

Supplemental Appendix

Congestion in Onboarding Workers and Sticky R&D

JUSTIN BLOESCH JACOB P. WEBER

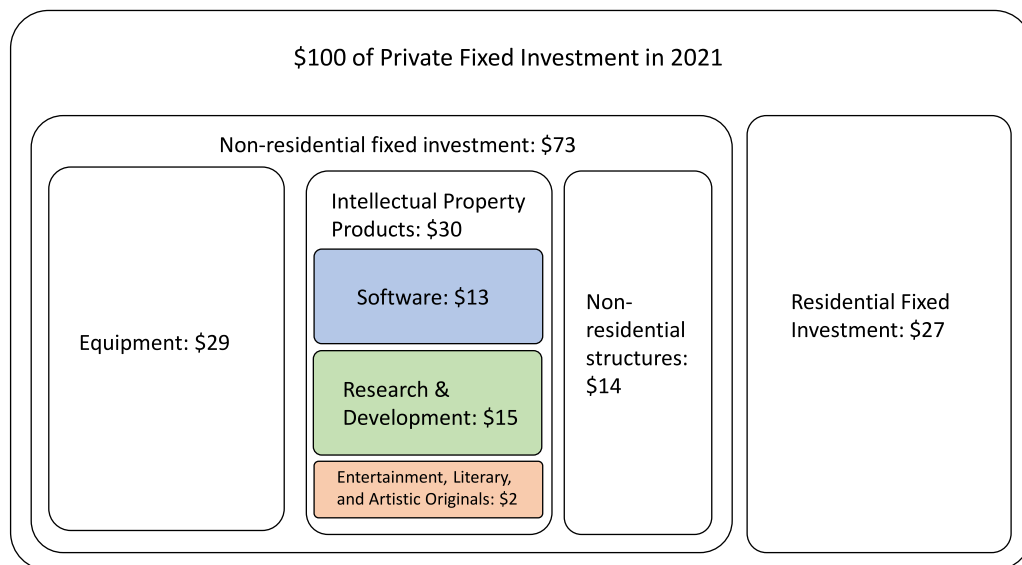
A BEA’s Treatment of Software and R&D Spending in Intellectual Property Products (IPP) Investment

This appendix presents summary statistics illustrating the growing importance of software and R&D in US investment spending, and elaborates on the various ways software spending appears in the NIPAs.

The BEA began capitalizing expenditures on software as investment in 1999, and other R&D expenses as investment in 2013, reflecting their growing importance. These are generally measured at cost, including e.g. the wages and salaries of workers involved in development; see the NIPA handbook Ch. 6 ([BEA, 2021](#)) for details. Non-residential, fixed investment (i.e. not counting inventories) thus now consists of structures, equipment, and a new category called “Intellectual Property Products” (IPP). IPP contains both software expenditures, R&D (including software R&D), and a small share of “literary arts and originals” investment, e.g. the production of films, books, etc. The BEA’s definition of intangible investment (IPP) is narrow in the sense that the NIPAs do not capitalize e.g. marketing or advertising expenses, finance and insurance costs of new product development, training costs, or organizational capital as investment; see [Koh, Santaeulàlia-Llopis and Zheng \(2020\)](#) for a discussion. Figure 11 illustrates this new breakdown for fixed investment quantitatively for the year 2021.⁴⁰ Ignoring the “Literary and Artistic Originals” component, which has remained stable as a share of investment, the remaining components of IPP have risen steadily as a share of US investment, as shown in Figure 12.

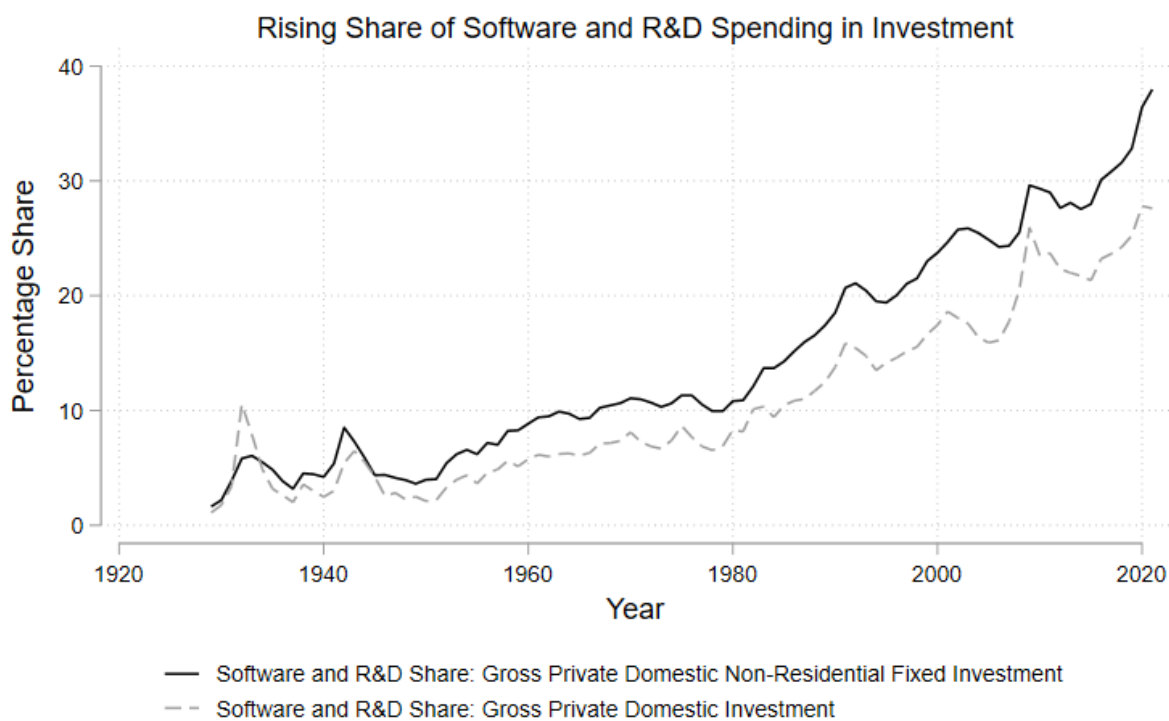
⁴⁰This figure replicates Figure 1 in [Howes and von Ende-Becker \(2022\)](#) but for the year 2021 instead of 2020. Note also that their Figure 1’s exact dollar amounts reflect outdated GDP statistics: as of the September 29th, 2022 revision IPP was larger than equipment in 2020 as claimed in the introduction (see NIPA table 1.1.5).

Figure 11: The Components of US Fixed Investment



Notes: Components of fixed investment (i.e. excluding inventories) in the US national accounts. Note that a large share of R&D is software R&D. Source: BEA and Authors' calculations.

Figure 12: Secular Rise in Software and R&D Investment



Source: BEA.

Notes: Software and R&D—the two largest components of IPP investment—have risen steadily as a share of US investment. The excluded category “Literary and Artistic Originals” is a small share of U.S. investment and has been stable over time. Source: BEA.

The “Software” category of IPP includes purchases of prepackaged software and of customized software from companies that are “primarily engaged in software development,” as well as expenditures for the own-account production of new or “significantly enhanced” software that a firm develops in-house.⁴¹ Own-account software does not include the development of software originals from which copies are made for sale (i.e. product development) or incorporated into other products (such as vehicles or appliances); these expenses are instead included in the R&D category of IPP (BEA, 2021) reflecting recent changes in 2018 (Moylan and Okubo, 2020). Roughly 1/3 of R&D is software R&D (32.2% in 2016), reflecting a secular increase over the past two decades (Moris, 2019).

While the BEA does not currently publish the components of R&D separately by type, underlying NSF survey data permits separating software R&D from other kinds of R&D for specific years. Table 2 presents a breakdown of IPP for 2016 using data from the NSF in Moris (2019) and the BEA to break out software R&D from other R&D, showing that software expenditures make up a majority of IPP.

Table 2: Composition of Non-Residential Investment in 2016

Category	Investment Share (Ppt.)
Software R&D and Other Software Investment	19.1
Software R&D	5.3
Other Software Investment	13.8
Non-software R&D	11.1
Literary and Artistic Originals	3.2
Equipment and Structures	66.7

Notes: Shares of US Gross Private Domestic Fixed, Non-residential Investment. Source: authors’ calculations from BEA data and NSF data in Moris (2019).

B Proof of Proposition 1 and Discussion

Statement: Consider the problem of a firm choosing paths $\{I_{t+1}, J_t, S_t\}_{t=0}^{\infty}$ subject to the law of motion (1) and the production function $I_t = S_{t-1}^{\nu}$ to maximize the expected, present discounted value of current and future profits (2). Assume $\rho(x) \in [0, 1]$ and $\rho'(x) < 0$ on some

⁴¹Prepackaged software excludes software embedded, or bundled, in computers and other equipment (Moris, 2019).

domain D that does not restrict the firm's optimal choice. Further, assume $f(x) \equiv \rho(x)x$ is strictly increasing on D , and that $f''(x) < 0$ on D .

Under these assumptions, then in any solution where (1) binds always and $J_t > 0$ always, so that the firm's problem can be written as:

$$\max_{\{I_{t+1}, J_t, S_t\}_{t=0}^{\infty}} \mathbf{E} \left[\sum_{t=0}^{\infty} \Lambda_{0,t} [P_t^k I_t - W_t(S_{t-1} + J_t)] \right]$$

subject to

$$\begin{aligned} I_t &= S_{t-1}^\nu \\ S_t &= (1-d)S_{t-1} + \rho \left(\frac{J_t}{S_{t-1}} \right) J_t, \end{aligned}$$

there exists an equivalent maximization problem yielding the same solution for I_t :

$$\max_{\{I_{t+1}\}_{t=0}^{\infty}} \mathbf{E} \left[\sum_{t=0}^{\infty} \Lambda_{0,t} \left[P_t^k I_t - W_t \left(1 + \underbrace{\Phi \left(\frac{I_{t+1}}{I_t} \right)}_{\text{Convex Adjustment Costs from Onboarding}} \right) I_t^{\frac{1}{\nu}} \right] \right]$$

and a domain G which does not restrict firm's optimal choice and where $\Phi' > 0$ on G . Further, under the assumption $f''(x) < 0$ on D , then it also follows that $\Phi'' > 0$ on G .

Proof: Since the law of motion (1) from the main text binds and $J_t > 0$ by assumption (so that the complementary slackness condition on this constraint can be ignored), rewrite the binding law of motion for S_t , equation (1), as:

$$\frac{S_t}{S_{t-1}} = (1-d) + \rho \left(\frac{J_t}{S_{t-1}} \right) \frac{J_t}{S_{t-1}} \equiv (1-d) + f \left(\frac{J_t}{S_{t-1}} \right) \quad (11)$$

Assume the function $f \left(\frac{J_t}{S_{t-1}} \right) = \rho \left(\frac{J_t}{S_{t-1}} \right) \frac{J_t}{S_{t-1}}$ is strictly increasing (and therefore invertible) and concave in $\frac{J_t}{S_{t-1}}$ on D . Note D is a subset of $[0, \infty)$ since $J_t \geq 0$ implies $J_t/S_{t-1} \geq 0$. Then (11) implies that $\frac{S_t}{S_{t-1}}$ is a concave, strictly increasing function of the term $(1-d)$ and the ratio $\frac{J_t}{S_{t-1}}$. Define this function as $F(\frac{J_t}{S_{t-1}})$, suppressing dependence on d , such that $F^{-1}(\frac{S_t}{S_{t-1}}) = \frac{J_t}{S_{t-1}}$ is the inverse of $F(\cdot)$.⁴² Then F^{-1} is convex and strictly increasing on $G \equiv F(x) \forall x \in D$. Pulling an S_{t-1} out of the final term in per-period profits in (2) and

⁴²This function was defined as $F^{-1}(\cdot) \equiv \mathcal{F}(\cdot)$ in the text's Section 2.

plugging this in for the resulting J_t/S_{t-1} term yields:

$$P_t^k I_t - W_t \left(S_{t-1} + F^{-1} \left(\frac{S_t}{S_{t-1}} \right) S_{t-1} \right)$$

Now note since $\frac{S_t}{S_{t-1}}$ is an increasing, convex function of $\frac{I_{t+1}}{I_t}$, i.e. $\frac{S_t}{S_{t-1}} = \left(\frac{I_{t+1}}{I_t}\right)^\frac{1}{\nu}$, it follows that F^{-1} is an increasing, convex function of $\frac{I_{t+1}}{I_t}$. Substituting in, we obtain the following for profits in period t :

$$P_t^k I_t - W_t \left(1 + \underbrace{\Phi \left(\frac{I_{t+1}}{I_t} \right)}_{\text{Convex Adjustment Costs from Onboarding}} \right) I_t^\frac{1}{\nu}$$

which yields the result.

Discussion: for $\Phi(\cdot)$ to be increasing and convex on G , we need $f(x) = \rho(x)x$ to be increasing and concave on D . Neither follows easily from the assumptions $\rho'(x) < 0$ and $\rho(x) \in [0, 1]$. To see this consider the expressions for f' and f'' in terms of ρ ,

$$\begin{aligned} f'(x) &= \rho'(x)x + \rho(x) \\ f''(x) &= \rho''(x)x + 2\rho'(x) \end{aligned}$$

Note that for x small enough, we can always find a neighborhood where $f'(x) > 0$ and $f''(x) < 0$ under the assumption that $\rho'(x) < 0$ and the additional assumptions that $\rho'(x)$ and $\rho''(x)$ are bounded as $x \rightarrow 0$, since under these added assumptions

$$\begin{aligned} \lim_{x \rightarrow 0} f'(x) &= \rho(0) > 0 \\ \lim_{x \rightarrow 0} f''(x) &= 2\rho'(0) < 0 \end{aligned}$$

The neighborhood with $x \approx 0$ is of interest because this corresponds to a steady state of the model: when $f(x) = d$, the model is in steady state where $S_t/S_{t-1} = 1$. When d is small, x may also be small (or zero if $d = 0$). So we can have local concavity of f (and convexity of Φ) under very mild assumptions about the boundedness of the first and second derivative of ρ at zero. This is relevant since many aggregate models log-linearize around a steady state, and only require that the adjustment cost function be convex when evaluated at that point. Of course when d is large, the model's steady state may be far from $x \approx 0$. Thus, we may

remain concerned that our assumptions on f may not hold for large x .

To alleviate these concerns, we note that the requirement that $f'(x) > 0$ and $f''(x) < 0$ does not put overly restrictive requirements on ρ in light of our empirical results. In practice, those results suggest that ρ :

- is a function of J_t/S_{t-1}
- satisfies $\rho(x) \in [0, 1]$ on $x \in [0, \infty)$ and $\rho'(x) < 0$
- is convex, i.e. $\rho''(x) > 0$.

One function that satisfies all these properties is $\rho(x) \equiv \frac{1}{ax+b} + c$, given appropriate choices of $a, c \geq 0$ and $b > 0$ so that $\rho(0)$ is well-defined. To see this, note ρ is decreasing and convex:

$$\begin{aligned}\rho'(x) &= -a \left(\frac{1}{ax+b} \right)^2 < 0 \quad \forall x \geq 0 \\ \rho''(x) &= 2a^2 \left(\frac{1}{ax+b} \right)^3 > 0 \quad \forall x \geq 0\end{aligned}$$

while f is increasing and concave:

$$\begin{aligned}f'(x) &= \rho'(x)x + \rho(x) \\ &= -a \left(\frac{1}{ax+b} \right)^2 x + \frac{1}{ax+b} + c \\ &= \left(\frac{1}{ax+b} \right) \left(\frac{-ax}{ax+b} + 1 \right) + c > 0 \quad \forall x \geq 0 \\ f''(x) &= \rho''(x)x + 2\rho'(x) \\ &= 2a^2 \left(\frac{1}{ax+b} \right)^3 x - 2a \left(\frac{1}{ax+b} \right)^2 \\ &= 2a \left(\frac{1}{ax+b} \right)^2 \left(\frac{ax}{ax+b} - 1 \right) < 0 \quad \forall x \geq 0.\end{aligned}$$

Additionally, a linear function or linear approximation will also work: if $\rho(x) = b - ax$ then $f(x) = bx - ax^2$ is quadratic, and no cost-minimizing firm will ever choose a point where $x > \frac{b}{2a}$. So for the domain $x \in D \equiv [0, \frac{b}{2a})$ which does not restrict the choices of a

cost-minimizing firm:⁴³

$$\begin{aligned} f'(x) &= b - 2ax > 0 \quad \forall x \in \left[0, \frac{b}{2a}\right) \\ f''(x) &= -2a < 0 \quad \forall x. \end{aligned}$$

B.1 Equivalence With Adjustment Costs in Kline et al. (2019)

The above section assumed a single wage and a dynamic problem. Here we consider the case where juniors and seniors are paid separate wages of W^j and W^s respectively, and show how we provide microfoundations for the convex adjustment costs in the one-period problem laid out in Kline et al. (2019).

In the one-period version of the firm's problem with congestion in onboarding, a firm begins the period with some stock of incumbent, senior workers S and chooses J and potentially prices P^k , W^j and W^s as well to maximize revenue: output will be produced at the end of the period by the juniors that onboard, $\rho\left(\frac{J}{S}\right)J$, and the seniors that remain, $G(W^s)S$, where $G(W^s) \in (0, 1)$ reflects the fact that the separation rate may be a function of the wage, though the main text assumes $G(W^s) = 1 - \delta$. The firm thus maximizes

$$P^k \left(G(W^s)S + \rho\left(\frac{J}{S}\right)J \right)^\nu - W^s S - W^j J \quad (12)$$

where $\nu < 1$ reflects the presence of diminishing marginal returns in our setting (where firms are price takers), or alternatively a downward sloping demand curve as in Kline et al. (2019); see their section II.B.

Adding and subtracting a J term yields:

$$P^k \left(G(W^s)S + J - c\left(\frac{J}{S}\right)S \right)^\nu - W^s S - W^j J \quad (13)$$

where $c\left(\frac{J}{S}\right) \equiv \left(1 - \rho\left(\frac{J}{S}\right)\right)\frac{J}{S}$ is twice differentiable in $\frac{J}{S}$ and convex on $[0, \infty]$ e.g. when $\rho(x) = b - ax$ as in the main text. With the problem written this way, we can see that the convex cost of hiring more J workers comes from the fact that hiring J workers has decreasing returns with respect to output at the end of the period: as the firm hires more

⁴³There are two cases for any $x > 0$ not in D . For any choice $x \in [\frac{b}{2a}, \frac{b}{a}]$, there is a choice $x \in D$ that weakly dominates because it achieves the same growth in S_t at a smaller cost. Choosing very large $x > \frac{b}{2a}$ means paying $W_t J_t$ to *reduce* the stock of S_t given S_{t-1} – assuming that the law of motion for S binds rules out the possibility that this is optimal (in our quantitative exercise with heterogeneous firms in Appendix G, we allow for “free disposal” of S which guarantees firms would never choose $x > \frac{b}{2a}$).

and more J workers, fewer and fewer of them will become productive S workers. This is why our convex cost function shows up as a convex penalty to *output* as opposed to a separable cost term. However, with linear revenue products, the equivalence is exact up to the price term P^k : $\nu = 1$ implies

$$P^k (G(W^S)S + J) - W^S S - W^J - P^k c \left(\frac{J}{S} \right) S \quad (14)$$

as in that paper.

C Narrative Evidence on “Project-Specific Capital”

The returns to project-specific tenure that we document reflects a combination of skill-acquisition and earned trust or reputation within a team, which our model in Section 2 is general enough to encompass. We emphasize the acquisition of project-specific skills, as this frequently arises in interviews with practitioners. While human capital acquired while working on a specific project may in principle be portable (and imperfectly captured by our controls for overall programming experience) in general the evidence that knowledge gained by working on OSS projects is applicable elsewhere is weak: in a longitudinal study of contributors to the Apache project, [Hann et al. \(2004\)](#) found that increases in human capital as measured by total contributions to the project, did not lead to increased wages.⁴⁴ Moreover, in Appendix E.6 we find that estimated congestion is *larger* for projects with a longer history of design decisions—i.e., more pull requests—consistent with the idea that congestion is driven by the acquisition of project-specific capital, as opposed to just solely screening newcomers for ability or basic competence.

What is this non-portable, “project-specific” human capital acquired in the first few months? Even for experienced developers, joining a new project or a new team entails acquiring knowledge specific to how that team operates and how existing code is structured (“software architecture”). Interviews with developers reveal that project-specific knowledge such as learning about the needs and requirements of end users, the “dos and don’ts” of design for a particular project or company, and – for tacit knowledge – “knowing who knows what” all plays a role in making a newcomer productive on a software development team ([Stol et al., 2014](#)).

⁴⁴The authors took advantage of the Apache project’s unusual hierarchy, which includes five rankings, to show that instead these earned credentials explained wage growth, which they interpret as consistent with a signaling theory of the benefits to contributing to an OSS project.

Consistent with this, early studies recommending the adoption of OSS software development practices within private firms highlighted the advantages of adapting OSS development methods for private industry because of their ability to reduce onboarding times: these methods include making the entire history of design decisions and code base accessible to newcomers, and also to assign them relatively easier tasks that they can use to build skills and demonstrate their newly-acquired competence. As noted by [Torkar et al. \(2011\)](#),

“It is important to have a predefined path that allows new developers to learn while doing productive activities... If this issue is left unattended, there is a risk of placing newcomers in positions for which they are unqualified or making their learning curve unnecessarily long. With proper support from experienced developers, bug fixing and technical debt reducing activities are a good entry point for new developers. Such tasks allow new developers to familiarize themselves with the software architecture... Following this strategy, they would be ready to be incorporated sooner in regular development project activities. Additionally, resourceful developers would have a greater chance to stand out sooner, reducing employee frustration...”

Moreover, having all changes and discussions publicly logged, as GitHub and the Merge/Pull model enable, would both serve to improve onboarding and mitigate the damage done when senior workers left:

“This archive [of past design and implementation decisions] would form a useful knowledge base that can be used to lower the learning curve for newcomers and ground further decision-making for experienced developers. Moreover, this knowledge would be permanent and independent of key employees leaving a project.”

[Torkar et al. \(2011\)](#) contrasted these OSS practices with existing practices at Ericsson, a global telecommunications company, which had typical problems acclimating new workers to their in-house software development methodology, “Streamline:” it took 38% of newcomers over a month just to acclimate to the in-house methodology, and a majority never graduated from the initial “software testing” tasks that Ericsson commonly assigned to newcomers as part of the onboarding process.⁴⁵

⁴⁵Ericsson was not alone in this: in a series of workshops, [Torkar et al. \(2011\)](#) asked representatives from three large, multinational software companies to rank a list of their suggested benefits of adopting OSS methods by desirability, and found that both “Define an entry path for newcomers” and “Increase information availability and visibility” were consistently prioritized as the most important potential benefits.

OSS software projects also use simple initial tasks to build and assess competency, as [Torkar et al. \(2011\)](#) pointed out. In the words of one developer and OSS project founder surveyed and quoted by [Kalliamvakou et al. \(2015\)](#):

“Even if you are not sure if the other dev[eloper] is capable of contributing good code, you can review pull requests and if the fifth pull request is good you give him/her commit bit [the power to make direct changes in the OSS repository, i.e. merge others’ pull requests].”

Consistent with this, empirical studies find that initial OSS contributions are often more trivial tasks ([Subramanian, 2020](#)) and that a developer’s track record with a project is the single most important predictor for time-to-merge in OSS projects on GitHub ([Gousios et al., 2014](#)).⁴⁶ In short, both OSS projects and the private sector use simple initial tasks to build competence and evaluate the performance of newcomers, with private industry adopting OSS development practices in no small part because they facilitated this process. The increase in productivity we document, via a decrease in approval times, reflects this process by which juniors take on tasks, learn from seniors (e.g., during code review), and eventually both write better code (which is merged faster) and build trust, allowing them to take on more serious tasks and have their changes merged with less scrutiny.

All this suggests that attention from seniors, both for education and evaluation, is critical for onboarding juniors.

D GitHub Data Appendix

This section describes how we worked with GHTorrent data to create the datasets necessary to estimate the main text’s regressions. Before doing so, we first discuss the structure of GitHub’s data, which [Gousios \(2013\)](#) accesses through GitHub’s API to build GHTorrent.

In GitHub, every repository can be conceived of as a collection of events on a timeline that keeps track of changes to the repository’s files and the discussion surrounding those changes. Figure 13 presents an example of workflow on a GitHub project being developed using pull requests. Each grey circle represents a discrete event, or action, taken on the

⁴⁶Secondary factors were project-specific or measures of pull request size and complexity. [Gousios et al. \(2014\)](#) also investigate the determinants of pull request *acceptance*, finding that almost all pull requests are eventually merged and suggest the number may be as high as 90% once one corrects for merges occurring outside of GitHub. They report that the single most relevant factor for eventual merging is whether the files touched have been modified recently (i.e. are relevant to ongoing development). We thus do not consider this as an outcome variable for the purposes to determining whether and when a developer achieves competency.

part of some user. These events are timestamped and stored as JSON files with associated characteristics that depend on the action in question; see [Gousios and Spinellis \(2012\)](#) for a more technical description. These files can be accessed directly through GitHub’s API, subject to request limits, and some researchers, studying a small number of repositories, obtain data from GitHub’s API directly.

To obtain data on all public repositories, GHTorrent surmounts the request limit by crowdsourcing API keys from multiple donors and processes the raw, timestamped event data into a MySQL database; see Figure 1 in [Gousios \(2013\)](#) for the database schema. The result is a “snapshot” of the timeline for all public repositories on GitHub, taken at different points in time.⁴⁷ Various snapshots are currently publically available on Google BigQuery, which allows a user to query GHTorrent’s MySQL dataset using SQL. The main text uses the 2019Q2 ([Gousios, 2019](#)) snapshot; an earlier draft obtained similar results on the 2016Q3 snapshot ([Gousios, 2016](#)).⁴⁸ This broad coverage, ease of access, and preprocessing explains the popularity of GHTorrent with researchers ([Cosentino et al., 2016](#)).⁴⁹

The first step in processing the GHTorrent data is to merge data on pull request comments and actions (open, merge, etc.) to create a table of pull request activity on each repository. We then drop all repositories that have less than a specified number of “merge” events: 100 in the main text.⁵⁰ This effectively keeps repositories that are actually using GitHub to jointly develop code; as discussed in the main text, repositories that are actually personal projects or websites generally do not use pull requests, and some repositories on GitHub are “mirrors” of projects actually being developed elsewhere, and thus may have only open and closed pull requests, but no merge events.⁵¹ Effectively, by conditioning on projects with many merge events, we select only projects that are being *developed* on GitHub.

We then drop all events initiated by bots using regex filters on logins, following [Wyrich](#)

⁴⁷For example, the GHTorrent sample, or “snapshot” used in the main text of this paper is from 2019 ([Gousios, 2019](#)), so that a project that began in 2008 but was deleted in 2018 would not appear in that dataset. However, the project would appear in the 2016 sample ([Gousios, 2016](#)) used in an earlier version of this paper, and we could see the timeline of events from 2008 to 2016 there.

⁴⁸For this project, we accessed the GHTorrent dataset through BigQuery’s API in Python. To access the 2016 snapshot, the project id is “ghtorrent-bq” and the 2016 vintage has the dataset id “ght”. To access the 2019 snapshot, the dataset id is “ghtorrentmysql1906” and the project id is “MySQL1906”. For more documentation, see <https://github.com/ghtorrent/ghtorrent.org>.

⁴⁹GH Archive (<https://www.gharchive.org/>) also records this public GitHub timeline data. For a comparison of the costs and benefits of various methods of accessing GitHub’s data, see [Mombach \(2019\)](#).

⁵⁰An early version of this paper used 120 merge events as the threshold, which corresponded to keeping exactly 25% of all pull requests from the largest projects in the 2016 GHTorrent snapshot. Results in the main text are not sensitive to the choice of threshold; see Appendix [E.6](#).

⁵¹Missing merge *events* in GitHub for pull requests that are actually merged is very common—see [Gousios et al. \(2014\)](#) for a discussion.

et al. (2021). We also drop events initiated by organizational accounts (that stand in for groups of users) and events for accounts that can't be linked to a user ("fake" accounts in GHTorrent parlance; these are real users who have not configured a GitHub account and whose commit activity is tracked via email address in GHTorrent). Finally, we drop events where the user is missing. Collectively, this means dropping only 2.8% of the PR activity data, and the vast majority of this is "bot" activity (2.1% of all activity) .

There are some errors in GHTorrent: for example, we also drop duplicate events. However, having made all of the above cuts, there are a few instances of multiple recorded "open" or "merge" events for the same pull request at different times. We resolve this by keeping the earliest open event and last merge event, but these constitute a tiny fraction of the data (a few hundred events collectively in a final dataset of over 65 million events). As a last step, we remove some "test" repositories that have the word "test" in the repository name and an average time to merge measured in seconds (less than one minute) though this only removes 16 repositories.

Having made these cuts, we are left with a dataset of over 65 million pull request events on 36,537 large repositories that we consider joint attempts to develop code, which we now call *projects*.⁵² We can use this dataset in turn to create two different panel datasets that are used to produce the figures in the main text.

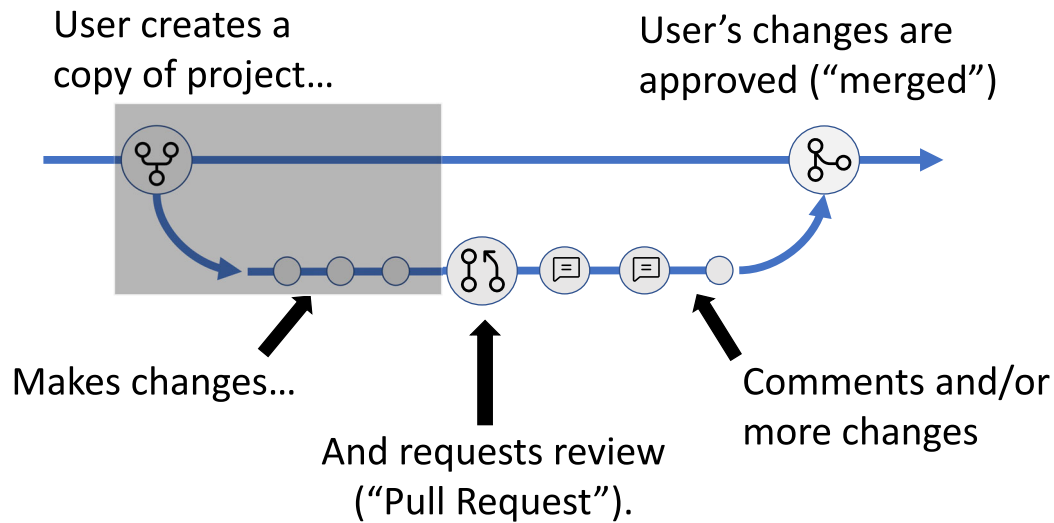
First, we build a panel dataset on individual contributions (merged pull requests) from the events in this dataset. The goal is to estimate equation (4) in the main text, reproduced here: letting $y_{i,p,t}$ be either the approval time or total comments received for a contribution opened by user i on project p at time t , we estimate the following via OLS:

$$\begin{aligned} y_{i,p,t} = & \sum_{j=1}^{13} D(\text{Months Project Experience} = j)_{i,p,t} \\ & + \sum_k D(\text{Months Programming Experience} = k)_{i,t} \\ & + D_{i,p} + \beta_{PA,p} ProjectAge_{p,t} + \epsilon_{i,p,t}. \end{aligned}$$

The first sum consists of dummy variables for having between one and thirteen or more months of experience on project p at time t , and the second sum consists of dummy variables for overall programming experience measured by GitHub account age at time t , which is capped at 49 months, so that the dummy for having exactly 49 months includes all individuals with more than four years (48 months) of experience on GitHub measured using account

⁵²We thus treat forked repositories as separate projects, if they have enough merge events; see Section 3.1.

Figure 13: Workflow on GitHub: Simplified Example



Notes: A GitHub repository can be conceived of as a collection of events on a timeline tracking changes to the repository's files and the discussion surrounding those changes. Each grey circle represents a discrete event, or action, taken by some user. These events are timestamped and stored as JSON files with additional information depending on the type of event. In this example, a user creates a copy of a repository, and then makes changes (grouped into individual "commit" events by the user) to the copy before opening a pull request to have their changes merged into the original. The grey box over this part of the workflow reflects that we may not observe this part of the contribution process: if the repository is copied as a new repository on GitHub, for example, then we will observe these changes. But if the project is e.g. downloaded by the user and then changed locally, we may not be able to observe this activity.

age.⁵³ We also allow for individual-by-project fixed effects ($D_{i,p}$) and project-specific linear time trends ($\beta_{PA,p}ProjectAge_{p,t}$). Table 3 presents example of this dataset, with fictitious data, used to estimate this regression. Recall that Figure 3 in the main text uses the marginal effects estimated from equation (4) to show that approval time and the number of comments recieve fall precipitously in the first six months of project-specific experience. This provides evidence of a nontrivial onboarding period for juniors, as discussed in the main text.

Table 3: Panel Data on Individual Contributions (Example)

n	Opened t	Contributor i	Project p	Project Experience	Approval Time	Total Comments
1	1/1/15	Jake	Project A	0 days	6 days	4
2	1/9/15	Jake	Project A	8 days	5 days	2
3	1/9/15	Jake	Project B	0 days	17 days	5
4	1/12/15	Jake	Project B	3 days	15 days	3
5	1/12/15	Justin	Project B	0 days	1 days	0

Notes: An illustrative example of the panel dataset used to estimate regression (4) in the main text. The actual dataset has $N = 10,881,355$ observations (approved contributions) on 36,537 large projects (with at least 100 approved contributions).

Next, we build a panel dataset of juniors joining new projects that is used to estimate equation (5) in the main text, yielding the estimate for ρ plotted in Figure 6. We begin by identifying junior J type and senior S type workers using their pull request activity. In each calendar month t and each project p , we assign each user with activity on at least one pull request in p at t into either category J or category S . We drop users who never contribute, and restrict attention to those who open at least one pull request that is merged. A J type transitions to an S type on a particular project either when they have reached a tenure of six months on that project, or when we observe them reviewing code (i.e., when we observe them merging/closing/commenting on pull requests opened by others). Project tenure is measured as the length of time between a user’s first observed activity and their last observed activity on a project. This definition implies that some workers are S types from their first month on a project. Also note that once a J type worker transitions on a project, they are counted as an S type in any calendar month when they “re-appear” on that project in the pull request data.

We define the quantity of J types on project p at time t as $J_{p,t}$, tabulated as the number of users who have contributed to that project (i.e. authored at least one pull request that was

⁵³A small number of user accounts and projects have pull request activity predating the reported date of account or project creation. We correct these using the time of earliest observed pull request activity.

eventually merged) at time t with less than six months of tenure and who have not reviewed code written by others (i.e. who have not been observed merging/closing/commenting on a pull request opened by someone else). The other active users are summed into $S_{p,t}$. Table 4 presents an example of this dataset, with fictitious data, used to estimate equation (5), reproduced here: letting $\mathbf{1}(i \text{ joining } p \text{ at } t \text{ onboards})$ denote an indicator function for whether a Junior i on project p (counted in the sum $J_{p,t}$) will eventually transition to being an S type on project p , we estimate:

$$\begin{aligned} \mathbf{1}(i \text{ joining } p \text{ at } t \text{ onboards}) &= \sum_b D\left(\frac{J_{p,t}}{S_{p,t}} \text{ in bin } b\right) \\ &+ D_p + \beta_{PA,p} ProjectAge_{p,t} + X_t + \gamma_{i,t} + \epsilon_{i,p,t}. \end{aligned}$$

We estimate the effect of $J_{p,t}/S_{p,t}$ non-parametrically via OLS by measuring it as a set of dummy variables representing equidistant bins for junior-senior ratios. Project specific dummies D_p control for unobservable project-specific features that may make some projects easier to join, while $ProjectAge$ is a project-specific time trend meant to capture the project life cycle, since some projects may become harder to join as they age; X_t are year fixed effects, and $\gamma_{i,t}$ captures Junior-specific controls, which include fixed effects for the year user i 's account was created, and dummies for the age of user i 's account.

We cannot include user-project specific fixed effects here because they are collinear with the outcome variable (we only observe one outcome per project for each individual: either they onboard, or they do not). Relatedly, we cannot well-estimate individual fixed effects because in practice most individuals join very few OSS projects in sample over time. Note if someone joins only one project in our sample of large OSS projects, we cannot estimate a fixed effect for them. Appendix E.2 discusses this and shows that our results are qualitatively unchanged by adding individual fixed effects, though the sample size shrinks.

We linearly approximate the non-parametric estimates of congestion in Figure 6 using OLS for use in calibrating our DSGE model. This is the blue line in that figure.

Table 4: Panel Data on Juniors Joining Projects (Example)

n	Month Joined t	Junior i	Project p	Cohort Size $J_{p,t}$	$S_{p,t}$	$\frac{J_{p,t}}{S_{p,t}}$	Onboards?
1	1/2015	Jake	Project A	2	4	.5	Y
2	1/2015	Justin	Project A	2	4	.5	N
3	1/2015	Justin	Project B	1	10	.1	Y

Notes: An illustrative example of the panel dataset used to estimate regression (5) in the main text. The actual dataset has $N = 1,044,039$ observations, one for every junior on each project joined.

E Robustness of Empirical Analysis with GitHub Data

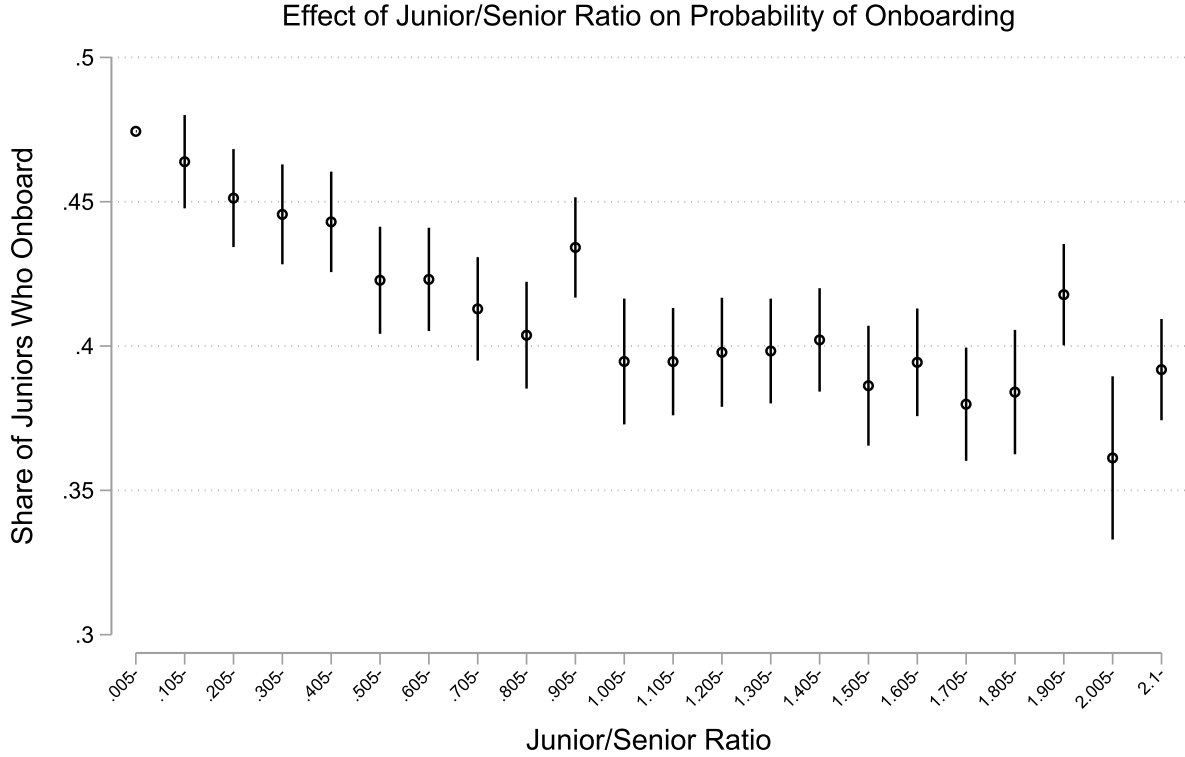
E.1 Congestion Results with More Bins

Figure 14 extends the congestion analysis of Figure 6 to allow for J/S ratios greater than 2. We prefer our headline calibration because we view it as a local linear approximation of the nonlinear function over a space which is more relevant for optimizing firms in our model: the steady-state ratio of J/S is generally much, much less than one in both the representative agent model in the main text and the heterogeneous firm model in Appendix G. We obtain a qualitatively similar negative slope.

E.2 Adding Individual Fixed Effects to the Congestion Regression

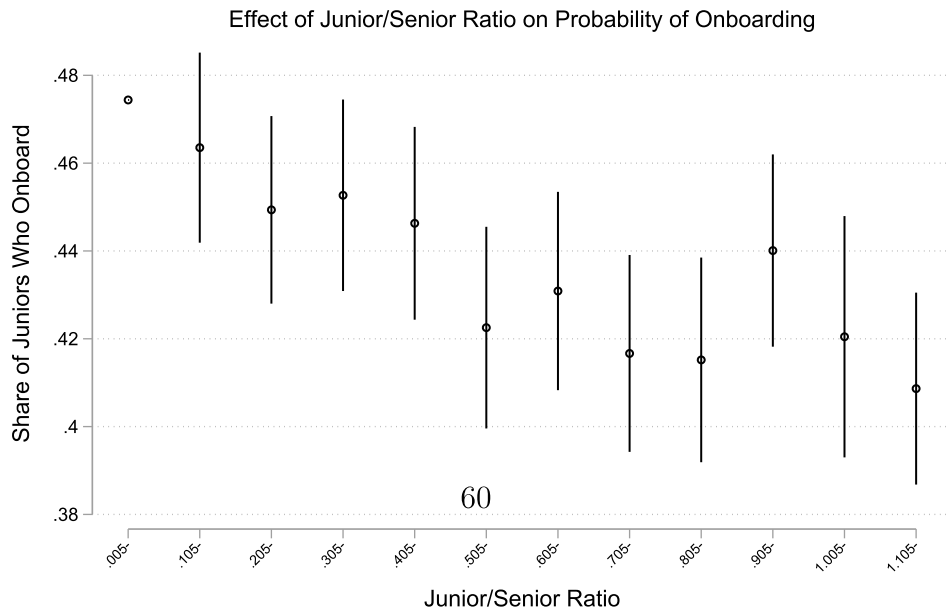
Adding individual fixed effects to the congestion regressions in the main text shrinks the sample size by about a third. This is because we can only identify individual fixed effects for contributors who successfully merge pull requests on multiple large open source projects (recall we only consider projects with many pull requests; see Section 3). Including fixed effects drops individuals who e.g. only contribute and join one major open source project. We include fixed effects here and note that in spite of the diminished sample size ($N = 646,843$ as opposed to $N = 1,044,39$ in the headline results in Figure 6), the results are qualitatively similar. See Figure 15.

Figure 14: Non-parametric Estimate of the Onboarding Function ρ



Notes: Estimates from: $\mathbf{1}(i \text{ joining } p \text{ at } t \text{ onboards}) = \sum_b D\left(\frac{J_{p,t}}{S_{p,t}} \text{ in bin } = b\right) + D_p + \beta_{PA,p}ProjectAge_{p,t} + X_t + \gamma_{i,t} + \epsilon_{i,p,t}$. The “spikes” in bins containing $J/S = 1$ or $J/S = 2$ reflects the fact that being “one-on-one” or “two-on-one” with an incumbent worker is particularly helpful for successful onboarding. Note over 75% of all project-month observations have $J/S \leq 1$ and over 90% have $J/S \leq 2$.

Figure 15: Non-parametric Estimate of the Onboarding Function ρ with Individual Contributor Fixed Effects



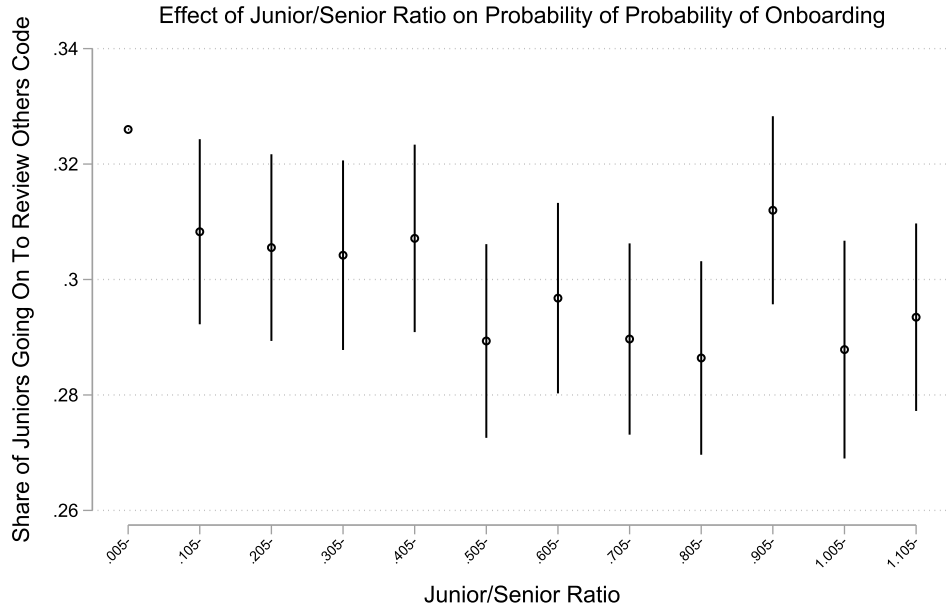
E.3 Congestion Results for Narrower Definitions of Onboarding

We count juniors as “onboarding” successfully in the main text using two observables:

- A junior goes on to remain with the project at least six months
- A junior eventually begins commenting on/merging/closing pull requests opened by others (i.e. code review).

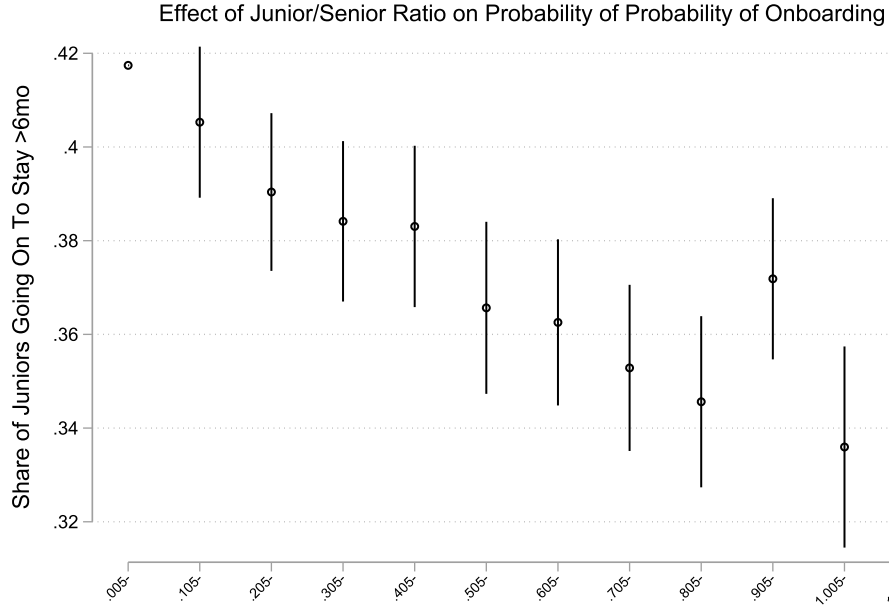
If any junior eventually satisfies one of these two conditions, they get counted as “onboarded.” As discussed, many do not and leave within a month of joining the project and without reviewing anyone else’s code. Our main specification uses both of these definitions, but using just one or the other yields qualitatively similar results: i.e., an “onboarding function” that looks downward sloping, consistent with congestion.

Figure 16: Non-parametric Estimate of the Onboarding Function with Onboarding Success Determined by Junior’s Activity Only



Notes: Estimates from: $\mathbf{1}(i \text{ joining } p \text{ at } t \text{ onboards}) = \sum_b D\left(\frac{J_{p,t}}{S_{p,t}} \text{ in bin } = b\right) + D_p + \beta_{PA,p}ProjectAge_{p,t} + X_t + \gamma_{i,t} + \epsilon_{i,p,t}$. The “spike” in the bin which contains $J/S = 1$ reflects the fact that being “one-on-one” with an incumbent worker is particularly helpful for successful onboarding. Note over 75% of all project-month observations have $J/S \leq 1$.

Figure 17: Non-parametric Estimate of the Onboarding Function with Onboarding Success Determined by Eventual Project Tenure Only

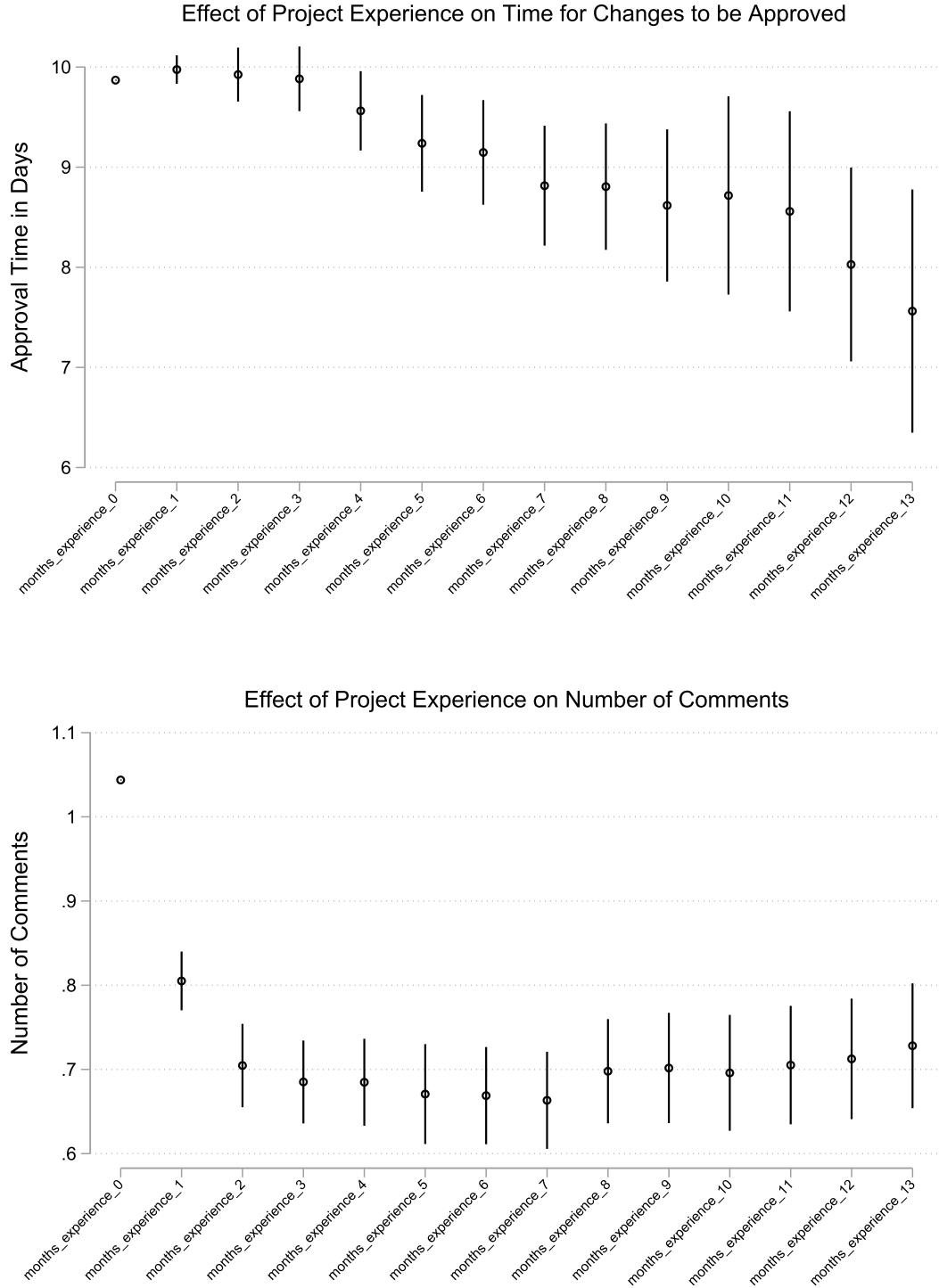


Notes: Estimates from: $\mathbf{1}(i \text{ joining } p \text{ at } t \text{ onboards}) = \sum_b D\left(\frac{J_{p,t}}{S_{p,t}} \text{ in bin } = b\right) + D_p + \beta_{PA,p}ProjectAge_{p,t} + X_t + \gamma_{i,t} + \epsilon_{i,p,t}$. The “spike” in the bin which contains $J/S = 1$ reflects the fact that being “one-on-one” with an incumbent worker is particularly helpful for successful onboarding. Note over 75% of all project-month observations have $J/S \leq 1$.

E.4 Long-Tenure Workers Who Don’t Review Code as Senior Improve Productivity with Tenure

In the main text, we treat long-tenure workers who have been with a project for at least 6 months as senior workers. This reflects the fact that these workers do appear to onboard: specifically, we re-estimate our regressions with productivity (measured as time-to-merge) and number of comments just for the subset of workers who never do code-review in Figure 18. The results suggest that these workers do get more productive and should be counted as “seniors” from the point of view of productivity, after about six months tenure.

Figure 18: Workers That Stick With Projects But Don't Review Code Still Onboard.



Notes: Unconditional mean values for a user with zero months of project-specific experience (first hollow dot) compared to predictions for an otherwise identical user with one or more months of project-specific experience (capped at ≥ 13 months). Over time, new contributors' proposed changes are approved faster and with less discussion, even if we never see them go on to review others' code. Standard errors are clustered at the project level. Source: GHTorrent and authors' calculations.

E.5 Congestion Results Dropping Juniors with only One Contribution

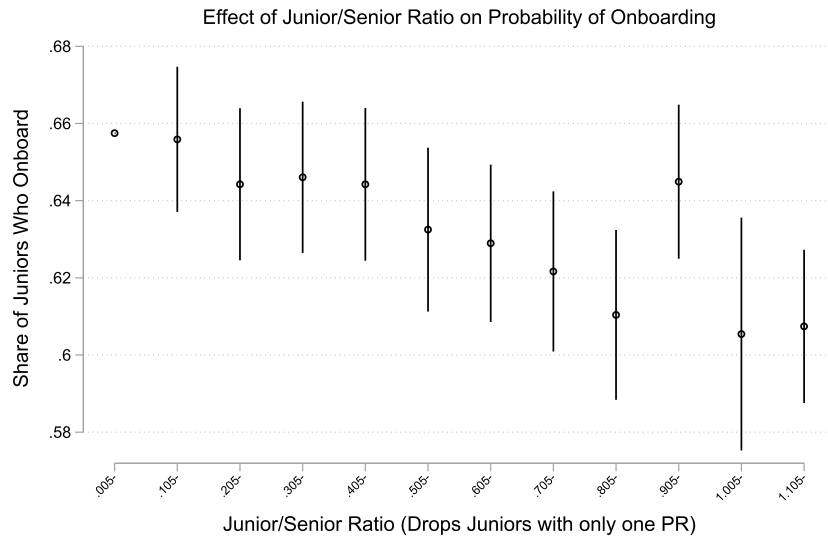
This section re-estimates our congestion regression but dropping junior workers who only attempt to contribute once (i.e., we keep juniors who open at least two or more pull requests, and have at least one merged). While these juniors make up a non-trivial share of the sample, dropping them does not affect the congestion results, meaning the slope in Figure 19 appears similar to the slope in our benchmark regression. However, the constant is mechanically higher: we are dropping J workers who—almost by definition—do not onboard, because they do not stick around for six months.⁵⁴ That raises the level of the line because the unconditional average rate of success jumps upward when we drop these onboarding “failures.”

E.6 Robustness Checks on Changing the Project Size Threshold

This section re-estimates key results from the main text, but varying the threshold for minimum project size: while in the main text we keep all repositories with at least 100 merged PRs, here we also consider thresholds of 200, 500 and 1,000 merged PRs. Table 5 presents summary statistics on the number of users, projects and project size (measured as mean PRs per project) across these four thresholds. Table 6 summarizes our estimates for the calibrated ρ function across these four samples. Continuing to control for project fixed effects, project-specific time trends, and other variables as in the main text equation (5), we find that estimated congestion is *larger* for projects with a longer history of design decisions—i.e., more pull requests—consistent with the idea that congestion is driven by the acquisition of project-specific capital, as opposed to just solely screening for newcomers’ ability.

⁵⁴It is only *almost* by definition since it is technically possible that a user could contribute exactly once and then move on to reviewing others code, and hence be counted as onboarded under our definition without ever contributing again; this is very uncommon in practice.

Figure 19: Non-parametric Estimate of the Onboarding Function ρ : Only Juniors with More than One PR



Notes: This replicates the congestion estimate in the main text but dropping users who open only one (merged) pull request. As before, the figure reports the unconditional mean onboarding probability for a junior joining a project at date t in the lowest bin for junior to senior ratio $\frac{J}{S}$ at t (first hollow dot) compared to predicted probabilities for an otherwise identical user as the ratio $\frac{J}{S}$ increases (capped at 1.105). We count juniors as successfully onboarding if they remain with the project for at least six months or begin reviewing code written by others (merging/closing/commenting on pull requests authored by other users). SEs clustered at the project level. Source: GHTorrent and authors' calculations.

Table 5: Summary Statistics for Subsamples Varying our Threshold for Large Projects

Minimum Number of PRs	100	200	500	1,000
Total Projects	36,537	14,779	4,472	1,662
Total Users	922,751	742,711	540,939	404,523
Total PRs	11,665,456	8,730,955	5,682,778	3,792,175
Mean PRs per Project	319.3	590.8	1,270.7	2,281.7

Notes: Summary statistics for four different subsamples of the GHTorrent Pull Request (PR) data, where we keep only projects with at least 100, 200, 500 or 1,000 merged PRs, respectively. The first column is the benchmark sample used in the main text.

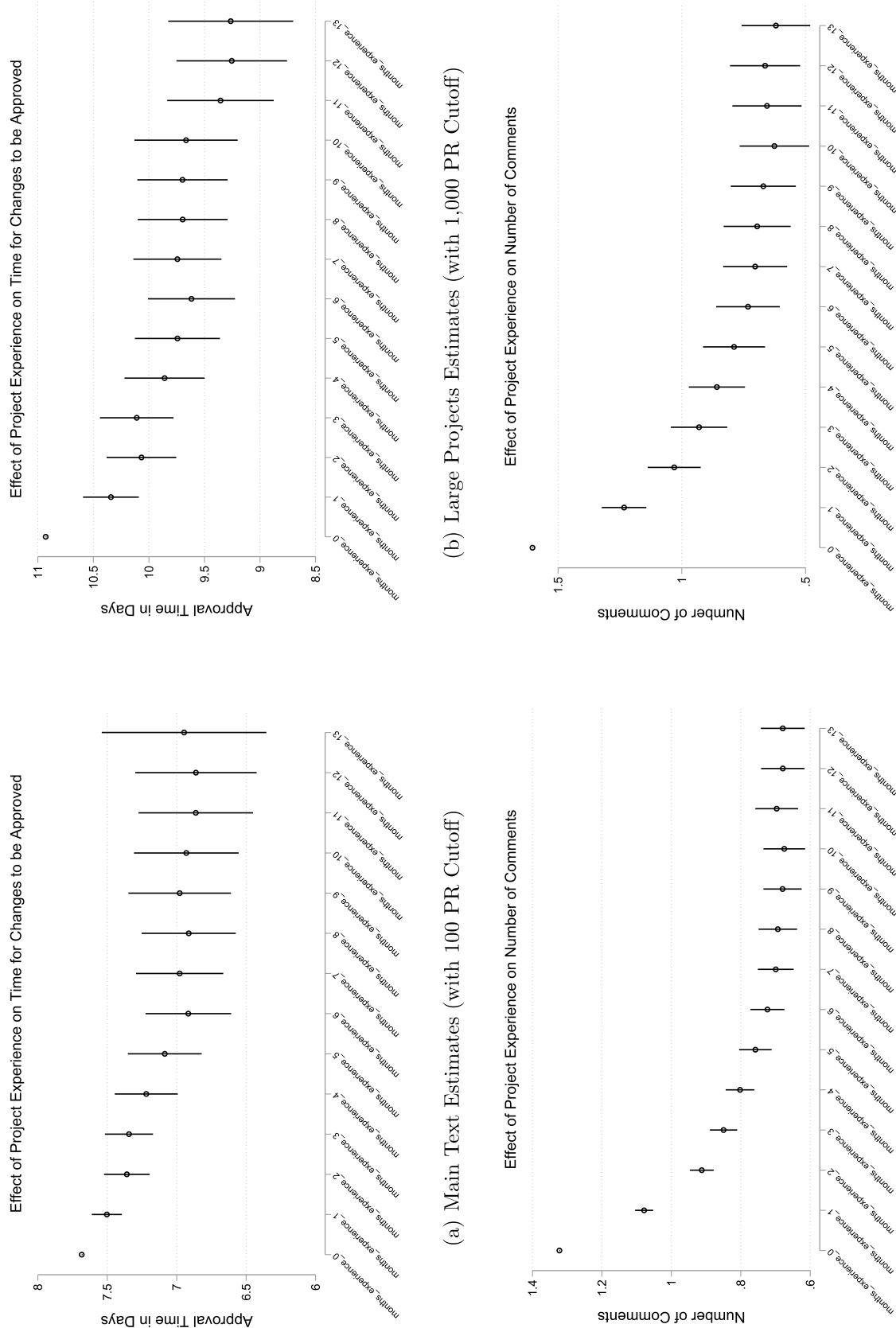
Table 6: Estimates for $\rho(x) = b + ax$ When Varying our Threshold for Large Projects

Minimum Number of PRs	100	200	500	1,000
Estimate for constant b	0.47	0.47	0.48	0.49
Estimate for slope a	-0.06	-0.08	-0.10	-0.13

Notes: This table reports estimates of ρ for projects with different minimum numbers of pull requests (PRs): 100, 200, 500, and 1000, measured by the cumulative lifetime number of PRs on the project. The estimates for the slope of ρ become more negative in large projects. The first column is the benchmark sample used in the main text. These results support the hypothesis that worker onboarding is at least partly due to learning project specific skills, in which junior workers benefit more from senior workers' attention in large and complicated projects.

Figure 20 plots results for time to onboard for the largest threshold (1,000 PRs) against the baseline threshold of 100 PRs, showing that the time to onboard is similar for larger projects. However, Figure 21 compares the nonlinear estimates for congestion underlying the linear results in Table 6, demonstrating that the congestion profile appears notably steeper (note the change in scale).

Figure 20: Estimates of Time to Onboard are Similar for Larger Projects

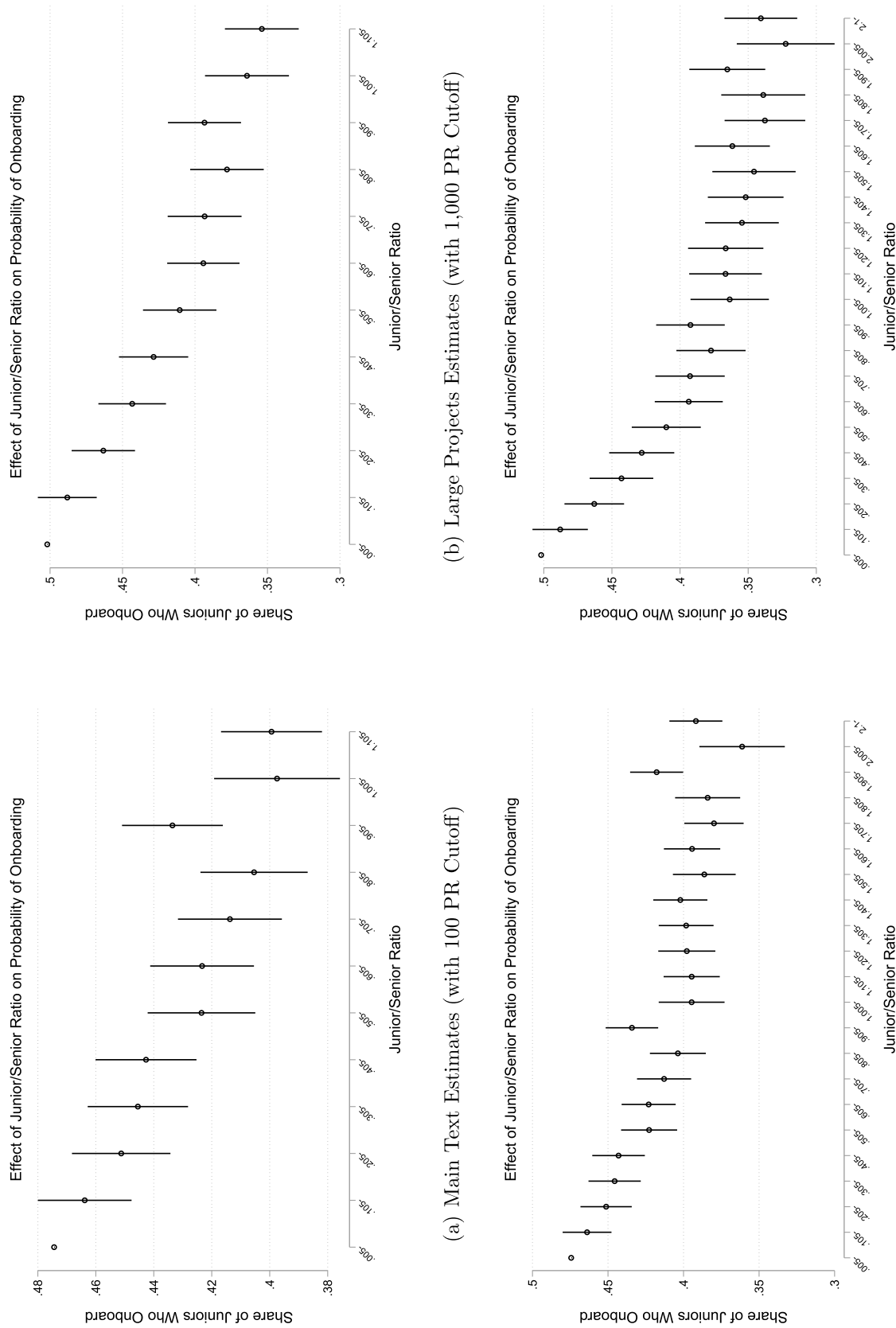


(c) Main Text Estimates (with 100 PR Cutoff)

(d) Large Projects Estimates (with 1,000 PR Cutoff)

Notes: Unconditional mean values for a user with zero months of project-specific experience (first hollow dot) compared to predictions for an otherwise identical user with one or more months of project-specific experience (capped at ≥ 13 months). SEs clustered at project level. The samples are described by the first and last columns of Table 5.

Figure 21: Estimates of Congestion (ρ) Using Only Larger Projects are Steeper



(c) Main Text Estimates (with 100 PR Cutoff): More Bins

(d) Large Projects Estimates (with 1,000 PR Cutoff): More Bins

Notes: Juniors' onboarding probabilities across various projects by junior-senior ratio at the time they joined. The left hand side uses projects with at least 100 PRs as in the main text; the right hand side uses only projects with at least 1,000 PRs. The samples and estimates of ρ are described by the first and last columns of Table 5 and Table 6. SEs clustered at project level.

F Relaxing the Equal Wage Assumption

Instead of assuming J and S workers get the same wage, W_t , we now assume that S workers get some wage $W_t^s \geq W_t$ because they have “hold up” power and can threaten to leave the firm. We assume that workers care only about maximizing wage income (since they are part of a representative household and enjoy equal consumption). The ad-hoc wage rule is:

$$W_t^s = (1 - \theta) \times W_t + \theta \times \underbrace{\nu P_t^k Z_t S_{t-1}^{\nu-1}}_{\text{Marginal Product of } S}$$

where θ reflects the worker’s bargaining weight, so that $\theta = 0$ nests our benchmark model in the main text. This is a tractable way of introducing a wage increase for S workers that is endogenous to their value to the firm.⁵⁵ Formally, the representative firm now chooses paths for $\{INTAN_{t+1}, J_t, S_t\}_{t=0}^{\infty}$ to solve⁵⁶

$$\max_{\{INTAN_{t+1}, J_t, S_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t [P_t^k INTAN_t - W_t^s S_{t-1} - W_t J_t]$$

subject to

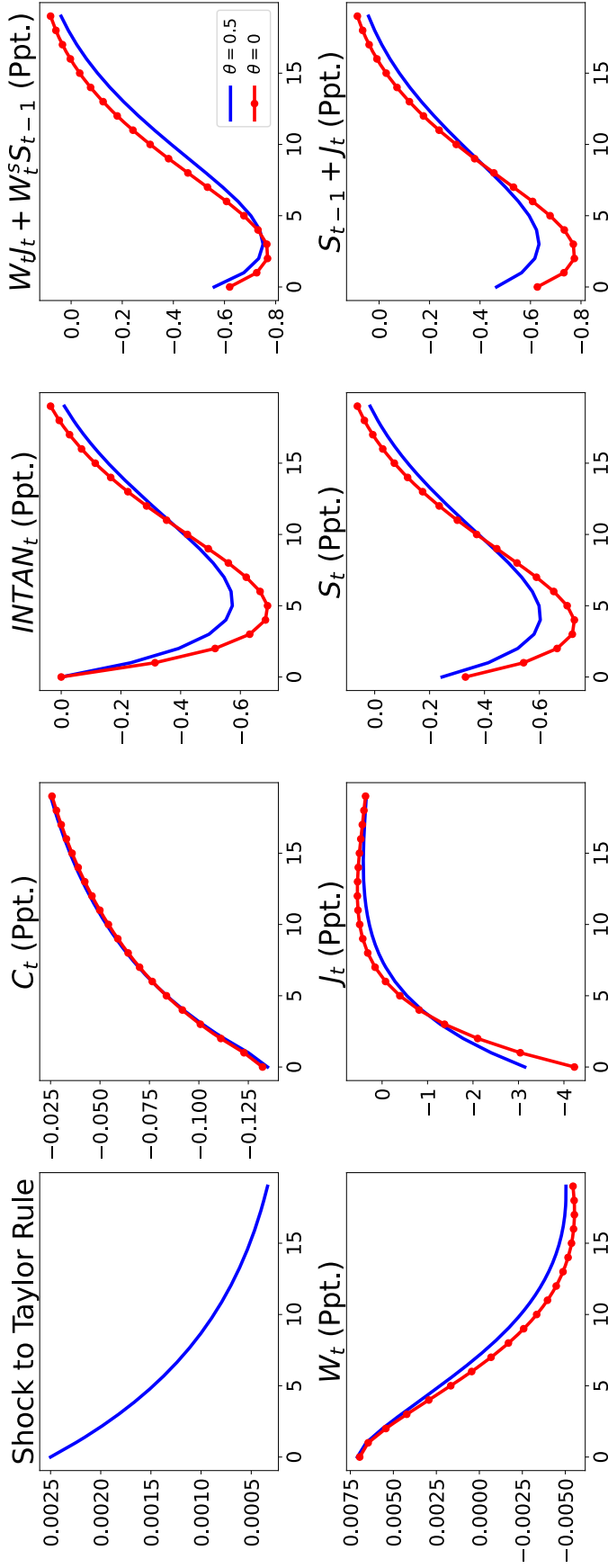
$$\begin{aligned} INTAN_t &= Z_t S_{t-1}^\nu \\ S_t &\leq (1 - d)S_{t-1} + \rho \left(\frac{J_t}{S_{t-1}} \right) J_t \\ J_t &\geq 0 \\ W_t^s &= (1 - \theta)W_t + \theta \nu P_t^k Z_t S_{t-1}^{\nu-1}. \end{aligned}$$

Note that W_t is still determined according to the same wage Phillips curve as in Section 4. We plot results for both TFP shocks and monetary policy shocks as in the main text, comparing the model with an equal weight $\theta = 0.5$ (which implies a real wage increase of 16% upon onboarding and becoming an S , in steady state) against the benchmark model with $\theta = 0$ and note that the results are qualitatively similar; see Figure 22 and Figure 23 below. The employment response of senior workers in the investment good sector is dampened when senior workers have some bargaining power, consistent with a large literature showing that wage cyclicality from bargaining dampens employment fluctuations (Shimer, 2005).

⁵⁵We view this as an ad-hoc simplification because the value to the firm *exceeds* the marginal revenue product of an onboarded S worker due to the presence of an endogenous onboarding function, etc.

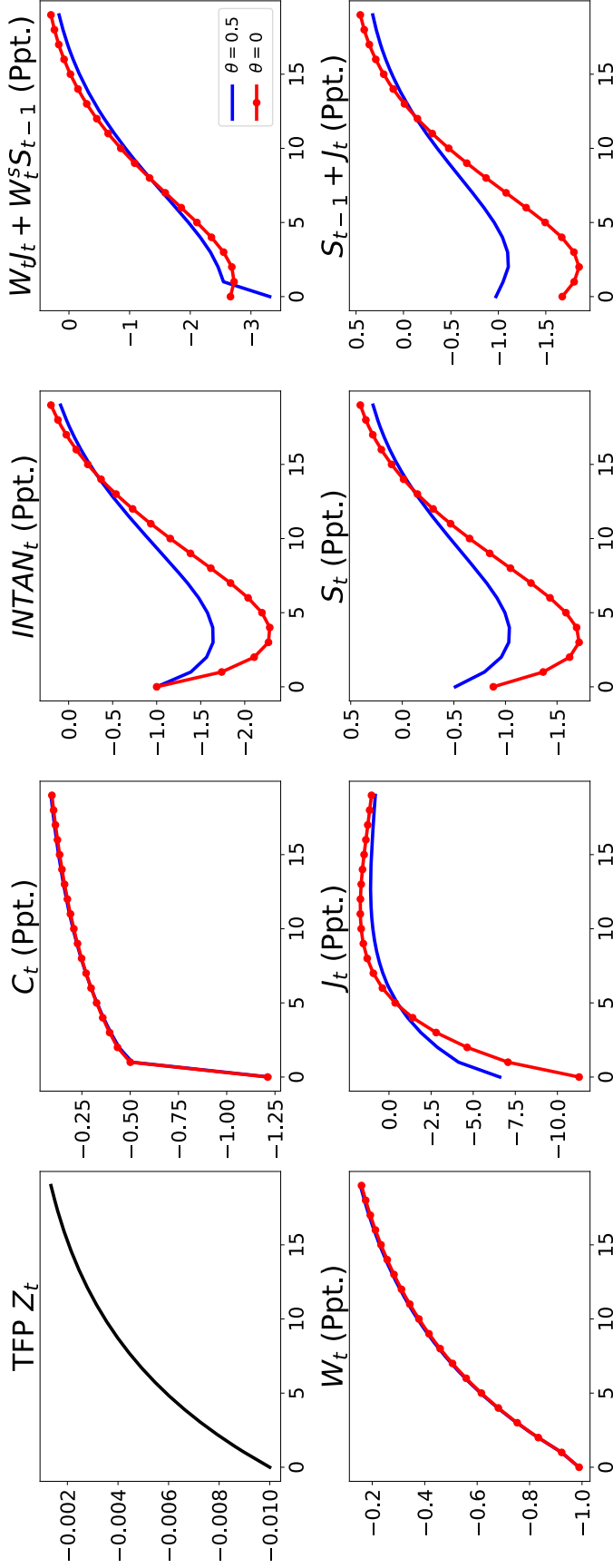
⁵⁶Intangible investment in this section is denoted $INTAN$ to align with notation in Section 4.

Figure 22: Model Responses to a Contractionary Monetary Policy Shock: Bargained S Wages With $W^s > W$



Notes: Quarterly impulse responses in the model with calibrated $\rho = .47 - .06 \left(\frac{J_t}{S_{t-1}} \right)$, where the wage for S workers is higher than for J workers and determined by an ad-hoc bargaining condition with weight θ . We compare an equal weight $\theta = 0.5$ with the benchmark model with $\theta = 0$ and note that the results are qualitatively similar.

Figure 23: Model Responses to a Contractionary TFP Shock: Bargained S Wages With $W^s > W$



Notes: Quarterly impulse responses in the model with calibrated $\rho = .47 - .06 \left(\frac{J_t}{S_{t-1}} \right)$, where the wage for S workers is higher than for J workers and determined by an ad-hoc bargaining condition with weight θ . We compare an equal weight $\theta = 0.5$ with the benchmark model with $\theta = 0$ and note that the results are qualitatively similar.

G Congestion in Onboarding With Large Shocks: Heterogeneous Firm Model

The main text solves the congestion model assuming a representative firm and linearizing around a steady-state where the firm never lays off senior workers (the law of motion for S binds) or implements a hiring freeze ($J > 0$). Under these assumptions, we can prove a theoretical equivalence between the congestion specification and $\dot{I}$ adjustment costs.

This section relaxes those assumptions by linearizing around a stochastic steady state where a continuum of R&D firms produce subject to idiosyncratic risk, and may endogenously choose $J = 0$ and S less than the quantity required by the law of motion if optimal. In this setting, our theoretical equivalence with $\dot{I}$ adjustment costs no longer holds. However, this section demonstrates that in practice, the congestion model with $\rho'(x) < 0$ still delivers a hump-shaped response of investment to TFP and monetary policy shocks.

Section G.1 describes the firm's problem in partial equilibrium, taking prices as given, and shows how congestion mutes the response of individual firms' hiring decisions to shocks. Section G.2 then solves Section 4's model but assuming heterogeneous firms and aggregate risk. We solve directly for the IRFs with respect to monetary policy and TFP shocks using the sequence space Jacobian approach in Auclert et al. (2021), and show that the response for investment remains hump shaped, as in a model with $\dot{I}$ investment adjustment costs.

G.1 Firm's Response to Idiosyncratic Productivity Shocks: PE

Relaxing the assumption that the R&D firm never lays off workers or implements a hiring freeze does not qualitatively change the results. To see this, consider a numerical solution to the firm's problem where prices (P_t^k and W_t) are taken as given and constant, but there is exogenous risk in the production process for R&D. Output is now given by:

$$I_t \equiv e_t S_{t-1}^\nu,$$

where e_t is a firm-specific productivity shock which follows a persistent Markov process. For simplicity in this section, we assume e_t only takes on two states: high or low. We also assume the firm discounts the future at an interest rate $1 + r$ also taken as given and constant. We can then solve for the firm's optimal choices given an appropriate calibration. Critically, this calibration assumes $\rho(x)$ is a decreasing function. Note that other than the number of states in the Markov process, the calibration for $\rho(x)$ and parameters ν and d follows the

general equilibrium model in Section 4. Prices W , P^k and r are also calibrated here to the endogenous steady-state solutions that arise from this calibration when solving the model in Section G.2 below, with a ρ function calibrated to what we find in GitHub data.

A firm at time t with idiosyncratic productivity e_t and incumbent, senior workers S_t has the following value function: plugging in the constraint $I_t = e_t S_{t-1}^\nu$,

$$V_t(e_t, S_{t-1}) = \max_{J_t, S_t} \left\{ P^k e_t S_{t-1}^\nu - W(S_{t-1} + J_t) + \frac{E_t[V_{t+1}(e_{t+1}, S_t)]}{1+r} \right\}.$$

Senior workers separate at rate d and juniors J_t become productive at endogenous rate ρ :

$$\begin{aligned} S_t &\leq (1-d)S_{t-1} + \rho \left(\frac{J_t}{S_{t-1}} \right) J_t \\ J_t &\geq 0. \end{aligned}$$

An increase in e_t is a positive shock to the marginal revenue product of the firm's workers. Accordingly, transitioning from the low to the high state for e_t will cause the firm to increase in size. The grey arrows in Figure 24 illustrate the adjustment of a firm that has been in the low productivity state for a long time transitioning to the high productivity state. Conditional on remaining in the high state, the firm slowly hires new workers, since congestion in onboarding means that hiring many J 's at once is costly, eventually converging to the long run optimum given by S_∞ .

To show the delayed response to negative shocks, we can work through the opposite case of a firm that has been in the high productivity state for a long time (choosing S_∞ in Figure 24) and transitions to the bad state with low marginal revenue products. The firm "hoards" S workers and responds by implementing a hiring freeze ($J = 0$), letting exogenous separations bring the firm to the long run optimum for the low productivity state (S_0 in Figure 24). This behavior is optimal because the S workers have option value: if the firm returns to the high state, it will have to pay heavy costs to rebuild the team, and so it avoids letting the size of the team get too small too quickly. Indeed, the subtle kink in the firm's policy function in the low state (the blue dotted line) reflects the point at which setting $J = 0$ sees the firm shrink too quickly, so the firm chooses $J > 0$.

Note that this labor hoarding behavior does not depend on congestion, and would be present even in a standard fixed hiring cost model. However, without congestion, there is a strong asymmetry as the firm adjusts immediately to *positive* shocks. Figure 25 shows this by repeating the exercise in Figure 24 but for the case where $\rho(x)$ is nearly constant, i.e.

$\rho'(x) \approx 0$, which is identical to a model with a fixed cost of hiring new workers.

G.2 Aggregate Response to TFP and MP Shocks with Heterogeneous Firms and Idiosyncratic Risk: GE

We now solve Section 4’s GE model but assuming heterogeneous firms. We solve directly for the IRFs with respect to monetary policy and aggregate TFP shocks using the sequence space Jacobian approach in Auclert et al. (2021), and show that the response for investment remains hump shaped, as in a model with I-dot investment adjustment costs.

As in Section 4, we abstract from standard frictions used to fit the data (e.g. endogenous capital utilization, habit formation, etc.) to isolate the effect of congestion in creating persistent dynamics in the model. We will solve and compare two different models: one with “I-dot” investment adjustment costs following Christiano et al. (2005) and standard investment production, and one where the only friction in investment production comes from congestion as described above. However, in this section, the production of investment under congestion will occur in the presence of idiosyncratic risk, as described in Section G.1.

In both models, a representative household solves a standard optimization problem, as in the main text. The household accumulates intangible capital K_t through intangible investment $INTAN_t$. We assume this is the only capital used by firms and abstract from tangible capital for simplicity. It also trades a riskless bond in zero net supply B_t which pays real interest rate r . The household solves

$$\max_{\{C_t\}_{t=0}^{\infty}, \{B_t\}_{t=0}^{\infty}, \{INTAN_t\}_{t=0}^{\infty}, \{K_t\}_{t=0}^{\infty}} \mathbf{E} \left[\sum_{t=0}^{\infty} \beta^t \left(\frac{C_t^{1-\sigma}}{1-\sigma} - \omega \frac{L_t^{1+\eta}}{1+\eta} \right) \right] \quad (15)$$

subject to standard budget and capital accumulation constraints,

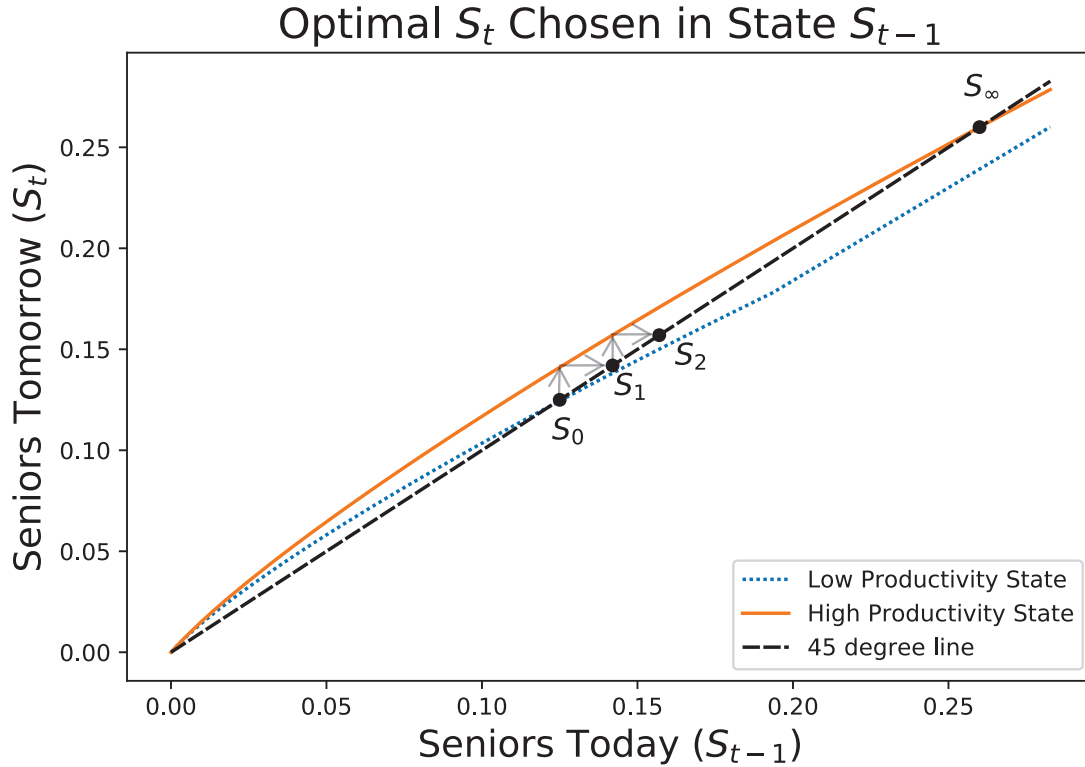
$$C_t + B_t + P_t^k INTAN_t = (1 + r_{t-1})B_{t-1} + R_t^k K_{t-1} + W_t L_t + DIV_t \quad (16)$$

$$K_t = (1 - \delta)K_{t-1} + \left(1 - \frac{\phi}{2} \left(\frac{INTAN_t}{INTAN_{t-1}} - 1 \right)^2 \right) INTAN_t \quad (17)$$

where $\phi = 0$ in the congestion model with heterogeneous firms. The household earns income from supplying capital K_t and labor L_t to firms, and also potentially from dividends paid by investment-goods producing firms, DIV_t .

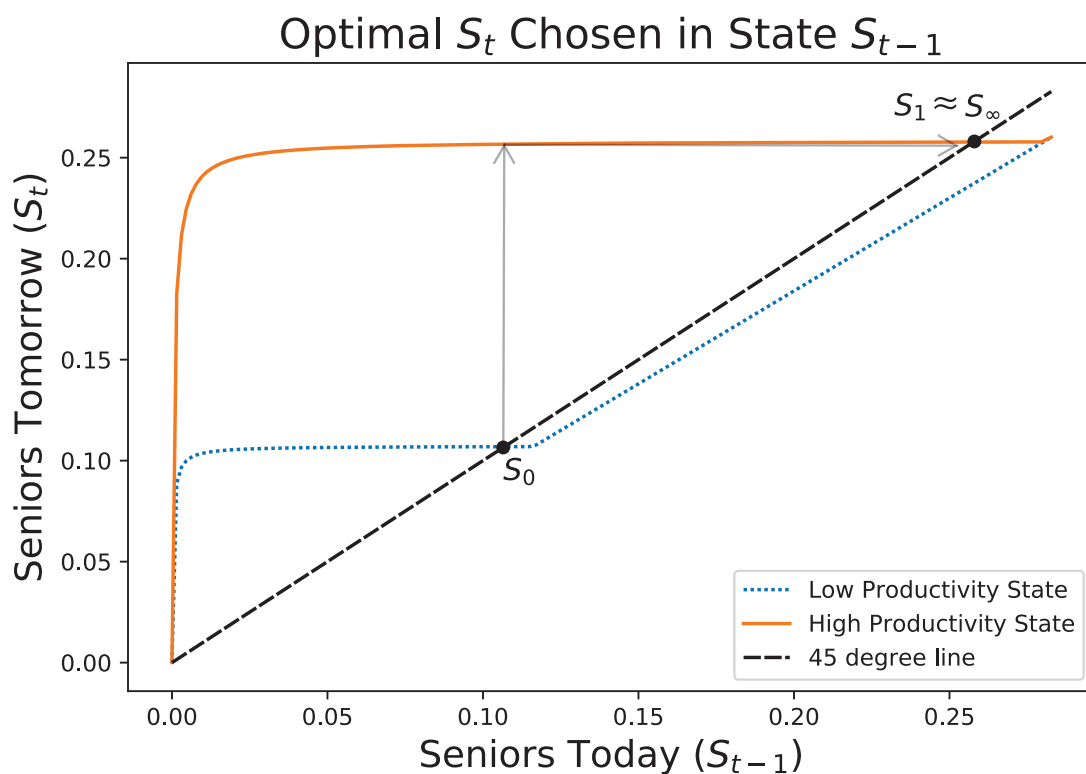
As in the main text, a perfectly competitive, representative final consumption good firm

Figure 24: **With Congestion ($\rho(x)$ Decreasing):** Firm Hires Slowly in Response to Positive Shocks



Notes: The firm's optimal choice of senior workers tomorrow, S_t , given seniors today, S_{t-1} . There are two lines because this choice depends on workers' productivity, which can be low or high. The figure illustrates slow adjustment for a firm with S_0 senior workers transitioning from the low productivity to the high productivity state in period $t = 1$. Grey arrows trace out the firm's choices at $t = 1$ and then $t = 2$ assuming it remains in the high state. Note that these choices S_1 and S_2 remain far below the long-run value S_∞ . The firm also slowly adjusts to negative shocks by “hoarding labor” in case it transitions back to the good state. Adjustment is slow because the firm implements a hiring freeze ($J_t = 0$) and lets exogenous separations slowly reduce the stock of senior workers S_t .

Figure 25: **Without Congestion ($\rho(x)$ Flat):** Firm Immediately Adjusts to Positive Shocks



Notes: This figure repeats the exercise in Figure 24 for the case where $\rho(x)$ is nearly constant, i.e. $\rho'(x) \approx 0$. This is identical to a fixed cost hiring model, yielding slow adjustment to negative shocks from labor hoarding (not shown), but rapid adjustment to positive shocks (grey arrows).

produces Cobb-Douglas,

$$C_t = Z_t K_{t-1}^\alpha J_{c,t}^{1-\alpha}$$

where $J_{c,t}$ denotes labor hired in the consumption goods sector, which we denote with a “ J ” to signify that workers are immediately productive in that sector, and Z_t is an exogenous total factor productivity (TFP) shock. This yields the following standard factor demands for capital and labor:

$$\begin{aligned} R_t^k &= \alpha Z_t \left(\frac{K_{t-1}}{J_{c,t}} \right)^{\alpha-1} \\ W_t &= (1 - \alpha) Z_t \left(\frac{K_{t-1}}{J_{c,t}} \right)^\alpha. \end{aligned}$$

A continuum of investment goods firms indexed by $i \in [0, 1]$ produce intangible investment. Their problem is described below, and involves hiring juniors $j_t(i)$ and seniors $s_t(i)$ to produce intangible investment. We thus define aggregate labor used in the intangible sector as

$$J_t + S_{t-1} \equiv \int_0^1 (j_t(i) + s_{t-1}(i)) di,$$

so that aggregate labor demand from all firms is given by

$$L_t = J_{c,t} + J_t + S_{t-1}.$$

Regarding wages, we continue to make the simplifying assumption that all workers receive the same wage W_t . To introduce wage stickiness and consider the role of monetary policy shocks, we also continue to assume the same wage Phillips curve and monetary policy rule as in Section 4. The two models we compare here in this section differ only in their production of investment goods and in the presence of adjustment costs, ϕ .

Model 1: Representative Firm with I-dot Adjustment Costs ($\phi > 0$)

The first model assumes simply that all investment firms i are identical and that $INTAN_t = Z_t S_{t-1}^\nu$ where S_t is chosen freely each period. J_t is zero here always, so aggregate labor demand is simply $L_t = S_{t-1} + J_{c,t}$. To get hump-shaped impulse response functions, this model needs I-dot adjustment costs with $\phi > 0$. This model serves as a benchmark for the congestion model, described next. It is identical to the benchmark model used in Section 4.

Model 2: Heterogeneous Firms with Congestion in Onboarding

Intangible investment goods firms are owned by households (or, equivalently, a representative venture capital firm that maximizes household utility) and maximize the expected present value of current and future dividends. These firms solve the same optimization problem described in Section G.1, but with new notation since we now consider a continuum of firms $i \in [0, 1]$ optimizing given time-varying prices. These firms choose individual stocks of senior workers $s_t(i)$ and junior workers $j_t(i)$ which aggregate up to total J_t and S_t in the intangible investment sector. They face a common price for their output, P_t^k , but productivity shocks are now both firm specific, $e_t(i)$, and aggregate, Z_t . The idiosyncratic, firm-specific shocks will imply that the onboarding constraint and non-negativity constraint on J_t will bind for some firms but not others in the stochastic steady state that we linearize around.

Each firm $i \in [0, 1]$ takes the price of intangible capital P_t^k , wages W_t , aggregate productivity Z_t , and interest rates r_t as given. Production of aggregate investment is $INTAN_t \equiv \int_0^1 Z_t e(i) s_{t-1}(i)^\nu di$, where idiosyncratic productivity $e_t(i)$ takes on discrete values and follows a Markov process calibrated to match a persistent AR(1) process. A firm with idiosyncratic productivity $e_t(i)$ and incumbent, senior workers $s_{t-1}(i)$ has the following value function:

$$V_t(e_t(i), s_{t-1}(i)) = \max_{j_t(i), s_t(i)} \left\{ P_t^k Z_t e_t(i) s_{t-1}(i)^\nu - W_t (s_{t-1}(i) + j_t(i)) + \frac{E_t[V_{t+1}(e_{t+1}(i), s_t(i))]}{1 + r_t} \right\}$$

where workers separate at rate d and new hires j_t become specialized at endogenous rate ρ :

$$\begin{aligned} s_t(i) &\leq (1 - d)s_{t-1}(i) + \rho \left(\frac{j_t(i)}{s_{t-1}(i)} \right) j_t(i) \\ j_t(i) &\geq 0. \end{aligned}$$

Finally, in this model we set $\phi = 0$ to eliminate adjustment costs in the household budget constraint.

G.3 Calibration and Results

For our quarterly calibration, we use the same as in Section 4: specifically, we choose the same values for ρ and for all parameters listed in Table 1.⁵⁷

In Model 2 with idiosyncratic risk and congestion in onboarding, we assume e_t follows a nine-state Markov process calibrated to match a persistent AR(1) process.⁵⁸ Recall for ρ we

⁵⁷Using slightly higher values for δ and d do not materially change the results.

⁵⁸ e_t follows a nine-state Markov process calibrated to discretize $X_t = .95X_{t-1} + \gamma_t$ with $\gamma \sim \mathcal{N}(0, .025)$.

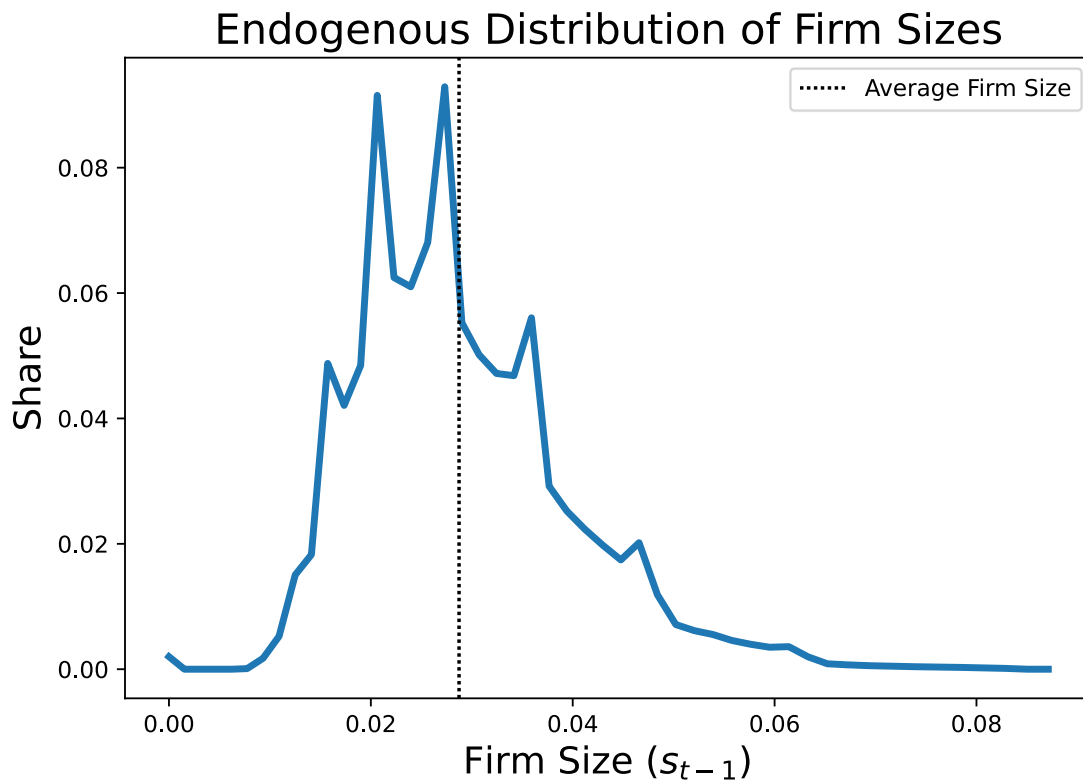
use a linear form as described above in Section 3 calibrated to $\rho = .47 - .06 \left(\frac{j_t}{s_{t-1}} \right)$, as in Section 4.

We solve the model in sequence space to first order around this steady state with idiosyncratic risk given an exogenous path for a particular shock, either monetary policy shocks, ϵ_t or for TFP, Z_t (Auclert et al., 2021). That steady state features an endogenous distribution of R&D firms, which Figure 26 plots. The distribution of firm sizes is right-skewed, despite the fact that idiosyncratic shocks are symmetric, because it is harder for firms to grow than to shrink: firms scale up in the face of positive shocks more slowly than they downsize in response to negative shocks.

The following figures present the congestion model's quarterly impulse responses of investment $INTAN_t$ and other aggregate variables to both a contractionary monetary policy shock ϵ_t (in Figure 27) and contractionary TFP shock (in Figure 28) calibrated to decay at rate of 10% per quarter. These responses are the red, dotted lines in Figure 8, and in each case we can see that the response of investment is hump-shaped, comparable to the standard *ad hoc* model of I-dot adjustment costs, which are shown by the blue, crossed lines in Figure 8. This shows that the results in Section 4 are not reversed in a general equilibrium setting where large idiosyncratic shocks violate the assumptions made in Proposition 1. We conclude that our congestion model provides a highly plausible explanation, or microfoundations, for the investment adjustment costs used in quantitative DSGE models to capture the dynamics of R&D and other intangible investment.

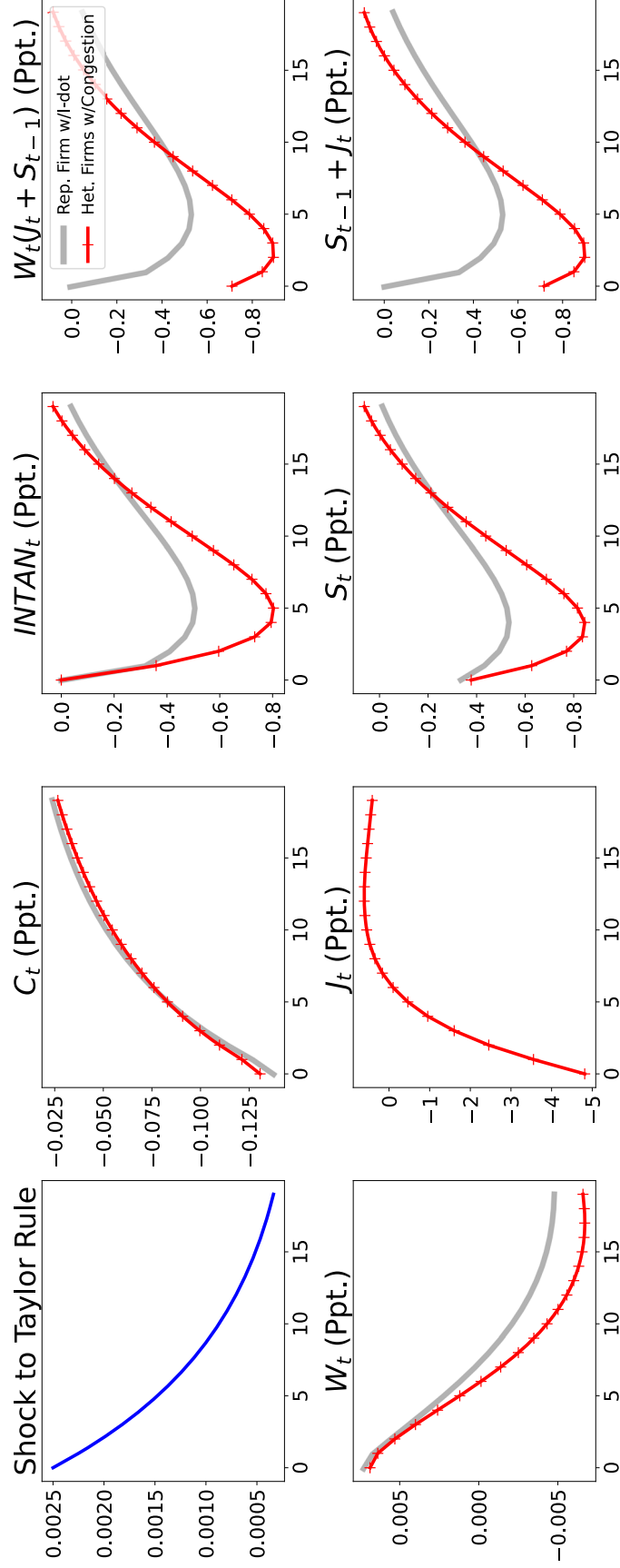
For the *two-state* calibration presented in Section G.1, this implies a high-state productivity of 5% more than in the low state. In practice, the level of idiosyncratic risk barely matters for the aggregate model's dynamics.

Figure 26: Right-Skewed Endogenous Firm Distribution When ρ Slopes Down



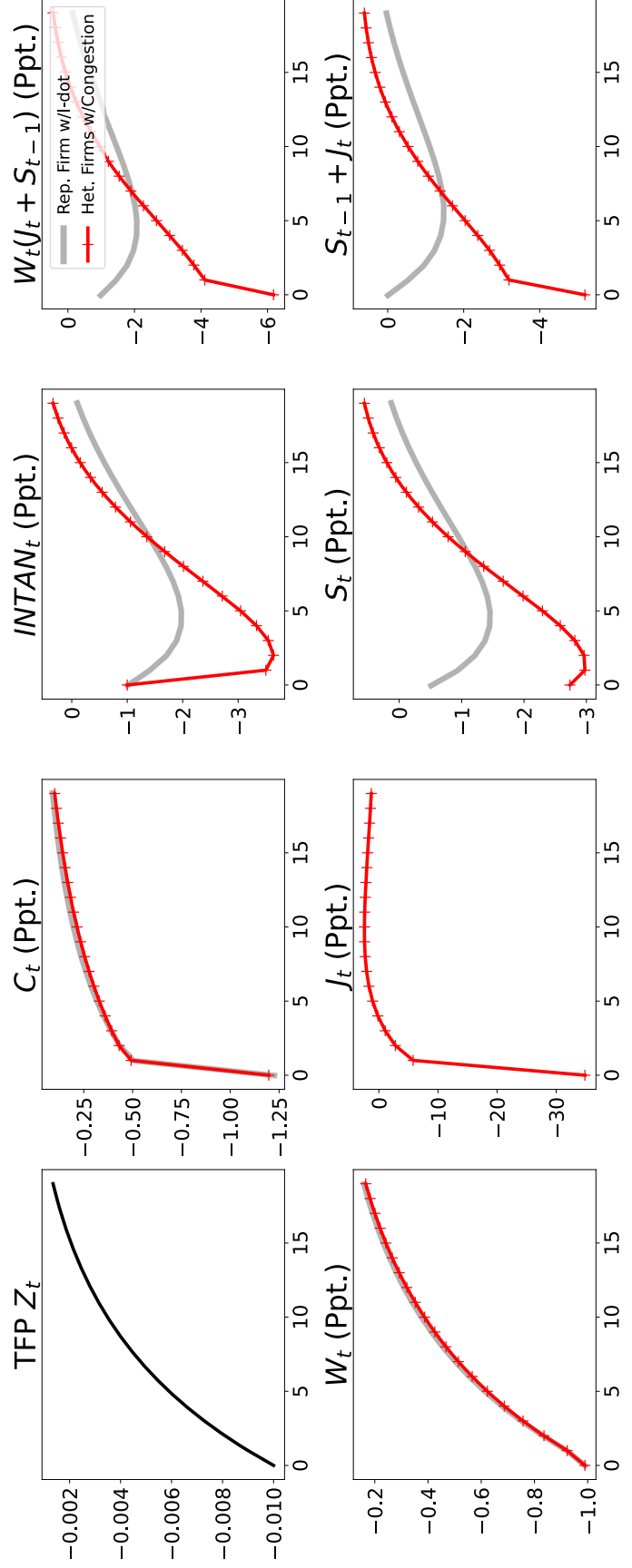
Notes: Endogenous distribution of firms in the congestion model. Firms are ex-ante identical but ex-post different in size due to idiosyncratic productivity shocks which follow a nine-state Markov process. This is calibrated to match a persistent AR(1) process, and the “spikes” in the distribution are the long-run values for firms that have been in a particular productivity state for a long time (and could be “smoothed out” by adding more states). The vertical line plots the average value of S across all firms (the steady state value of S in the model). The distribution is right-skewed because scaling up in the face of positive shocks takes a long time (due to congestion) but layoffs can happen more quickly.

Figure 27: Model Responses to a Contractionary Monetary Policy Shock



Notes: Quarterly impulse responses in the congestion model with heterogeneous firms and idiosyncratic risk, compared to a simple representative firm model with convex investment adjustment costs.

Figure 28: Model Responses to a Contractionary TFP Shock



Notes: Quarterly impulse responses in the congestion model with heterogeneous firms and idiosyncratic risk, compared to a simple representative firm model with convex investment adjustment costs.