Learning on the Job

Jacob Adenbaum Fil Babalievsky William Jungerman*

University of Edinburgh Census Bureau UNC, Chapel Hill

This version: November 1, 2024

CLICK HERE FOR MOST RECENT VERSION

Abstract

What are the sources of worker learning within the firm? How much of a worker's human capital growth comes from firm specific factors, such as the learning environment, as opposed to their own ability, and the composition of their coworkers? In this paper, we introduce a novel labor search model with multi-worker firms, learning from coworkers, and heterogeneity in learning-by-doing rates that can vary at the worker and firm level. Despite its complexity, we show that it is possible to solve such a model by leveraging recent advances from the machine learning literature. We use French administrative data to discipline the parameters of the model, specifically by targeting how wage growth varies across workers, firms, and the distribution of coworker wages. With the calibrated model in hand we perform a series of structural and statistical decompositions to test how much of the variance of human capital is driven by learning from coworkers and by heterogeneous learning-by-doing, and find that learning from coworkers is the dominant source of learning in the economy. Switching off learning from coworkers lowers human capital and wages by more than 25%, and differences in the composition of coworkers accounts for more than 50% of the variance of human capital growth rates. Finally, we use the model to calculate dynamic markdowns that price in the benefits of learning on the job and find that markdowns are less than 5% on average for workers with more than 1 year of experience at their current firm.

JEL codes: C63, E24.

Keywords: Machine learning, labor search, human capital, knowledge diffusion.

*Adenbaum: University of Edinburgh (jacobadenbaum@gmail.com), Babalievsky: Census Bureau (filipfba@gmail.com), Jungerman: University of North Carolina, Chapel Hill (wjunger@unc.edu). We thank Mike Elsby, Kyle Herkenhoff, Gregor Jarosch, Jeremy Lise, Rafael Lopes de Melo, Stan Rabinovich, Can Tian, Ludo Visschers, and numerous conference and seminar participants for helpful comments. We are very grateful to Sandeep Sarangi for help with the UNC Longleaf cluster. This work is generously supported by a public grant overseen by the French National Research Agency (ANR) as part of the "Investissements d'Avenir" program (reference: ANR-10-EQPX-17 - Centre d'accès sécurisé aux données – CASD) and by the Google Cloud Research Credits program (award GCP19980904).

Introduction

In his seminal paper, Rosen (1972a) hypothesized that a substantial amount of workers' market skills are acquired through their participation in the labor market:

> *Clearly, "education" is not produced only in schools and learning does not cease after graduation. Instead, it is economical to transfer its location to the market; for after some point work and learning are complementary, and knowledge is more efficiently acquired in conjunction with work experience rather than school.*

However, jobs differ in the sort of learning environments they offer, and the composition of one's coworkers. How does the learning of a shift-worker at a fast food restaurant compare to the junior colleague of a master electrician? Would the electrician's apprentice be willing to accept a lower wage in exchange for the opportunity to learn from such a more skilled colleague? Or consider two young paralegals: the first works at a high-powered corporate law firm, interacting with some of the smartest colleagues in the profession, but is given no time to think or study; the second works at a much smaller firm, with far more mentorship and space to take time and learn, but has far fewer interactions with the best and brightest. Workers constantly make decisions about which jobs to take, and what wages to demand. It is clear that when doing so, they pay close attention to the composition of their coworkers, the temperament of their manager, and their opportunities for growth and development.

In this paper, we seek to disentangle the sources of worker learning within the firm. We aim to distinguish between three potential mechanisms: first, some workers may be better able to learn on the job; second, working for different employers may affect the rate at which workers learn; third, the composition of a worker's coworkers may matter over-and-above the fixed characteristics of the worker and firm. The core challenge is one of identification: human capital is unobservable, and our main proxy, wages, is the outcome of a forward-looking bargaining process. Workers select which firms they wish to join, and negotiate their wages based on what they expect to gain in the future. If a worker anticipates rapid human capital growth from joining a firm, they will tend to be willing to accept a lower wage than they would otherwise.

To address this, and allow us to interpret the data, we develop a structural model of frictional job search and wage bargaining that accommodates the rich patterns of sorting that we expect to see in real world. We extend the standard Postel–Vinay and Robin (2002) model of labor search to allow for firms that differ in their productivity and learning environments, and workers who are heterogeneous in their human capital and learning ability. This enables us to embed a notion of learning-by-doing (as in Arrow 1962) at rates that vary both by firm and by worker. Moreover, as we want to distinguish between a firm's learning environment and the specific composition of potential coworkers at the firm, we must accommodate large multi-worker firms, with heterogeneous distributions of workers within the firm, and complementarities between workers in both production of the final good, and in the accumulation of human capital. Our model shares many similarities with the continuous-time limit of Herkenhoff, Lise, Menzio, and Phillips (2024), while allowing for firms to have many workers and allowing for richer heterogeneity among firms and workers, which allows us to separately disentangle firm and worker learning effects from the learning effects of coworkers.

To solve such a model would be infeasible using standard methods. When the composition of your coworkers affects your future human capital growth, you must keep track of how that distribution evolves over time. This means that for even modestly sized firms the state space that agents must track comprises dozens of states, and for larger firms it may require them to track several hundred. The curse of dimensionality (Bellman 1966) has historically made problems of this size intractable. We show how to use recent advances in machine learning to solve the problem. The key insight, due to Maliar, Maliar, and Winant (2021) and Azinovic, Gaegauf, and Scheidegger (2022), is that we can approximate the key model objects of interest with deep neural networks, which are a class of universal function approximators with *much* better scaling properties, and which can handle such high dimensional state spaces. In doing so, we contribute to a growing list of economic applications of scientific machine learning.

Moreover, we show how to solve our model while accommodating heterogeneously sized firms: as firms add and subtract workers over time, the number of distinct states they must track varies dynamically. The key insight is that the correct way to think about the firm's state is not as a fixed vector, but instead as a *set* of coworker states, which can vary over time. This set theoretic representation of the firm encodes an important feature of firms in our framework: the underlying economic structures are permutation invariant. That is, if we re-order the set of workers within the firm, it is still the same firm. We then exploit recent advances from Kahou, Fernandez-Villaverde, Perla, and Sood (2022) that provide a constructive approach to design neural network architectures that can represent such functions in a computationally efficient way.

We calibrate our model to French administrative data, which features two-year panels for every worker in France with identifiers scrambled to prevent linkages and a full panel for workers born in October. Calibration is challenging, as our model can generate many plausible sorting patterns that threaten the identification assumptions behind statistical approaches that measure the determinants of learning. For example, we allow for the possibility that initial human capital and learning ability are correlated, and we allow the production function to be either submodular or supermodular. This means that high-skilled workers with high innate learning abilities could sort into the same firm to maximize static output, which would generate a spurious correlation between coworker human capital and wage growth rates. Our calibration strategy therefore relies on indirect inference: we generate simulated data from the model and match a set of statistics that are informative about the underlying structural parameters. Most importantly, we regress wage growth on the average difference in wages between a worker and their higher-paid and lower-paid coworkers, following Jarosch, Oberfield, and Rossi-Hansberg (2021), adding firm fixed effects to this regression to capture the firm-specific component of learning. Intuitively, the coefficients discipline the effect of coworkers on learning, the variance of firm fixed effects helps identify the variance in firm learning environments, and the variance of the worker-level error terms pins down the variance of worker learning abilities.

With the calibrated model in hand, we perform a structural decomposition of the sources of learning in the economy by switching off learning from coworkers, variation in firm learning environments, and variation in worker learning ability individually and together. Switching off learning from coworkers lowers wages and human capital by more than 25%, whereas switching off the other sources of learning has more minor effects. We augment this with a statistical variance decomposition and find that differences in the extent of learning from coworkers explain more than 50% of the variance of human capital growth rates. This is overwhelmingly due to differences in learning from more-skilled coworkers.

We can also use our model to study the distribution of markdowns across workers. The correct notion of markdowns in our model is the gap between a worker's value and their contribution to the joint value of the firm, which takes into account that part of a worker's compensation is the training that they get at their current firm. We find that this markdown is less than 5% on average for workers who have been with their current firm for more than 1 year.

Related Literature. Our paper contributes to a longstanding literature that studies how modeling worker learning can affect our understanding of the economy. Classic examples include Becker (1964), Ben-Porath (1967), Rosen (1972a), Rosen (1972b), and Mincer and Jovanovic (1981). More recently, many papers have revisited these classic questions using a combination of modern computational techniques, administrative data, and structural methods. Of these papers, the closest to us in the literature are Herkenhoff, Lise, Menzio, and Phillips (2024), Hong (2022), Jarosch, Oberfield, and Rossi-Hansberg (2021), and Gregory

(2023).

Herkenhoff, Lise, Menzio, and Phillips (2024) consider a frictional labor market with learning on-the-job and learning from coworkers, and Hong (2022) extends it to accommodate heterogeneity in firm productivity. Our model shares many similarities to theirs, but because we are interested in disentangling firm from coworker effects, we must model a much richer structure of firm and worker heterogeneity in order to match the sorting patterns we see in the data. We also substantially relax their assumption that firms consist of at most two-worker teams, which helps to capture higher order learning dynamics among the workers. Jarosch, Oberfield, and Rossi-Hansberg (2021) study worker learning in large, multi-worker firms, but rely on the assumption of competitive markets in order to make the model tractable for estimation. We build on their learning function and embed it into a model of frictional markets with firm and worker learning effects, which has the added benefit of allowing us to use the model to more easily solve counterfactuals. Gregory (2023) builds on Lise and Postel-Vinay (2020) and allows for heterogeneous learning rates across firms and workers. We allow for these same effects, while also incorporating a rich structure of coworker learning effects and selection on the composition of coworkers.

Other papers in this literature have considered different mechanisms, which may be fruitful to model in future extensions of this framework. Ma, Nakab, and Vidart (2024) consider a model with both learning from coworkers and external training. Crews (2023) and Martellini (2022) study spatial models where the distribution of human capital levels within a city can matter for human capital growth. Freund (2024) extends Herkenhoff, Lise, Menzio, and Phillips (2024) to incorporate worker specialization. Lise and Postel-Vinay (2020) study a model with multidimensional skills. Pastorino (2024) considers whether learning about worker ability can drive dispersion in wages and finds that it has a significant but indirect effect operating through worker sorting. There is also a substantial empirical literature on the effects of coworkers on learning. For example, Nix (2020) finds a meaningful effect of coworker education on own wage growth.

We also contribute to the literature on search models, and in particular to multi-worker search models. Our model extends the classic work of Postel–Vinay and Robin (2002) to include multiple workers and sources of on-the-job learning. Other examples of multi-worker search models include Elsby and Michaels (2013), Kaas and Kircher (2015), Acemoglu and Hawkins (2014), Schaal (2017), and Bilal, Engbom, Mongey, and Violante (2022). To the best of our knowledge, ours is the first search model to combine firms with more than two workers with coworker learning.

Our solution method builds on a growing literature that uses neural networks to solve dynamic macroeconomic models, as in Maliar, Maliar, and Winant (2021) and Azinovic, Gaegauf, and Scheidegger (2022). We also build on the results from Kahou, Fernandez-Villaverde, Perla, and Sood (2022) for the representation of dynamically sized state spaces. Whereas Kahou, Fernandez-Villaverde, Perla, and Sood (2022) exploit the permutation invariance of Wagstaff, Fuchs, Engelcke, Posner, and Osborne (2019) to efficiently represent functions where the aggregate state contains a large number of agents (for instance, a distribution over the states of agents), we use a related result that allows us to apply the same approach to represent the states of firms that contain a distribution of agents within the firm's idiosyncratic state. When solving the model, we target the residual of the Hamilton-Jacobi-Bellman equation, similar to how Duarte, Duarte, and Silva (2023) solve continuous time finance models with neural networks. Jungerman (2023) uses reinforcement learning to solve a dynamic monopsony model where firms are large relative to the overall economy.

1 Model

In this section, we present a search model of multi-worker firms with nontrivial production and learning complementarities. We extend a standard labor search model with random search, and Postel–Vinay and Robin (2002) bargaining, by augmenting it in two key ways in order to capture nontrivial interactions across different workers within the firm. First, we adopt a more general production technology, where workers can be either substitutes or complements within the firm. Second, we add interactions between workers in their human capital accumulation process. Specifically, we adopt a similar functional form to Jarosch, Oberfield, and Rossi-Hansberg (2021), allowing for workers' learning on the job to depend in a flexible fashion on the entire distribution of human capital among their coworkers. We further augment this learning production technology to allow for fixed worker and firm heterogeneity in learning ability. This necessarily involves tracking the entire distribution of human capital among the workers as part of the state space, and therefore poses some considerable computational challenges, which we will address in detail in Section 2.

1.1 Agents

Consider a model where time is continuous, which is populated by a continuum of workers and a continuum of firms. We normalize the measure of firms to 1, and denote by $N_w \in \mathbb{R}$ the measure of workers. Workers are indexed by $i \in [0, N_w]$, and are differentiated by their *general* human capital h_i, and their time-invariant learning ability a_i. Firms are indexed by $k \in [0,1]$, and are similarly heterogeneous along two dimensions: their productivity z_k and their learning environment q_k. Both of these dimensions are immutable, and are jointly distributed according to $G_f(z_k, q_k)$. Workers and firms exit the model at a constant rate: workers retire exogenously and are replaced by new workers at a rate δ_r, and firms shut down at a rate δ_f, and are replaced with new firms. New workers are "born" unmatched with an initial draw from a distribution G_w, and new firms are "born" unmatched from any workers (i.e, with a firm size of 0) and with an initial draw from G_f[1]. A firm consists of a single manager, and n_k matched workers, who contribute to production. The employment rate is $e \in [0,1]$ and the non-employment rate is $(1-e)$. Although formally speaking, the firm and worker states are also indexed by the time t, we will omit time subscripts for clarity, except where necessary. All agents discount the future at common rate $\rho \in (0,1)$ and have linear preferences over the numeraire final good.

1.2 Technology

There are three types of technology in this economy: meeting, production, learning. We will discuss each of them in turn.

Meetings and Matches. Firms cannot produce without workers. Instead, they must match with one another in a frictional labor market. We adopt a straightforward matching technology: each worker generates matches at a rate ψ^N if unmatched or ψ^E if matched. These matches are allocated uniformly to workers, both matched and unmatched, and are allocated to firms in proportion to their size (including the manager, i.e, $n_k + 1$).[2] Defining e as the share of workers who are employed, we get that the total mass of matches is $e\psi^E + (1-e)\psi^N$, and that the share of matches generated by employed workers s^E and nonemployed workers s^N are as follows:

$$s^E = \frac{e\psi^E}{e\psi^E + (1-e)\psi^N} \qquad\qquad s^N = \frac{(1-e)\psi^N}{e\psi^E + (1-e)\psi^N}$$

When a firm meets a worker, they may agree upon a mutually acceptable wage, and form a match. Once they agree upon a wage, the firm is obliged to pay them the agreed flow wage w_i on an ongoing basis, either until the match ends, or until one of the parties can credibly

[1] In practice, we assume that both of these distributions are joint log normals, as discussed in Subsection 3.4.

[2] The assumption that managers are included in firm size when allocating matches is a technical one, since otherwise unmatched firms would never receive meetings with workers.

threaten to end the match. Firms and workers cannot dissolve the match at will. Instead, they must wait for exogenous renegotiation events to arise which occur at a rate λ, or for the worker to meet another firm.[3] A worker can match with only one firm, but a firm may match with many workers.

This is somewhat of a departure from existing matching technologies, which is necessary when modeling firms of heterogenous size if we want to allow for Gibrat's Law (Gibrat 1931). Recall that Gibrat's Law is the observation that firms grow at a rate that does not depend on their absolute size. If large and small firms received matches at a constant rate, then smaller firms would be presented with opportunities to hire a new worker at a *much* faster rate, proportional to their size, since a one worker firm would expect to meet as many workers in a given time period as a firm with hundreds of workers, which is clearly implausible. Of course, as we will see, there is no reason in our model that all firms *must* grow at the same rate, but this departure is necessary in order to allow the possibility.

The Firm State. We will need to define a bit of notation so that we can represent the production relevant states of the firm. Let us denote by $\mathcal{W}_k$ the set of all workers who are matched to firm k, and we will collect all of the worker's states together in a vector $\mathbf{x}_i := (h_i, a_i, w_i) \in \mathbb{R}^3$. We can then describe the firm's workforce as $X_k := \{\mathbf{x}_i \mid i \in \mathcal{W}_k\}$. Further, we will denote by $S_k := (z_k, q_k, X_k)$ the full state of the firm. In a slight abuse of notation, we will continue to refer to a worker's state as $\mathbf{x}_i$ even when the worker is not matched with a firm.

Production. Firms produce units of the final good by combining their productivity z_k with their workforce. Once firms are matched to workers, they produce a flow output that follows a CES production function across worker human capital levels:

$$F(\underbrace{z_k, q_k, X_k}_{S_k}) := z_k \left(\sum_{i \in \mathcal{W}_k} h_i^\eta \right)^{\frac{1}{\eta}} \tag{1}$$

where $\eta > 0$ controls the elasticity of substitution between workers. We do not further restrict η, allowing both for super-modular production functions (when $\eta > 1$) and sub-modular

[3] We assume away the possibility of continuous renegotiations between the firm and their workers for the sake of tractability. If multiple workers are able to simultaneously negotiate with the firm, it introduces a host of tricky questions around their ability to bargain collectively, or form cartels in order to extract a larger share of the surplus from the firm. While a theoretical treatment of these sorts of multilateral negotiations is interesting in its own right, a proper treatment is beyond the scope of this paper. By assuming that matches can only be dissolved at fixed intervals which arrive according to a Poisson process, we ensure that events where the firm needs to engage in multilateral negotiations with its workers occur with measure zero.

production functions (when $\eta < 1$). For a more extensive discussion of the importance of supermodularity in multi-worker production function, see Herkenhoff, Lise, Menzio, and Phillips (2024).

Unmatched workers have access to a home-production technology, which produces output proportional to their human capital bh_i.

Learning. We use the functional form from Jarosch, Oberfield, and Rossi-Hansberg (2021). Let us denote by $\mathcal{W}_i^+ := \left\{ j \in W_{k(i)} \mid h_j > h_i \right\}$ the set of worker i's coworkers with higher levels of human capital. Similarly, let $\mathcal{W}_i^- := \left\{ j \in W_{k(i)} \mid h_j < h_i \right\}$ denote worker i's less skilled coworkers.[4] When a worker is matched to a firm, their human capital evolves in the following way. At a constant arrival rate γ^E, a learning event occurs where all the workers at the firm update their human capital simultaneously.[5] The growth rate of human capital for a worker i depends on a drift term a_i that encodes the worker's learning ability, a drift term q_k, corresponding to the firm's learning environment, and terms that reflect the impact of the worker's learning from all of their coworkers, split into the different effects of more skilled coworkers and less skilled coworkers:

$$\log\left(\frac{h_i'}{h_i}\right) = a_i + q_k + \underbrace{\frac{\theta^+}{n_k - 1} \sum_{j \in \mathcal{W}_i^+} \log\left(\frac{h_j}{h_i}\right)}_{\text{Effect of More Skilled Workers}} + \underbrace{\frac{\theta^-}{n_k - 1} \sum_{j \in \mathcal{W}_i^-} \log\left(\frac{h_j}{h_i}\right)}_{\text{Effect of Less Skilled Workers}} \tag{2}$$

When we present the value functions, we will use $H(S_k)$ to denote the new state of firm k after all the workers have learned.

Unmatched workers cannot learn.

Wage Bargaining. We impose a similar bargaining structure to Postel–Vinay and Robin (2002), which entails the following four assumptions:

[4] We have technically excluded the case where a different worker at the firm has exactly the same human capital level as i, however, we will exclude this case since it occurs with probability zero. Alternatively, we could have defined

$$\mathcal{W}_i^+ := \left\{ j \in W_{k(i)} \mid h_j \geq h_i, j \neq i \right\}$$

to resolve the ambiguity.

[5] In principle, we could imagine workers updating their human capital either continuously, or one at a time. In practice, we found that updating the human capital for all workers at once yields substantial benefits in the computation of the model, which we discuss in Section 2. Since the data that we will be estimating the model against is at a yearly frequency, we argue that this assumption is relatively innocuous, relative to the computational gains.

(A1) Firms can condition their wage offers on worker states (h_i, a_i) as well as the states of the firm at which the worker is currently matched, if the firm is meeting an employed worker.

(A2) Firms can make counter-offers when a rival firm attempts to hire one of their workers.

(A3) Firms make take-it-or-leave-it wage offers.

(A4) Wage contracts can only be renegotiated by mutual consent, during the arrival of either a renegotiation shock, or when a worker meets another firm.

(A5) When considering hiring and firing decisions, firms maximize the joint value of the full coalition (the firm and all its current workers).

Of these, Assumptions (A1) to (A3) are entirely standard, and Assumption (A4) is a modification of the original bargaining protocol to rule out situations where the firm would need to bargain multilaterally with several workers at once. In previous models, this has not been a concern, because when the matches are additively separable across the firm, we can analyze workers at the match level. In our model, this is not the case because there are spillovers and complementarities between workers, both through the production function and the learning function.

As in Postel–Vinay and Robin (2002), these assumptions ensure that workers' decisions about whether to accept or reject a job offer are bilaterally efficient. Assumptions (A1) and (A2) ensure that firms enter a Bertrand auction over a worker, where the worker accepts the offer from the firm with the highest willingness to pay, and receives a value corresponding to the losing firm's willingness to pay. Assumption (A3) ensures that workers have no bargaining power (in the traditional Nash sense) when negotiating with the firm, other than their outside option at the time of the negotiation. Assumption (A4) ensures that firms cannot unilaterally rescind a wage offer that they have already negotiated after the worker has turned down a competing offer. However, frequent renegotiation shocks ensures that a worker's compensation cannot persist for long above the firm's maximum willingness to pay. For instance, a worker whose value to the firm is high due to their ability to "train" others may find themselves less valuable over time as the coworker composition of the firm changes. If some of their less skilled coworkers leave, or accumulate enough human capital that their training is less effective, the firm may find that a worker is paid more than their marginal product at the firm. In this case, the firm can credibly threaten to separate unless the worker takes a pay cut, but only at the regular renegotiation intervals.

Assumption Assumption (A5) is similar to the key assumption in Herkenhoff, Lise, Menzio, and Phillips (2024), and it means that hiring and firing decisions maximize the joint value of the firm and its workers. Hence, we can focus our attention on that joint value rather than the firm and worker values individually.

1.3 Non-employed Value

Let $U(h_i, a_i)$ be value of an non-employed worker i. They receive flow benefits bh, make contact with firms at a rate ψ^N, and retire at a rate δ_r. Let $W_0^\star(h_i, a_i, S_k)$ denote the value that a firm with state S_k will offer to an non-employed worker with state (h_i, a_i). As in Postel–Vinay and Robin (2002), since the firm makes the worker a take-it-or-leave it offer, we know that $W_0^\star(h_i, a_i, S_k) = U(h_i, a_i)$, and so the change in the worker value is zero, regardless of whether they accept or reject the wage offer. This is equivalent to workers having zero bargaining power out of non-employment. We define $\pi(S_k)$ to be the probability that a firm with state S_k receives a meeting,

$$\pi(S_k) := \frac{n(S_k) + 1}{\int (n(S_k) + 1) \mathrm{d}\chi(S_k)}$$

where $n(S_k) = n_k$ is the firm size and $\chi(S_k)$ is the distribution of firms across states. Denoting by $\Pi(S_k)$ the cdf of $\pi(S_k)$, we can now write the nonemployed worker's value recursively as

$$(\rho + \delta_r) U(h_i, a_i) = bh_i \tag{3}$$

This recursive formulation yields a closed-form expression:

$$U(h_i, a_i) = \frac{bh_i}{\rho + \delta_r} \tag{4}$$

1.4 Joint Value

Before we can proceed, it will be useful to introduce some notation to describe the change to a firm's state when they add or remove a worker from the firm. Consider the firm k with state S_k and workforce X_k. Since each worker's state $\mathbf{x}_i$ includes the worker's index i by definition, we can understand adding or removing a worker as the set union and set exclusion operators on the set X_k, respectively[6]. We will denote these by

$$\begin{aligned} S_k \oplus \mathbf{x}_i &:= (z_k, q_k, X_k \cup \{\mathbf{x}_i\}) \\ S_k \ominus \mathbf{x}_i &:= (z_k, q_k, X_k \backslash \{\mathbf{x}_i\}) \end{aligned} \tag{5}$$

Let $V(S_k)$ denote the present value of a firm with state S_k and all of its matches, including the continuation values of both the workers and the firm if the match were to terminate. The flow value of this match is just the flow output of the firm $F(S_k)$, since both the firm and the workers have linear utility, and so the wage payments exactly offset each other. A match can be destroyed exogenously if firms die, workers retire, or it receives a match-break shock. However, there are two ways that the match can break endogenously:

1. A renegotiation shock can occur, where the firm and the worker choose to separate endogenously.

2. The worker can be poached by a rival firm.

Let us consider each of these cases in turn for a worker $\mathbf{x}_i \in X_k$.

Renegotiation. At the time of a renegotiation shock, the firm and the worker may choose to adjust the worker's wage w_i, however they will choose to separate if and only if

$$\underbrace{V(S_k)}_{\text{Joint Value with match}} < \underbrace{V(S_k \ominus \mathbf{x}_i) + U(\mathbf{x}_i)}_{\text{Joint Value without match}}$$

If we define the match surplus $\Delta(S_k, \mathbf{x}_i) := V(S_k) - V(S_k \ominus \mathbf{x}_i) - U(\mathbf{x}_i)$, then we can write this condition compactly as

$$\underbrace{\Delta(S_k, \mathbf{x}_i)}_{\text{Match Surplus}} < 0$$

If this occurs, the change in the joint value is exactly $-\Delta(S_k, \mathbf{x}_i)$. So, we can write the change in the value resulting from a renegotiation shock as $\max\{-\Delta(S_k, \mathbf{x}_i), 0\}$.

Poaching. Consider the case where a worker j at firm p meets a new firm k. We will refer to firm p as the poached or incumbent firm and firm k as the poacher or poaching firm. There are three cases to consider:

– *Case 1.* If $\Delta(S_p, \mathbf{x}_j) < 0$ then the firm is unwilling to keep the worker on at any wage,

[6] Technically, the set union and exclusion operations are ill-defined in the case that we are adding or removing a worker with identical human capital, ability, and wages as another worker in the firm. However, this occurs with probability 0 as all three are continuous variables.

and will not make a counteroffer.[7] Regardless of whether $\Delta(S_k \oplus \mathbf{x}_j, \mathbf{x}_j) > 0$ or not, the worker will get $U(\mathbf{x}_j)$ since they are negotiating with firm k with an outside option of $U(\mathbf{x}_j)$, and the incumbent firm will get $V(S_p \ominus \mathbf{x}_j)$. Therefore, the change in joint value is $-\Delta(S_p, \mathbf{x}_j)$.

- *Case 2.* If $\Delta(S_p, \mathbf{x}_j) > 0$, and $\Delta(S_p, \mathbf{x}_j) > \Delta(S_k \oplus \mathbf{x}_j, \mathbf{x}_j)$, then the worker will stay at the incumbent firm and the joint value remains unchanged.

- *Case 3.* If $\Delta(S_p, \mathbf{x}_j) > 0$, and $\Delta(S_p, \mathbf{x}_j) < \Delta(S_k \oplus \mathbf{x}_j, \mathbf{x}_j)$, then the worker will leave the incumbent firm for the poaching firm. In this case, the worker moves to the new firm, being delivered a value equal to the incumbent firm's maximum willingness to pay: $V(S_p) - V(S_p \ominus \mathbf{x}_j)$, and so the change in the joint value is

$$\underbrace{V(S_p) - V(S_p \ominus \mathbf{x}_j)}_{\text{Payment to } j \text{ from } k} - \underbrace{\big(V(S_p) - V(S_p \ominus \mathbf{x}_j)\big)}_{\text{Loss of match value}} = 0$$

So, from the perspective of an incumbent match, the change in the joint value following a poaching event is $\max\{-\Delta(S_p, \mathbf{x}_j), 0\}$. Conversely, from the perspective of a poacher, the change in the joint value is[8]

$$\begin{aligned}
\mathcal{B}(S_k, S_p, \mathbf{x}_j) :&= \max\{V(S_k \oplus \mathbf{x}_j) - V(S_k) - \max\{V(S_p) - V(S_p \ominus \mathbf{x}_j), U(\mathbf{x}_j)\}, 0\} \\
&= \max\Big\{V(S_k \oplus \mathbf{x}_j) - V(S_k) - \Big(\max\{V(S_p) - V(S_p \ominus \mathbf{x}_j) - U(\mathbf{x}_j), 0\} + U(\mathbf{x}_j)\Big), 0\Big\} \\
&= \max\{V(S_k \oplus \mathbf{x}_j) - V(S_k) - U(\mathbf{x}_j) - \max\{V(S_p) - V(S_p \ominus \mathbf{x}_j) - U(\mathbf{x}_j), 0\}, 0\} \\
&= \max\{\Delta(S_k \oplus \mathbf{x}_j, \mathbf{x}_j) - \max\{\Delta(S_p, \mathbf{x}_j), 0\}, 0\}
\end{aligned}$$

We collect the results from the preceding discussion in the following proposition:

Proposition 1 (Separations)**.** *When a worker j at firm p receives a poaching event with firm $k \neq p$, the increment to the joint value is* $\max\{-\Delta(S_p, \mathbf{x}_j), 0\}$. *The change in the poaching*

[7] The possibility of this case is the reason why it is *not* sufficient to just consider the difference in the marginal products:

$$\Delta(S_p, \mathbf{x}_j) - \Delta(S_k \oplus \mathbf{x}_j, \mathbf{x}_j)$$

In principle, both firms may derive negative surplus from a worker, and neither would be willing to bid on them, although we expect that such a pathological case happens only rarely. We rule out the possibility of an EE transition in this case by assuming that each EE contact is immediately preceded by a renegotiation shock which allows the incumbent firm to separate from the worker. Otherwise, one could wind up with a perverse situation where an incumbent firm would be willing to make a side payment to the poaching firm in order to hire an unproductive worker from whom they would prefer to separate.

[8] In the second line here, we make use of the identity that $\max\{a - b, 0\} = \max\{a, b\} - b$. Also note that $\max\{x, 0\} = \mathbb{1}\{x \geq 0\} \times x$ which, in this context, can be equivalently interpreted as the product of the probability and the value of separating.

firm's value net of their payment to the worker is

$$\mathcal{B}(S_k, S_p, \mathbf{x}_j) = \max\left\{\Delta(S_k \oplus \mathbf{x}_j, \mathbf{x}_j) - \max\left\{\Delta(S_p, \mathbf{x}_j), 0\right\}, 0\right\}$$

Let ω to be the rate at which each employee of a firm (including the manager) generates meetings for the firm. We also use $\chi^N(\mathbf{x}_i)$ to denote the distribution of non-employed workers and $\chi^E(\mathbf{x}_j, S_{p(j)})$ to denote the distribution of workers across firms. With this in hand, we are now ready to write the HJB equation for the joint value of a firm and its matches:

$$\begin{aligned}
\rho V(S_k) = {} & \underbrace{F(S_k)}_{\text{Flow output}} \underbrace{- \delta_f \left(V(S_k) - \sum_{i \in \mathcal{W}_k} U(\mathbf{x}_i) \right)}_{\text{Firm Death}} + \underbrace{\sum_{i \in \mathcal{W}_k} (\lambda + \psi^E) \max\left\{-\Delta(S_k, \mathbf{x}_i), 0\right\}}_{\text{Quit Opportunities and Poaching}} \\
& + \underbrace{\gamma^E \Big[V(H(S_k)) - V(S_k) \Big]}_{\text{Learning}} + \underbrace{\sum_{i \in \mathcal{W}_k} (\delta_r + \delta_m) \Big[V(S_k \ominus \mathbf{x}_i) - V(S_k) \Big] + \delta_m U(\mathbf{x}_i)}_{\text{Match Breaks and Retirement}} \\
& + (n_k + 1)\omega \Bigg[\underbrace{s^N \int \max\left\{\Delta(S_k \oplus \mathbf{x}_j, \mathbf{x}_j), 0\right\} \mathrm{d}\chi^N(\mathbf{x}_j)}_{\text{Meet Unmatched}} \\
& + \underbrace{s^E \int \mathcal{B}(S_k, S_{p(j)}, \mathbf{x}_j) \mathrm{d}\chi^E(\mathbf{x}_j, S_{p(j)})}_{\text{Meet Matched}} \Bigg]
\end{aligned} \tag{6}$$

Note that the scaling factor $\frac{1}{N_w}$ in the last term ensures that we are computing an expected value over all possible meetings, since the set of workers has measure N_w. We also do not formally exclude a firm meeting its own workers in our notation, but merely note that since the firm has a finite number of workers, the set of all such meetings has measure zero, and therefore does not contribute to the integral.

As is standard in this class of sequential auction models (e.g. Lise and Robin 2017), our model retains the feature that when the workers have zero bargaining power, the joint value can be written entirely independent of the distribution of wages within the firm. Since utility is transferable, and negotiations are bilaterally efficient, wages serve a purely allocative role within the model, and as we have shown, we can characterize the separation decisions entirely in terms of the joint value of the firm. This is an extremely convenient feature of the model, and we will return to it in Section 2, when we discuss the computational algorithm for solving it.

1.5 Worker Value

Having characterized the separation policies in terms of the joint value V in the previous section, we are now in a position to write down the HJB equation for the value of a worker i employed at a firm k which we denote $W_i(S_k)$. Worker i receives a flow value w_i, and the set of possible events is the same as in the case of the joint value. However, for worker i, we need to track the changes in their value that are induced by wage renegotiations.

When a coworker $j \in \mathcal{W}_k \setminus \{i\}$ leaves the firm, the change in worker i's value is indirect: it only comes through future changes in learning opportunities: $W_i(S_k \ominus \mathbf{x}_j) - W_i(S_k)$. A similar expression governs the indirect change in the worker's value when the firm hires a worker j from outside the firm: $W_i(S_k \oplus \mathbf{x}_j) - W_i(S_k)$. However, we need to pay close attention to how the worker's value evolves when events arrive that directly change the value through renegotiation. Let us consider the two cases in turn.

Value of Renegotiation. According to Assumption (A4), when the worker i receives a renegotiation shock, wages can only be renegotiated by mutual consent which means that we can restrict our attention to cases where one party can credibly threaten to terminate the match. Consistent with Thomas and Worrall (1988) and Hall (2005), we impose the following assumption to deal with this case:

> (A6) When either a worker i or their employer k can credibly threaten to end the match, the wage adjusts so that $W_i(S_k)$ attains the closest boundary of the bargaining set $[U(\mathbf{x}_i), V(S_k) - V(S_k \ominus \mathbf{x}_i)]$.

Assumption (A6) ensures that the wages changes by the smallest amount necessary to satisfy the binding constraint and keep the match alive.

There are three cases we must consider, which we also summarize in Figure 1:

- *Case 1.* If the surplus $\Delta(S_k, \mathbf{x}_i)$ is negative, then both the worker and the firm are better off separating, and there is no wage payment that can make the match viable. In this case, the change in the worker value is

$$U(\mathbf{x}_i) - W_i(S_k)$$

- *Case 2.* If the surplus is positive, but the worker's participation constraint binds – that is, if $W_i(S_k) < U(\mathbf{x}_i)$ and the worker would prefer to quit to nonemployment unless given a raise – then the firm will increase the wage payment to the worker to deliver a value of $U(\mathbf{x}_i)$. Thus, the change in the worker value will again be

$$U(\mathbf{x}_i) - W_i(S_k)$$

– *Case 3.* However, if the firm's participation constraint binds, and $W_i(S_k) > V(S_k) - V(S_k \ominus \mathbf{x}_i)$ – that is, if the worker is being paid more than their marginal product at the firm – then the worker's wage will be decreased until they receive exactly their marginal product. And so the change in the worker value will be

$$V(S_k) - V(S_k \ominus \mathbf{x}_i) - W_i(S_k)$$

We can equivalently write this down as a piecewise function for the change in value following a renegotiation shock:

$$Q_i(S_k) = \begin{cases} U(\mathbf{x}_i) - W_i(S_k) & \text{if } \Delta(S_k, \mathbf{x}_i) < 0 \text{ or } W_i(S_k) < U(\mathbf{x}_i) \\ V(S_k) - V(S_k \ominus \mathbf{x}_i) - W_i(S_k) & \text{if } \Delta(S_k, \mathbf{x}_i) \geq 0 \text{ and } W_i(S_k) > V(S_k) - V(S_k \ominus \mathbf{x}_i) \\ 0 & \text{otherwise} \end{cases}$$

Value of Poaching. We now turn to characterize the firm and worker wage payments during a poaching event. Consider a worker i at incumbent firm k, who is poached by a firm p. We already know from Proposition 1 that the worker will go to the firm at which their match surplus is highest, and that they will separate to nonemployment if and only if their surplus is negative at both firms. We want to characterize the wages that the worker will be offered following the poaching event, however it will be most convenient to describe this in terms of the value that the firm delivers to the worker.[9] We will proceed in several steps:

First, since every worker contact is preceded by a renegotiation shock, we can dispense with the case where the surplus at the incumbent firm $\Delta(S_k, \mathbf{x}_i)$ is negative. In this case, the worker and incumbent firm separate, and the worker immediately bargains with the poaching firm p with an outside option of nonemployment. As in the case of nonemployed workers, since the firm can make the worker a take-it or leave-it offer (Assumption (A3)) they offer the worker a wage that delivers the value $U(\mathbf{x}_i)$. Hence, the worker receives the value $U(\mathbf{x}_i)$ regardless of whether or not they choose to form a match with the poaching firm.

Second, if the surplus at the incumbent firm is positive then, as in Postel–Vinay and Robin

[9] We will see in Section 2 that there is a one to one mapping between the value delivered and the wage.

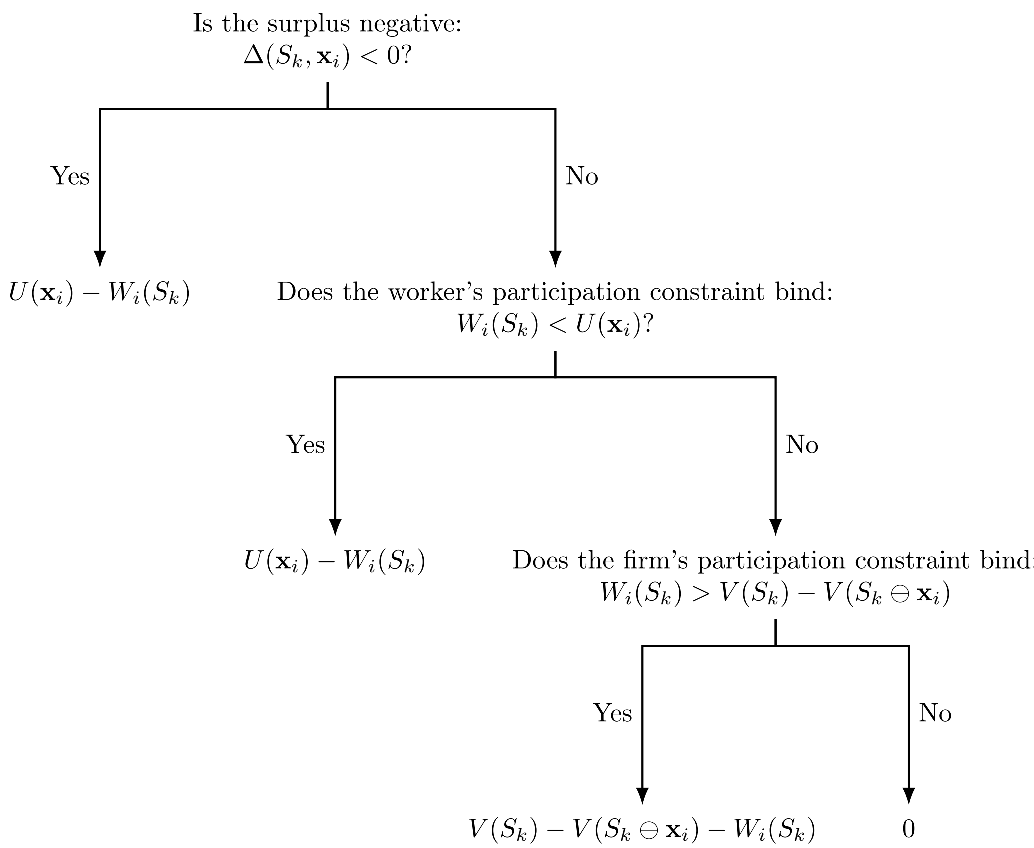


Figure 1: Possible values for $Q_i(S_k)$

(2002), the wage behavior depends on which of four cases we are in:

- *Case 1.* If $\Delta(S_p \oplus \mathbf{x}_i, \mathbf{x}_i) > \Delta(S_k, \mathbf{x}_i)$, then the surplus is higher at the poaching firm, and the worker leaves. The poaching firm must offer the worker their marginal product at the incumbent firm, and so the worker receives a value of $\Delta(S_k, \mathbf{x}_k) + U(\mathbf{x}_i)$

- *Case 2.* Otherwise, if the surplus is higher at the incumbent firm, the value the worker receives depends on their current value. The worker has an outside option that is the maximum of their value of nonemployment $U(\mathbf{x}_i)$ and their marginal product at the poaching firm $\Delta(S_p \oplus \mathbf{x}_i, \mathbf{x}_i) + U(\mathbf{x}_i)$. If the worker's value $W_i(S_k) < \max\{\Delta(S_p \oplus \mathbf{x}_i, \mathbf{x}_i), 0\} + U(\mathbf{x}_i)$, then the incumbent firm must increase their wage to deliver the outside option $\max\{\Delta(S_p \oplus \mathbf{x}_i, \mathbf{x}_i), 0\} + U(\mathbf{x}_i)$.

- *Case 3.* If the surplus is higher at the incumbent firm, but the worker's value $W_i(S_k)$ lies between their outside option $\max\{\Delta(S_p \oplus \mathbf{x}_i, \mathbf{x}_i), 0\} + U(\mathbf{x}_i)$ and their marginal product at the incumbent firm $\Delta(S_k, \mathbf{x}_i) + U(\mathbf{x}_i)$, then neither party can credibly threaten to dissolve the match, and the worker's wage remains unchanged.

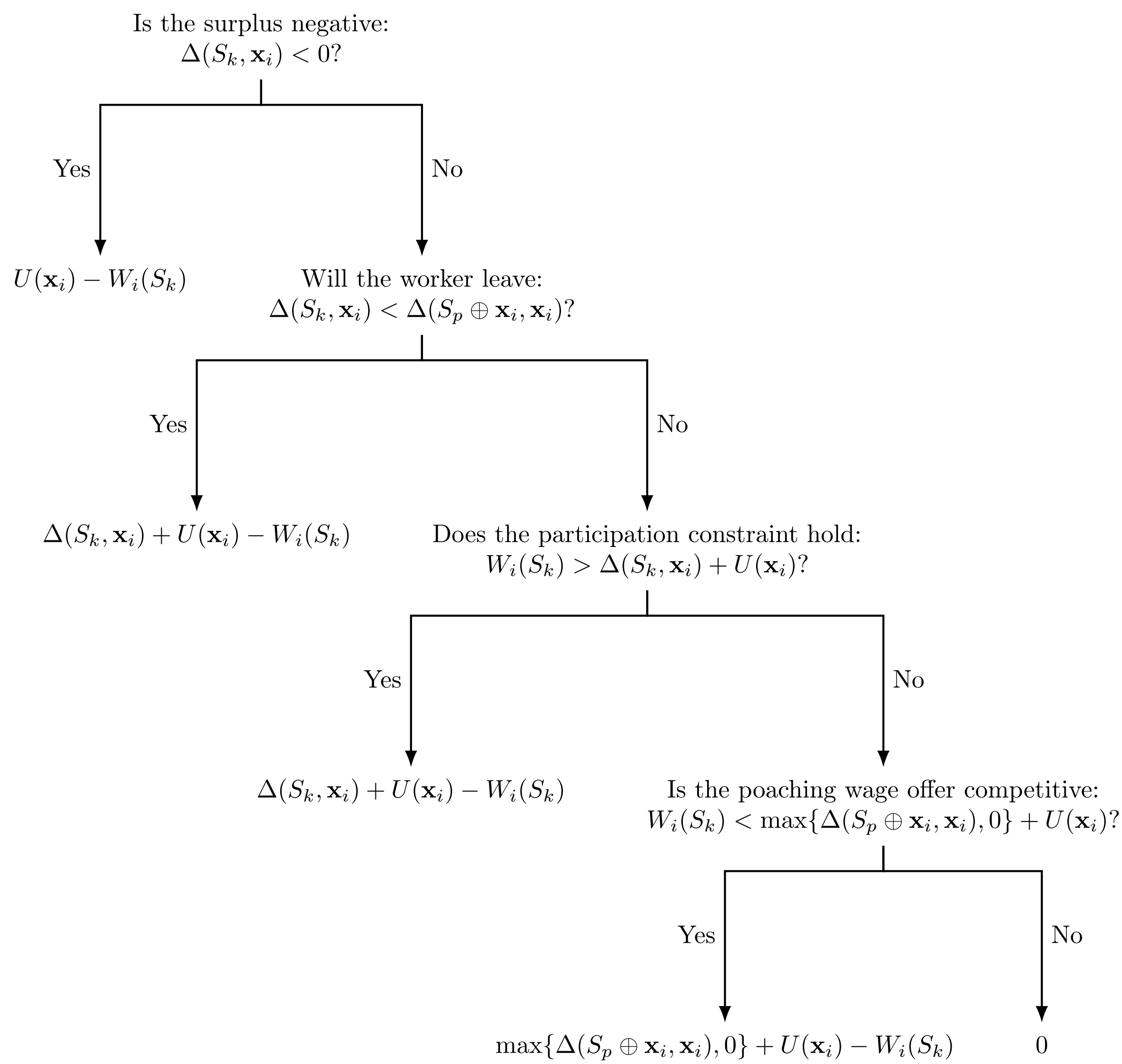


Figure 2: Possible values for $P_i(S_k, S_p)$

- *Case 4.* If the worker's value $W_i(S_k)$ is greater than their marginal product, $\Delta(S_k, \mathbf{x}_i) + U(\mathbf{x}_i)$, but the surplus is still greater at the incumbent firm than at the poacher, then the incumbent firm can renegotiate their wage back down to the worker's full marginal product. That is, the worker's wage falls until it delivers the value $\Delta(S_k, \mathbf{x}_i) + U(\mathbf{x}_i)$.

We summarize this sequence of decisions in Figure 2. In all of these cases, the worker's change in value $P_i(S_k, S_p)$ is equal to the difference between the value delivered by the new wage and their old value $W_i(S_k)$. Since each of these decisions depends on a simple condition on the relative values, let us name each of the conditions as follows:

	Condition	Description
C_1	$\Delta(S_k, \mathbf{x}_i) < 0$	Surplus is negative
C_2	$\Delta(S_k, \mathbf{x}_i) < \Delta(S_p \oplus \mathbf{x}_i, \mathbf{x}_i)$	Worker leaves for p
C_3	$W_i(S_k) > \Delta(S_k, \mathbf{x}_i) + U(\mathbf{x}_i)$	Firm participation constraint
C_4	$W_i(S_k) < U(\mathbf{x}_i)$	Worker participation constraint
C_5	$W_i(S_k) < \max\{\Delta(S_p \oplus \mathbf{x}_i, \mathbf{x}_i), 0\} + U(\mathbf{x}_i)$	Poacher offer is competitive

We can then collect the results from the preceding discussion in the following proposition:

Proposition 2 (Poaching)**.** *When a worker i at firm k receives a poaching event from firm p, Then the change in the worker i's value upon receiving a poaching offer from p is given by:*

$$P_i(S_k, S_p) = \begin{cases} U(\mathbf{x}_i) - W_i(S_k) & \text{if } C_1, \\ \Delta(S_k, \mathbf{x}_i) + U(\mathbf{x}_i) - W_i(S_k) & \text{if } \neg C_1 \text{ and } C_2, \\ \Delta(S_k, \mathbf{x}_i) + U(\mathbf{x}_i) - W_i(S_k) & \text{if } \neg C_1, \neg C_2, \text{ and } C_3, \\ \max\{\Delta(S_p \oplus \mathbf{x}_i, \mathbf{x}_i), 0\} + U(\mathbf{x}_i) - W_i(S_k) & \text{if } \neg C_1, \neg C_2, C_4, \text{ and } C_5, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Worker HJB Equation. With these in hand, we can write the full worker's HJB equation.

$$
\begin{aligned}
\rho W_i(S_k) = w_i &+ \underbrace{\gamma^E\Big(W_i(H(S_k)) - W_i(S_k)\Big)}_{\text{Learning}} + \underbrace{\delta_f\Big(U(\mathbf{x}_i) - W_i(S_k)\Big)}_{\text{Firm Death}} \\
&+ \underbrace{\sum_{j\neq i\in\mathcal{W}_k}(\delta_r+\delta_m)\Big(W_i(S_k\ominus\mathbf{x}_j) - W_i(S_k)\Big)}_{\text{Coworker Match Breaks and Retirement}} \\
&+ \underbrace{(n_k+1)\omega s^E\int\Big(\mathbb{1}\left\{\mathcal{B}(S_k, S_{p(j)}, \mathbf{x}_j) > 0\right\}\Big)\Big(W_i(S_k\oplus\mathbf{x}_j) - W_i(S_k)\Big)\mathrm{d}\chi^E(\mathbf{x}_j, S_{p(j)})}_{\text{Potential new co-worker from employment}} \\
&+ \underbrace{(n_k+1)\omega s^N\int\Big(\mathbb{1}\left\{\Delta(S_k\oplus\mathbf{x}_j, \mathbf{x}_j) > 0\right\}\Big)\Big(W_i(S_k\oplus\mathbf{x}_j) - W_i(S_k)\Big)\mathrm{d}\chi^N(\mathbf{x}_j)}_{\text{Potential new co-worker from non-employment}} \\
&+ \underbrace{\lambda\sum_{j\neq i\in\mathcal{W}_k}\Big(\mathbb{1}\left\{\Delta(S_k, \mathbf{x}_j) < 0\right\}\Big)\Big(W_i(S_k\ominus\mathbf{x}_j) - W_i(S_k)\Big)}_{\text{Coworker Quit Opportunities}} \\
&+ \underbrace{\psi^E\int\sum_{j\neq i\in\mathcal{W}_k}\mathbb{1}\left\{\mathcal{B}(S_p, S_k, \mathbf{x}_j) > 0\right\}\Big(W_i(S\ominus\mathbf{x}_j) - W_i(S_k)\Big)\mathrm{d}\Pi(S_p)}_{\text{Coworker Poacher Meetings}} \\
&+ \underbrace{\delta_m\Big(U(\mathbf{x}_i) - W_i(S_k)\Big) - \delta_r W_i(S_k)}_{\text{Own Match Breaks and Retirement}} + \underbrace{\lambda Q_i(S_k)}_{\text{Own Renegotiation Shocks}} + \underbrace{\psi^E\int P_i(S_k, S_p)\mathrm{d}\Pi(S_p)}_{\text{Own Poacher Meetings}}
\end{aligned} \tag{8}
$$

Notice that there are three distributions that workers must integrate over here:

1. The distribution of nonemployed worker states: $\chi^N(\mathbf{x}_j)$,
2. The joint distribution of firm states and worker states at that firm: $\chi^E(S_p, \mathbf{x}_j)$, and
3. The size weighted distribution of firm states: $\Pi(S_p)$

1.6 Equilibrium

Definition 1 (Equilibrium)**.** *A stationary equilibrium is a set of value functions $\{V, U\}$, distributions $\{\chi, \chi^N\}$, and a firm match rate ω such that the values solve Equations (3) and (6), conditional on the distributions, the distributions are stationary and consistent with the decisions implied by the values, and the market for matches clears:*

$$
\omega\int(1+n(S_k))\mathrm{d}\chi(S_k) = N_w(e\psi^E e + (1-e)\psi^N)
$$

Note that the distribution of employed workers across states $\chi^E(S_k, \mathbf{x}_i)$ is embedded within the distribution over firm states χ, since the worker states are included within the firm states. The employment rate is also implied by χ, because if we know the total measure of workers employed at all of the firms, and we know the total measure of workers, we can back out e as their ratio. We derive the flow equation for e as well in Appendix A.2. Note also that these distributions also imply what shares of matches come from employment (s^E) and nonemployment (s^N). As in Herkenhoff, Lise, Menzio, and Phillips (2024), our notion of an equilibrium does not make reference to wages, as every real decision in the economy is determined by the decision rules implied by the joint values. In addition, it does not make reference to the worker value W, which is only required to back out the wage, allocating the surplus between the firm and the worker.

2 Computation

In this section, we give a broad overview of the computational algorithm used to numerically solve and simulate a model with such a rich firm structure. Our goal is to numerically approximate the functions V and W, along with the ergodic distribution χ, that are consistent with our notion of equilibrium (Definition 1). As previously discussed, all of the real allocations in our model are fully characterized by V and χ. Wages are purely allocative so we only back them out, along with the worker value W, after we have found the equilibrium.

Although much of our approach is fairly standard, especially in the context of continuous time models, there are three key features which make this problem challenging. First, the state space is extremely high dimensional: a large firm needs to track the full distribution of many workers' human capital and learning abilities. As has been well known since Bellman (1966), high dimensional dynamic problems pose considerable challenges, often known as the "curse of dimensionality." Second, the firm state spaces are heterogeneously sized: a firm with thirty workers needs to track more state variables than a firm with only one. We see this in the fact that in our model, the firm state S_k contains the *set* of their worker states, X_k, whose cardinality is not known ahead of time, and, in fact, is expected to vary over the lifetime of the firm. This poses considerable challenges, since it is not obvious how to even represent such a function on the computer. Finally, in a somewhat related problem, the integrals in Equations (6) and (8) require taking expectations over draws from an extremely high dimensional ergodic distribution of potential matches. Standard numerical integration techniques would prove computationally infeasible, and so we must rely on Monte Carlo integration methods.

At a high level, we solve these challenges by making two key computational choices. First, we approximate the value functions V and W with deep neural networks. Neural networks are, at their core, an extremely convenient kind of function approximator, with several desirable features that make them well suited to approximating functions over high dimensional input spaces. They have been shown to work remarkably well at approximating the solutions to high dimensional dynamic programs (see Maliar, Maliar, and Winant 2021 for an example). As we will see, the use of neural networks also helps us with the third challenge, by making it feasible to use Monte Carlo methods to approximate the relevant integrals, since the standard methods for training these networks are robust to mean-zero i.i.d. approximation errors.

Second, we make use of the permutation invariance results from Kahou, Fernandez-Villaverde, Perla, and Sood (2022) to show how to represent a function over a heterogeneously dimensional state space such as ours. The key intuition is that within a firm, the joint value of the firm is permutation invariant with respect to the states of its workers. That is, if we were to swap the order in which we list the workers in the firm, it would still be the same firm: the order of the workers is irrelevant to the underlying economic structure of the firm. In results that are reminiscent of Krusell and Smith (1998), Kahou, Fernandez-Villaverde, Perla, and Sood (2022) show that for functions such as this, we can actually write our value functions as a function of all the fixed states, and an average of a set of "moments" that are calculated for each of the worker states. The key insight of their paper is that these function representation results tell us *exactly* what sort of neural network architecture is appropriate for the problem. In our context, we build on their work by observing that these permutation invariance results apply not only to high dimensional distributions of agents across the economy, but also work equally well for distributions of states within an individual agent, and allow us to accommodate functions where the cardinality of the set over which we are calculating these moments varies dynamically.

2.1 Neural Networks

Although much has been written about neural networks and their uses across a wide array of domains (e.g. Silver, Lever, Heess, Degris, Wierstra, and Riedmiller 2014), for our purposes it will be most helpful to abstract away from many of the details, and understand them instead as a kind of parametric universal function approximator with three key characteristics:

1. A "large enough" neural network can approximate any continuous function from $\mathbb{R}^n \to \mathbb{R}^n$ to an arbitrary degree of accuracy (Hornik, Stinchcombe, and White 1989).

2. The number of parameters needed (and therefore the computational cost of evaluating

and training the network) grows linearly in the dimensions of the input and output spaces, rather than exponentially.

3. The functions are differentiable, and can be trained with variants of gradient descent when given an appropriately defined loss function.

Although they are often treated as a very flexible nonparametric estimator, strictly speaking, they are parameterized by many parameter weights. The literature refers to the process of fitting these parameters to minimize an objective function as "training" the neural network, and we will follow this convention. For a more formal definition of neural networks and a discussion of their properties and how to train them, see Appendix C. Although a more detailed treatment of deep neural networks is beyond the scope of this paper, the interested reader should see Bishop (2006) and Goodfellow, Bengio, and Courville (2016) for classic textbook treatments, or Maliar, Maliar, and Winant (2021) and Jungerman (2023) for a treatment more geared towards economists.

2.2 Permutation Invariance

One of the more distinctive features of our model is that each firm's state S_k contains not only the idiosyncratic firm characteristics (productivity z_k and learning environment q_k), but also contains the set of all the individual state vectors of each worker within the firm X_k. Since each worker's state is characterized by the three values (h_i, a_i, w_i) this means that a firm with n_k workers has $3n_k + 2$ states. For example, a firm with $n_k = 50$ workers must account for 152 state variables, whereas a firm with $n_k = 2$ workers must only account for 8 state variables. Moreover, since the size of the firm is endogenous and evolves dynamically over time, we require a representation of the value functions V and W that can accommodate this without being extravagantly costly from a computational perspective.[10]

To solve this problem, we exploit the results on permutation invariance from Kahou, Fernandez-Villaverde, Perla, and Sood (2022). We transcribe the key result, originally due to Wagstaff, Fuchs, Engelcke, Posner, and Osborne (2019), here:

Proposition 3. *Let* $f : \mathbb{R}^{N+1} \to \mathbb{R}$ *be a continuous, permutation invariant function under*

[10] A naive approach here might be to pad the state vector of the firm with a large number of additional zeros, to accommodate the largest possible firm that could ever reasonably be encountered. Since, in principle, neural networks tend to do quite well at handling redundant states, this could work. However, such an architecture exhibits performance that scales with the worst case scenario (i.e, the largest possible firm in the economy). In many realistic calibrations, the largest firm may be one or two orders of magnitude larger than the average firm, making such an approach extraordinarily costly.

the symmetric group S_N, i.e, for all $(x, X) \in \mathbb{R}^{N+1}$ and all $\pi \in S_N$:

$$f(x, \pi X) = f(x, X)$$

Then there exists an $L \leq N$ and continuous functions $\rho : \mathbb{R}^{L+1} \to \mathbb{R}$ and $\phi : \mathbb{R} \to \mathbb{R}^L$ such that

$$f(x, X) = \rho\left(x, \frac{1}{N}\sum_{i=1}^{N}\phi(X_i)\right) \tag{9}$$

where X_i is the ith element of X.

This proposition generalizes trivially to cases where x and X_i have dimension greater than 1.[11] The intuition here is that if we can reorder the elements of the state vector, without changing the underlying value, then there must be some latent space in which we can represent the contribution of each individual state additively. We can think about the function ϕ as generating the relevant "moments" that the agents need to track, in a similar spirit to Krusell and Smith (1998). There are two main benefits relative to the existing approach: First, there is no need to hard-code the relevant set of moments. Instead, the training process recovers the set of moments automatically. Second, we can use an as many moments as like, whereas in Krusell and Smith (1998) it becomes increasingly difficult to add moments because of the curse of dimensionality.

These results provide a clear, constructive path to design architectures that can efficiently approximate value functions defined over set-valued state spaces, as we have in our model. We specify an inner neural network ϕ, which operates on each worker's state variables, and an outer neural network ρ, which takes as inputs the non-permutation invariant states x, and the average of the moments $\phi(X_i)$. We use this architecture to represent both the joint value V and the worker value W.

In doing so, we build on the results from Kahou, Fernandez-Villaverde, Perla, and Sood (2022), who demonstrate the utility of these permutation invariant architectures to approximate the solution to dynamic programs where a distribution over many agents must be tracked as part of the *aggregate state*. Here, we observe that these architectures work equally well in models where the distribution over a set of agents is part of the idiosyncratic state of the firm (and its workers). Crucially, this means that our algorithm allows us to handle firms of arbitrary size, and the costs of evaluating the value functions scale linearly with the number of workers in the firm.[12]

[11] It also generalizes to the cases where the cardinality of the input space is not fixed, but is some number less than N (see Wagstaff, Fuchs, Engelcke, Posner, and Osborne 2019, Theorem 4.4).

[12] In practice, we still do impose a cap on the total number of workers we allow at firms in our model, for

In practice, we augment the automatic moment generation process by adding several hand-picked moments computed over the distribution of worker states within the firm. We found that including the firm's flow output $F(S_k)$, the number of workers employed at the firm, as well as all the first and second moments of the worker states, averaged over the firm, helps substantially to speed convergence and improve the accuracy of our solutions. Since neural networks tend to handle redundant and co-linear states quite efficiently, this has a negligible computational overhead, relative to the gains.

2.3 Solving the Joint Value and Ergodic Distribution

To solve for the equilibrium of the model, we begin by randomly initializing the parameters of the neural network and by initializing a guess for the distribution of firms and nonemployed workers. Next we alternate between two steps: improving our approximation of the joint value function, and improving our approximation of the ergodic distribution. We go back-and-forth between these two steps for many iterations, to ensure that the two have jointly converged. In the next sections we describe each of these steps in turn.

Solving the Joint Value. We approximate the joint value V defined in Equation (6) with a neural network $\widehat{V}(\cdot;\theta_V)$ which is parameterized by a large vector of parameters θ_V. At each iteration of the training loop, commonly referred to as an *epoch*, we generate a set of training data using our current approximation of the ergodic distribution χ. We define $R_V(S_k;\theta_V^{\text{old}})$ to be the residual of Equation (6) for a particular firm with state S_k, using $\widehat{V}(\cdot,\theta_V)$ in place of V. We then define the loss function[13] to be

$$\mathcal{L}_V(\theta_V^{\text{old}}) := \int R_V(S_k;\theta_V^{\text{old}})^2 \mathrm{d}\Omega(S_k) \tag{10}$$

purely technical reasons. However, we set a relatively large value for the cap (100 workers) which ensures that it is non-binding for almost all teams at most reasonable calibrations of the model. See Table 4 for a description of typical team sizes in the economy that we calibrate to. In principle, this cap could be increased substantially without a material impact on the computation, except that it would require substantially larger and (mostly) redundant allocations of memory.

[13] In practice, in order to minimize the effect from outliers, we use the Huber loss which coincides with the Mean Squared Error for sufficiently small residual values. See Appendix C for the definition and a more detailed discussion.

where Ω is some arbitrary distribution over the firm states.[14] An update to θ_V typically looks like a standard gradient descent step:

$$\theta_V^{\text{new}} = \theta_V^{\text{old}} - \alpha_l \nabla_{\theta_V^{\text{old}}} \mathcal{L}_V(\theta_V^{\text{old}}) \tag{11}$$

where α_l is the learning rate for the gradient descent algorithm (not to be confused with the rate of learning from coworkers). The learning rate is a key hyperparameter in training neural networks. It controls the size of the step that the algorithm takes in the direction of the gradient. If the learning rate is too large, the algorithm may overshoot the minimum. If it is too small, the algorithm may get stuck in local minima and not converge to the true solution. We use the popular `NAdam` optimizer (Dozat 2016) which improves on stochastic gradient descent by incorporating Nesterov momentum into the standard `Adam` algorithm and has the convenient property of not having requiring us to exogenously set the learning rate α_l.

We cannot directly evaluate the integral in Equation (10).[15] Instead, we follow the typical approach in the machine learning literature and approximate it using Monte Carlo integration over small batches of points sampled from our approximation of Ω. This approach is called *stochastic gradient descent* (Amari 1993). In particular, we replace $\nabla_{\theta_V^{\text{old}}} \mathcal{L}_V(\theta_V^{\text{old}})$ with

$$\widehat{\nabla}_{\theta_V^{\text{old}}} \mathcal{L}_V(\theta_V^{\text{old}}) := \frac{1}{N_{\text{batch}}} \sum_{j=1}^{N_{\text{batch}}} \nabla_{\theta_V^{\text{old}}} R_V(S_k^j; \theta_V^{\text{old}})^2 \tag{12}$$

where $S_k^j \sim \Omega$. In general, gradient descent will work on the neural network parameters as long as the gradient that we feed in is unbiased. Moreover, it has been noted (Masters and Luschi 2018) that often smaller batches perform better than larger batches, especially when training neural networks. This may be because smaller batches lead to higher variance in the direction of the updates, which helps the optimizer to break out of local minima in the neural network's parameter space.

An important feature of the model from a computational perspective is the fact that wages

[14] In principle, many distributions would work as long as we achieve the desired accuracy. In practice, it is common to use the ergodic distribution χ as the measure Ω. However, we find it useful to draw from the set of all states that are reachable in a single event from the points sampled from in χ. This can be thought of as an augmented version of χ, which is supplied with synthetic data that is particularly relevant to the decisions that agents face. In particular, this helps the model by ensuring that the neural network accurately approximates the joint value even in decision relevant off-equilibrium states.

[15] We do not use the right-hand-side as training data for the left-hand-side, as one would do in value function iteration, but rather update both sides of the equation simultaneously. As in Duarte, Duarte, and Silva (2023), we find that this is more stable.

paid to the workers are not payoff relevant states from the perspective of the joint value. Consider Equation (6): wages do not appear in the flow value $F(S_k)$ since they are transfers between the firm and the workers, who all have linear utility. Moreover, our bargaining assumptions (and in particular assumption (A3) that firms make take-it-or-leave-it offers) ensure that the wage currently being offered does not affect the match's continuation value. To see why, imagine the problem of a worker being poached: because the firms Bertrand compete over them, their value following a poaching event depends only on the maximum willingness to pay of the "losing" firm. As a result, their decisions about whether to accept or reject a wage offer *does not depend on their current wage.*

This stark result is a common feature of sequential auction models where workers have no bargaining power in the Nash sense (see the discussion in Lise and Robin 2017). Although it can be relaxed, as in Cahuc, Postel-Vinay, and Robin (2006), doing so would substantially complicate the computation, as it would require us to jointly approximate both the nonemployed value and the joint value in order to recover the policy functions, and we would need to ensure that both value functions converged jointly with our approximation of χ. In particular, this result implies that the current wage of all of the workers drop out of the HJB equation for the joint value, and we can approximate it using only the non-wage states. Moreover, we can solve the joint value V independent of the worker value W and the wages. We impose this property directly through the neural network architecture by only supplying the approximation $\widehat{V}$ with the non-wage states.

Approximating the Ergodic Distribution. We need to approximate the ergodic distribution of firms across states χ and the distribution of nonemployed workers χ^N.[16] However, as is made clear by the previous discussion, we do not need the full distribution χ: we need the *marginal distribution* $\tilde{\chi}$ which integrates out over all of the non-wage states, and the distribution of nonemployed workers χ^N. In principle one could approximate this directly using the Kolmogorov Forward Equation. However, this introduces a number of complications and does not work as well with our approach to approximating the integrals.[17] Instead, we approximate this distribution with a finite sample approximation, obtained through simulation.

We initialize a fixed pool of firms of size N_f, and a pool of nonemployed workers of size N_N, and then update them by simulating forward, taking as given the current approximation of $\widehat{V}$. Proposition 1 implies the policy functions for meetings and separations. We use

[16] We can always back out the distribution of employed workers χ^E from the distribution of firm states.

[17] For instance, it is not entirely obvious how, even after recovering an approximation of the probability distribution χ, one would go about sampling from it, given that the underlying state space is set-valued.

these implied policy functions to step the distribution forward, at intervals Δt. We then treat the empirical CDF of the simulated pool of firms as our approximation for $\tilde{\chi}$, and likewise for the pool of nonemployed workers and χ^N.[18] When we must take draws from $\tilde{\chi}$, as when approximating the integrals over future meetings, we sample from our finite sample approximation with replacement.

For the purposes of simulation, we treat each individual particle as a measure zero draw from the true distribution. This means that we do not allow for interactions *between* individuals in the pool; every time one of our firms meets a new worker, they are drawn with replacement from the relevant pool. This has the substantial benefit of allowing us to simulate all of the members of the pool in parallel, but requires some care to ensure that all the relevant flows between the employed and nonemployed pools are consistent.

When working with small batch sizes for training, it can be helpful to control the relative ratios of training to simulation. We parameterize this in the algorithm with the hyperparameter N_S, and only simulate forward $\tilde{\chi}$ every N_S training epochs.

2.4 Solving for Wages

As wages are purely allocative and do not influence the policy functions, we do not solve for them until after the joint value and non-wage ergodic distribution $\tilde{\chi}$ have converged. We approximate the worker value HJB defined in Equation 8 in broadly the same way that we approximate the joint-value HJB, except that since we have already found an approximation of $\tilde{\chi}$, we do not need an approximation of the ergodic distribution to converge jointly with our approximation of W. A similar argument as with the joint value shows that the only payoff relevant wage state for a worker i is their own wage w_i. And so, relative to the solution method for V, we have one additional state.

We draw training data for all of the non-wage states from $\tilde{\chi}$. However, we must therefore take a stance on how to sample training data for each worker's wage. We do not know what the wage distribution will be ex-ante, but we can guess plausible upper and lower bounds, based on the worker's marginal product at the firm:

$$J_i(S_k) = V(S_k) - V(S_k \ominus \mathbf{x}_i)$$

This marginal product pins down the maximum wage that the firm would be willing to pay the worker. A lower bound is more tricky, since wages can be negative if workers expect high

[18] Recall that the empirical CDF is a consistent approximation of the true distribution.

human capital growth in the future and firms backload the value of the wage payments. We choose a deliberately conservative range of potential wages to sample from, since the only cost to having a range that is too broad is wasted computation (i.e, a solution that is accurate at states that are not visited in equilibrium) and sample wages for training uniformly over the wages that deliver a net present value in the interval $[-2 \cdot J_i(S_k) + U(\mathbf{x}_i), 2 \cdot J_i(S_k)]$, where the net present value is calculated using a conservative estimate of the effective discount rate, assuming that the worker never chooses to leave the match and that the wage is never renegotiated.

Since all we must do here is generate training data for the neural networks, we do not need the training data to approximate the ergodic distribution. The location of the training data determines where the neural network approximation will be the most accurate. In fact, we *do not* want to only train the neural network on wages that are observed in equilibrium: as with our choice of how to generate the synthetic training distribution Ω, we specifically want to approximate W well at wages that are *not* visited in equilibrium. This motivates our choice to sample uniformly over a deliberately wide range of wages.

Finally, once the neural network approximating W is trained, we can recover wages by inverting W in the w_i argument, to find the wage payment w_i to worker i that implements a given value. Since this is a one dimensional problem, we solve it with a brute force grid search, which ensures a robustly accurate solution. We do this only at the very end of the algorithm, when recovering an approximation of χ to match against the data, since it is quite computationally costly.

Solution Algorithm. The above discussion makes clear: we can solve this model sequentially in blocks, where each block only depends on the solution to the previous blocks (this is reminiscent of the block recursive structure in Menzio and Shi 2011). We proceed in steps: First, we jointly solve for the joint value V and the non-wage components of the ergodic distribution $\tilde{\chi}$ and χ^N. Then using the training data implied by $\tilde{\chi}$, we solve for the worker value W. Finally, we simulate and use W to back out wages, and recover an approximation of the full ergodic distribution χ. We show the high level description in Algorithm 1.

3 Data and Measurement

In this section, we describe how we calibrate the model using French administrative data. We first present the data and then go over our identification strategy.

Algorithm 1: Full Solution Algorithm

Train the Joint Value V:
for $i = 1, \ldots N_V$ **do**
  Generate the synthetic training distribution Ω from $\tilde{\chi}$
  Draw sample a batch of training data $S_k^j \sim \Omega$ for $j = 1, \ldots, N_{\text{batch}}$
  Compute the gradient $\widehat{\nabla}_{\theta_V^{\text{old}}} \mathcal{L}_V(\theta_V^{\text{old}})$
  Update θ_V according to Equation (11)
  if $\mod(i, N_S) = 0$ **then**
    Simulate forward and update the non-wage ergodic distribution $\tilde{\chi}$ and χ^N
  end
end
Train the Worker Value W:
for $i = 1, \ldots, N_W$ **do**
  Sample $S_k^j \sim \Omega$ for $j = 1, \ldots, N_{\text{batch}}$ and choose a worker $i \in \mathcal{W}_k$
  Draw wages w_i uniformly over a credible range implied by $J_i(S_k^j)$
  Compute the gradient $\widehat{\nabla}_{\theta_W^{\text{old}}} \mathcal{L}_W(\theta_W^{\text{old}})$
  Update θ_W according to Equation (11)
end
Simulate the ergodic distribution χ, backing out wages using W

3.1 French Administrative Data (DADS)

We use annual administrative data made available to researchers by the French National Statistical Institute, *Institut national de la statistique et des études économiques* (INSEE). Every year, firms in France are required to submit the *Déclarations annuelles de données sociales* (DADS), a tax form with information on the firm's employees. INSEE uses the DADS forms to produce matched employer-employee data. We use two main datasets: a repeated cross-section which we refer to as the *short panel* and a *long panel* containing the full employment history of a subset of French workers.

Short panel. The short panel is the DADS-Postes.[19] The unit of observation is a job spell, i.e. a worker-job-year tuple, and so workers with multiple jobs appear multiple times in the dataset. Worker IDs are scrambled every two years so it is not possible to follow workers for more than two periods at a time. This means it is not well-suited for longitudinal analysis, but given it contains the near-universe of workers, we use it to compute cross-sectional (or very short-run) moments such as the wage distribution across firms and labor market transition rates.

[19] "Poste" translates to "match" or "job" in French.

Long panel. The second dataset we use is the DADS-Panel, which contains the full employment history of all workers in France who are born in October. We use this dataset to infer non-employment spells which allows us to compute the employment rate and transition rates.

These two datasets contain detailed information on both workers and firms. On the worker side, these include wages, hours worked (allowing us to distinguish between the extensive and intensive margins), tenure, labor force experience, occupation codes (up to 4-digit), and standard demographics like gender and age. We do not observe education. On the firm side, we observe sector and industry codes, firm size, and employment bins along several dimensions (e.g. wages and occupation codes). The main data limitations we face is we have no direct measures of human capital, on-the-job learning, and firm productivity.

Sample Selection. In both datasets, we restrict our sample to private sector workers between the ages of 16 and 65 in mainland France.[20] By construction, the short panel only includes primary jobs so that a worker is only present once per year. We replicate as best we can the primary job selection process used by the INSEE in the long panel.[21] Finally, to compute moment targets that are growth rates or changes, we restrict the sample to stayers, i.e. workers who did not switch teams. All other unconditional moments are calculated on the full sample. In Table 1, we report summary statistics for both samples. Although in principle, subsetting the data by month of birth is not a random selection, it does not seem to introduce any systematic bias for the moments we are targeting.

[20] We also conduct standard dataset cleaning procedures such as removing workers with negative or zero wages or hours worked, fictitious establishment IDs, and missing values for occupation and demographic variables like gender.

[21] We do this by sorting the worker jobs within a (worker ID, year) pair by duration of employment (in days) first, then salary, and then number of hours worked. As a tiebreaker, if observations are identical on all these characteristics (which is exceedingly rare), we sort by the firm ID, and then choose the first one.

Table 1: Sample Summary Statistics

	Short Panel	Long Panel
Sample Size		
Workers	15.35m	17.38m
Firms	3.29m	N/A
Year	2015	2015
Hourly Wages		
Mean	18.59	17.88
p25	11.88	11.08
Median	14.98	13.98
p75	21.11	19.43

Note: This table reports summary statistics for the DADS-Postes 2015 and the DADS-Panel 2015. Both samples are constructed as described in Section 3.1. Hourly wages are in 2015 euros. Our notion of a firm is an establishment times an occupation code, and as establishment is not recorded in the long panel, we cannot distinguish firms in that dataset.

3.2 Defining a Team

Integral to our analysis is the concept of a team, by which we mean the set of coworkers that you interact and learn from. The tension in defining a team is analogous to defining a local labor market: too narrow of a boundary risks omitting coworkers that you interact with and are relevant for learning, while too broad of a boundary risks including workers that you never engage with. Since we cannot directly observe team composition, we make the conservative assumption that the set of coworkers is all the other workers at the establishment with the same 1-digit occupation code (OCC1).[22]

In Table 2, we report the 1-digit occupation codes used in the DADS data which we use to define teams, as well a representative sample of the underlying 2-digit and 4-digit occupation codes for *Executives and High-Level Professionals*. If we instead conditioned on more granular occupation codes to define our notion of a team, we would be restricting workers to not interact and learn from workers in similar occupations. For instance, it seems unreasonable to assume that dentists are not interacting with residents in dentistry or that lawyers cannot interact with legal professionals and journalists. If we were to categorize teams at the 2-digit or 4-digit level, we would exclude both of these cases.

[22] Note that we intentionally do not also condition on a worker's *echelon*, which we do observe in the data, to allow for workers to potentially learn from coworkers at different levels of the firm hierarchy.

Table 2: Occupation Coding in DADS Data

1			**Farmers**
2			**Craftsmen, Tradespeople, and Business Owners**
3			**Executives and High-Level Professionals**
	31		*Independent Professionals*
		311c	Dentists
		311d	Psychologists and Therapists
		311e	Veterinarians
		3121	Lawyers
	33		*Public Sector Executives*
	34		*Professors, Scientific Professionals*
		342b	Research Professors
		344a	Hospital Doctors Without an Independent Practice
		344c	Residents in Medicine, Dentistry and Pharmacy
		344d	Salaried Pharmacists
	35		*Careers in Media, Arts, and Entertainment*
		352a	Journalists
		352b	Literary Authors, Screenwriters
		353a	Newspaper Editors, Media Executives, Publishing Directors
	37		*Corporate Administrative and Commercial Managers*
		372e	Legal Professionals
		375a	Advertising Executives
		376a	Financial Market Executives
	38		*Engineers and Technical Managers*
		382b	Salaried Architects
		384b	Mechanical Manufacturing Engineers and Metalworking Managers
		387c	Production Process Engineers and Managers
		387d	Quality Control Engineers and Managers
4			**Intermediate Professions**
5			**Clerical Workers**
6			**Manual Laborers**
9			**Non-Coded**

Note: This table illustrates occupation coding in the DADS. There are 6 distinct 1-digit occupation codes. To illustrate the granularity of 2-digit occupation coding, we also report a selection of the 2-digit and 4-digit codes for a representative example. The full list of occupation codes can be found on INSEE's website at `https://www.insee.fr/fr/information/2497952`.

To further validate our definition, in Table 3, we present self-flow rates, which we decompose into losses from establishment switching versus occupation switching. In the model, we

Table 3: Self-Flow Rates

	Rate (%)
OCC1	89.92
Firm	83.64
Establishment	79.16
Establishment × OCC1	74.11

Note: This table reports self-flow rates, the empirical probability that a worker stays at the same group from one year to the next. Calculated in the DADS-Postes from 2014 to 2015.

can account for team switches due to establishment switching as those get counted as an EE transition. What the model can not account for is switches due to occupation switching since we have no notion of occupation choice. From 2014 to 2015, 74.11% of workers remain in the same team. The majority of the switching is due to establishment switching. Moreover, as expected, the self-flow rate for firms is higher than for establishments, with the difference capturing workers who switch establishments within the same firm. We treat workers who stay in the same firm but change establishments as having switched teams since the set of coworkers they interact with will change.[23]

In Table 4, we provide summary statistics for our team definition, both unweighted and employment-weighted, as well as a comparison to alternative definitions. The unweighted average team size in our sample is 4.66 with a 5% of teams being larger than 14 workers. However, the employment weighted mean is 152 with a long right tail. For instance, the 75th percentile is 83 and the 95th percentile is 535. To better visualize the pareto tail, in Figure 3 we plot the unweighted and employment-weighted complementary cumulative distribution functions of the firm size distribution. Finally, we compute the mean number of 1-digit occupations per establishment, which is 3, suggesting that establishments on average have a very diverse set of workers.

3.3 Model-Generated Data

Here we briefly explain how we generate simulated data from the model that matches the structure of our data. Once we have solved for V and W, and have recovered an approxi-

[23] Our analysis predates the Covid-19 pandemic which saw a large shift to remote or hybrid work which could in principle imply that workers interact with people in other establishments. This could still be the case in 2015, but we do not observe whether a worker works remotely or not in the data so we abstract away from this.

Table 4: Team Definition Summary Statistics

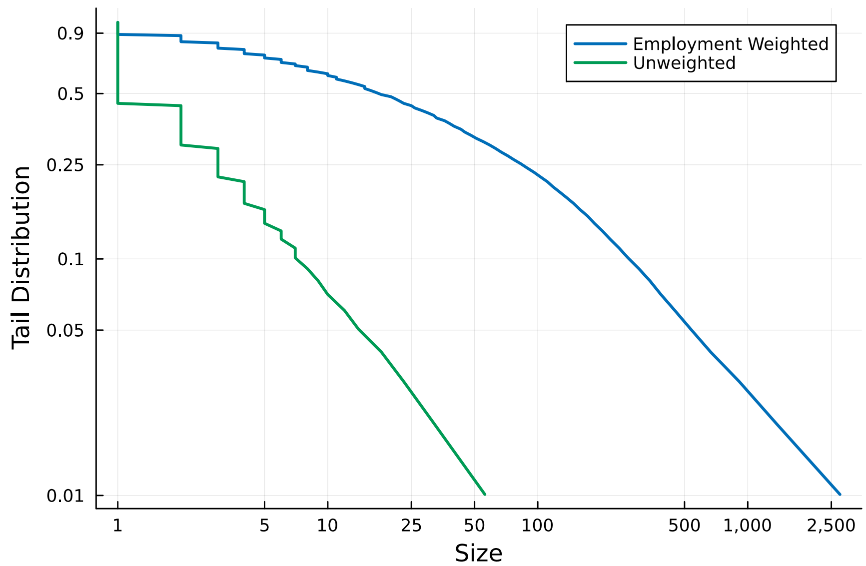
	Mean	**SD**	**p25**	**Median**	**p75**	**p95**
Unweighted						
Firm Size	8.30	202.06	1.00	1.00	4.00	20.00
Establishment Size	7.15	43.96	1.00	1.00	4.00	24.00
Establishment × OCC1 Size	4.66	26.18	1.00	1.00	3.00	14.00
Employment-weighted						
Firm Size	4,925.21	17,113.05	12.00	96.00	930.00	34,005.00
Establishment Size	277.45	1,088.03	8.00	37.00	162.00	986.00
Establishment × OCC1 Size	151.72	646.95	4.00	17.00	83.00	535.00
OCC1 per Establishment	3.06	1.26	2.00	3.00	4.00	5.00

Note: Summary statistics for our definition of a firm, the intersection of an establishment code and a 1-digit occupation code (OCC1), as well as alternative definitions, both unweighted and employment-weighted. Computed in the 2015 DADS-Postes.

mation of the ergodic distribution χ, we can simulate panels that look exactly like what we observe in the data. This is particularly important because our estimation strategy relies heavily on indirect inference: we include the same selection mechanisms in the model as we expect to see in the real world, and rely on the model to disentangle the relevant effects. This requires ensuring a careful correspondence between the model and the data.

Short Panel. The short panel is the most straightforward to compute, given our approximation of χ. We sample a pool of firm states from χ with replacement, and simulate them forward for 2 years. If a firm dies within that period, it is replaced with a newborn firm that is not matched with any workers and whose states are sampled from G_f. We record *snapshots* of the states of the firm at the end of the first and second year. In the data, we do not observe flow wages, but rather we observe annual earnings and hours worked, which we use to calculate the hourly wage. Therefore, when simulating data from the model, we integrate wages over the course of the year, between snapshots, in order to capture the average hourly wage implied by the model. Otherwise, the wages would be biased upwards, because workers tend to receive outside offers during the course of the year, and so their end of year flow wages are typically higher than their average wage.

Because we treat each firm as a measure zero particle when simulating, it means that a worker who joins the firm during the year does not have a history of their previous wage. Rather, we only see their flow wage at the moment when they are hired. Additionally, this means that we do not see a worker who leaves the firm during the course of the simulation.



Note: This figure plots the complementary cumulative distribution function (tail distribution), defined as the share of firms (establishment × 1-digit occupation) with more than x employees, for both the unweighted and employment-weighted firm size distributions. Computed in the 2015 DADS-Postes.

Figure 3: Pareto Tails of the Firm Size Distribution

Hence, in both the model and the data, we restrict all of the moments involving changes in wages to be calculated only with workers who remain at the firm over the two snapshots.

Long Panel. To simulate a panel of workers, we must separately simulate a pool of workers, and track them through spells of non-employment. We draw a pool of workers with replacement from χ and χ^N, in proportion to the employment rate e, and simulate them forward. In contrast to the approach we take for the short panel, when a worker leaves the firm, we cease to track their coworkers on an ongoing basis. This mirrors the structure of the long panel in our data, where we observe worker histories, but we do not observe the histories of their coworkers. As with firms, we replace them with unmatched newborn workers, whose states are drawn from G_w when they retire.

3.4 Parameterization

Workers draw their initial human capital h_i^0 and their permanent learning ability a_i from a joint log normal distribution $G_w(h_i^0, a_i)$:

$$\begin{pmatrix} \log h_i^0 \\ \log a_i \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} \mu_h \\ \mu_a \end{pmatrix}, \begin{pmatrix} \sigma_h^2 & \sigma_{ha}^2 \\ \sigma_{ha}^2 & \sigma_a^2 \end{pmatrix}\right]$$

Table 5: Externally-Calibrated Parameters

	Description	Value	Explanation
δ_r	Worker retirement rate	0.05	40 year career
δ_h	Human capital depreciation rate	1.0	Match data frequency
λ	Renegotiation shock arrival rate	1.0	Match data frequency
γ^E	Learning event arrival rate	1.0	Match data frequency
ρ	Annual discounting rate	0.05	Standard
μ_h	Mean log initial human capital	0.0	Normalization
μ_z	Mean log firm productivity	0.0	Normalization
μ_a	Mean log worker learning ability	0.0	Normalization

Note: This table reports the externally-calibrated parameters and their source.

We also assume a joint log normal process $G_f(z_k, q_k)$:

$$\begin{pmatrix} \log z_k \\ \log q_k \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} \mu_z \\ \mu_q \end{pmatrix}, \begin{pmatrix} \sigma_z^2 & \sigma_{zq}^2 \\ \sigma_{zq}^2 & \sigma_q^2 \end{pmatrix}\right]$$

3.5 Calibration Strategy

Exogenously Set. We set several parameters exogenously. We set the worker retirement rate δ_r to .025 to match an average 40 year career length, and the employer death rate to .05 to get an average 20-year lifespan. As our data is annual, we set the human capital depreciation rate δ_h, renegotiation rate λ, and learning arrival rate γ^E to 1. Firm productivity and worker human capital are multiplicative, hence the mean log firm productivity μ_z and mean log initial worker human capital μ_h are not separable from each other, or from the level of prices in the economy. We therefore normalize both to 0 and re-scale model-generated wages to match mean wages in the data[24]. Likewise, the mean log worker learning ability μ_a and mean log firm learning environment μ_q are not separable, so we normalize μ_a to 0. Finally, we set the discount factor ρ to a standard value of 0.05. We summarize these externally-calibrated parameters in Table 5.

We calibrate the model to match a set of informative statistics in the data. There is no direct mapping between parameters and empirical targets, but some parameters more directly affect certain targets. Hence, we will list our targets together with the most relevant parameter. Unless otherwise specified, all targets come from the two-year panel. Also, we use wage

[24] Some of our calibration targets are not scale-invariant, meaning that we need to get the level of wages right.

ranks rather than raw or logged wages unless otherwise specified. Our model can generate negative wages, ruling out the use of logged wages, and raw wages are more susceptible to outliers.

The measure of workers per employer, N_w, is pinned down by the average number of workers per employer. We cannot read N_w directly off the data as some employers may have zero workers, but the average number of workers in employers with at least one employee constrains this parameter. The firm death rate δ_f and the rate at which employed workers receive meetings ψ^E control how much time firms have to grow and how quickly they can poach workers, hence they control the firm size distribution. We therefore use the median and 90th percentiles of the firm size distribution to pin these parameters down. We cannot directly measure δ_f for the same reason that we cannot read N_w off the data: some firms may have zero employees in some years, meaning we do not know how old they are or when they die. The EN rate helps discipline the exogenous match break rate δ_m. The elasticity of substitution across worker human capital levels, η, affects incentives for workers to sort together with coworkers of similar skill levels. Hence, we set η to target the between-firm wage variance, similar to the strategy in Herkenhoff, Lise, Menzio, and Phillips (2024).

The mean firm learning environment μ_q helps match the mean wage growth rate. The variance of firm productivity σ_z helps match the correlation of firm size and mean wage rank within the firm. To gain additional information on σ_z, we target the mean EE wage rank change in the October panel. The variance of mean wage rank changes within a firm helps pin down σ_q. The covariance of firm mean wage rank and firm mean wage rank growth rates helps discipline σ_{zq}. To pin down both the initial variance of human capital σ_h and the value of nonemployment b, we target the P75-P50 and P50-P25 raw wage gaps. The impact of the initial variance of human capital on the dispersion of wages is clear, but the value of nonemployment b has a more subtle effect: it affects worker bargaining power out of nonemployment, pushes up wages for currently-nonemployed workers bargaining with firms, and constrains the lower end of the worker wage distribution. The variances of wage rank changes disciplines σ_a, and the covariance of these rank changes with wage levels helps pin down σ_{qa}. The nonemployed contact rate ψ_N controls the employment level. As we do not observe workers unless they are working, we calculate a proxy for the employment level in both our real and simulated data by calculating the share of years between the first and last time when we see an employee where the employee has a job. Bigot and Goux (2003), a report by INSEE, find labor force participation rate of 69% for the first trimester of 2003 using the French Labor Force Survey which is sampled weekly. This is in line with our results.

We cannot directly identify the parameters of the learning function $(\theta^+, \theta^-,)$ as we do not observe human capital. Furthermore, the two-year panel only lets us observe a worker's coworkers as of $t-1$ if those coworkers are still employed in t[25]. Therefore, we use an auxiliary model that replaces human capital in Equation 2 with wages. We define $\mathbb{W}^+_{i,t} := \left\{j \in W_{k(i)} \mid w_{j,t-1} > w_{j,t-1}\right\}$ as the set of worker i's coworkers who had higher wages as of $t-1$. Similarly, let $\mathbb{W}^-_{i,t} := \left\{j \in W_{k(i)} \mid w_{j,t-1} < w_{j,t-1}\right\}$ denote the set of worker i's coworkers with lower wages.[26] Note that we only include workers and coworkers who stayed at the same firm in both $t-1$ and t in this regression. We also include firm fixed effects $\alpha_{k(i)}$. We continue to use wage ranks rather than log wages as our model can generate negative wages.

$$
\begin{aligned}
w_{i,t} - w_{i,t-1} = \alpha_{k(i)} &+ \underbrace{\tilde{\theta}^+_1 \sum_{j \in \mathbb{W}^+_{i,t}} \frac{w_{j,t-1} - w_{i,t-1}}{n_{k(i)} - 1}}_{\text{Higher-Wage Coworkers}} + \underbrace{\tilde{\theta}^-_1 \sum_{j \in \mathbb{W}^-_{i,t}} \frac{w_{j,t-1} - w_{i,t-1}}{n_{k(i)} - 1}}_{\text{Lower-Wage Coworkers}} \\
&\underbrace{+ \tilde{\theta}^+_2 \sum_{j \in \mathbb{W}^+_{i,t}} \frac{(w_{j,t-1} - w_{i,t-1})^2}{n_{k(i)} - 1} + \tilde{\theta}^-_2 \sum_{j \in \mathbb{W}^-_{i,t}} \frac{(w_{j,t-1} - w_{i,t-1})^2}{n_{k(i)} - 1}}_{\text{Nonlinear Effects}} + \epsilon_{i,t}
\end{aligned}
\tag{13}
$$

Note that $n_{k(i)}$ refers to the size of firm $k(i)$ at time $t-1$, not at time t. The coefficients $\tilde{\theta}^+_1, \tilde{\theta}^+_2, \tilde{\theta}^-_1, \tilde{\theta}^-_2$ are informative about the underlying structural parameters θ^+, θ^- but do not directly identify them. We add a quadratic term because our human capital accumulation process is log-additive, but we cannot run a regression on log wages in model-generated data as there is a possibility that some wages will be negative. The quadratic terms therefore help us match the nonlinearity of the learning function. The variance of the firm fixed effects $\alpha_{k(i)}$ and variance of error terms $\epsilon_{i,t}$ also help identify σ_q and σ_a, respectively.

3.6 Calibration Results

We present the results of the calibration in Table 7. Our model does a reasonable job of matching the empirical targets despite having fewer parameters than targets. As in Herkenhoff, Lise, Menzio, and Phillips (2024), we find that the impact of higher-skilled coworkers is stronger than the impact of lower-skilled coworkers.

[25] Our model-generated firm simulations imposes an even tighter constraint and only allow us to observe coworkers who stay at the same firm at $t-1$ and t. We do not treat our sample of firms as being the entire economy but rather as a measure zero draw from the ergodic distribution. Therefore, when a worker leaves a firm in our simulation, there is no probability that they end up at one of the other firms in our sample.

[26] Note that, as with our definitions of $\mathcal{W}^+_i$ and $\mathcal{W}^-_i$ in Section 1.2, we exclude the case where a coworker has exactly the same wage since the contribution in our specification would be 0.

Table 6: Estimates of Equation (13)

	Change in Hourly Wage			
	(1)	(2)	(3)	(4)
$\tilde{\theta}_1^+$	0.391***	0.340***	0.332***	0.157***
	(0.001)	(0.003)	(0.003)	(0.004)
$\tilde{\theta}_2^+$		0.001***		0.065***
		(0.000)		(0.002)
$\tilde{\theta}_1^-$	−0.021***	0.003	−0.001	0.061***
	(0.001)	(0.003)	(0.002)	(0.004)
$\tilde{\theta}_2^-$		0.000***		−0.003
		(0.000)		(0.004)
Quadratic	*N*	*Y*	*N*	*Y*
Log or Rank	Rank	Rank	Log	Log
R^2	0.364	0.364	0.283	0.292
RMSE	7.287	7.286	0.156	0.155
Variance of Firm FE	34.640	34.409	0.011	0.010
N	7,954,029	7,954,029	7,954,029	7,954,029

Note: This table reports the results of estimating Equation (13) on the 2015 DADS-Postes. We report robust standard errors only. We report both linear and quadratic specifications for log wages and wage ranks for completeness, but we only target the quadratic rank specification (Column 2). The difference in the total number of workers is due to our sample restriction: we only include workers who have stayed at the same firm for more than a year. See the discussion in Section 3.3.

3.7 Solution Accuracy

We find that our solution method accurately approximates the joint value V and the worker value W. We check for correctness along two key dimensions: convergence of the HJB residuals to zero, and stationarity of our approximation of the ergodic distribution $\tilde{\chi}$. In Figure 4, we report the HJB residuals for V and W along the solution path. We compute all of the errors in relative terms (i.e, as a percentage of the relevant value $V(S_k)$ or $W_i(S_k)$) so that they are interpretable. We find that our solution method reliably solves the joint and worker values, with the L^2 errors dropping below 10^{-4} within 10,000 training epochs. This corresponds to a relative error of less than 1% of the value, which is quite accurate for problems of this size. The joint value V achieves L^2 errors of 10^{-6}, which correspond to relative errors of no more than 0.1% of the value.

We also report, in Figure 5, a set of simulated moments from our approximation of $\tilde{\chi}$ along the training path. We find that the moments generally settle down by the end of training, indicating that the ergodic distribution has converged.

Table 7: Internally-Calibrated Parameters

	Description	Value	Target	Data	Model
Short panel					
N_w	Workers per firm	5.371	Average employer size (unweighted)	4.660	5.190
b	Nonemployment flow value	0.141	p50 - p25 Wages	3.090	6.780
η	Production elasticity	0.939	Between-firm wage variance share (rank)	0.843	0.463
μ_q	Average Learning Environment	-0.016	Mean wage rank change	1.819	3.737
σ_z	Firm productivity variance	0.342	Correlation firm size vs. wage rank	0.038	0.164
σ_q	Firm learning environment variance	0.013	Variance of firm mean wage rank change	54.773	70.098
			Variance of $\alpha_{k(i)}$ in Equation 13	34.409	67.962
σ_{zq}	Firm learning-productivity covariance	0.013	Firm mean wage level-growth covariance	0.131	0.091
σ_h	Initial worker human capital variance	0.157	p75 - p50 Wages	6.165	3.262
σ_a	Worker learning ability variance	0.008	Wage rank change variance	73.354	174.555
			Variance of $\epsilon_{i,t}$ in Equation 13	7.286	9.428
σ_{ha}	Worker learning-initial productivity covariance	6.505e-04	Worker wage level-growth covariance	0.109	-0.005
θ^+	Learning from higher-ability coworkers	0.165	$\tilde{\theta}_1^+$ in Equation 13	0.340	0.384
			$\tilde{\theta}_2^+$ in Equation 13	0.001	-0.003
θ^-	Learning from lower-ability coworkers	0.034	$\tilde{\theta}_1^-$ in Equation 13	0.003	-0.002
			$\tilde{\theta}_2^-$ in Equation 13	0.000	-0.002
δ_f	Employer death rate	0.001	P50 employer size	1	5
ψ^E	Employed Contact Rate	1.048	P90 employer size	8	7
Long panel					
δ_m	Match break rate	0.127	EN rate	0.149	0.070
ψ^N	Nonemployed Contact Rate	0.314	NE Rate	0.311	0.353
	Inferred employment rate	0.733	0.823		

Note: This table reports the internally-calibrated parameters and compares the relevant model-generated empirical targets with those in the data. Unconditional moments are computed before the sample is restricted to stayers.

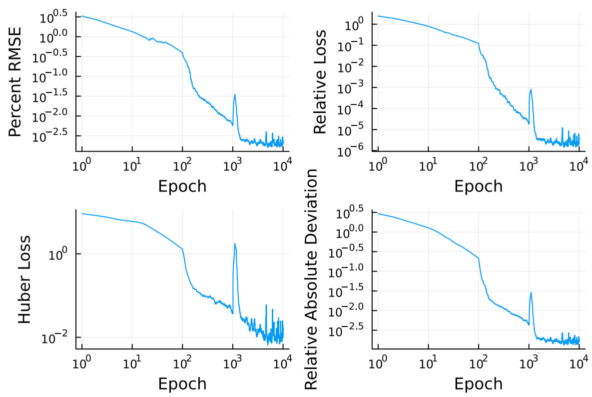
4 Results

4.1 The Determinants of On-the-Job Learning

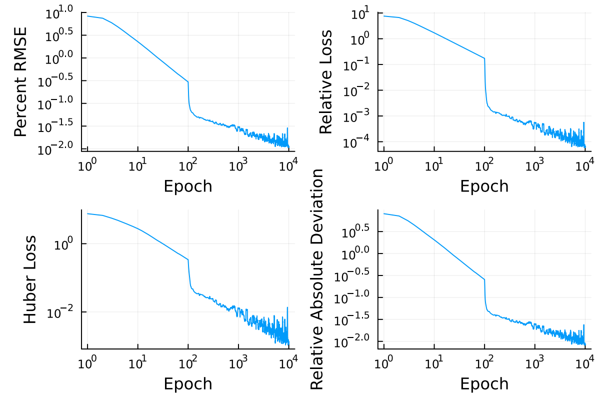
In this section, we unpack the key drivers of on-the-job learning with both a structural and statistical decomposition. We describe each of these in turn.

Structural Decomposition. The key parameters driving on-the-job learning are the mean learning-by-doing rate μ_q[27], the worker-level standard deviation of learning-by-doing σ_a, the firm-level standard deviation σ_q, and the parameters governing learning from coworkers θ^+ and θ^-. In order to quantify the contribution of each of these elements, we re-solve the model after setting them to zero, either one-at-a-time or in combination. We then calculate the mean and variance of human capital and wages in each counterfactual economy and report the results in Table 8. Learning from coworkers is the most important source of learning: without it, wages and human capital would be more than 25 percent lower on average.

[27] Recall that μ_q and μ_a are not separable, hence we normalize the latter to 0.



(a) Joint value V



(b) Worker value W

Note: For both the joint value V and the worker value W we report four measures of training errors. The top left panel shows the relative RMSE (interpretable as percentage error). The top right panel shows the relative L^2 errors. The bottom right panel shows the mean absolute errors (similar interpretation to RMSE, but less sensitive to outliers). The bottom left panel reports the raw Huber loss.

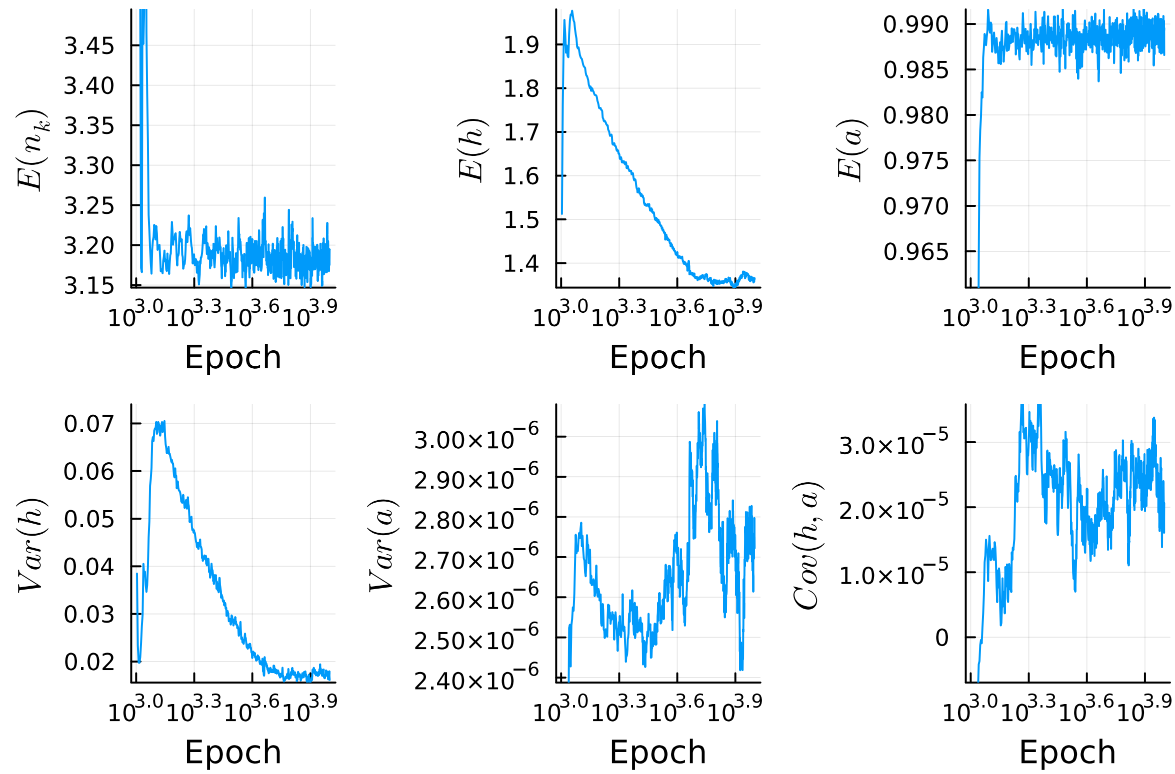
Figure 4: Training Errors along Solution Path

Table 8: Structural Decomposition of Learning

		Mean h	Var h	Mean w	Var w
Individual	σ_q	1.050	0.646	1.093	1.615
	σ_a	1.009	0.700	0.956	1.112
	(θ^-, θ^+)	0.686	5.582	0.722	0.698
Cumulative	σ_q	1.050	0.646	1.093	1.615
	σ_a, σ_q	1.004	0.518	0.924	0.900
	$\sigma_a, \sigma_q, (\theta^-, \theta^+)$	0.858	2.564	0.861	0.904

Note: This table reports the results of the structural decomposition of the drivers of learning. The first column indicates which parameters are set to zero. Note that any related covariance terms are also set to zero, e.g. σ_{ha} is set to zero whenever σ_a is set to zero. The other columns report certain key statistics from the counterfactual equilibrium, all normalized so that their value in the initial equilibrium is 1.0.

Statistical Decomposition. Next, we perform a statistical decomposition of the sources of variation in the learning rate. Taking the variance of Equation (2), we arrive at Equation (14).



Note: Variances and covariances are computed within firm, and then averaged across all firms.

Figure 5: Simulated moments from $\tilde{\chi}$ along the training path

$$
\begin{aligned}
\mathrm{Var}\left(\log\left(\frac{h_i'}{h_i}\right)\right) &= \mathrm{Var}(a_i) + \mathrm{Var}(q_k) + 2\mathrm{Cov}(a_i, q_k) \\
&+ \left(\frac{\theta^+}{n_k - 1}\right)^2 \mathrm{Var}\left(\sum_{j \in \mathcal{W}_{i,k}^+} \log\left(\frac{h_j}{h_i}\right)\right) + \left(\frac{\theta^-}{n_k - 1}\right)^2 \mathrm{Var}\left(\sum_{j \in \mathcal{W}_{i,k}^-} \log\left(\frac{h_j}{h_i}\right)\right) \\
&+ \frac{2\theta^+}{n_k - 1}\left(\mathrm{Cov}\left(a_i, \sum_{j \in \mathcal{W}_{i,k}^+} \log\left(\frac{h_j}{h_i}\right)\right) + \mathrm{Cov}\left(q_k, \sum_{j \in \mathcal{W}_{i,k}^+} \log\left(\frac{h_j}{h_i}\right)\right)\right) \\
&+ \frac{2\theta^-}{n_k - 1}\left(\mathrm{Cov}\left(a_i, \sum_{j \in \mathcal{W}_{i,k}^-} \log\left(\frac{h_j}{h_i}\right)\right) + \mathrm{Cov}\left(q_k, \sum_{j \in \mathcal{W}_{i,k}^-} \log\left(\frac{h_j}{h_i}\right)\right)\right) \\
&+ \frac{2\theta^+\theta^-}{(n_k - 1)^2}\mathrm{Cov}\left(\sum_{j \in \mathcal{W}_{i,k}^+} \log\left(\frac{h_j}{h_i}\right), \sum_{j \in \mathcal{W}_{i,k}^-} \log\left(\frac{h_j}{h_i}\right)\right)
\end{aligned}
\tag{14}
$$

Table 9: Variance-Covariance Matrix for the Statistical Decomposition of Learning

	a_i	q_i	$\frac{\theta^+}{n_k-1}\sum_{j\in\mathcal{W}_i^+}\log\left(\frac{h_j}{h_i}\right)$	$\frac{\theta^-}{n_k-1}\sum_{j\in\mathcal{W}_i^-}\log\left(\frac{h_j}{h_i}\right)$
a_i	0.157	0.002	-0.113	-0.030
q_i		0.355	-0.011	0.002
$\frac{\theta^+}{n_k-1}\sum_{j\in\mathcal{W}_i^+}\log\left(\frac{h_j}{h_i}\right)$			0.532	0.073
$\frac{\theta^-}{n_k-1}\sum_{j\in\mathcal{W}_i^-}\log\left(\frac{h_j}{h_i}\right)$				0.033

Note: This table reports the share of the variance of learning rates attributable to each of the terms in Equation 14. The diagonal terms correspond to the variance of each component, and the off-diagonal terms correspond to the covariances. The terms are normalized so that they all sum to 1.0. Note that we report the upper triangular half only to not over-clutter the table.

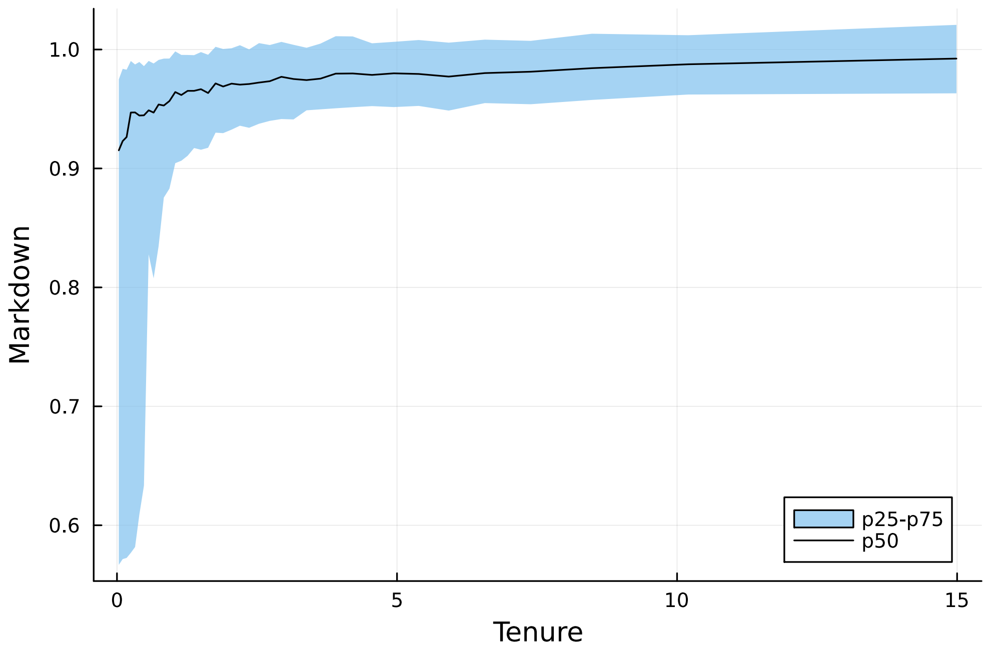
In Table 9, we report the share of the variance of learning rates attributable to each element of Equation (14). We find that learning from more-skilled coworkers is the dominant source of variation in learning rates, accounting for more than half the variance by itself.

4.2 Monopsony and On-the-Job Learning

As our model features frictional search, workers will not be paid their full contribution to joint value. In this section, we quantify the markdown between their contribution and their compensation, i.e. the gap between the worker's value $W_i(S_k)$ and their contribution to the joint value $J_i(S_k) \equiv V(S_k) - V(S_k \ominus x_i)$. The distribution of markdowns over a worker's tenure is of particular interest in a model featuring coworker learning. Low-skilled workers may be willing to take a very large markdown initially to benefit from learning on-the-job, a sort of "internship". This suggests that markdowns should fall over time. Conversely, firms may be willing to pay more-skilled workers a high wage to "train" their coworkers, but they may be less and less willing to do so as less-skilled coworkers catch up. Hence, the relationship between worker tenure and markdowns is ambiguous. We therefore plot the average markdowns by worker tenure categories in Figure 6. We find that markdowns fall over time, and that they decline to less than 5 percent for workers who have been at their current job for more than 1 year.

Conclusion

In this paper, we have introduced a novel model of frictional search that is well-suited to understanding the drivers of human capital growth. The model features multi-worker firms, learning from coworkers, and learning-by-doing rates that vary by worker and by firm. We



Notes: This figure plots dynamic markdowns against the worker's tenure at their current firm.

Figure 6: Markdowns by Tenure

have solved the model by using neural networks to approximate the HJB equation that characterizes the joint value of a firm and all its workers. Using French administrative data, we have disciplined the structural parameters of the model using indirect inference. We used the calibrated model to decompose the sources of learning in the economy and found that learning from coworkers is the dominant driver of human capital growth. Without it, human capital and wages would both fall by more than 25 percent on average. A variance decomposition of learning rates revealed that differences in the rate of learning from more-skilled coworkers account for half the variance of learning rates across workers. We then used the model to calculate dynamic markdowns (the worker's value over their contribution to joint value) and found that they are small and decline to less than 5 percent on average for workers who have been at their job for more than 1 year.

References

Acemoglu, D. and W. B. Hawkins (2014): "Search with multi-worker firms," *Theoretical Economics*, 9, 583–628.

Amari, S.-i. (1993): "Backpropagation and stochastic gradient descent method," *Neurocomputing*, 5, 185–196.

Arrow, K. J. (1962): "The Economic Implications of Learning by Doing," *The Review of Economic Studies*, 29, 155–173.

Azinovic, M., L. Gaegauf, and S. Scheidegger (2022): "Deep Equilibrium Nets," *International Economic Review*, 63, 1471–1525.

Becker, G. S. (1964): *Human Capital: A Theoretical and Empirical Analysis with Special Reference to Education, First Edition*, NBER Books, National Bureau of Economic Research, Inc.

Bellman, R. (1966): "Dynamic programming," *Science*, 153, 34–37.

Ben-Porath, Y. (1967): "The Production of Human Capital and the Life Cycle of Earnings," *Journal of Political Economy*, 75, 352–365.

Bigot, J.-F. and D. Goux (2003): "Enquête sur l'emploi du premier trimestre 2003," Tech. rep., Institut National de la Statistique et des Études Économiques (INSEE), iNSEE report.

Bilal, A., N. Engbom, S. Mongey, and G. L. Violante (2022): "Firm and worker dynamics in a frictional labor market," *Econometrica*, 90, 1425–1462.

Bishop, C. M. (2006): *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Berlin, Heidelberg: Springer-Verlag.

Cahuc, P., F. Postel-Vinay, and J.-M. Robin (2006): "Wage Bargaining with On-the-Job Search: Theory and Evidence," *Econometrica*, 74, 323–364.

Crews, L. (2023): "A Dynamic Spatial Knowledge Economy," Working Paper.

Dozat, T. (2016): "Incorporating Nesterov Momentum into Adam," in *Proceedings of the 4th International Conference on Learning Representations*, 1–4.

Duarte, V., D. Duarte, and D. H. Silva (2023): "Machine Learning for Continuous-Time Finance," Working Paper.

Elsby, M. W. L. and R. Michaels (2013): "Marginal Jobs, Heterogeneous Firms, and Unemployment Flows," *American Economic Journal: Macroeconomics*, 5, 1–48.

Freund, L. (2024): "Superstar Teams," Working Paper.

Gibrat, R. (1931): "Les inégalités économiques," *Sirey*.

Goodfellow, I., Y. Bengio, and A. Courville (2016): *Deep Learning*, MIT Press, http://www.deeplearningbook.org.

Gregory, V. (2023): "Firms as Learning Environments: Implications for Earnings Dynamics and Job Search," Working Paper.

Hall, R. E. (2005): "Employment Fluctuations with Equilibrium Wage Stickiness," *American Economic Review*, 95, 50–65.

Herkenhoff, K., J. Lise, G. Menzio, and G. M. Phillips (2024): "Production and learning in teams," *Econometrica*, 92, 467–504.

Hong, L. (2022): "Coworker Sorting, Learning, and Inequality," Working Paper.

Hornik, K., M. Stinchcombe, and H. White (1989): "Multilayer feedforward networks are universal approximators," *Neural networks*, 2, 359–366.

Jarosch, G., E. Oberfield, and E. Rossi-Hansberg (2021): "Learning From Coworkers," *Econometrica*, 89, 647–676.

Jungerman, W. (2023): "Dynamic Monopsony and Human Capital," Working Paper.

Kaas, L. and P. Kircher (2015): "Efficient Firm Dynamics in a Frictional Labor Market," *American Economic Review*, 105, 3030–60.

Kahou, M. E., J. Fernández-Villaverde, S. Gómez-Cardona, J. Perla, and J. Rosa (2022): "Spooky Boundaries at a Distance: Exploring Transversality and Stability with Deep Learning," Working Paper.

Kahou, M. E., J. Fernandez-Villaverde, J. Perla, and A. Sood (2022): "Exploiting Symmetry in High-Dimensional Dynamic Programming," Tech. rep.

Kingma, D. and J. Ba (2015): "Adam: A Method for Stochastic Optimization," in *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA.

Krusell, P. and A. A. Smith, Jr (1998): "Income and wealth heterogeneity in the macroeconomy," *Journal of political Economy*, 106, 867–896.

Lise, J. and F. Postel-Vinay (2020): "Multidimensional Skills, Sorting, and Human Capital Accumulation," *American Economic Review*, 110, 2328–76.

Lise, J. and J.-M. Robin (2017): "The Macrodynamics of Sorting between Workers and Firms," *American Economic Review*, 107, 1104–35.

Ma, X., A. Nakab, and D. Vidart (2024): "How do Workers Learn? Theory and Evidence on the Roots of Lifecycle Human Capital Accumulation," Working Paper.

Maliar, L., S. Maliar, and P. Winant (2021): "Deep Learning for Solving Dynamic Economic Models." *Journal of Monetary Economics*, 122, 76–101.

Martellini, P. (2022): "Local labor markets and aggregate productivity," *Manuscript, UW Madison.[470].*

Masters, D. and C. Luschi (2018): "Revisiting Small Batch Training for Deep Neural Networks," *CoRR*, abs/1804.07612.

Menzio, G. and S. Shi (2011): "Efficient Search on the Job and the Business Cycle," *Journal of Political Economy*, 119, 468–510.

Mincer, J. and B. Jovanovic (1981): "Labor Mobility and Wages," in *Studies in Labor Markets*, National Bureau of Economic Research, Inc, NBER Chapters, 21–64.

Nix, E. (2020): "Learning Spillovers in the Firm," Working Paper.

Pastorino, E. (2024): "Careers in Firms: The Role of Learning about Ability and Human Capital Acquisition," *Journal of Political Economy*, 132, 1994–2073.

Postel–Vinay, F. and J. Robin (2002): "Equilibrium Wage Dispersion with Worker and Employer Heterogeneity," *Econometrica*, 70, 2295–2350.

Rosen, S. (1972a): "Learning and Experience in the Labor Market," *The Journal of Human Resources*, 7, 326–342.

——— (1972b): "Learning by Experience as Joint Production," *The Quarterly Journal of Economics*, 86, 366–382.

Schaal, E. (2017): "Uncertainty and Unemployment," *Econometrica*, 85, 1675–1721.

Silver, D., G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller (2014): "Deterministic Policy Gradient Algorithms," in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, JMLR.org, ICML'14.

Thomas, J. and T. Worrall (1988): "Self-Enforcing Wage Contracts," *The Review of Economic Studies*, 55, 541–554.

Wagstaff, E., F. Fuchs, M. Engelcke, I. Posner, and M. A. Osborne (2019): "On the Limitations of Representing Functions on Sets," in *Proceedings of the 36th International Conference on Machine Learning*, ed. by K. Chaudhuri and R. Salakhutdinov, PMLR, vol. 97 of *Proceedings of Machine Learning Research*, 6487–6494.

Appendix

Table of Contents

A Model Appendix

A.1 Model Notation

Table A.1: Model Notation

Symbol	Description
N_w	measure of workers
$i \in [0, N_w]$	worker index
$k \in [0,1]$	firm index
(h_i, a_i)	*general* human capital and learning ability, with $(h_i^0, a_i) \sim G_w$
$k \in [0,1]$	firm index
$(z_k, q_k) \sim G_f$	immutable states of firm k: productivity and learning environment
δ_r	rate at which workers exogenously retire
δ_f	rate at which firms exogenously shut down
n_k	size of firm k
$e \in [0,1]$	employment rate
$\rho \in (0,1)$	discount rate
ψ^E	match arrival rate for employed workers
ψ^N	match arrival rate for non-employed workers
w_i	wage of worker i
λ	rate at which a renegotiation event occurs
$\mathcal{W}_k$	set of workers employed by firm k
$\mathbf{x}_i := (h_i, a_i, w_i)$	worker i's states, with $\mathbf{x}_i \in \mathbb{R}^3$
$X_k := \{\mathbf{x}_i \mid i \in \mathcal{W}_k\}$	set of states of firm k's workforce
$S_k := (z_k, q_k, X_k)$	state of firm k
$F(S_k)$	output of firm k with state S_k
$\sigma > 0$	elasticity of substitution between workers
b	home production
$\mathcal{W}_i^+ := \left\{j \in W_{k(i)} \mid h_j > h_i\right\}$	workers at $k(i)$ with higher levels of human capital than worker i
$\mathcal{W}_i^- := \left\{j \in W_{k(i)} \mid h_j < h_i\right\}$	workers at $k(i)$ with lower levels of human capital than worker i
γ^E	arrival rate of a learning event for employed workers
$H(S_k)$	new state of firm k with state S_k after all workers have learned
γ^N	arrival rate of a learning event for non-employed workers
$U(\mathbf{x}_i) = U(h_i, a_i)$	value of a non-employed worker with state (h_i, a_i)
$W_0^\star(h_i, a_i, S_k)$	value k with state S_k offers to an non-employed worker with state (h_i, a_i)
$\pi(S_k)$	probability firm with state S_k receives a meeting, with associated cdf $\Pi(S_k)$
$\Pi(S_k)$	cdf of $\pi(S_k)$
$S_k \oplus \mathbf{x}_i := (z_k, q_k, X_k \cup \{\mathbf{x}_i\})$	new state of firm k with state S_k after adding worker i with state $\mathbf{x}_i$
$S_k \ominus \mathbf{x}_i := (z_k, q_k, X_k \backslash \{\mathbf{x}_i\})$	new state of firm k with state S_k after removing worker i with state $\mathbf{x}_i$
$V(S_k)$	value of a firm with state S_k and all of its matches
$\Delta(S_k, \mathbf{x}_i) := V(S_k) - V(S_k \ominus \mathbf{x}_i) - U(\mathbf{x}_i)$	surplus of a match between worker i and firm k
$\mathcal{B}(S_k, S_p, \mathbf{x}_j)$	poaching firm k's payment to worker j at poached firm p
ω	rate at which each employee of a firm generates meetings
$\chi^N(\mathbf{x}_i)$	distribution of non-employed workers
$\chi^E(\mathbf{x}_j, S_{p(j)})$	distribution of employed workers across firms
$W_i(S_k)$	value of a worker i employed at a firm k
$Q_i(S_k)$	change in value following a renegotiation shock
$P_i(S_k, S_p)$	change in value to i employed at k of being poached by p

A.2 Flow Equations

To pin down the evolution of the employment rate and the rate at which firms receive matches, we must consider the flow equations both for employment e and for the flow of matches m. Changes in the employment rate can come from four sources:

1. Exogenous separations due to match break shocks, or firm death,
2. Exogenous separations due to retirement, which result in newborn workers replacing the employed workers in a state of nonemployment,
3. Endogenous separations due to renegotiation events, and
4. Endogenous separations due to poaching events.

This means that at any given time, net flows into employment can be written as

$$\begin{aligned}\dot{e} = &\underbrace{(1-e)\psi^N \int\int \mathbb{1}\left\{\Delta(S_p \oplus \mathbf{x}_i, \mathbf{x}_i) > 0\right\} \mathrm{d}\Pi(S_p) \mathrm{d}\chi^N(\mathbf{x}_i)}_{\text{Matches from non-employment}} \\ &- \underbrace{e\lambda \int \mathbb{1}\left\{\Delta(S_k, \mathbf{x}_i) < 0\right\} \mathrm{d}\chi^E(S_k, \mathbf{x}_i)}_{\text{Renegotiation separations}} \\ &- \underbrace{e\psi^E \int\int \mathbb{1}\left\{\Delta(S_k, \mathbf{x}_i) < 0\right\} \mathbb{1}\left\{\Delta(S_p \oplus \mathbf{x}_i, \mathbf{x}_i) < 0\right\} \mathrm{d}\chi^E(S_k, \mathbf{x}_i) \mathrm{d}\Pi(S_p)}_{\text{Poaching separations}} \\ &- \underbrace{e(\delta_f + \delta_m + \delta_r)}_{\text{Exogenous separations}}\end{aligned} \tag{A.1}$$

The first term corresponds to nonemployed workers who meet a firm. In every unit of time, a measure $\psi^N(1-e)$ of these workers arrive, and we must integrate out over their choice of whether or not to match with the firm. We know that they will form a match if and only if the match surplus is positive. The second term corresponds to *employed* workers who receive a renegotiation shock, and discover that the surplus at their current firm is negative. In this case, they separate and join the pool of nonemployed workers. The third term corresponds to the unusual case where a worker meets a new firm through on the job search, but discovers that their surplus at both the incumbent *and* the poaching firm is negative. In this case, they also separate to nonemployment. The final term is the contribution to EU flows of exogenous separations, whether due to firm death, exogenous match break shocks, or net flows to nonemployment due to worker retirement.

In principle, we could write down the Kolmogorov Forward Equations for the joint distribution over firm and worker states χ^E, the distribution of worker human capital levels χ^N, and the poaching distribution Π. However, this introduces a number of complications and, as we saw in Section 2, it is not necessary to characterize the solution of the model. Instead, we will show that it is possible to solve the model and approximate these distributions by simulation.

B Data Appendix

B.1 Variable List

- Hourly Wages: We use hourly wages to control for variation in hours worked. Wages are `S_BRUT` in the short panel and `SB` in the long panel. These are gross annual wages which include overtime and bonuses. We divide these by `NBHEUR`, the number of hours worked in the year, to get hourly wages. Finally, we use the CPI provided by INSEE to deflate all wages to 2015 euros.
- Establishment: We use the INSEE provided establishment codes `SIRET` to define the boundaries of a team.
- Occupation: We also use 1-digit occupation codes, the first digit of `PCS4`, to define a team.

C Neural Networks

Neural networks are a class of parametric, differentiable function approximators which can handle high-dimensional inputs and outputs. They have become extremely popular in recent years precisely because they exhibit these properties. In fact, neural networks that are sufficiently "deep" (we will make this precise shortly) are a class of universal function approximators (Hornik, Stinchcombe, and White, 1989): they can, in principle, approximate any continuous function to an arbitrary degree of accuracy. Furthermore, they have a non-linear structure, which allows them to represent a wide range of functions that are of interest in economic (and non-economic) applications, including discrete choices and functions that are kinked or non-differentiable at points. As we will see, there is a sense in which these approximators are sparsely parameterized: the number of parameters required scales linearly with the dimension of the input, which means they are well-equipped to handle very high dimensional input spaces. In contrast, this scales exponentially for series approximations like polynomials. Moreover, because they have become so widely used in the field of machine learning, there exist well-developed and mature software libraries for working with them.

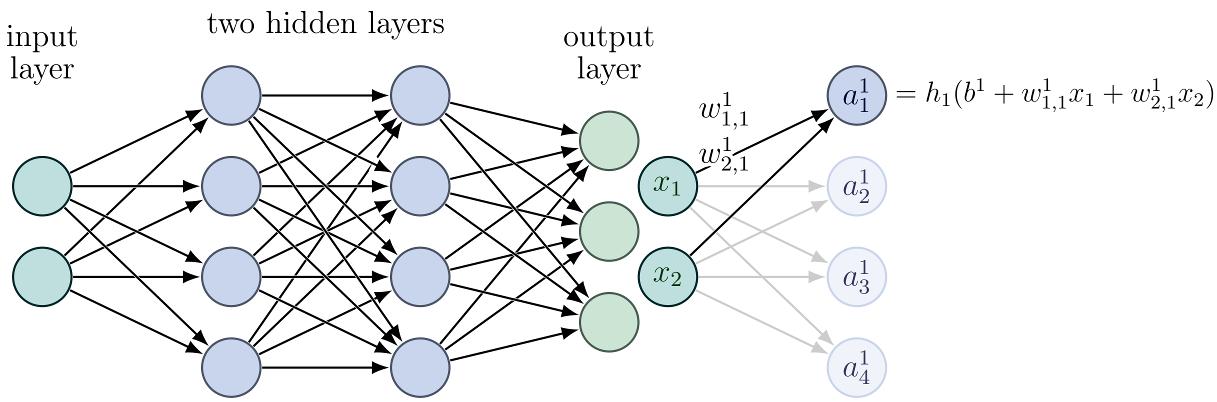
In this section, we will provide a brief overview of the basics of neural networks. We recommend Bishop (2006) and Goodfellow, Bengio, and Courville (2016) for a more detailed textbook treatment.

Definition. We build up to the definition of a neural network step-by-step. First, a *neuron* is a function $y : \mathbb{R}^m \to \mathbb{R}^1$ that takes in a list of inputs x, multiplies them by *weights* $w \in \mathbb{R}^m$, adds a *bias* $b \in \mathbb{R}$, then applies an *activation function* $h : \mathbb{R}^1 \to \mathbb{R}^1$, and returns the output.

$$y(x; w, b) = h\left(b + \sum_{i=1}^{m} w_i x_i\right)$$

A *layer* is a list of n neurons. The *width* of a layer is the number of neurons in that layer. The simplest neural network consists of a single layer mapping inputs to the final output. A *deep* neural network has multiple layers, where the output of one layer becomes the input of the next layer. The number of layers is the *depth* of the network. All layers excluding the final output layer are said to be *hidden*. The kth output of a network with two layers, an input of dimension m, a first layer with n_1 neurons, a second layer with n_2 neurons, and ℓ output dimensions is:

$$y_k(x; w, b) = h_3\left(b^3 + \sum_{j_2=1}^{n_2} w^3_{j_2,k} h_2\left(b^2 + \sum_{j_1=1}^{n_1} w^2_{j_1,j_2} h_1\left(b^1 + \sum_{i=1}^{m} w^1_{i,j_1} x_i\right)\right)\right) \tag{C.1}$$



(a) Network diagram (b) Neuron

Note: Tikz code adapted from `https://tikz.net/neural_networks/`.

Figure C.1: Example of a Neural Network as in Equation (C.1)

In Figure C.1 we show an example of a neural network with two inputs, two hidden layers (each of width four) and three outputs. There are many types of neural network architectures, but this is an example of a *dense* neural network: every node is fully connected to all the nodes in the previous layer. Because we have built this approximator up in blocks, it is easy to add more layers, or make an individual layer wider. Although dense neural networks are often a good place to start, many problems benefit substantially from more highly specialized neural network architectures.

Training. "Training" a neural network NN involves choosing parameters θ (i.e. the weights w and biases b) to minimize a "loss function".

Some common loss functions used to train value functions are the mean squared error (MSE),

$$\mathcal{L}^{MSE}(y;\theta) := \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2,$$

and the Huber loss

$$\mathcal{L}^{Huber}(y;\theta) := \frac{1}{N}\sum_{i=1}^{N}\begin{cases} 0.5(\hat{y}_i - y_i)^2 & \text{if} \quad |\hat{y}_i - y_i| < \delta, \\ \delta|\hat{y}_i - y_i| - 0.5\delta^2 & \text{otherwise,} \end{cases}$$

where N is the size of the "batch" or sample, x_i is ith element of the input vector, $\hat{y}_i = NN(x_i;\theta)$ is the output of the neural network parameterized by θ, and y_i is the truth. The main benefit of using the Huber loss relative to the MSE is that it is less sensitive to outliers. In our context, where we are minimizing the squared residual from an HJB equation, $\hat{y}_i$ is the model predicted residual, and since we know that at the true solution, the residual is zero, we set $y_i = 0$.

The parameters are updated in the direction of the gradient of the loss with respect to the parameters. The simplest version of this stochastic[28] gradient descent approach (see Amari 1993) is

$$\theta_{new} = \theta + \alpha_l \frac{\partial \mathcal{L}(x;\theta)}{\partial \theta},$$

where η is the *learning rate*, which controls the speed with which the parameters move in the direction of the gradient. Generally, a low value of the learning rate will converge more robustly but more slowly. Setting a high value of the learning rate can also help the network jump over "fitness valleys" in parameter space, i.e. regions close to a local minimum but far from any of the better local minima. There are many variations on the "vanilla" gradient descent algorithm that adjust the learning rate over the course of the training run or use a moving average of the gradient rather than the current value. Many are variants of the

[28] Because we only use a finite randomly-sampled batch, the gradient is not deterministic.

popular ADAM optimizer described in Kingma and Ba (2015).

Choice of Activation Functions. There are many choices of activation functions available to researchers, and the right activation functions for the hidden layers are not always the right ones for the output layer. One attractive feature of activation functions is they impose model constraints. For example, a non-negativity constraint can be enforced by choosing the common softplus function, $f(x) = \ln(1 + e^x)$, as the activation function for the outer layer of the neural network. In problems with a bounded range (for example, a consumption-savings problem where the policy is what share of cash-on-hand is consumed), the sigmoid activation function, $f(x) = 1/(1+e^x)$, is a popular choice. Finally, the softmax function $f : \mathbb{R}^k \to (0, 1)^k$ defined by $f(\mathbf{x}) = e^{x_i} / \sum_{j=1}^{k} e^{x_j}$ ensures that the output of the network belongs to the unit simplex. However, in many applications, a simple linear activation function for the outer layer performs well enough.

There are also many different activation functions to choose from for the inner layers. A common choice is the rectified linear unit (ReLU), defined as

$$f(x) = \begin{cases} x & \text{if} \quad x \geq 0, \\ 0 & \text{otherwise,} \end{cases}$$

which only "activates" past a certain threshold and is a very rough analogue for the functioning of a neuron in the brain. However, ReLU also has a gradient of zero below that threshold, which can cause issues with standard gradient-based training algorithms. Therefore the Leaky ReLU activation function,

$$f(x) = \begin{cases} x & \text{if} \quad x \geq 0, \\ \alpha x & \text{otherwise,} \end{cases}$$

where $\alpha \in (0, 1)$ is a popular alternative. Finally, the Swish activation function, $f(x) = x$ sigmoid(αx), acts like a smooth version of Leaky ReLU and often performs better.

Dealing with the Curse of Dimensionality. Neural networks have been shown to perform well on very high-dimensional problems where standard methods based on rectangular grids and interpolants quickly become intractable. Their success in these high-dimensional problems is not fully understood and is still a topic of study. Mechanically, increasing the input length only increases the size of the first layer of the network, and it only does so at a linear rate, although it is often advisable to increase the width of the other layers and the depth of the network to handle larger problems.

An interesting feature of neural networks is that they are over-parameterized: in general, they have more than enough parameters to fit their training data perfectly. Kahou, Fernández-Villaverde, Gómez-Cardona, Perla, and Rosa (2022) show that they have a built-in bias towards parameterizations that generate smaller gradients, which distinguishes them from polynomial approximations. In general, the solutions of economic problems tend to be fairly "smooth": the values and policies in between two grid points are often close to a convex combination of the values and policies at those grid points. Hence, it is unsurprising that function approximators biased towards low gradients would perform well on such problems.