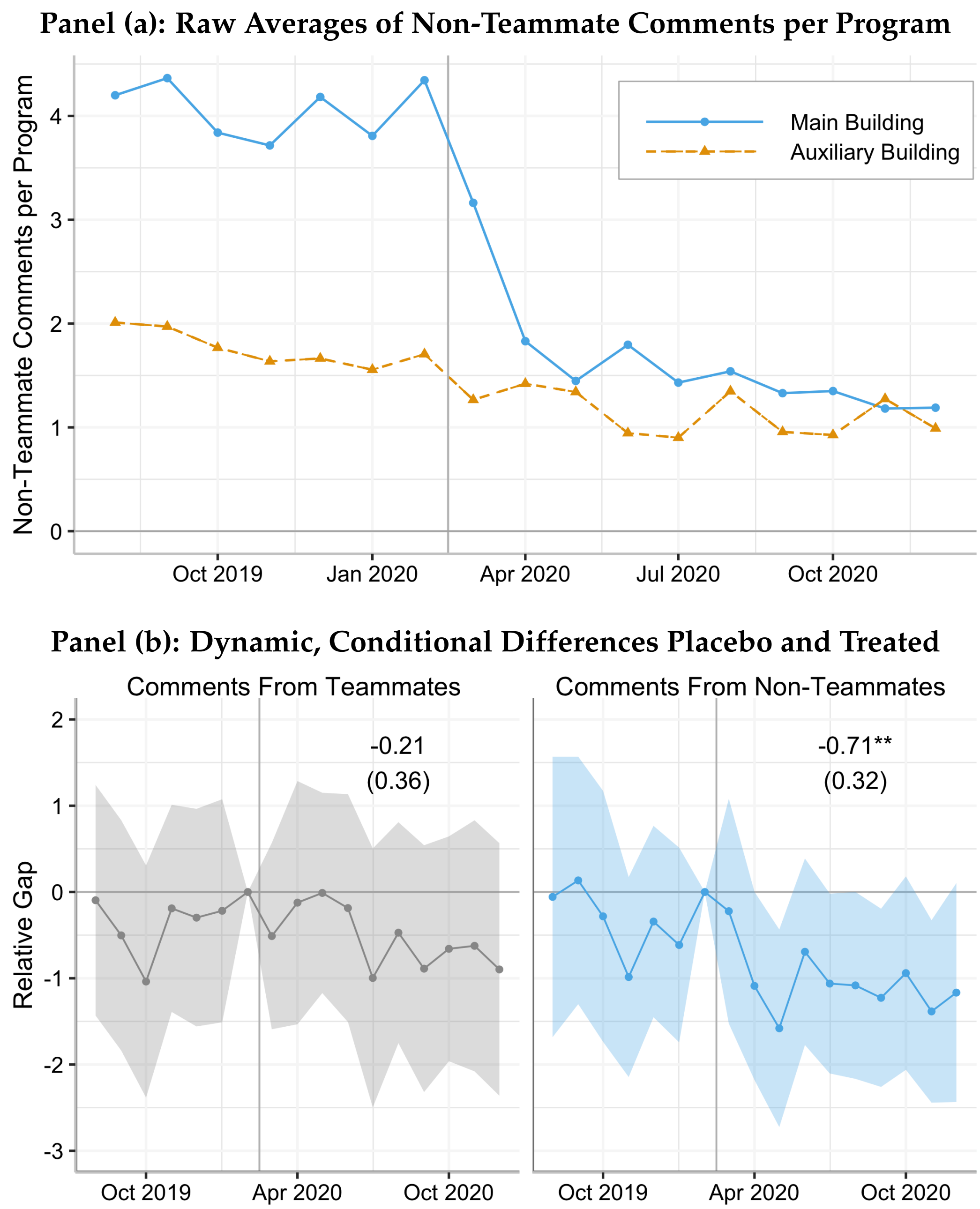


Panel (c): Placebo Check with Comments from Teammates or Non-Teammates



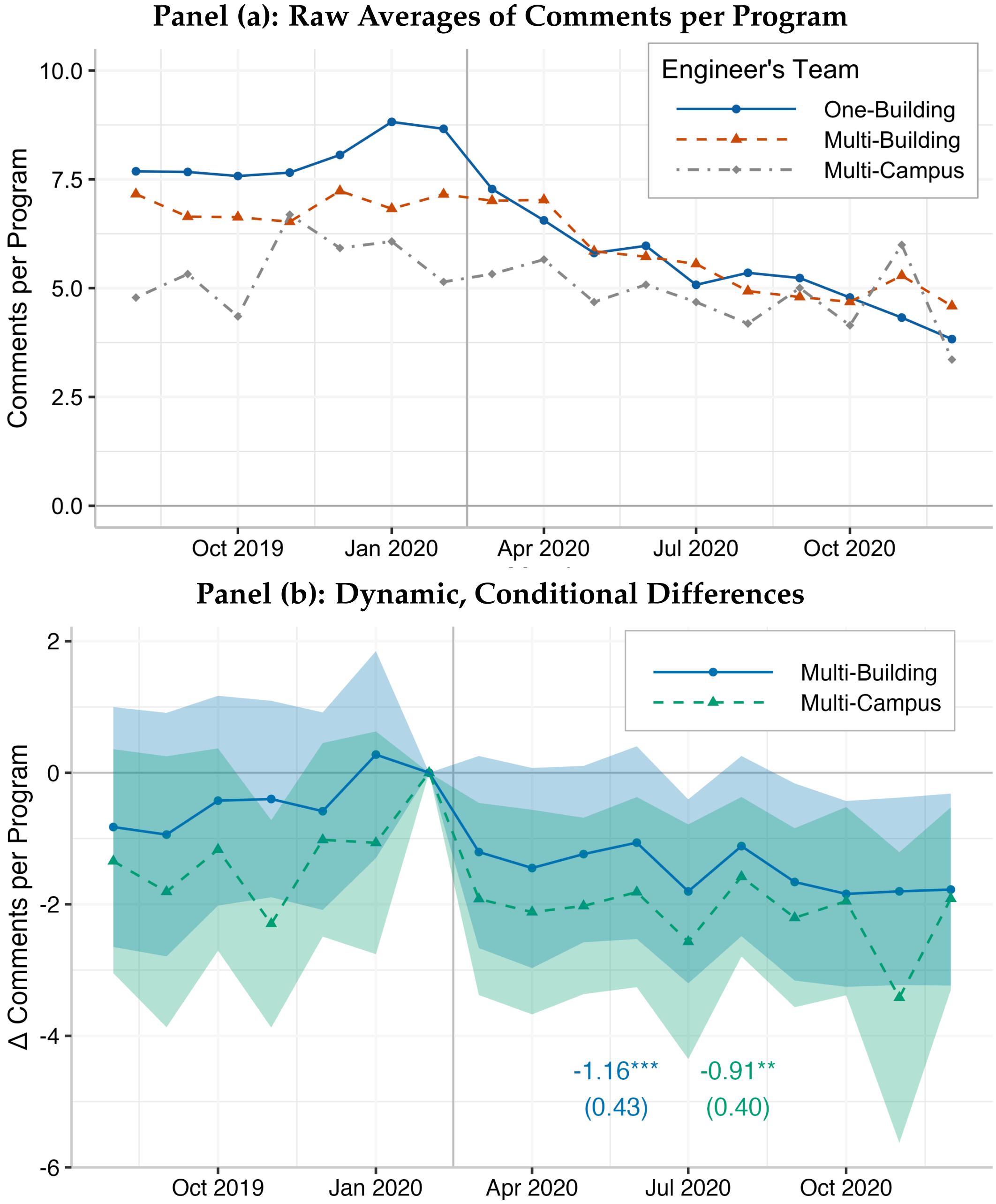
Notes: This figure replicates Figure 1 but limits the sample to the 588 engineers who built internal tools (i.e., software used by others in the firm). Since it is useful for these workers to sit near those who use their tools (e.g., sales workers), it is more likely that their teams end up split across the two office buildings. The analysis compares engineers who had sat with all their teammates in the same building before the offices closed (N=215) to engineers on multi-building teams (N=373). *p<0.1; **p<0.05; ***p<0.01.

Figure A.9: Proximity to Non-Teammates



Notes: This figure compares the feedback received by the engineers who sat in the main building (N=788) and the engineers who sat in the auxiliary building (N=277) around the office closures. Panel (a) plots raw averages in the number of non-teammate comments that engineers receive on their code. Panel (b) presents the conditional differences between engineers in the main and auxiliary buildings, controlling for our preferred controls and the engineers' proximity to their teammates (the analogue of Equation 2 for building rather than team-type). The left plot shows a placebo check with teammate comments. The right plot shows non-teammate comments which should be impacted. Ribbons are 95% confidence intervals with clustering by engineering team. Only engineers whose teammates all worked in the main campus are included. The grey vertical lines mark the COVID-19 office closures. *p<0.1; **p<0.05; ***p<0.01.

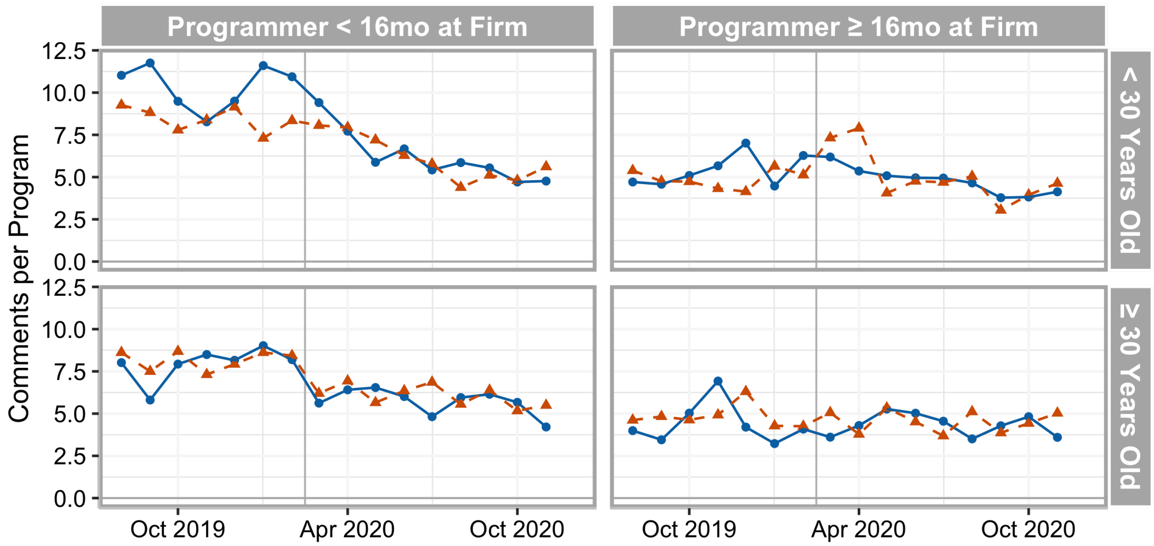
Figure A.10: Proximity to Teammates, by Distance



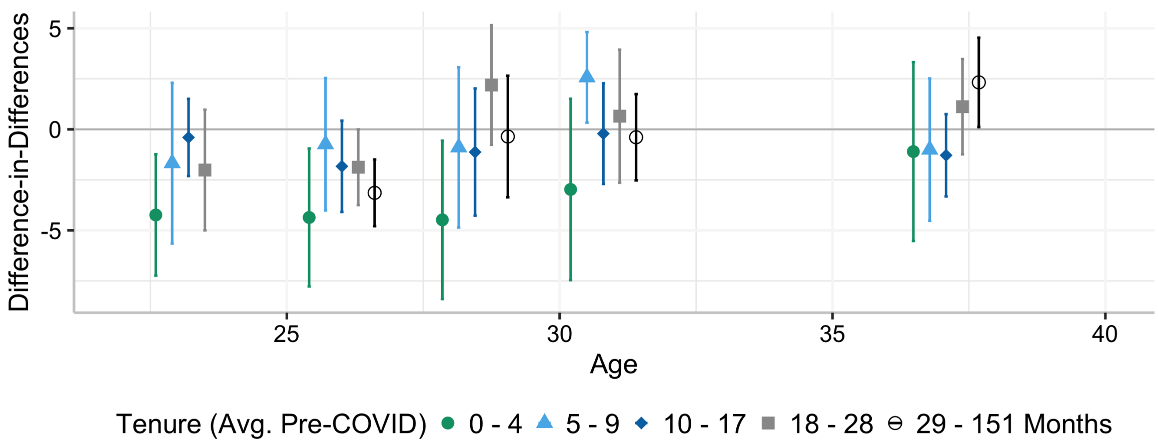
Notes: This figure illustrates the online feedback received by engineers in one-building teams (N=637), multi-building, single-campus teams (N=418), and multi-campus teams (N=215) before and after the offices closed for COVID-19 (the grey vertical lines). The sample includes all engineers who are all themselves in the main campus, regardless of their teammates' locations. Panel (a) plots the raw averages; Panel (b) plots the differences from Equation 3, conditional on our preferred set of controls listed in Subsection III.B. The ribbons reflect 95% confidence intervals with clustering by engineering team. The annotated coefficients come from the analogue of Equation 2. *p<0.1; **p<0.05; ***p<0.01.

Figure A.11: Proximity and Mentorship of Young Engineers

Panel (a): Comments per Program by Program Writer's Tenure and Age



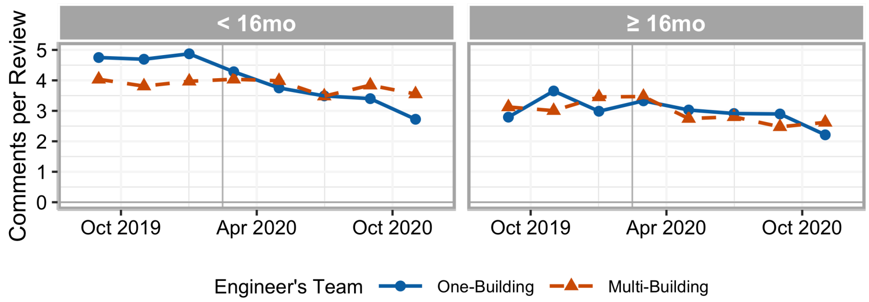
Panel (b): Diff-in-Diff in Comments Per Program by Age and Tenure



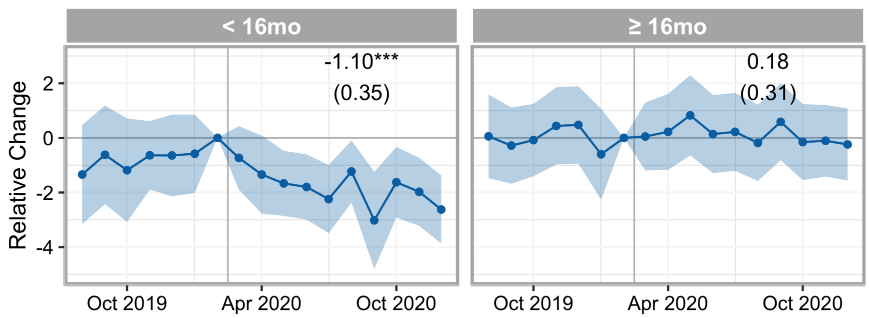
Notes: Panel (a) shows the effects of proximity on online feedback received by engineers of different tenures and ages. It shows the raw monthly averages of comments received per program for engineers on one- and multi-building teams, separately by those below and above the median tenure of 16 months and those below and above the average age of 30. Panel (b) shows the estimated difference-in-differences coefficient from Equation 2 for different age quintiles separately by tenure quintile. Each specification includes our preferred controls for program scope, programmer tenure, and engineering type. Error bars represent 95% confidence intervals with standard errors clustered by team.

Figure A.12: Externalities from Distant Teammates By Tenure

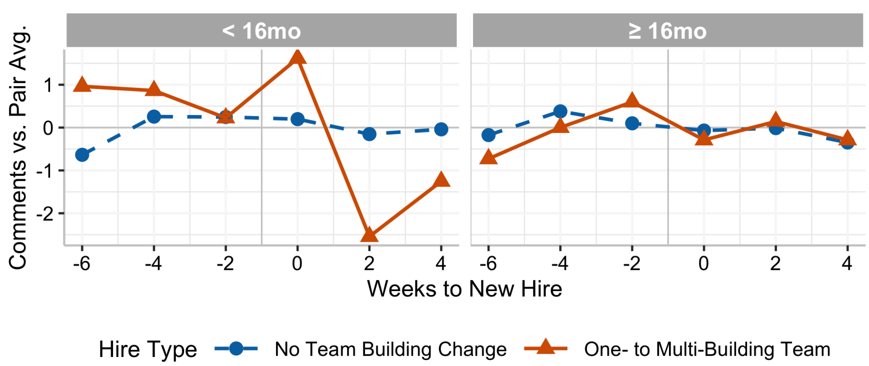
Panel (a): Raw Comments from Same-Building Teammates around Closures



Panel (b): Dynamic, Conditional Differences



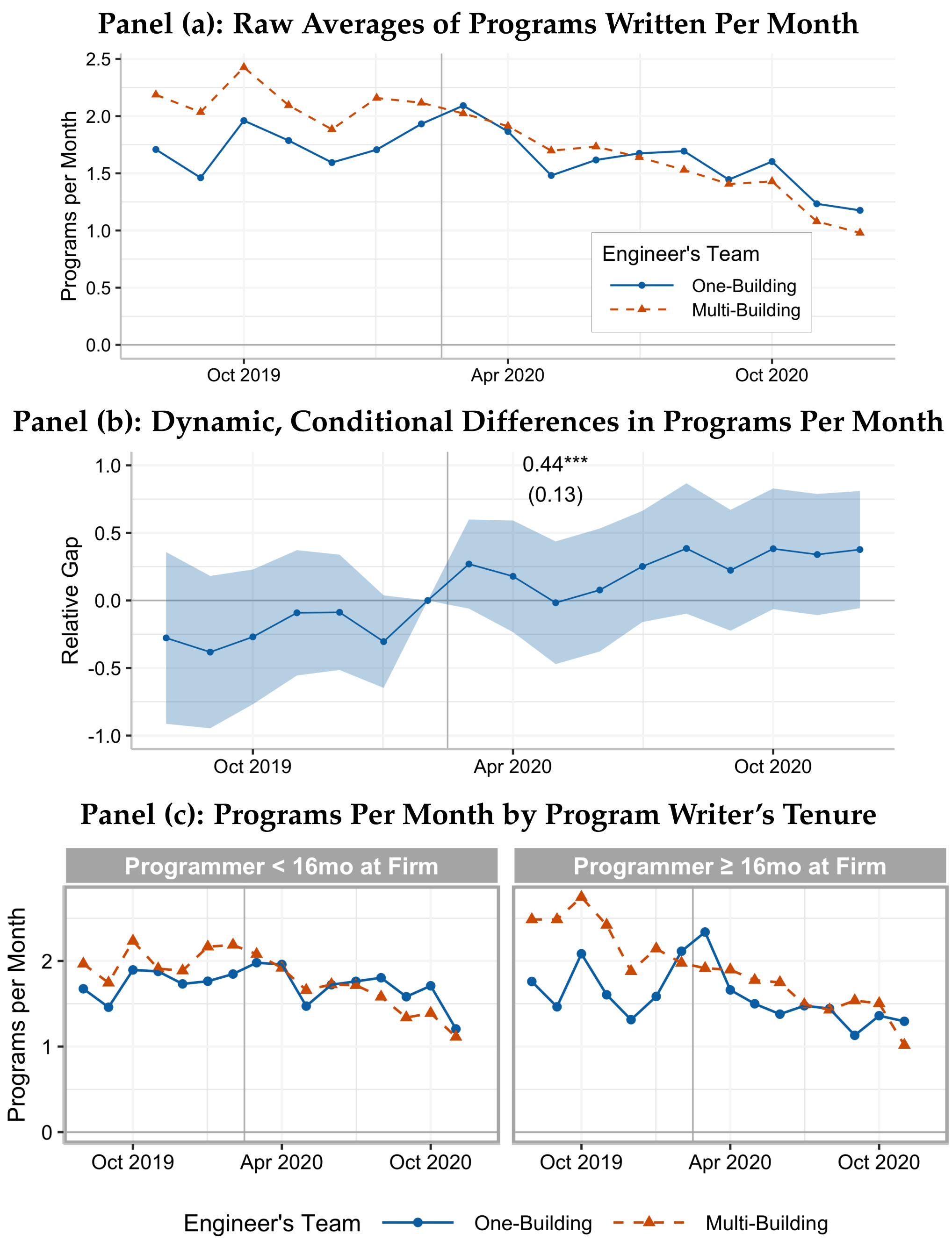
Panel (c): Raw Comments/Review from Same-Building Teammate around a Hire



Panel (d): Dynamic, Conditional Differences

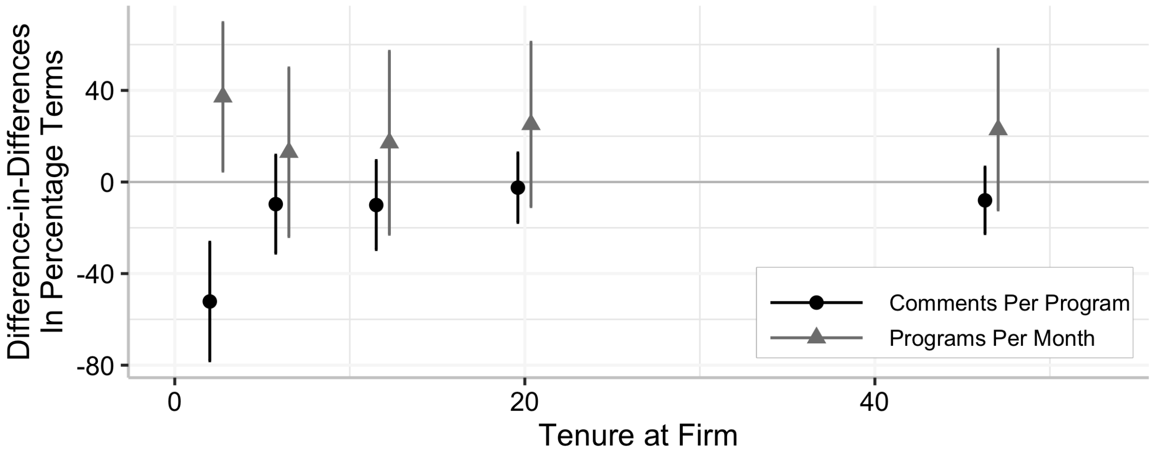
Notes: This figure replicates Figure 3 but differentiates between more and less experienced engineers.

Figure A.13: Proximity to Teammates and Engineer Output for Engineers who Build Internal Tools



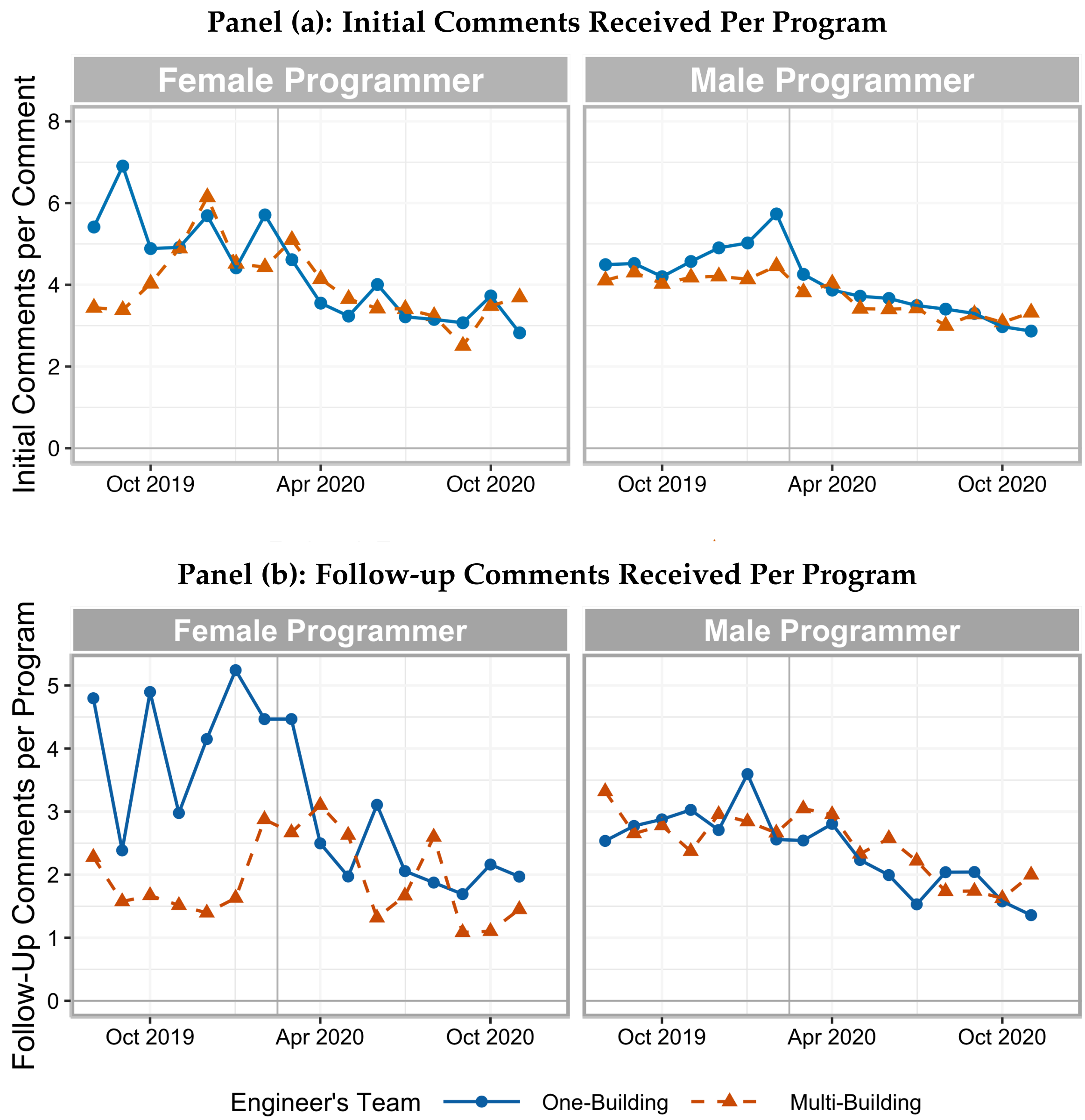
Notes: This figure replicates Figure 4 but limits the sample to the 588 engineers who worked on internal tools (i.e., software used by others in the firm). Since it is useful for these workers to sit near those who use their tools (e.g., sales workers), it is more likely that these teams are split across the two office buildings. The analysis compares engineers who had sat with all their teammates in the same building before the offices closed (N=215) to engineers on multi-building teams (N=373). Standard errors are clustered by engineering team. *p<0.1; **p<0.05; ***p<0.01.

Figure A.14: Proximity to Teammates and Engineer Output and Feedback by Baseline Tenure



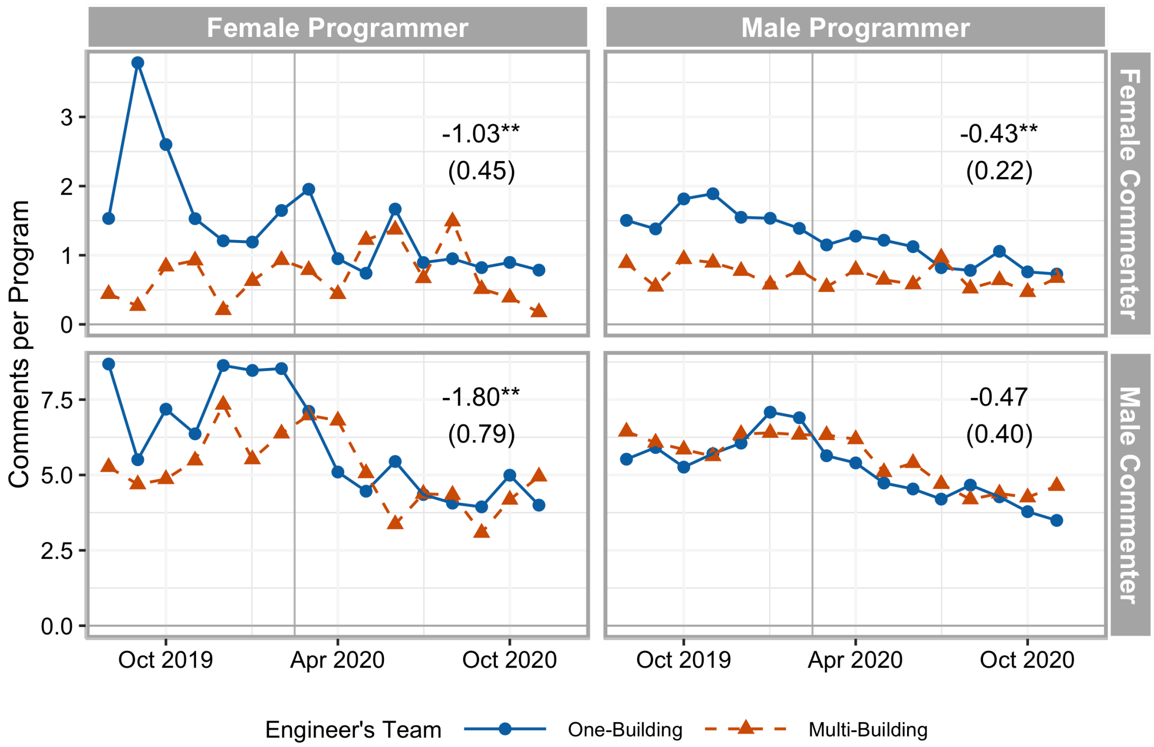
Notes: This figure illustrates the effects of losing proximity to teammates on feedback and programming output for engineers of different tenure at the firm. Each point represents a quintile of baseline tenure at the time that the offices closed. The y-axis plots the difference-in-differences coefficient from Equation 2 with our preferred set of controls for the scope of the program (quartics in files changed, lines added, and lines deleted), the engineer's tenure at the firm (in months) and the engineering role (e.g., website design versus database management). The error bars represent 95% confidence intervals with standard errors clustered by team.

Figure A.15: Gendered Impacts of Proximity to Teammates on Initial and Follow-up Feedback



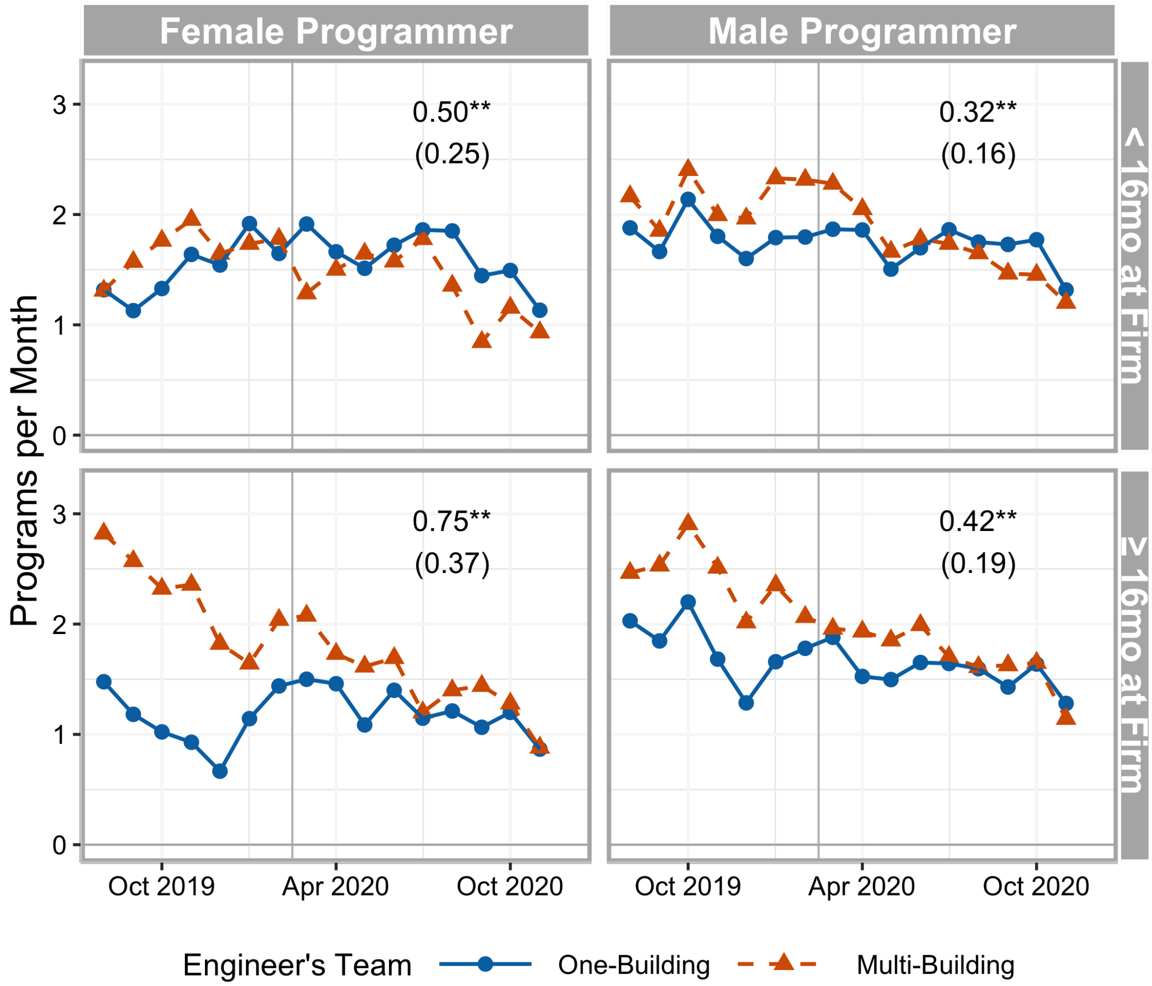
Notes: This figure illustrates the gendered impact of proximity on the feedback that an engineer receives on their code. Panel (a) plots the initial feedback on their code, defined as the average number of comments that an engineer receives on their programs before they send a follow-up reply. Panel (b) plots the average number of comments that an engineer receives on their code after they send a reply. The sample limits to engineers whose teammates all worked in the main campus.

Figure A.16: Gendered Effects of Proximity on Comments from Male and Female Commenters



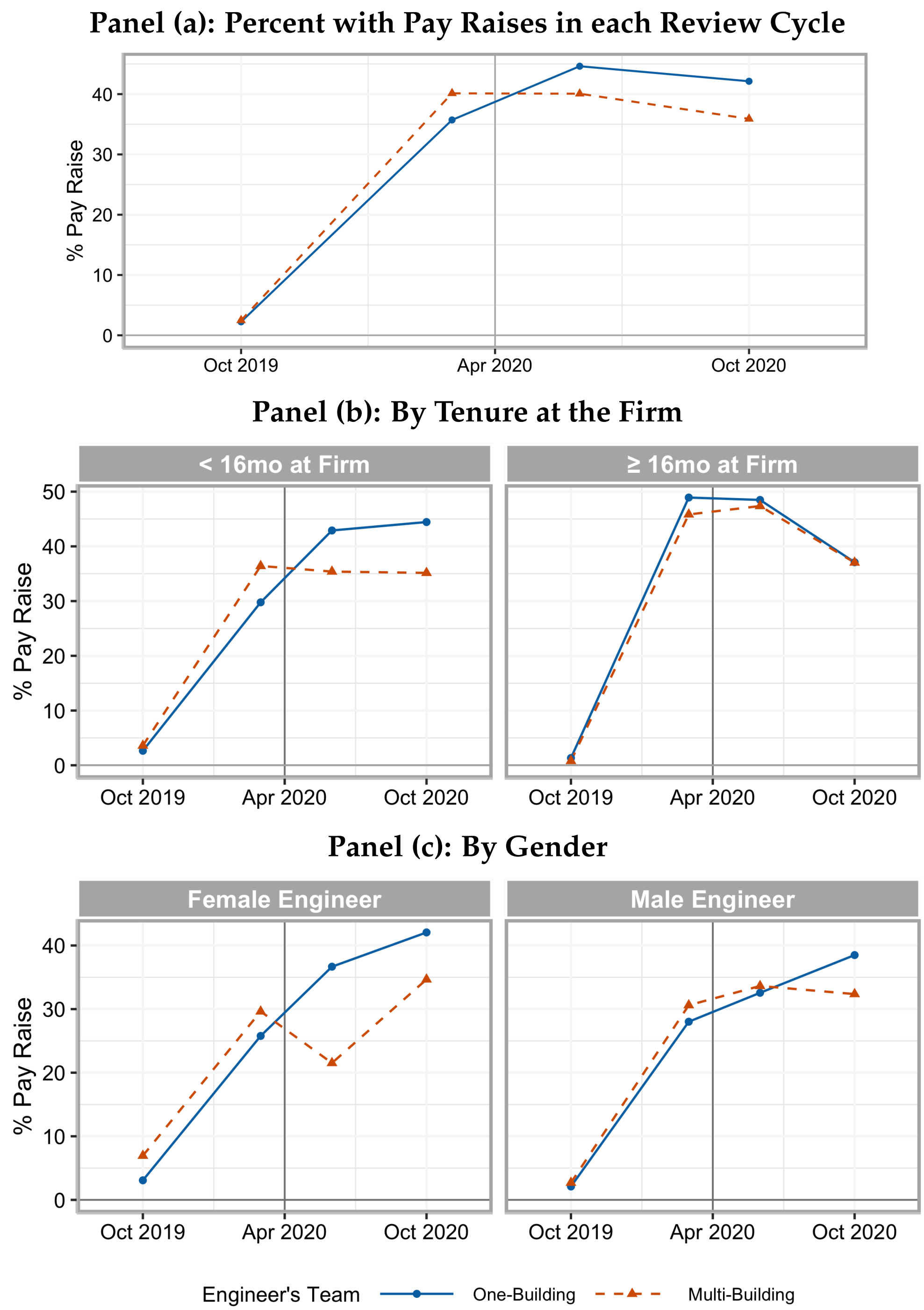
Notes: This figure illustrates the gendered impact of proximity on feedback from male and female commenters. Each plot shows the raw monthly averages of comments received per program for engineers on one- and multi-building teams, separately by female and male engineers and for comments from male and female commenters. The sample limits to engineers who submitted a program to the main code-base in that month and whose teammates all worked in the main campus. The annotated coefficient reflects Equation 2 with our preferred set of controls for program scope, engineering type, and engineer tenure. Standard errors are clustered by team. $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$.

Figure A.17: Gendered Impacts of Proximity to Teammates on Programming Output by Seniority



Notes: This figure illustrates the gendered impact of proximity on programming output. Each plot shows the raw monthly averages of programs per program for engineers on one- and multi-building teams, separately by female and male engineers and for those with above and below the median tenure of 16 months. The sample limits to engineers whose teammates all worked in the main campus. The annotated coefficient reflects Equation 2 including our preferred set of controls for engineering type and engineer tenure and project scope. Standard errors are clustered by team. *p<0.1; **p<0.05; ***p<0.01.

Figure A.18: Impacts of Proximity on Pay Raises

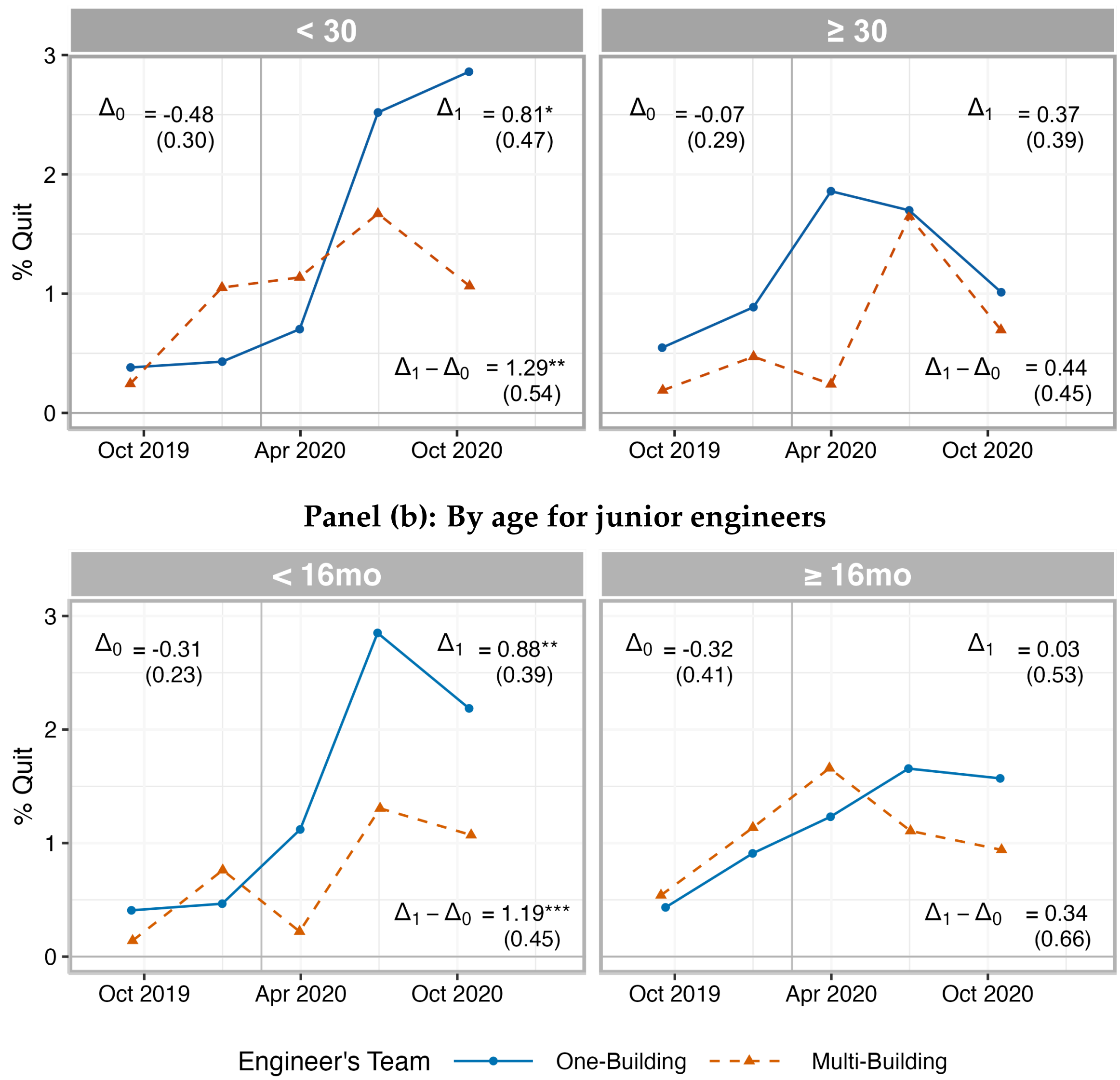


Notes: This figure illustrates the impact of proximity on the likelihood of a pay raise. Each point reflects the percent of engineers with pay raises at the end of each tri-annual review period. The post period is defined as starting in April 2020 since March 2020 pay raises were based on winter 2019-2020 reviews. The sample includes engineers who worked on the main code-base, whose teammates were all in the main campus, and who were hired before the start of our sample.

Figure A.19: Impacts of Proximity on Quits by Age

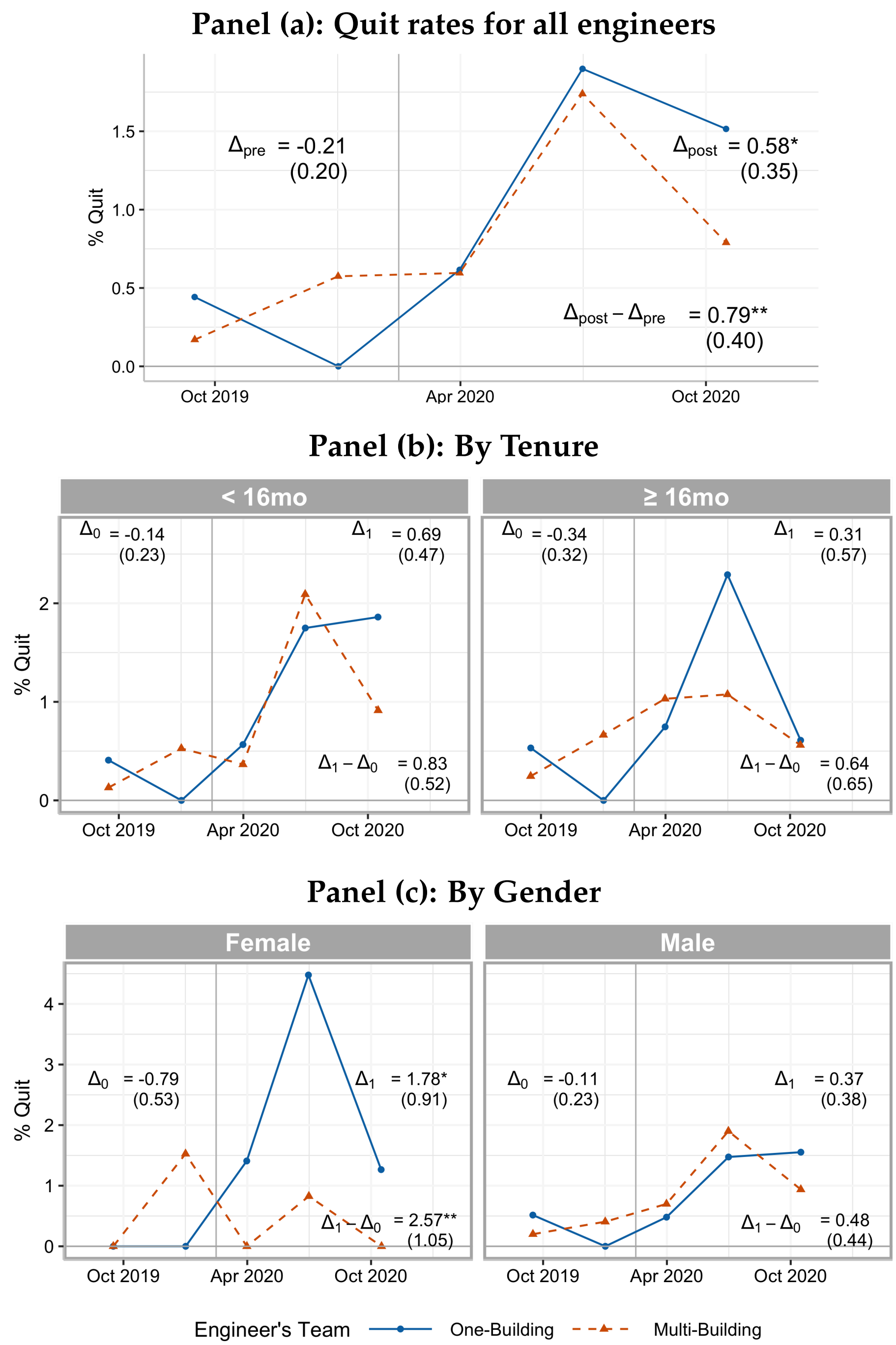
Panel (a): By age for all engineers

Panel (b): By age for junior engineers



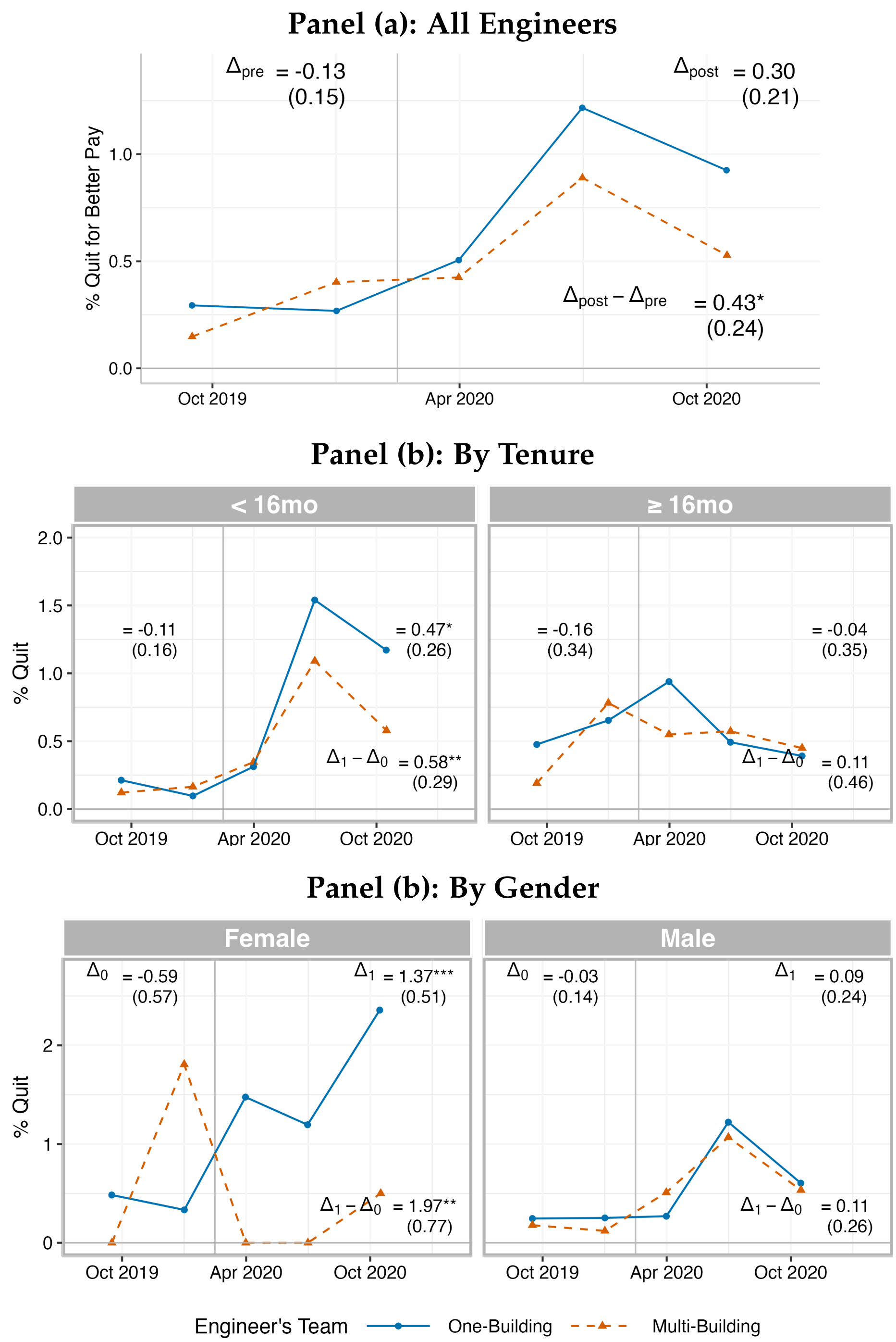
Notes: This figure illustrates the effects of proximity on quits for older and younger engineers (a) overall and (b) for relatively junior engineers before COVID (with less than the median sixteen months of experience). Each plot shows the raw quit rates for engineers on one-building and multi-building teams. The annotated coefficients use our preferred set of controls for engineering group and engineer tenure. Standard errors are clustered by engineering team. The sample includes engineers who worked on the main code-base, whose teammates were all in the main campus, and who were hired before the start of our sample. *p<0.1; **p<0.05; ***p<0.01.

Figure A.20: Impacts of Proximity on Quits for Engineers who Work on Internal Tools



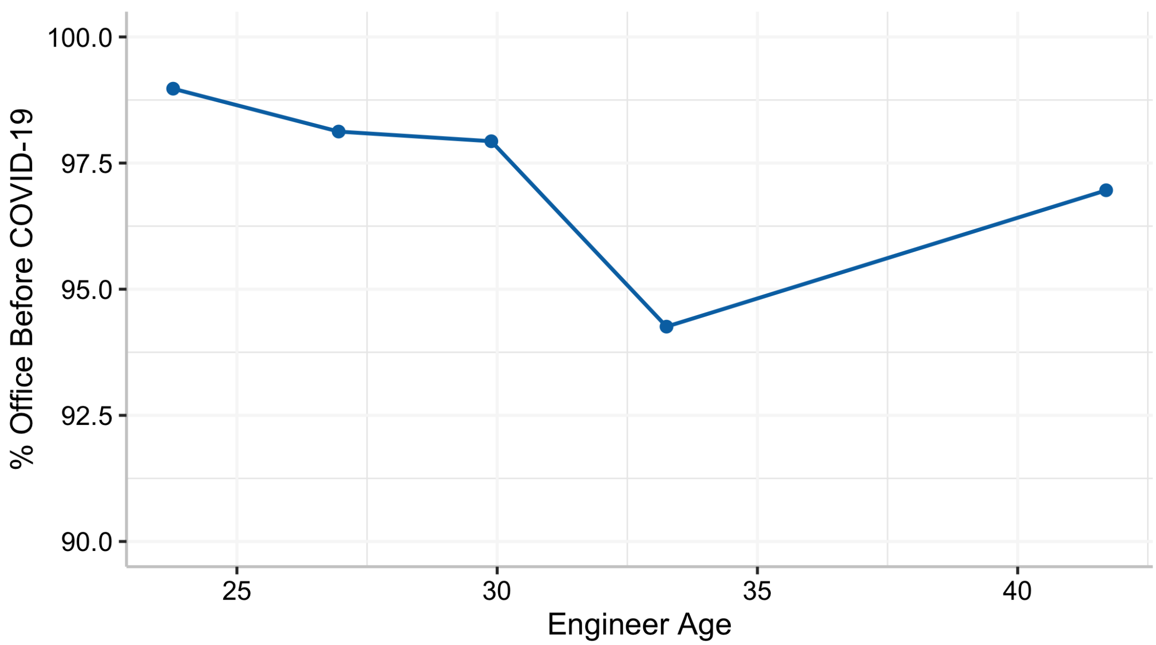
Notes: This figure replicates Figure 6 but limits the sample to the 588 engineers who worked on internal tools. Since it is useful for these workers to sit near those who use their tools (e.g., sales workers), it is more likely that these teams are split across the two office buildings. The analysis compares engineers who had sat with all their teammates in the same building before the offices closed (N=215) to engineers on multi-building teams (N=373). Standard errors are clustered by engineering team. *p<0.1; **p<0.05; ***p<0.01.

Figure A.21: Impacts of Proximity on Quits for Higher Paying Jobs



Notes: This figure replicates Figure 6 focusing on quits to jobs with higher average compensation on Glassdoor than the current firm's average salaries on this site. We compare workers who quit for higher pay to workers who did not quit or quit for a position with lower average compensation. We exclude workers who quit but for whom we do not know the destination compensation. The sample includes engineers who worked on the main code-base, whose teammates were all in the main campus, and who were hired before the start of our sample. The annotated coefficients use our preferred set of controls for engineering group and engineer tenure. Standard errors are clustered by engineering team. *p<0.1; **p<0.05; ***p<0.01.

Figure A.22: Office Work at the Firm by Engineer Age



Notes: This figure plots the share of engineers at the firm who were working in the office rather than from home before the pandemic as a function of the engineer's age.

Table A.1: Example Emblematic Comments of Principal Components

PCA	Total Load	Text of Comment
1	1.25	This **logic** is very confusing to **read** for the first **time**. Please add a paragraph in the comment section to clearly explain the **logic**.
1	0.98	So, you are still **reading** all jobs and then **counting** in PHP. I would think you want to make a **count** query to **SQL server**.
1	0.90	If you change it to **SELECT COUNT**(*) AS **COUNT** without a GROUP BY, would that work for you? GROUP BY is an expensive operation on our **server**
1	0.87	Do we want to add the price range **logic** in there then? I'm confused why that **logic** hasn't changed but this **logic** here has.
1	0.85	Do you think it's getting close to time to separate the filtering **logic** and the transfor ming **logic** here?
2	1.29	We agree and approve the **ownership** transfer from Jane Doe's **team** to our **team**.
2	1.08	Can you please get **approval** from John Doe's **team** first? We have a pr in review which makes John Doe's **team codeowners** of this file.
2	0.95	Instead of giving static **team** id, can we give random **team** id. Get **team** id from tabl e and do ORDER BY `ORDER BY newid()` to get random **teams**
2	0.92	If this **team** is yours, it would be great if you could migrate the - **owners** in this change to the appropriate **owner team** and **reviewer** group
2	0.85	LGTM note: should we move this file **ownership** in **codeowners** to your **team**?
2	−0.48	This if **return**, else if **return** else **return** statement can be turned into if **return**, if **re** **turn**, **return**
2	−0.41	Type hint **return** int, if it is **nullable** you can use ?int. ``` **public function** id_supplier(): int { **return** $this->id_supplier; } ```
2	−0.39	Can we just **return** this **bool** check? Rather than if () {**return false**} else {**return true**}
2	−0.38	Stick this in the **return** below so we have `**return** a || b || c;` instead of `if(a) {**retur** **n** true}; **return** b || c;`
2	−0.38	Should update this **return** type to match the actual **return** type. Especially since you added a **return** type hint.
2	−0.38	Could we add the **return** type hint? Also in the doc string above, the **return** is listed as **string** but this **function** returns a **function**.
3	1.06	Why do we want to do this? If a **class** has a **test class** and non-**test class**, it should be able to run the **test** file, no?
3	0.99	Could we create a **function** in conversation model and put the **DAO** function there? Then here, we'll call the **model function**.
3	0.91	Any reason to make this as a static **function** and moving it to **DAO**? Looks more like an helper **function** instead of a **DAO function**?
3	0.88	If this **method** and the **method** below are only used in **test**, could we add them in th e **test DAO class**?
3	0.86	Make this into a wrapper around the real **method**, and pass the request into the real **method**. Then you can create a unit **test** for that **method**.

Notes: This table shows five emblematic examples with high loadings on each principal component. The second principal component has identified two clusters of comments: where each cluster includes words that often appear together but rarely appear with words in the other cluster. These comments were selected because they have the maximum loadings of all comments less than 150 characters long. See Section I.A for details on principal component analysis.

Table A.2: Example Comments Evaluated by Software Engineers

Comment	Helpful?	Change code?	Actionable steps?	Explain reasoning?	Rude?
nitpick: If would have been nice if you designed this so that we can have the table in a place other than csn_service. But that's ok if you don't want to do this for this ticket.	Yes	No	Yes	Yes	No
clean up this comment	Not enough info	Not enough info	No	No	A little bit
Are you certain once the variables are binded to the SQL string that the null variables aren't going to get casted as string and result in NULL which wouldn't be null? i.e. `THEN ISNULL(NULL, 0)`	Yes	No	No	Yes	No
A general comment that should be applied everywhere. We should be keeping the naming of these entities consistent. Instead of orderSplits and orderItemSplits we should be calling things inventoryOrderSplits and inventoryOrderItemSplits.	Yes	Yes	Yes	Yes	No
This maybe required	Yes	Yes	Yes	Yes	A little bit
IIRC you can update the collection class phpdoc so that PHPStorm can detect the model type from `getDeliveryZones()` automatically instead of writing it manually ^ - [link]	Yes	Yes	Yes	Yes	No
Looks like the key 'must_not_ship_before_date' is used in a couple of places, here and shipping_label_helper.php is it possible to create a constant for it somewhere and reuse it?	Yes	Yes	Yes	No	No
What does this look like on the frontend, will it just be a blank view? Can we make it just show an error message ,something along the lines of Failed to generate preview. Please click 'Preview Creative' to try again? If that is possible/easy?	Yes	Not enough info	Yes	Yes	No
Function description	Yes	Yes	Yes	Yes	No
This should be changed back to `getNavHeaderHtml()`	Yes	Yes	Yes	Yes	No
Leave all these properties unset and instead make them required in the constructor as parameters. Then child constructs can just	Yes	Yes	Yes	Yes	No

Notes: This table continued on the subsequent pages shows examples that were rated by software engineers.

pass them in the `parent::` call (while the dependencies like `App Nexus Client` and `DAO` can be injected)					
Good improvement I think.	Yes	No	No	No	No
Capitalize EXISTS	Yes	Yes	Yes	No	A little bit
can you import the logger to not use the fully qualified name?	Yes	Not enough info	Yes	No	A little bit
`'web'` probably should be a constant	Yes	Yes	Yes	Yes	No
This doesn't seem right -- why are we defaulting to `Feature::is_enabled` instead of an instance of `Toggle`? Shouldn't we be avoiding static calls to `Feature` in this manner?	Yes	Yes	Yes	Yes	No
Are SOP's going to have any idea how to handle an error thrown from a try/catch block/will they have any action they can take after one of these errors? Might be better to just log this and display a generic error rather than the specific one.	Yes	Yes	Yes	Yes	No
This guy wants to be an object	Not enough info	Not enough info	Not enough info	Not enough info	A little bit
Hm, there's no `ORDER BY` on the DAO call here, so maybe the limit solution doesn't actually work, since you may just return the same `N` tran_logs every time...We'd have to add date-based looping combined with was there at least N returned from the last set to ensure we've validated every tran log we need to look at has been. What's the current rate for 158's per day? It may be worth doing a loop over each day or couple days	Not enough info	Not enough info	Yes	Yes	No
nit: extra white space	Yes	Yes	Yes	No	No
Why ISNULL() here? I believe in your :delivery_date and :ship_date will never actually be null, and imo it makes the fn less intuitive if someone tries to set the value of one of these fields to NULL	Yes	Yes	Yes	Yes	No
might also want to check for an actual null value here, not just the word 'null' or an empty string	Yes	Yes	Yes	Yes	No
Slight nitpick, but is `skuName` set in stone as a requirement? It seems like	Yes	Yes	Yes	Yes	No

`productName` should map to `$product-name` and `displayName` should map to `$product-display_name`, for example (and consistency with the product models)					

What if we keep the cartons with no SSCC and remove the cartons with no SSCC in the `serialize` function? Or implement a validate function on the model and act on it according to what the validate function says - [link]
 	
``
if (!$collection-validate_models()) return array_filter($collection, 'exclude_empty_sscc'))
return $collection
``
 	
This way, we still keep cartons with no sscc in the collection. And don't have to guess any point in our code if we've removed cartons with no sscc or not.
	Yes	Yes	Yes	Yes	No

nice!!	No	No	No	No	No
I know this is how the current faux joins do it... but why do we fill an array with null?	Yes	No	No	No	No
I assume the commented out client logger and stub is left over from local troubleshooting?	Yes	Not enough info	No	Not enough info	Moderately
should we add info level kibana log here	Yes	Yes	No	No	No
Should we keep the `- Zones_To_Pick[]` since we already have it as the param type?	Yes	Yes	Yes	Yes	No
LGTM: Only formatting comments for me	Yes	Yes	Yes	Yes	No
I'm wondering if this is more complicated than necessary and if we instead should be taking advantage of the functions already tried and true in the bulk helper file	Yes	Yes	Yes	Yes	No
HTTP errors are logged in L1926. This code is only for timeouts	Yes	Yes	Yes	Yes	No
Change year	Yes	Yes	Yes	Yes	No
WF3 sniff seems to be failing. Can you fix this?	Yes	Not enough info	No	No	No
Add more info on description/testing done. Run ```firm sniff``` / ```firm format``` and address issues: ```	Yes	Yes	Yes	Yes	No

FILE: .../wms/dimension_update_tool/dimension_update_tool_detail_view.php -- *emojii* | -- *emojii* | FOUND 3 ERRORS AFFECTING 2 LINES *emojii* | -- *emojii* | 19 [link]| fully-specified classpath *emojii* | -- ```					
another nit: don't love the name here, because it will _have_ to be changed if the test wins	No	Not enough info	No	No	No
just wanted to make sure Adam is warned of this change	No	Not enough info	No	No	No
our sniffer prefers `random_int()` instead of `rand`	Yes	Yes	Yes	Yes	No
nit: can remove `public` here	Not enough info	Not enough info	Yes	No	No
Agree with John here, those store constants don't exist in every context and using them has got us into trouble in the past (looking at you, Category Builder)	Yes	Yes	Yes	Yes	No

Table A.3: Gender Gaps in Online Feedback: Coded Comments

Panel (a): Overall gender gap

	% Of Comments									
	Helpful		Change Code		Actionable		Explain Reason		Rude	
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
Female	2.90**	3.56***	2.41*	2.97**	2.35*	2.58*	−0.22	0.63	−0.71	−0.72
	(1.14)	(0.92)	(1.41)	(1.45)	(1.37)	(1.36)	(1.36)	(1.41)	(1.08)	(1.12)

Panel (b): Gender gap by team-type and period

	Helpful		Change Code		Actionable		Explain Reason		Rude	
Female										
One-Building, Pre	2.03	4.20**	3.50	5.82**	1.79	3.40	0.78	3.75	0.53	0.76
	(2.37)	(2.10)	(2.80)	(2.80)	(2.69)	(2.48)	(2.50)	(2.65)	(2.32)	(2.36)
Multi-Building, Pre	1.85	4.18*	0.27	−0.77	−0.56	0.91	−2.94	−0.95	−1.69	−1.48
	(2.54)	(2.22)	(2.86)	(3.25)	(2.83)	(2.77)	(3.11)	(3.21)	(2.05)	(2.13)
One-Building, Post	4.23*	3.81**	1.50	1.32	2.87	3.35	−0.85	−0.23	−1.69	−1.96
	(2.29)	(1.74)	(2.88)	(3.07)	(2.84)	(3.04)	(2.55)	(2.81)	(2.10)	(2.17)
Multi-Building, Post	3.97	2.28	5.26*	6.68**	6.06**	3.12	2.87	0.36	0.13	−0.18
	(2.48)	(1.85)	(2.85)	(2.96)	(2.91)	(2.86)	(2.63)	(2.64)	(1.80)	(1.87)
Preferred Controls	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Drop Not Enough Info		✓		✓		✓		✓		✓
Dependent Mean	76.2%	87.0%	52.0%	70.4%	60.3%	68.4%	51.4%	58.4%	14.7%	15.2%
# Comments	5,377	4,708	5,377	3,974	5,377	4,741	5,377	4,740	5,377	5,174
# Teams	258	255	258	253	258	254	258	254	258	258

Notes: This table evaluates the gender gap in feedback for the random subset of comments that engineers evaluated on their helpfulness and content. Panel (a) considers the overall gender differences with our preferred set controls. Panel (b) allows the gender difference to vary by the team-type (one- versus multi-building) both before and after the office closed (pre vs. post). Standard errors are clustered by team. See the note of Table A.7 for the definitions of the outcomes. $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$.

Table A.4: Experience Gaps in Online Feedback: Coded Comments

Panel (a): Overall experience gap

	% Of Comments									
	Helpful		Change Code		Actionable		Explain Reason		Rude	
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
Junior (< 16mo)	3.40***	1.59	4.90***	6.41***	6.66***	6.62***	4.27***	4.40***	−2.40**	−2.43**
	(1.23)	(1.07)	(1.46)	(1.61)	(1.61)	(1.64)	(1.56)	(1.63)	(1.04)	(1.08)

Panel (b): Experience gap by team-type and period

	Helpful		Change Code		Actionable		Explain Reason		Rude	
Junior (< 16mo)										
One-Building, Pre	5.07*	1.47	4.68	5.43	3.08	2.07	0.29	−0.15	−3.36	−3.39
	(2.73)	(2.31)	(3.11)	(3.33)	(2.74)	(2.81)	(3.01)	(3.09)	(2.63)	(2.67)
Multi-Building, Pre	4.30	2.39	5.35	4.46	9.32**	9.86***	6.63*	6.21*	−1.76	−1.41
	(2.90)	(2.33)	(3.62)	(3.73)	(3.68)	(3.48)	(3.42)	(3.42)	(2.07)	(2.17)
One-Building, Post	1.54	1.88	3.48	4.92*	5.42**	5.36**	1.94	2.84	−2.13	−2.15
	(2.29)	(1.72)	(2.60)	(2.74)	(2.54)	(2.68)	(2.70)	(2.79)	(1.82)	(1.94)
Multi-Building, Post	2.50	0.13	5.15**	9.01***	7.22***	7.66***	6.33**	6.82**	−2.46	−2.76
	(2.10)	(1.96)	(2.14)	(2.84)	(2.65)	(2.74)	(2.86)	(3.16)	(1.84)	(1.90)
Preferred Controls	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Drop Not Enough Info		✓		✓		✓		✓		✓
Dependent Mean	76.2%	87.0%	52.0%	70.4%	60.3%	68.4%	51.4%	58.4%	14.7%	15.2%
# Comments	5,377	4,708	5,377	3,974	5,377	4,741	5,377	4,740	5,377	5,174
# Teams	258	255	258	253	258	254	258	254	258	258

Notes: This table evaluates the experience gap in feedback for the random subset of comments that engineers evaluated on their helpfulness and content. Panel (a) considers the overall differences between junior engineers with less than the median tenure at the firm (16 months) and senior engineers with more experience. We include our preferred set of controls, sans the usual tenure controls, which would render the junior indicator not identified. Panel (b) allows the experience difference to vary by the team-type (one- versus multi-building) both before and after the office closed (pre vs. post). Standard errors are clustered by team. See the note of Table A.7 for the definitions of the outcomes. *p<0.1; **p<0.05; ***p<0.01.

Table A.5: Testing Robustness of Results to Local-Linear Time-Trends

	Comments per Program					
	(1)	(2)	(3)	(4)	(5)	(6)
Post x In One-Building Team	-1.29^{***}	-1.72^{**}	-1.16^{***}	-1.41^{**}	-1.14^{**}	-1.36^{**}
	(0.48)	(0.86)	(0.43)	(0.63)	(0.47)	(0.64)
One-Building Team	1.16^{**}	1.82^{**}	1.80^{***}	2.50^{***}		
	(0.52)	(0.93)	(0.49)	(0.66)		
Post	-1.22^{***}	-0.24				
	(0.36)	(0.63)				
Pre-Mean in One-Building Teams	8.04	8.04	8.04	8.04	8.04	8.04
Percentage Effects						
Post x One-Building Team	-16.06%	-21.35%	-14.45%	-17.58%	-14.15%	-16.89%
One-Building	14.46%	22.63%	22.38%	31.15%		
% One-Building Team	58.33	58.33	58.33	58.33	58.33	58.33
Local-Linear Time-Trends		✓		✓		✓
Preferreed Controls			✓	✓	✓	✓
All Controls					✓	✓
# Engineers	1,055	1,055	1,055	1,055	1,055	1,055
# Months	17	17	17	17	17	17
# Engineer-Months	9,304	9,304	9,304	9,304	9,304	9,304
R^2	0.02	0.02	0.36	0.36	0.50	0.50

Notes: This table tests the robustness of the results in Table 2 to the inclusion of local-linear time-trends on each side of the office closures for engineers on one- and multi-building teams. The odd columns repeat the results from Table 2 for reference. The even columns include local-linear time-trends that allow comments on each program to evolve differentially over time for engineers on one- and multi-building teams both before and after the offices closed for the pandemic. The preferred controls are those in Column 4 of Table 2 and all controls are those in Column 8. See Table 2's note for details on controls. Standard errors are clustered by engineering team. $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$.

Table A.6: Proximity to Teammates and Online Feedback for Engineers who Work on Internal Tools

	Comments per Program							
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Post x One-Building Team	-1.52^{***}	-1.52^{***}	-1.58^{***}	-1.38^{***}	-1.39^{***}	-1.16^{**}	-1.78^{***}	-1.06^{**}
	(0.59)	(0.59)	(0.51)	(0.50)	(0.50)	(0.48)	(0.50)	(0.50)
One-Building Team	1.16	1.16	2.28^{***}	1.98^{***}	1.98^{***}			
	(0.74)	(0.74)	(0.65)	(0.59)	(0.59)			
Post	-1.36^{***}							
	(0.39)							
Pre-Mean, One-Building Team	8.27	8.27	8.27	8.27	8.27	8.27	8.27	8.27
Percentage Effects								
Post x One-Building Team	-18.3%	-18.3%	-19.1%	-16.7%	-16.8%	-14%	-21.5%	-12.8%
One-Building	14.1%	14.1%	27.6%	24%	24%			
% One-Building Team	58.3	58.3	58.3	58.3	58.3	58.3	58.3	58.3
Engineer Group x Post FE		✓	✓	✓	✓	✓	✓	✓
Program Scope Quartics			✓	✓	✓	✓	✓	✓
Months at Firm x Post FE				✓	✓	✓	✓	✓
Team Size x Post					✓	✓	✓	✓
Engineer FE						✓	✓	✓
Engineer Traits x Post FE							✓	✓
Main Building x Post FE								✓
# Teams	140	140	140	140	140	140	140	140
# Engineers	588	588	588	588	588	588	588	588
# Engineer-Months	5,630	5,630	5,630	5,630	5,630	5,630	5,630	5,630
R^2	0.02	0.02	0.28	0.36	0.36	0.49	0.50	0.50

Notes: This table replicates Table 2 but limits the sample to engineers who built internal tools for others in the firm. Since it was useful for these engineers to sit near the other engineers who used their tools, these engineers' teams were more likely to end up being split across the two buildings on the firm's main campus. Standard errors are clustered by team. $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$.

Table A.7: Proximity to Teammates and Online Feedback for Engineers: Coded Comments

Panel (a): One- vs. Multi-Building Teams

	% Of Comments									
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
	Helpful		Implemented in Code		Actionable Feedback		Reasoning Explained		Rude	
One-Building x Post	2.09	−0.23	0.81	−0.01	3.43	1.06	−2.62	−3.16	−1.79	−1.91
	(3.38)	(2.51)	(4.41)	(4.48)	(4.04)	(3.83)	(3.50)	(3.84)	(2.56)	(2.64)
One-Building Team	0.79	1.32	2.81	3.58	0.86	1.14	3.85	3.80	1.30	1.05
	(2.43)	(1.83)	(2.64)	(2.67)	(2.49)	(2.43)	(2.50)	(2.71)	(1.88)	(1.92)
Dependent Mean	76.2%	87.0%	52.0%	70.4%	60.3%	68.4%	51.4%	58.4%	14.7%	15.2%

Panel (b): By Gender

	Helpful		Implemented in Code		Actionable Feedback		Reasoning Explained		Rude	
Female: One-Building x Post	2.27	0.45	−2.98	−6.62	0.65	−0.14	−6.59	−6.31	−4.06	−4.21
	(4.55)	(3.36)	(6.15)	(6.20)	(5.36)	(5.03)	(4.15)	(4.61)	(3.55)	(3.66)
Female: One-Building	1.08	1.74	4.82	7.58**	2.25	2.76	5.88*	6.63**	2.52	2.30
	(3.27)	(2.37)	(3.78)	(3.60)	(3.47)	(3.15)	(3.03)	(3.15)	(2.64)	(2.72)
Male: One-Building x Post	2.19	−1.07	4.01	5.32	6.18	2.13	0.85	−1.01	−0.01	−0.18
	(3.99)	(3.11)	(4.34)	(4.59)	(4.54)	(4.35)	(4.54)	(4.77)	(2.94)	(3.02)
Male: One-Building	0.90	1.72	1.59	1.00	−0.09	0.27	2.15	1.93	0.29	0.06
	(2.76)	(2.31)	(2.86)	(3.16)	(2.93)	(2.95)	(3.32)	(3.58)	(2.31)	(2.35)
Female: Dependent Mean	77.4%	88.7%	53.6%	72.1%	61.7%	69.8%	51.5%	58.7%	14.5%	15.0%
Male: Dependent Mean	74.9%	85.4%	50.4%	68.7%	58.9%	66.9%	51.4%	58.0%	14.8%	15.4%

Panel (c): By Tenure

	Helpful		Implemented in Code		Actionable Feedback		Reasoning Explained		Rude	
Junior (< 16mo): One-Building x Post	1.13	0.43	0.45	−1.65	4.68	2.67	−3.09	−3.57	−2.02	−1.94
	(3.69)	(2.82)	(4.79)	(4.95)	(4.56)	(4.38)	(4.12)	(4.65)	(2.90)	(2.96)
Junior (< 16mo): One-Building	1.45	1.37	2.59	3.60	−0.45	−0.55	2.99	3.12	0.90	0.54
	(2.67)	(2.02)	(2.98)	(3.14)	(2.92)	(2.66)	(2.80)	(2.98)	(2.00)	(2.04)
Senior (≥ 16mo): One-Building x Post	4.52	−1.07	0.95	2.26	−0.48	−3.89	−3.43	−3.66	−2.17	−2.70
	(4.93)	(3.87)	(5.93)	(6.04)	(5.50)	(5.75)	(4.98)	(5.35)	(4.26)	(4.43)
Senior (≥ 16mo): One-Building	−1.22	1.13	3.47	3.58	4.84	6.20	6.53	5.83	2.57	2.62
	(3.86)	(3.23)	(4.43)	(4.65)	(4.65)	(5.11)	(3.99)	(4.41)	(3.49)	(3.61)
Junior (< 16mo): Dependent Mean	77.5%	87.7%	54.2%	73.0%	62.9%	70.9%	53.3%	60.2%	13.8%	14.4%
Senior (≥ 16mo): Dependent Mean	74.0%	85.9%	48.5%	66.1%	56.0%	64.1%	48.4%	55.2%	16.0%	16.6%
Preferred Controls	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Drop Not Enough Info		✓		✓		✓		✓		✓
# Comments	5,377	4,708	5,377	3,974	5,377	4,741	5,377	4,740	5,377	5,174
# Teams	258	255	258	253	258	254	258	254	258	258

Notes: This table considers the random subset of comments that engineers evaluated on their helpfulness and content. Each column replicates our preferred specification in Column 4 of Table 2. The odd columns include all evaluated comments. The even columns exclude comments where the evaluator said that there was "not enough information" to determine the best response. For columns 1−2, the evaluators, were asked, "Would you find this comment helpful?" (Yes/No/Not enough info). For columns 3−4, "Do you think you would change your code because of this comment?" For columns 5−6, "Does this comment suggest actionable steps to change your code?" For columns 7−8, "Does this comment explain the reason for changing your code?" For columns 9−10, "Is the tone of this comment rude?" Standard errors are clustered by team. *p<0.1; **p<0.05; ***p<0.01.

Table A.8: Proximity to Teammates and Mentorship

Panel (a): Comments Received by Seniority

	Comments per Program					
	Received by Junior (< 16mo)			Received by Senior (≥ 16mo)		
	(1)	(2)	(3)	(4)	(5)	(6)
Post x One-Building Team	-1.40^{**}	-1.72^{***}	-2.01^{***}	-0.41	-0.38	0.25
	(0.65)	(0.60)	(0.63)	(0.45)	(0.42)	(0.67)
One-Building Team	1.27^{*}	2.61^{***}		0.13	0.58	
	(0.68)	(0.69)		(0.39)	(0.41)	
Post	-2.17^{***}			-0.04		
	(0.50)			(0.32)		
Pre-Mean, One-Building Team	9.56	9.56	9.56	4.99	4.99	4.99
Percentage Effects						
Post x One-Building Team	-14.7%	-18%	-21%	-8.2%	-7.7%	4.9%
One-Building Team	13.2%	27.3%		2.5%	11.7%	
Preferred Controls		✓			✓	
All Controls			✓			✓
# Engineer-Months	6,056	6,056	6,056	3,248	3,248	3,248

Panel (b): Comments Written by Seniority

	Comments per Program					
	Written by Junior (< 16mo)			Written by Senior (≥ 16mo)		
	(1)	(2)	(3)	(4)	(5)	(6)
Post x One-Building Team	-0.41^{*}	-0.25	-0.37	-0.95^{**}	-1.00^{***}	-0.84^{*}
	(0.24)	(0.26)	(0.32)	(0.38)	(0.36)	(0.45)
One-Building Team	0.15	0.39		1.07^{***}	1.47^{***}	
	(0.28)	(0.30)		(0.37)	(0.40)	
Post	-0.74^{***}			-0.42		
	(0.18)			(0.30)		
Pre-Mean, One-Building Team	2.94	2.94	2.94	3.73	3.73	3.73
Percentage Effects						
Post x One-Building Team	-13.9%	-8.6%	-12.7%	-25.4%	-26.8%	-22.4%
One-Building Team	5.1%	13.3%		28.7%	39.4%	
Preferred Controls		✓			✓	
All Controls			✓			✓
# Engineer-Months	9,304	9,304	9,304	9,304	9,304	9,304

Notes: This table investigates the relationship between sitting near teammates and (a) the feedback received by junior and senior engineers and (b) the feedback given by these engineers. The preferred controls reflect Column 4 of Table 2; all controls reflect Column 7. Standard errors are clustered by team. *p<0.1; **p<0.05; ***p<0.01.

Table A.9: Externalities from a Distant Teammate on Online Feedback from Proximate Teammates

	# Comments			
	All Per Program	From Proximate Teammates Per Review	All Per Program	From Proximate Teammates Per Review
	(1)	(2)	(3)	(4)
Post x On One-Building Team	-1.16^{***}	-0.54^{**}	-1.56^{***}	-0.71^{**}
	(0.43)	(0.26)	(0.46)	(0.28)
On One-Building Team	1.80^{***}	0.71^{***}		
	(0.49)	(0.28)		
Pre-Mean, One-Building Team	8.04	4.17	8.04	4.17
Percentage Effects				
Post x On One-Building Team	-14.45%	-12.98%	-19.36%	-17.1%
On One-Building Team	22.38%	17.07%		
Avg. on Multi-Building Teams				
# Teammate Commenters	1.71	1.71	1.71	1.71
% From Proximate Teammates	39.39	39.39	39.39	39.39
# Proximate Teammate Commenters	0.67	0.67	0.67	0.67
Back-of-the-envelope Calculations				
% Initial Gap Explained		26.59%		
% Differential Change Explained		31.33%		30.8%
Controls	Preferred	Preferred	All	All
# Engineers	1,055	934	1,055	934
# Engineer-Months	9,304	7,174	9,304	7,174
R^2	0.36	0.24	0.50	0.46

Notes: This table investigates whether having a teammate in a different building impacts the online feedback an engineer receives from her proximate (same-building) teammates. The odd columns consider all comments on each program. The even columns consider the average length of reviews from proximate teammates, conditional on them leaving reviews. The first two columns include the preferred controls. The next two columns include all controls. The back-of-the-envelope calculations consider how much feedback from proximate teammates can explain overall effects on comments in the preceding column, based on the share of comments that come from proximate teammates. Each column estimates Equation 2. Standard errors are clustered by team. $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$.

Table A.10: Difference-in-Differences Around New Hires in a Different Building From Teammates vs. Other Hires Before COVID-19

	Comments per Review from Same-Building Teammate		
Post Hire x One- to Multi-Building Team	-1.483^{**}	-1.710^{*}	-1.332
	(0.640)	(1.017)	(1.062)
Post Hire	-0.302	-0.068	0.010
	(0.987)	(1.151)	(1.008)
Bandwidth = 6 weeks	✓	✓	✓
Pre-Period Mean for Treated	4.329	4.329	4.329
Engineer x Event FE	✓	✓	✓
Engineer x Commenter x Event FE		✓	✓
Program Content			✓
# Teams	126	126	126
# Treated Teams	16	16	16
# Engineers	400	400	400
# Treated Engineers	46	46	46
# Engineer-Commenter Pairs	1159	1159	1159
# Treated Engineer-Commenter Pairs	142	142	142
Observations	4,017	4,017	4,017
R^2	0.236	0.407	0.552

Notes: This table compares the change in comments per review in teams where a new hire converts the team from a one-building team to a multi-building team relative to teams where a new hire does not change whether they are a one- or a multi-building team. Each observation is the comments that a particular commenter left on a coder's program. The analysis compares the change in the length of the peer-reviews in the commenter-coder pair around the two types of new hires as in Equation 4. Standard errors are clustered by team. *p<0.1; **p<0.05; ***p<0.01.

Table A.11: Proximity to Teammates and Engineer Output for Engineers who Build Internal Tools

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
	Panel (a): Programs per Month						
Post x One-Building Team	0.47***	0.47***	0.44***	0.44***	0.39***	0.38***	0.44***
	(0.14)	(0.14)	(0.13)	(0.13)	(0.13)	(0.13)	(0.15)
One-Building Team	−0.42**	−0.42**	−0.39**	−0.41**			
	(0.19)	(0.19)	(0.19)	(0.19)			
Pre-Mean, One-Building Team	1.82	1.82	1.82	1.82	1.82	1.82	1.82
Post x One-Building Team as %	25.8%	25.8%	24.2%	24.2%	21.3%	20.9%	24%
One-Building Team as %	-23.2%	-23.2%	-21.6%	-22.7%			
R^2	0.01	0.01	0.04	0.05	0.42	0.43	0.43
	Panel (b): Lines Added per Month						
Post x One-Building Team	100**	100**	103**	105**	97**	86**	153***
	(43)	(43)	(44)	(42)	(42)	(43)	(46)
One-Building Team	−197***	−197***	−202***	−209***			
	(52)	(52)	(53)	(53)			
Pre-Mean, One-Building Team	332	332	332	332	332	332	332
Post x One-Building Team as %	30.1%	30.1%	30.8%	31.6%	29.2%	26%	46.2%
One-Building Team as %	-59.4%	-59.4%	-60.9%	-63%			
R^2	0.02	0.02	0.05	0.05	0.34	0.35	0.35
	Panel (c): Files Changed per Month						
Post x One-Building Team	1.50	1.50	1.41	1.50	1.10	0.90	2.11*
	(1.05)	(1.05)	(1.04)	(1.03)	(1.02)	(1.03)	(1.11)
One-Building Team	−4.12***	−4.12***	−4.11***	−4.33***			
	(1.25)	(1.25)	(1.27)	(1.26)			
Pre-Mean, One-Building Team	9.03	9.03	9.03	9.03	9.03	9.03	9.03
Post x One-Building Team as %	16.6%	16.6%	15.7%	16.6%	12.2%	10%	23.4%
One-Building Team as %	-45.7%	-45.7%	-45.5%	-47.9%			
R^2	0.01	0.01	0.04	0.04	0.32	0.33	0.33
Engineer Group x Post FE		✓	✓	✓	✓	✓	✓
Months at Firm x Post FE			✓	✓	✓	✓	✓
Team Size x Post				✓	✓	✓	✓
Engineer FE					✓	✓	✓
Engineer Traits x Post FE						✓	✓
Main Building x Post 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 replicates Table 4 but limits the sample to engineers who built internal tools for others in the firm. Since it was useful for these engineers to sit near the other engineers who used their tools, these engineers' teams were more likely to end up being split across the two buildings on the firm's main campus. Standard errors are clustered by engineering team. *p<0.1; **p<0.05; ***p<0.01.

Table A.12: Proximity to Teammates and Engineer Output by Seniority

	Monthly Programming Output					
	Programs		Lines of Code		Files Changed	
	(1)	(2)	(3)	(4)	(5)	(6)
Junior x Post x One-Building Team	0.35**	0.26*	92.29**	96.58**	0.93	0.82
	(0.14)	(0.14)	(46.23)	(45.75)	(1.16)	(1.11)
Junior x One-Building Team	−0.13		−160.40***		−2.86**	
	(0.18)		(53.49)		(1.28)	
Senior x Post x One-Building Team	0.49***	0.47**	90.34*	157.40**	2.11	2.88**
	(0.18)	(0.20)	(53.16)	(61.63)	(1.29)	(1.40)
Senior x One-Building Team	−0.64***		−176.00***		−4.54***	
	(0.21)		(57.58)		(1.40)	
Junior Pre-Mean, One-Building Team	1.75	1.75	339.27	339.27	9.62	9.62
Senior Pre-Mean, One-Building Team	1.65	1.65	278.17	278.17	8.32	8.32
Percentage Effects						
Junior x Post x One-Building Team	19.9%	14.8%	27.2%	28.5%	9.7%	8.5%
Junior x One-Building Team	-7.3%		-47.3%		-29.7%	
Senior x Post x One-Building Team	29.5%	28.4%	32.5%	56.6%	25.3%	34.6%
Senior x One-Building Team	-38.5%		-63.3%		-54.6%	
Preferred Controls	✓	✓	✓	✓	✓	✓
All Controls		✓		✓		✓
# Teams	304	304	304	304	304	304
# 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 programming output, separately for junior and senior engineers. Junior engineers are defined as those with less than 16 months of experience at the firm before the office closures and senior engineers as those with at least 16 months at the firm (the median tenure). The preferred controls and full set of controls are described in Section III. Standard errors are clustered by engineering team. *p<0.1; **p<0.05; ***p<0.01.

Table A.13: Gendered Impacts of Proximity

	Panel (a): Difference-in-Differences in Feedback Per Program				
	Comments	# Commenters	Commenter Initial Comments	Author Follow-up Questions	Commenter Follow-up Comments
	(1)	(2)	(3)	(4)	(5)
Female: Post x One-Building Team	-2.83^{***}	-0.31^{***}	-1.02^{**}	-0.17^{*}	-1.81^{***}
	(0.81)	(0.09)	(0.52)	(0.10)	(0.70)
Female: One-Building Team	3.84^{***}	0.37^{***}	1.15^{**}	0.23^{**}	2.69^{***}
	(0.77)	(0.09)	(0.48)	(0.09)	(0.66)
Male: Post x One-Building Team	-0.83^{*}	-0.06	-0.37	-0.14^{***}	-0.46
	(0.45)	(0.05)	(0.27)	(0.05)	(0.34)
Male: One-Building Team	1.38^{***}	0.17^{***}	0.68^{**}	0.09^{*}	0.70^{*}
	(0.52)	(0.06)	(0.27)	(0.05)	(0.40)
Female Pre-Mean, One-Building Team	9.56	1.87	5.38	0.32	4.18
Male Pre-Mean, One-Building Team	7.67	1.75	4.79	0.22	2.88
Percentage Effects					
Female: Post x One-Building Team	-29.6%	-16.4%	-18.9%	-54.6%	-43.4%
Female: One-Building	40.1%	19.8%	21.3%	72.6%	64.3%
Male: Post x One-Building Team	-10.8%	-3.5%	-7.8%	-61.3%	-15.8%
Male: One-Building	18%	9.7%	14.2%	41%	24.3%

	Panel (b): Triple Difference in Feedback Per Program				
	Comments	# Commenters	Commenter Initial Comments	Author Follow-up Questions	Commenter Follow-up Comments
	(1)	(2)	(3)	(4)	(5)
Female x Post x One-Building Team	-2.01^{**}	-0.25^{***}	-0.65	-0.04	-1.36^{**}
	(0.82)	(0.08)	(0.50)	(0.09)	(0.67)
Female x One-Building Team	2.46^{***}	0.20^{***}	0.47	0.14^{*}	1.99^{***}
	(0.76)	(0.08)	(0.47)	(0.08)	(0.61)
Percentage Effects					
Female x Post x One-Building Team	-21%	-13.1%	-12%	-12.2%	-32.5%
Female x One-Building	25.7%	10.7%	8.7%	44.2%	47.6%
# Teams	304	304	304	304	304
# Engineers	1,055	1,055	1,055	1,055	1,055
# Engineer-Months	9,304	9,304	9,304	9,304	9,304

Notes: This table investigates the gendered impacts of proximity to teammates. Panel (a) shows difference-in-differences designs, conditional on our preferred set of controls (as in Table 3), for male and female programmers. This allows the difference-in-differences coefficients to be different by gender, while allowing the controls to be estimated jointly. Panel (b) shows the triple difference design, testing the difference in the estimated effects for female and male engineers. The sample includes engineers who submit programs to the firm's main code-base in the month and whose teams are all in the firm's main campus. Standard errors are clustered by team. $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$.

Table A.14: Proximity, Gender and Dimensions of Online Feedback

Panel (a): Feedback Length, Delay, and Mentions of Other Online Conversations

	Comments per Program	Total Characters	Hours to Comment	% Other Online Convo
	(1)	(2)	(3)	(4)
Female: Post x One-Building Team	-2.83^{***}	-338.20^{***}	1.93	0.90
	(0.81)	(110.70)	(1.29)	(1.63)
Female: One-Building Team	3.84^{***}	469.90^{***}	-2.60^{***}	1.59
	(0.77)	(102.10)	(0.96)	(1.31)
Male: Post x One-Building Team	-0.83^{*}	-98.23	0.83	-1.66^{*}
	(0.45)	(62.61)	(0.62)	(0.92)
Male: One-Building Team	1.38^{***}	147.10^{**}	-1.20^{**}	2.55^{***}
	(0.52)	(66.25)	(0.61)	(0.95)
Pre-Mean, Female, One-Building Team	9.56	1048.15	14.78	3.56
Pre-Mean, Male, One-Building Team	7.67	781.09	16.33	4.18
Percentage Effects				
Female: Post x One-Building Team	-29.6%	-32.3%	13.1%	25.4%
Female: One-Building Team	40.1%	44.8%	-17.6%	44.7%
Male: Post x One-Building Team	-10.8%	-12.6%	5.1%	-39.7%
Male: One-Building Team	18%	18.8%	-7.4%	61%

Panel (b): Intensive and Extensive Margins

	Comments per Program	Total Characters	Hours to Comment	% Other Online Convo
	(1)	(2)	(3)	(4)
Female x Post x One-Building Team	-2.01^{**}	-240.00^{**}	1.11	2.56
	(0.82)	(119.10)	(1.32)	(1.80)
Female x One-Building Team	2.46^{***}	322.80^{***}	-1.39	-0.96
	(0.76)	(104.90)	(0.94)	(1.37)
Percentage Effects				
Female x Post x One-Building Team	-21%	-22.9%	7.5%	71.9%
Female x One-Building	25.7%	30.8%	-9.4%	-26.8%
# 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 considers alternative metrics of the extent and timeliness of feedback by gender. Panel (a) shows difference-in-differences designs, conditional on our preferred set of controls (as in Table 3), for male and female programmers. This allows the difference-in-differences coefficients to be different by gender, while allowing the controls to be estimated jointly. Panel (b) shows the triple difference design, testing the difference in the estimated effects for female and male engineers. The sample includes engineers who submit programs to the firm's main code-base in the month and whose teams are all in the firm's main campus. Standard errors are clustered by team. $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$.

Table A.15: Gendered Impacts of Proximity Conditional on Tenure & Age

	Comments per Program					
	(1)	(2)	(3)	(4)	(5)	(6)
Female x Post x One-Building Team	-2.01^{**}	-1.98^{**}	-1.83^{**}	-1.65^{*}	-1.83^{*}	-1.68^{*}
	(0.82)	(0.81)	(0.86)	(0.88)	(0.97)	(1.01)
Female x One-Building Team	2.46^{***}	2.42^{***}	2.47^{***}	2.32^{***}	2.42^{***}	2.28^{**}
	(0.76)	(0.75)	(0.80)	(0.82)	(0.93)	(0.97)
Female Pre-Mean, One-Building Team	9.56	9.56	9.56	9.56	9.56	9.56
Male Pre-Mean, One-Building Team	7.67	7.67	7.67	7.67	7.67	7.67
Percentage Effects						
Female x Post x One-Building Team	-21%	-20.7%	-19.2%	-17.3%	-19.1%	-17.6%
Female x One-Building	25.7%	25.3%	25.8%	24.3%	25.3%	23.8%
Diff-in-Diff Interaction						
Junior (Months at Firm < 16mo)		✓	✓	✓	✓	✓
Months at Firm FE			✓	✓	✓	✓
Age < 30				✓	✓	✓
Age FE					✓	✓
Age < 30 x Junior						✓
# 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 investigates the gendered impacts of proximity to teammates while allowing for differential effects of proximity by gender and age. Column 1 repeats Column 1 of Table A.13(b) for reference. Column 2 allows for the effect of being on a one-building team to vary for junior versus senior engineers. Column 3 allows this interaction to vary by the precise number of months that the worker has been at the firm. Column 4 further allows the effect of proximity to differ for young engineers under 30. Column 5 allows for the effect of proximity to differ for engineers of each age. Column 6 allows for an interaction between being junior and young. Standard errors are clustered by team. *p<0.1; **p<0.05; ***p<0.01.

Table A.16: Gendered Impacts of Proximity on Mentorship and Output

	Panel (a): Difference-in-Differences in Mentorship & Output			
	Comments per Review	Programs	Lines of Code	Files Changed
	(1)	(2)	(3)	(4)
Female: Post x One-Building Team	-1.31^{**}	0.62^{***}	134.70^{**}	3.77^{**}
	(0.60)	(0.22)	(57.73)	(1.51)
Female: One-Building Team	1.13^{**}	-0.39^{*}	-128.50^{*}	-3.89^{**}
	(0.57)	(0.24)	(68.82)	(1.81)
Male: Post x One-Building Team	-0.20	0.37^{***}	82.21^{*}	0.88
	(0.22)	(0.14)	(43.28)	(1.06)
Male: One-Building Team	0.32	-0.32^{*}	-176.00^{***}	-3.50^{***}
	(0.27)	(0.17)	(48.84)	(1.14)
Female Pre-Mean, One-Building Team	4.15	1.38	295.56	7.78
Male Pre-Mean, One-Building Team	3.96	1.8	322.75	9.51
Percentage Effects				
Female: Post x One-Building Team	-31.5%	44.7%	45.6%	48.5%
Female: One-Building	27.2%	-28.4%	-43.5%	-49.9%
Male: Post x One-Building Team	-5%	20.4%	25.5%	9.3%
Male: One-Building	8.1%	-17.9%	-54.5%	-36.8%

	Panel (b): Triple Difference in Mentorship & Output			
	Comments per Review	Programs	Lines of Code	Files Changed
	(1)	(2)	(3)	(4)
Female x Post x One-Building Team	-1.11^{*}	0.25	52.48	2.89^{*}
	(0.62)	(0.24)	(62.84)	(1.64)
Female x One-Building Team	0.81	-0.07	47.52	-0.39
	(0.57)	(0.24)	(74.43)	(1.86)
Percentage Effects				
Female x Post x One-Building Team	-26.7%	18.2%	17.8%	37.2%
Female x One-Building	19.5%	-5%	16.1%	-5%
# Teams	304	304	304	304
# Engineers	1,055	1,055	1,055	1,055
# Engineer-Months	16,058	16,058	16,058	16,058

Notes: This table investigates the gendered impacts of proximity to teammates on mentorship given and programming output. Panel (a) shows difference-in-differences designs, conditional on our preferred set of controls for engineering group and tenure at the firm, for male and female programmers. This allows the difference-in-differences coefficients to be different by gender, while allowing the controls to be estimated jointly. Panel (b) shows the triple difference design, testing the difference in the estimated effects for female and male engineers. The sample includes engineers who ever submit programs to the firm's main code-base and whose teams are all in the firm's main campus. Standard errors are clustered by team. *p<0.1; **p<0.05; ***p<0.01.

Table A.17: Effect of Proximity to Teammates on Pay Raises for Engineers Who Built Internal Tools

	% Pay Raise					
	Offices Open			Offices Closed		
	(1)	(2)	(3)	(4)	(5)	(6)
One-Building Team	-3.78^{**}			4.72		
	(1.62)			(4.11)		
Junior (< 16mo) x One-Building Team		-5.42^{**}			4.90	
		(2.38)			(4.61)	
Senior ($\geq$ 16mo) x One-Building Team		-1.59			1.73	
		(5.70)			(7.12)	
Female x One-Building Team			-15.68^{**}			9.17
			(6.21)			(11.05)
Male x One-Building Team			-2.97			4.04
			(2.36)			(3.96)
Dependent Mean	13.53	13.53	13.53	38.72	38.72	38.72
Junior (< 16mo) Mean	17.21	17.21	17.21	17.21	17.21	17.21
Senior ($\geq$ 16mo) Mean	24.46	24.46	24.46	42.97	42.97	42.97
Female Mean	21.67	21.67	21.67	36.36	36.36	36.36
Male Mean	19.19	19.19	19.19	39.1	39.1	39.1
Percentage Effect						
One-Building Team	-27.9%			12.2%		
Junior (< 16mo) x One-Building Team		-31.5%			13.3%	
Senior ($\geq$ 16mo) x One-Building Team		-6.5%			4%	
Female x One-Building Team			-72.4%			25.2%
Male x One-Building Team			-15.5%			10.3%
Preferred Controls	✓	✓	✓	✓	✓	✓
# Teams	262	262	262	256	256	256
# Engineers	801	801	801	720	720	720
# Engineer-Review	1,988	1,988	1,988	1,851	1,851	1,851

Notes: This table replicates Table 5 but limits the sample to engineers who worked on internal tools used by other workers at the firm. Since it was useful for these engineers to sit near the other engineers who used their tools, these engineers' teams were more likely to end up being split across the two buildings on the firm's main campus. Standard errors are clustered by team. *p<0.1; **p<0.05; ***p<0.01.

Table A.18: Effect of Proximity to Teammates on Salary Increases

	Offices Open			Offices Closed		
	(1)	(2)	(3)	(4)	(5)	(6)
			(a): Salary Increase (in $s)			
One-Building Team	-238			239		
	(221)			(303)		
Junior (< 16mo) x One-Building Team		-142			228	
		(316)			(341)	
Senior ($\geq$ 16mo) x One-Building Team		-453			361	
		(661)			(495)	
Female x One-Building Team			-923			924
			(764)			(798)
Male x One-Building Team			-86			111
			(313)			(324)
Dependent Mean	1,626	1,626	1,626	2,867	2,867	2,867
(Junior, Senior) Means	(1.9K,3.2K)	(1.9K,3.2K)	(1.9K,3.2K)	(2.6K,3.4K)	(2.6K,3.4K)	(2.6K,3.4K)
(Female, Male) Means	(2.8K,2.2K)	(2.8K,2.2K)	(2.8K,2.2K)	(3.4K,2.8K)	(3.4K,2.8K)	(3.4K,2.8K)
Percentage Effect						
One-Building Team	-14.6%			8.4%		
Junior (< 16mo) x One-Building Team		-7.5%			8.9%	
Senior ($\geq$ 16mo) x One-Building Team		-14.3%			10.6%	
Female x One-Building Team			-32.5%			27.3%
Male x One-Building Team			-3.9%			4%
		(b): Inverse Hyperbolic Sine of Salary Increase				
One-Building Team	-0.32^{**}			0.36		
	(0.15)			(0.31)		
Junior (< 16mo) x One-Building Team		-0.47^{**}			0.62^{*}	
		(0.21)			(0.34)	
Senior ($\geq$ 16mo) x One-Building Team		-0.08			-0.09	
		(0.38)			(0.45)	
Female x One-Building Team			-0.69			1.04
			(0.47)			(0.65)
Male x One-Building Team			-0.28			0.23
			(0.20)			(0.32)
Percentage Effect						
One-Building Team	-24.5%			9.7%		
Junior (< 16mo) x One-Building Team		-28.9%			17%	
Senior ($\geq$ 16mo) x One-Building Team		-3.4%			-2.4%	
Female x One-Building Team			-34.5%			26.8%
Male x One-Building Team			-15.1%			6.2%
Preferred Controls	✓	✓	✓	✓	✓	✓
# Teams	262	262	262	256	256	256
# Engineers	801	801	801	720	720	720
# Engineer-Review	1,988	1,988	1,988	1,851	1,851	1,851

Notes: This table replicates the analysis of pay raises in Table 5 but considers the absolute changes in salaries in (a) and the inverse hyperbolic sine of these changes in (b). Standard errors are clustered by team. *p<0.1; **p<0.05; ***p<0.01.

Table A.19: Effect of Proximity to Teammates on Firings

	% Fired					
	Offices Open			Offices Closed		
	(1)	(2)	(3)	(4)	(5)	(6)
One-Building Team	-0.04			-0.18		
	(0.09)			(0.12)		
Junior ($<$ 16mo) x One-Building Team		-0.12			-0.17	
		(0.12)			(0.13)	
Senior ($\geq$ 16mo) x One-Building Team		0.11			-0.21	
		(0.11)			(0.29)	
Female x One-Building Team			0.01			-0.32
			(0.06)			(0.34)
Male x One-Building Team			-0.06			-0.15
			(0.11)			(0.13)
Dependent Mean	0.07	0.07	0.07	0.18	0.18	0.18
Junior ($<$ 16mo) Mean	0.08	0.08	0.08	0.08	0.08	0.08
Senior ($\geq$ 16mo) Mean	0.05	0.05	0.05	0.24	0.24	0.24
Female Mean	0	0	0	0.3	0.3	0.3
Male Mean	0.08	0.08	0.08	1.83	1.83	1.83
Percentage Effect						
One-Building Team	-59.2%			-97.2%		
Junior ($<$ 16mo) x One-Building Team		-150.2%			-101.9%	
Senior ($\geq$ 16mo) x One-Building Team		213.4%			-88.4%	
Female x One-Building Team			NA			-106.7%
Male x One-Building Team			-66%			-8%
Preferred Controls	✓	✓	✓	✓	✓	✓
# Teams	303	303	303	297	297	297
# Engineers	1,055	1,055	1,055	994	994	994
# Engineer-Month	6,812	6,812	6,812	9,288	9,288	9,288

Notes: This table investigates how the likelihood of being fired differs for engineers on one-building teams while the offices were open (Columns $1-3$) and after the offices closed (Columns $4-6$). Each column includes our preferred, time-varying controls for engineering type and firm tenure. Each observation is an engineer-month. The sample includes engineers who worked on the main code-base, whose teammates were all in the main campus, and who were hired before the start of our sample. Standard errors are clustered by team. *p<0.1; **p<0.05; ***p<0.01.

Table A.20: Effect of Proximity to Teammates on Quits to Higher and Lower Paying Jobs

	% Quit								
	All	Higher Pay	Lower Pay	All	Higher Pay	Lower Pay	All	Higher Pay	Lower Pay
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
Post x One-Building Team	0.67**	0.43*	0.24						
	(0.29)	(0.24)	(0.17)						
One-Building Team	−0.22	−0.13	−0.09						
	(0.20)	(0.15)	(0.12)						
Junior (< 16mo) x Post x One-Building				0.80**	0.58**	0.22			
				(0.35)	(0.29)	(0.22)			
Junior (< 16mo) x One-Building				−0.24	−0.11	−0.13			
				(0.22)	(0.16)	(0.17)			
Senior (≥ 16mo) x Post x One-Building				0.41	0.11	0.30			
				(0.50)	(0.46)	(0.19)			
Senior (≥ 16mo) x One-Building				−0.20	−0.16	−0.05			
				(0.34)	(0.34)	(0.06)			
Female x Post x One-Building							2.46***	1.97**	0.50*
							(0.81)	(0.77)	(0.30)
Female x One-Building							−0.53	−0.59	0.06
							(0.58)	(0.57)	(0.13)
Male x Post x One-Building							0.30	0.11	0.19
							(0.32)	(0.26)	(0.19)
Male x One-Building							−0.15	−0.03	−0.12
							(0.21)	(0.14)	(0.14)
Dependent Mean	0.70%	0.55%	0.15%	0.70%	0.55%	0.15%	0.70%	0.55%	0.15%
Preferred Controls	✓	✓	✓	✓	✓	✓	✓	✓	✓
# Engineer-Months	13,169	13,169	13,169	13,169	13,169	13,169	13,169	13,169	13,169
# Engineers	849	849	849	849	849	849	849	849	849
# Engineer Teams	276	276	276	276	276	276	276	276	276

Notes: This table investigates how the likelihood of a quit for a higher or lower paying job differs for engineers on one- and multi-building teams around the office closures. We categorize a worker as quitting for higher pay if they quit for a position with higher average compensation (including base salary, bonuses, and stocks) as measured by Glassdoor data. We compare those who quit for higher pay to workers who did not quit or quit for a position with lower average compensation. We exclude workers who quit but for whom we do not know the destination compensation. The sample includes engineers who worked on the main code-base, whose teammates were all in the main campus, and who were hired before the start of our sample. Each column includes our preferred, time-varying controls for engineering type and firm tenure. Standard errors are clustered by team. *p<0.1; **p<0.05; ***p<0.01.