

Deep Learning for Search and Matching Models*

Jonathan Payne[†] Adam Rebei[‡] Yucheng Yang[§]

April 29, 2024

Abstract

We develop a new method for characterizing global solutions to search and matching models with aggregate shocks and heterogeneous agents. We formulate general equilibrium as a high dimensional partial differential equation (PDE) with the distribution as a state variable. Solving this problem has previously been intractable because the distribution impacts agent decisions through the matching mechanism rather than through aggregate prices. We overcome these challenges by developing a new deep learning algorithm with efficient sampling in a high dimensional state space. This allows us to study search markets that are not “block recursive”. In applications to labor search models, we show that distribution feedback plays a more important role when aggregate shocks have an asymmetric impact across agents. Business cycles have a “cleansing” effect by amplifying positive assortative matching in recessions, and the magnitude of the countercyclicality depends on the bargaining process between workers and firms.

Keywords: Deep learning, Search and Matching, Block Recursive, Sorting, Business Cycles.

*We thank Fernando Alvarez, Isaac Baley, Jaroslav Borovička, Katarína Borovičková, Niklas Engbom, Jesús Fernández-Villaverde, Lukas Freund, Lars Hansen, Greg Kaplan, Ricardo Lagos, Guido Menzio, Jesse Perla, Xincheng Qiu, Tom Sargent, Edouard Schaal, Venky Venkateswaran, Gianluca Violante, and the seminar participants at ASU, Chicago, Minnesota, Norges Bank, NYU, Rice, DSE 2023, OzMac, Sargent Alumni Reading Group, Swiss Winter Macro Finance Workshop for their helpful comments and suggestions. We thank Yaqi Zeng for her research assistance.

[†]Princeton University, Department of Economics. Email: jepayne@princeton.edu

[‡]Stanford University. Email: rebei@stanford.edu

[§]University of Zurich, Department of Finance and SFI. Email: yucheng.yang@uzh.ch

1 Introduction

Heterogeneity and aggregate risks are important in markets with search frictions. However, modeling these markets has proven technically challenging. The existing literature either studies the deterministic steady state (e.g. [Shimer and Smith \(2000\)](#)) or imposes contracting restrictions to ensure that agent decisions are not affected by the distribution, thereby rendering the models “block-recursive” as seen in the seminal work of [Menzio and Shi \(2011\)](#) and [Lise and Robin \(2017\)](#).¹ Developments in the deep learning literature have opened up the possibility to relax these restrictions and solve high dimensional economic problems. In this paper, we present a general formulation of search and matching (SAM) models as high dimensional PDEs with distribution as a state variable. We then develop a new deep learning algorithm, which we refer to as DeepSAM, to solve this class of models. We apply our method to canonical models in the search and matching literature.

We focus on a class of models with the following features. The economy is populated by heterogeneous agents (e.g. workers or investors) and heterogeneous institutions (e.g. firms or financial intermediaries) that can be matched or unmatched. Matches generate utility that depends upon the idiosyncratic agent and institution types and an exogenous aggregate variable that follows a continuous time Markov chain. Unmatched agents and institutions engage in random search to meet each other. Upon meeting, they choose whether to accept the match and then bargain over the division of the matching surplus. We show that the equilibrium for this economy can be characterized recursively with a state space consisting of the exogenous aggregate variable and the distribution of matches across types in the economy. The match distribution impacts agent decisions because the opportunity cost of accepting a match depends on which other agents are looking for matches. This means that the partial differential equation (PDE) for the match surplus necessarily includes high dimensional terms capturing how the surplus changes as the distribution changes. By contrast, “block recursive” models impose restrictions to ensure these terms are zero.

We propose a new deep learning algorithm to solve this class of high dimensional PDEs. We approximate the surplus function with a neural network and then use gradient descent to train the neural network to minimize the average loss in the PDE for the surplus function on a random collection of sample points. To our knowledge, we are the first to apply deep learning to study search and matching models. In doing

¹Following [Lise and Robin \(2017\)](#), we call an equilibrium “block-recursive” if the agents’ value and policy functions are independent of the endogenous distribution of agents across employment states.

so, we face a number of technical challenges. First, unlike competitive market models (e.g. [Krusell and Smith \(1998\)](#)), the mean of the distribution is insufficient for calculating equilibrium. Instead, we need to integrate over the equilibrium surplus function, weighted by the population match density and the acceptance function. This means the shape of the distribution has a larger impact on the problem than in competitive market models and so we develop a sampling scheme that focuses on distributions close to the correct shape. Second, the surplus function can have sharp curvature in parts of the state space. For such cases, we use a “homotopy” approach combined with efficient sampling in a high dimensional state space. This involves training the neural network parameters that give low curvature and then gradually retraining the model with updated parameters.

In Section 3, we deploy our methodology to solve a “canonical” labor market search model. This model can be thought of as either the [Shimer and Smith \(2000\)](#) model with two-sided heterogeneity and aggregate shocks, or as the [Mortensen and Pissarides \(1994\)](#) model with worker and firm heterogeneity. We test our solution in a number of ways. We start by examining the neural network approximation to a model without aggregate shocks since the steady state of this model can be solved with existing fixed point solution techniques. We show that the average squared difference between our solution and the fixed point solution is in the order of magnitude of 10^{-6} . For the model with aggregate shocks, there are no existing solutions for us to compare to. Instead, we study the training error and stability of the solution. We show that the average numerical error on the differential equation over the full state space is in the order of 10^{-7} and that the standard deviation across many independent runs of our training algorithm is in the order of 10^{-4} at each point of the state space. We interpret these results as strong evidence that our neural network training algorithm can find an accurate solution.

We use our solution to the labor search model to revisit how heterogeneity interacts with aggregate labor market dynamics. We study a severe labor market crisis, where average job duration drops significantly. We compute the impulse response following the shock and compare the results to the “restricted dynamics” when agents make matching decisions believing they are always at the ergodic employment distribution. When the separation shock is symmetric across the population the restricted dynamics are similar to the full solution. However, when the separation shock is asymmetric and concentrated among low-skilled workers, the restricted model significantly overestimates the increase in unemployment during the recession and the rate of recovery following the recession. This is because agents in the restricted economy do not

internalize that the economy has more unemployed low-type workers and so the opportunity cost of waiting for better jobs is higher. Quantitatively, the restricted model overestimates peak unemployment by approximately 40%.

In Section 4, we introduce on-the-job search into our labor market model. Unlike [Lise and Robin \(2017\)](#) and subsequent papers, we are able to relax the assumption that firms have all the bargaining power in the market between unemployed workers and firms. We show that the firm bargaining power significantly skews the assortative matching pattern, with high firm bargaining power making high-type firms less picky and high worker bargaining power making high-type workers less picky. We further investigate the cyclicity of sorting patterns over the business cycles. We find that positive assortative matching is stronger in a recession and weaker in an expansion. This is because in a recession fewer agents are matched and so the opportunity cost of waiting for a high type is lower. The extent of countercyclical sorting over business cycles is significantly influenced by the bargaining process between workers and firms. With a standard calibration of the worker bargaining power used in the business cycle literature (e.g. by [Shimer \(2005\)](#)), the cyclical sorting dynamics are approximately 75% smaller than in an [Lise and Robin \(2017\)](#), where they impose that firms have all the bargaining power to ensure the model is block recursive. The intuition is that when one side of the market has all the bargaining power their decisions end up being more responsive to changes in the opportunity cost of waiting.

Literature Review: Over the past three decades, there have been two major advances in solving search and matching models with heterogeneity and aggregate risk. One is the Bertrand competition model of wage setting introduced in [Postel-Vinay and Robin \(2002\)](#) and deployed in many papers (e.g. [Cahuc et al. \(2006\)](#); [Lise and Robin \(2017\)](#)). The other is the directed search block recursive structure introduced by [Moen \(1997\)](#); [Menzio and Shi \(2010, 2011\)](#). Both these approaches impose contracting and entry assumptions that ensure that agent decisions are independent of the distribution of matches. Our paper relaxes these constraints and solves a general class of models where the distribution may impact agents' decision making.²

We are part of a growing computational economics literature using deep learning techniques to solve economic models and overcome the limitations of traditional solution techniques. These papers have focused on solving heterogeneous agent macroeconomic models with incomplete but competitive markets (e.g. [Duarte \(2018\)](#), [Azinovic et al.](#)

²Our work is also relevant to [Petrosky-Nadeau and Zhang \(2017\)](#), which proposes a global solution method to representative agent search models with aggregate shocks.

(2022), Maliar et al. (2021), Han et al. (2021), Kahou et al. (2021), Fernández-Villaverde et al. (2023), Gopalakrishna (2021), Sauzet (2021), Huang (2022), Gu et al. (2023), among others). Our contribution is to show how to undertake deep learning to solve a search and matching model. In doing so, we characterize the equilibrium as a high dimensional “master equation” similar to those discussed in Bilal (2023); Gu et al. (2023); Alvarez et al. (2023); Ahn et al. (2018). What makes the search and matching model different to train compared to a competitive incomplete markets model is that the shape of the distribution matters for calculating equilibrium rather than just calculating the evolution of the distribution. This is because, as summarized in Table 1, the distribution impacts agents’ decisions via the matching probability with other types rather than through aggregate prices. This imposes greater challenges on how we develop our numerical and sampling schemes to get an accurate solution.

	Distribution	How distribution affect agents’ decisions
HAM	Asset wealth and income	Via aggregate prices
SAM	Type (productivity) of agents in two sides of matching	Via matching probability with other types

Table 1: How distribution matters in heterogeneous agent models (HAM) vs search and matching (SAM) models.

Our paper is also connected to recent papers studying business cycle dynamics in heterogeneous agent labor search models (e.g. Krusell et al. (2017); Schaal (2017); Moscarini and Postel-Vinay (2018); Engbom (2021); Alves (2022); Qiu (2023)). Our contribution to this body of literature primarily centers on analyzing the feedback mechanisms generated by alterations in the distribution, which in turn influence agents’ matching decisions.

The paper is structured as follows. Section 2 describes our DeepSAM methodology in a general setup of search and matching models. Section 3 applies DeepSAM to a canonical labor market search model. Section 4 extends the model to incorporate on-the-job search. Section 5 concludes.

2 Methodology

In this section, we outline a general model that nests search and matching models from various streams of the literature. We then introduce our deep learning algorithm,

DeepSAM, to solve the model.

2.1 Environment

Setting: The economy is in continuous time with an infinite horizon. The economy is populated by a continuum of infinitely lived agents (e.g. workers or investors) indexed by type x , and a continuum of institutions (e.g. firms or dealers) indexed by type y . Agents are either employed in a match (e) or unmatched (u). Institutions are either producing in a match (p) or vacant (v). The distribution of matches between agents and institutions is endogenous, and determined by agent and institution decisions. Agents and institutions have a discount rate ρ . The aggregate state of the economy is indexed by $z_t \in \mathcal{Z}$, which follows a continuous time Markov chain with transition matrix Σ .

Match utility: If an agent is unmatched (unemployed), they get flow utility b . Agents match with institutions but not with each other. If an agent of type x is matched with an institution of type y , they generate transferable utility $F(x, y, z)$, where F is increasing in each variable and twice differentiable with uniformly bounded first partial derivatives on $\mathcal{Z} \times [0, 1] \times [0, 1]$. Matches are destroyed at exogenous rate $\delta(x, y, z)$ that potentially depends upon the match and the aggregate state.

Distributions: Let $g^w(x)$ denote the constant exogenous population function of agents. Let $g^f(y)$ denote the constant exogenous population function of institutions. We will relax the assumption on exogenous institution population later with a free entry condition in Section 2.2.6. Let $g_t(x, y)$ denote the function of matched workers. Let $g_t^e(x)$ denote the function of employed agents. Let $g_t^u(x)$ denote the function of unemployed agents. Let $g_t^p(y)$ denote the function of producing institutions. Let $g_t^v(y)$ denote the function of vacant institutions. The relationships between the densities are given in Table 2 below. We define the aggregate agent employment by $E_t := \int g_t^e(x)dx$, aggregate agent unemployment by $\mathcal{U}_t := \int g_t^u(x)dx$, aggregate producing institutions by $\mathcal{P}_t := \int g_t^p(y)dy$, and aggregate vacant institutions by $V_t := \int g_t^v(y)dy$. Observe that we can calculate all densities from g_t and so (z_t, g_t) is a sufficient aggregate state space for the economy.

Search and Matching Technology: Only and all unmatched agents engage in random search. We generalize to include “on-the-job” search in Section 4. A function $m : \mathcal{Z} \times \mathcal{G} \rightarrow \mathbb{R}^+, (z_t, g_t) \mapsto m(z_t, g_t)$ takes the state of the economy and gener-

Description	Function	Conditional Density
Matches	$g_t(x, y)$	
Employed workers	$g_t^e(x) = \int g_t^m(x, y) dy$	$g_t^e(x)/\mathcal{E}_t$
Unemployed workers	$g_t^u(x) = g_t^w(x) - g_t^e(x)$	$g_t^u(x)/\mathcal{U}_t$
Producing firms	$g_t^p(y) = \int g_t^m(x, y) dx$	$g_t^p(y)/\mathcal{P}_t$
Vacant firms	$g_t^v(y) = g_t^f(y) - g_t^p(y)$	$g_t^v(y)/\mathcal{V}_t$

Table 2: Summary of distributions

ates meetings. The rate at which a worker meets a potential institution is given by $\mathcal{M}_t^u := m(z_t, g_t)/\mathcal{U}_t$, while the rate at which a vacant firm meeting a potential hire is $\mathcal{M}_t^v := m(z_t, g_t)/\mathcal{V}_t$. The rate at which that an agent meets any institution $y \in Y \subset [0, 1]$ equals $\mathcal{M}_t^u(\int_Y (g_t^v(y)/\mathcal{V}_t) dy)$, where $g_t^v(y)/\mathcal{V}_t$ is the density conditional on being vacant. The rate at which an institution meets any worker $x \in X \subset [0, 1]$ equals $\mathcal{M}_t^v(\int_X (g_t^u(x)/\mathcal{U}_t) dx)$, where $g_t^u(x)/\mathcal{U}_t$ is the density conditional on being unemployed.

Surplus division: We impose that agents negotiate according to a generalized Nash Bargaining protocol so that agents get a fraction β of surplus and institutions get the remaining fraction $1 - \beta$. The contract is implemented by providing $w(x, y, z, g)$ flow utility to the agent and $f(x, y, z) - w(x, y, z, g)$ flow utility to the institution.

2.2 Recursive Characterization of Equilibrium

We now define and characterize a recursive equilibrium. The aggregate states are (z, g) , where z is the aggregate productivity and g is the distribution of matches.³ We guess (and later verify) that the law of motion for g takes the form:

$$dg_t(x, y) = \mu^g(x, y, z, g) dt.$$

2.2.1 Surplus Division

Let $V^u(x, z, g)$ denote the value of unemployment for a worker of type x . Let $V^e(x, y, z, g)$ denote the value of worker x employed at an institution of type y . Let $V^v(y, z, g)$ denote the value of a vacancy for firm y . Let $V^p(x, y, z, g)$ denote the value of firm y

³The mean field game literature has studied the mathematical difficulties involved in defining a recursive equilibrium with an infinite dimensional state (e.g. [Cardaliaguet et al. \(2015\)](#)). However, there is debate about whether these characterizations are appropriate for economic models. For our purposes, we are always going to work with a finite type space so the only relevant mathematical question is whether a limit exists as the type space becomes continuous.

employing a worker of type x . The surplus of a match is defined as:

$$S(x, y, z, g) := V^p(x, y, z, g) - V^v(y, z, g) + V^e(x, y, z, g) - V^u(x, z, g)$$

The Nash Bargaining protocol implies that the division of surplus is given by:

$$\begin{aligned} \beta S(x, y, z, g) &= V^e(x, y, z, g) - V^u(x, z, g) \\ (1 - \beta)S(x, y, z, g) &= V^p(x, y, z, g) - V^v(y, z, g) \end{aligned} \tag{2.1}$$

These equations must implicitly define a transfer or “wage” (x, y, z, g) . As in other papers, we assume that contract terms are indexed to the contracts of new hires so that $V^e(x, y, z, g) - V^u(x, z, g)$ and $V^p(x, y, z, g) - V^v(y, z, g)$ are the same for all agents with a particular (x, y) .

2.2.2 Agent Hamilton-Jacobi-Bellman (HJB) Equations

Suppose that agents believe that the evolution of the match distribution g is characterized by function $\tilde{\mu}^g(x, y, z, g)$. Given beliefs, the value functions V^u , V^e , V^v , and V^p satisfy the Hamilton Jacobi Bellman (HJB) Equations:

$$\begin{aligned} \rho V^u(x, z, g) &= b + \mathcal{M}^u \int \alpha(x, \tilde{y}, z, g) \beta S(x, \tilde{y}, z, g) \frac{g^v(\tilde{y})}{\mathcal{V}} d\tilde{y} \\ &\quad + \sum_{\tilde{z} \neq z} \lambda(z, \tilde{z})(V^u(x, \tilde{z}, g) - V^u(x, z, g)) + \langle D_g V^u, \tilde{\mu}^g \rangle \\ \rho V^e(x, y, z, g) &= w(x, y, z, g) - \beta \delta(x, y, z) S(x, y, z, g) \\ &\quad + \sum_{\tilde{z} \neq z} \lambda(z, \tilde{z})(V^e(x, \tilde{z}, g) - V^e(x, z, g)) + \langle D_g V^e, \tilde{\mu}^g \rangle \\ \rho V^v(y, z, g) &= \mathcal{M}^v \int \alpha(\tilde{x}, y, z, g) (1 - \beta) S(\tilde{x}, y, z, g) \frac{g^u(\tilde{x})}{\mathcal{U}} d\tilde{x} \\ &\quad + \sum_{\tilde{z} \neq z} \lambda(z, \tilde{z})(V^v(x, \tilde{z}, g) - V^v(x, z, g)) + \langle D_g V^v, \tilde{\mu}^g \rangle \\ \rho V^p(x, y, z, g) &= F(x, y, z) - w(x, y, z, g) - \delta(1 - \beta) S(x, y, z, g) \\ &\quad + \sum_{\tilde{z} \neq z} \lambda(z, \tilde{z})(V^p(x, \tilde{z}, g) - V^p(x, z, g)) + \langle D_g V^p, \tilde{\mu}^g \rangle. \end{aligned} \tag{2.2}$$

where $D_g V^j$ is the Frechet derivative of V^j with respect to the distribution g , $\langle f(y), h(y) \rangle = \int f(y)h(y)dy$ is the inner product, and α is an indicator for the acceptance of a match:

$$\alpha(x, y, z, g^m) := \begin{cases} 1, & \text{if } S(x, y, z, g) > 0 \\ 0, & \text{otherwise} \end{cases} \quad (2.3)$$

The acceptance function has this form because when surplus is positive and there is generalized Nash bargaining, both agents and institutions accept the match.

2.2.3 Distribution Evolution

Given the matching decisions of the agents, the measure of matches evolves according to:

$$dg_t(x, y) = -\delta(x, y, z)g_t(x, y)dt + \mathcal{M}_t^u g_t^u(x)\alpha(x, y, z, g^m) \frac{g_t^v(y)}{\mathcal{V}_t} dt$$

Given the state, g , then we can recover the other features of the distribution from Table 2. So, the KFE can be expressed as:

$$\begin{aligned} dg_t(x, y) &= -\delta(x, y, z)g_t(x, y) + \frac{m(\mathcal{U}_t, \mathcal{V}_t)}{\mathcal{U}_t \mathcal{V}_t} \alpha_t(x, y) \left(g^w(x) - \int g_t(x, y)dy \right) \\ &\quad \times \left(g^f(y) - \int g_t(x, y)dx \right) \\ &=: \mu^g(x, y, z, g^m)dt \end{aligned} \quad (2.4)$$

2.2.4 Equilibrium and Master Equation

Definition 1. A **(recursive) equilibrium** is a collection of functions $\{V^u, V^e, V^v, V^p, w, \alpha\}$ such that: (i) given a belief about the evolution of g_t , $(V^u, V^e, V^v, V^p, \alpha)$ solve the HJB equations (2.2), (ii) division of surplus satisfies (2.1), and (iii) agent beliefs about the evolution of g_t are consistent in the sense that $\tilde{\mu}^g = \mu^g$, where (2.2) is given by equation (2.4).

After combining the HJB equations and imposing belief consistency, the equilibrium

can be characterized by the “master equation” for the surplus:

$$\begin{aligned}
0 &= \mathcal{L}^S S \\
&=: -\rho S(x, y, z, g) + F(x, y, z) - \delta(x, y, z)S(x, y, z, g) \\
&\quad - (1 - \beta) \frac{m(z, g)}{\mathcal{V}(z, g)} \int \alpha(\tilde{x}, y, z, g) S(\tilde{x}, y, z, g) \frac{g^u(\tilde{x})}{\mathcal{U}(z, g)} d\tilde{x} \\
&\quad - b - \beta \frac{m(z, g)}{\mathcal{U}(z, g)} \int \alpha(x, \tilde{y}, z, g) S(x, \tilde{y}, z, g) \frac{g^v(\tilde{y})}{\mathcal{V}(z, g)} d\tilde{y} \\
&\quad + \langle D_g S(x, y, z, g), \mu^g(x, y, z, g) \rangle \\
&\quad + \sum_{\tilde{z} \neq z} \lambda(z) (S(x, y, \tilde{z}, g) - S(x, y, z, g))
\end{aligned} \tag{2.5}$$

where μ^g is given by (2.4), $(\mathcal{U}, \mathcal{V}, g^u, g^v)$ can be calculated by Table 2, and α is given by equation (2.3).

2.2.5 Relation to Environments in Other Papers

Block recursivity: We can compare this setup to well known papers in the search literature with “block-recursivity”. [Lise and Robin \(2017\)](#) sets $\beta = 0$ and introduces a “free” vacancy creation⁴ so that:

$$\alpha(x, y, z, g) = \alpha(x, y, z), \quad S(x, y, z, g) = S(x, y, z)$$

and the model has “block-recursivity”. [Menzio and Shi \(2011\)](#) has one-sided heterogeneity, competitive search, and “free” firm entry so:

$$S(x, y, z, g) = S(x, y, z)$$

and the model also has “block-recursivity”. The goal of our paper is to solve for α and S explicitly as a function of g .

Dimension reduction: For models with incomplete but competitive markets, [Krusell and Smith \(1998\)](#) suggests replacing the law of motion for the distribution by the law of motion of its mean (and potentially the law of motion of other low dimensional moments). This is a plausibly appealing approach for competitive market models because the distribution impacts agents’ decisions by changing aggregate prices and aggregate prices primarily depend upon the mean of the distribution. By contrast, in

⁴They also introduce on-the-job search, which we compare to in Section 4.

a search and matching model, the distribution impacts agents' decisions by changing the probability of which type of agent they meet. This ultimately enters the master equation on lines 3 and 4 of equation (2.5). We can see that there low dimensional moments of the distribution are sufficient for evaluating these terms. Instead, we need to the integral across the surplus function, weighted by the acceptance decision and the density of searching agents.

2.2.6 Free Entry

So far, we have described a model with a fixed population of firms. A common assumption in the labor literature is that firms can enter and create vacancies if they pay a cost. In this subsection, we show how this changes the surplus master equation.

Environment Changes: We now allow new firms to pay a cost c and enter the model with a draw of y from the uniform distribution $U(0, 1)$. This introduces an additional “free-entry” condition into the model that:

$$c = \mathbb{E}[V_t^v] = \int V^v(\tilde{y}, z, g) d\tilde{y}.$$

For simplicity, we also assume that the matching function is homothetic and only depends upon the ratio $\mathcal{V}_t/\mathcal{U}_t$ so that

$$\frac{m(z_t, g_t)}{\mathcal{V}_t} = \hat{m}\left(\frac{\mathcal{V}_t}{\mathcal{U}_t}\right).$$

Master Equation for Surplus with Free Entry Condition: Combining the free entry condition with the HJB equations gives:

$$\hat{m}\left(\frac{\mathcal{V}_t}{\mathcal{U}_t}\right) = \frac{\rho c}{\int \int \alpha(\tilde{x}, \tilde{y}) \frac{g_t^u(\tilde{x})}{\mathcal{U}_t} (1 - \beta) S_t(\tilde{x}, \tilde{y}) d\tilde{x} d\tilde{y}} \quad (2.6)$$

$$\Rightarrow \mathcal{V}_t = \mathcal{U}_t \hat{m}^{-1} \left(\frac{\rho c}{\int \int \alpha(\tilde{x}, \tilde{y}) \frac{g_t^u(\tilde{x})}{\mathcal{U}_t} (1 - \beta) S_t(\tilde{x}, \tilde{y}) d\tilde{x} d\tilde{y}} \right) \quad (2.7)$$

where $g_t^u = g_t^w - \int g_t^m(x, y) dy$ and so the RHS can be computed from g_t^m and S_t . For Equation (2.6), compared to the problem with exogenous institution population $g^f(y)$, the matching rate now depends upon the average surplus because new institutions enter the model until it is no longer profitable to do so. Since firm y draws are uniformly

distributed, g_t^f is given by:

$$g_t^f = \mathcal{V}_t + \mathcal{P}_t \quad (2.8)$$

where again \mathcal{V}_t and \mathcal{P}_t can be expressed in terms of g and S .

With the free entry condition, the master equation expression for surplus takes the same form as Equation (2.5) but with different definitions of $\mathcal{V}(z, g)$ and $g^v(y)$. $\mathcal{V}(z, g)$ is now calculated from Equation (2.7), and $g^v(y) = g^f(y) - \int g(x, y) dx$ can be calculated with z , g , and S using Equations (2.7) and (2.8). In addition, $\mu^g(x, y, z, g)$ also comes from the same expression of KFE (2.4) but with different definitions of $g^f(y)$ and \mathcal{V} . For more details, see Appendix A.

Even though the surplus master equation is more non-linear, we can still solve the master equation using similar techniques as the master equation without free entry.

2.3 Approximation with Finite Types

Our goal is to solve the master equation (2.5) numerically to obtain $S(x, y, z, g)$ and $\alpha(x, y, z, g)$. Then we could solve for the value and wage functions using Equation (2.2). The difficulty of solving Equation (2.5) is that the state space contains an infinite dimensional distribution, g , and so the master equation contains Frechet derivatives with respect to the distribution. To make progress on this problem, we discretize the type space so that equation (2.5) becomes a high, but finite dimensional partial differential equation that can be solved using deep learning.

Discrete type space and KFE: We restrict the possible types to a finite collection: $x \in \mathcal{X} = \{x_1, \dots, x_{n_x}\}$ and $y \in \mathcal{Y} = \{y_1, \dots, y_{n_y}\}$. With some abuse of notation, we let $\underline{\mathbf{g}}_t$ denote the vector of measures of matched agents at the points $(\mathcal{X}, \mathcal{Y})$, where $g_{tij} = g_t(x_i, y_j)$ is the function at type (x_i, y_j) . The aggregate state variables are now: $\{z, \underline{\mathbf{g}}_t\}$, the aggregate productivity and the density vector of matched agents. Under this discretization, the Riemann approximation to the KFE is given by:

$$\begin{aligned} dg_t(x, y)/dt &= \mu^g(x, y, z_t, \underline{\mathbf{g}}_t) \\ &= -\delta(x, y, z)g_t(x, y) + \frac{m(z_t, \underline{\mathbf{g}}_t)}{\mathcal{U}(z_t, \underline{\mathbf{g}}_t)\mathcal{V}(z_t, \underline{\mathbf{g}}_t)}\alpha(x, y, z, \underline{\mathbf{g}}_t) \\ &\quad \times \left(g^w(x) - \frac{1}{n_y} \sum_{j=1}^{n_y} \underline{\mathbf{g}}_t(x, y_j) \right) \left(g^f(y) - \frac{1}{n_x} \sum_{i=1}^{n_x} \underline{\mathbf{g}}_t(x_i, y) \right) \end{aligned} \quad (2.9)$$

Master equation: The discretized Master equation for the Surplus is given by:

$$\begin{aligned}
0 &= \mathcal{L}^S S \\
&= -(\rho + \delta(x, y, z))S(x, y, z, \underline{\mathbf{g}}) + F(x, y, z) - b \\
&\quad - (1 - \beta) \frac{m(z_t, \underline{\mathbf{g}})}{\mathcal{U}(z, \underline{\mathbf{g}})\mathcal{V}(z, \underline{\mathbf{g}})} \frac{1}{n_x} \sum_{i=1}^{n_x} \alpha(x_i, y, z, \underline{\mathbf{g}})S(x_i, y, z, \underline{\mathbf{g}})\underline{\mathbf{g}}^u(x_i) \\
&\quad - \beta \frac{m(z_t, \underline{\mathbf{g}})}{\mathcal{U}(z, \underline{\mathbf{g}})\mathcal{V}(z, \underline{\mathbf{g}})} \frac{1}{n_y} \sum_{j=1}^{n_y} \alpha(x, y_j, z, \underline{\mathbf{g}})S(x, y_j, z, \underline{\mathbf{g}})\underline{\mathbf{g}}^v(y_j) \\
&\quad + \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \partial_{g_{ij}} S(x, y, z, \underline{\mathbf{g}}) \mu^g(x_i, y_j, z, \underline{\mathbf{g}}) \\
&\quad + \sum_{\tilde{z} \neq z} \lambda(z)(S(x, y_j, \tilde{z}, \underline{\mathbf{g}}) - S(x, y_j, z, \underline{\mathbf{g}}))
\end{aligned} \tag{2.10}$$

where to ensure differentiability of the value function when there is a finite number of types, we approximate $\alpha(x, y, z, g)$ by:

$$\alpha(x, y, z, \underline{\mathbf{g}}) = \left(1 + e^{-\xi S(x, y, z, \underline{\mathbf{g}})}\right)^{-1}$$

which can be interpreted as a logit choice model where utility shocks come from an extreme value distribution with parameter ξ .

2.4 DeepSAM Algorithm

In this section, we outline our algorithm for solving the discretized master equation (2.10). Let $\omega = (x, y, z, \underline{\mathbf{g}}) \in \Omega^\omega$ denote the state space. Let $\Theta \in \Omega^\Theta$ denote a collection of parameters. We approximate the surplus function S by a “feed-forward” neural network:

$$\hat{S} : \Omega^\omega \times \Omega^\Theta \rightarrow \mathbb{R}, \quad (\omega, \Theta) \mapsto \hat{S}(\omega; \Theta)$$

with form:

$$\begin{aligned}
\mathbf{h}^{(1)} &= \phi^{(1)}(W^{(1)}\boldsymbol{\omega} + \mathbf{b}^{(1)}) && \dots \text{Hidden layer 1} \\
\mathbf{h}^{(2)} &= \phi^{(2)}(W^{(2)}\mathbf{h}^{(1)} + \mathbf{b}^{(2)}) && \dots \text{Hidden layer 2} \\
&\vdots \\
\mathbf{h}^{(H)} &= \phi^{(H)}(W^{(H)}\mathbf{h}^{(H-1)} + \mathbf{b}^{(H)}) && \dots \text{Hidden layer H} \\
\hat{S} &= \sigma(\mathbf{h}^{(H)}) && \dots \text{Surplus}
\end{aligned}$$

where, using the terminology of the deep learning literature, H is referred the number of hidden layers, the length of vector $\mathbf{h}^{(i)}$ is referred to as the number of neurons in hidden layer i , $\phi^{(i)}$ is referred to as the activation function for hidden layer i , and the collection $\Theta = (W^1, \dots, W^{(H)}, b^{(1)}, \dots, b^{(H)})$ are the parameters for the neural network.

Our goal is to train the parameters of the neural network to approximately solve equation (2.10). Our approach is summarized in Algorithm 1. Essentially, we use stochastic gradient descent to train the neural network to minimize the average loss in the master equation on a random collection of sample points.

2.4.1 Implementation Details

As with other neural network approaches, there are many implementation details involved with these generic steps. What makes the search and matching model different to solve compared to a competitive incomplete markets model? An important difference is that in a search and matching model, the shape of the distribution matters for calculating equilibrium rather than just for calculating the evolution of the distribution. This imposes greater restrictions on how we need to sample to get an accurate solution. We discuss how this impacts some key decisions below.

Sampling procedure: We first solve the model at the steady state for the different fixed z . We then start by drawing distributions that are a random combination of the steady state distributions for the different z . If required, once the error is small, we can then move to sampling from the ergodic distribution generated by the current solution. We can also increase sampling in regions of the state space (x, y) where errors are high.

Algorithm stability It is most difficult to stabilize the algorithm when $\hat{S}(x, y, z, g; \Theta)$ has sharp curvature. In this case, we use a “homotopy” approach. Step (1): Train NN for parameters that give low curvature in \hat{S}^1 . Step (2): Change parameters closer and

Algorithm 1: Generic Solution Algorithm

Input : Initial neural network parameters Θ^0 , number of points K to sample, sequence of learning rates $\{\zeta_n : n \geq 0\}$, precision threshold ϵ .

Output: A neural network approximation $(x, y, z, g) \mapsto \hat{S}(x, y, z, g; \Theta)$ of the surplus function S solving the discretized master equation.

1. Approximate surplus function by neural network $S(x, y, z, g) \approx \hat{S}(x, y, z, g; \Theta)$.
2. Start with initial parameter guess Θ^0 .
3. At iteration n with Θ^n :

- (a) Generate K sample points, $Q^n = \{(x_k, y_k, z_k, \{g_{ij,k}\}_{i \leq n_x, j \leq n_y})\}_{k \leq K}$.
- (b) Calculate the average mean squared error of surplus master equation (2.10) on sample points:

$$L(\Theta^n, Q^n) := \frac{1}{K} \sum_{k \leq K} \left| \mathcal{L}^S \hat{S}(x_k, y_k, z_k, \{g_{ij,k}\}_{i \leq n_x, j \leq n_y}) \right|^2$$

- (c) Update NN parameters with stochastic gradient descent (SGD) method:

$$\Theta^{n+1} = \Theta^n - \zeta^n \nabla_{\Theta} L(\Theta^n, Q^n)$$

- (d) Repeat until $L(\Theta^n, Q^n) \leq \epsilon$ with precision threshold ϵ .
 4. Once S is solved, we have α and can solve for worker and firm value functions.
-

retrain NN starting from previous $\hat{S}^2 = \hat{S}^1$. Step (3)+: keep changing parameters and retraining until at desired parameters.

Ultimately, our deep learning algorithm will allow us to solve models with high dimensionality. Sometimes this is referred to as “breaking the curse of dimensionality”. However, that is not the best description of what is happening. Instead, we are looking for the appropriate subspace of the high dimensional state space and then training the model on that subspace. In this sense, it is the flexibility of the neural network to train on an arbitrary subspace that opens up the possibility of solving high dimensional models. If we attempted to train the model on all conceivable distribution, then we hit the same “curse of dimensionality”.

3 Labor Search Model

In this section, we show how to use our algorithm to solve labor search and matching models. We focus on a canonical model that can be thought of as either [Shimer and Smith \(2000\)](#) with two-sided heterogeneity and aggregate shocks, or as [Mortensen and Pissarides \(1994\)](#) with worker and firm heterogeneity. Our presentation is similar to a continuous time version of [Hagedorn et al. \(2017\)](#) with aggregate shocks.

3.1 Environment Details

Our environment is a special case of the Section 2.1 with the following features. The agents are interpreted as “workers” and the institutions as “firms”. Matches are interpreted as producing output $F(z, x, y) = zf(x, y)$, where z is interpreted as aggregate productivity in the economy. Following [Shimer \(2005\)](#), we consider aggregate shocks to the aggregate productivity and the separation rate. The matching function $m : [0, 1] \times [0, 1] \rightarrow [0, \min(\mathcal{U}, \mathcal{V})], (\mathcal{U}, \mathcal{V}) \mapsto m(\mathcal{U}, \mathcal{V})$ is restricted to take the masses of unemployed workers \mathcal{U} and vacant firms \mathcal{V} as inputs. We allow for the free entry of firms.

3.2 Parameters

Economic parameters: The calibration of the economic parameters for our baseline model can be seen in Table 3. We calibrate the model at the annual frequency. Where possible, we take standard parameters from the literature. The only parameter that we calibrate “internally” is the scale parameter for the meeting function. We choose

Parameter	Interpretation	Value	Target/Source
ρ	Discount rate	0.05	Kaplan et al. (2018)
δ	Job destruction rate	0.2	BLS job tenure 5 years
ξ	Extreme value for α choice	2.0	
$f(x, y)$	Production function for match (x, y)	$0.6 + 0.4 (\sqrt{x} + \sqrt{y})^2$	Hagedorn et al. (2017)
β	Surplus division factor	0.72	Shimer (2005)
$m(\mathcal{U}, \mathcal{V})$	Matching function	$\kappa \mathcal{U}^\nu \mathcal{V}^{1-\nu}$	Lise and Robin (2017)
ν	Elasticity in meeting function	0.5	Lise and Robin (2017)
κ	Scale parameter for meeting function	5.4	Unemployment rate
b	Worker unemployment benefit	0.5	Shimer (2005)
n_x	Discretization of worker types	7	
n_y	Discretization of firm types	8	
c	Entry cost	4.86	Steady state $\mathcal{V}/\mathcal{U} = 1$
Steady State:			
\bar{z}	Steady state TFP	1	Shimer (2005)
$\bar{\delta}$	Steady state separation rate	0.2	Shimer (2005)
Productivity Shock:			
z_L, z_H	TFP shocks	0.985, 1.015	Lise and Robin (2017)
$\lambda_{z,LH}, \lambda_{z,HL}$	Poisson transition probability	0.08, 0.08	Shimer (2005)
Separation Shock:			
δ_L, δ_H	Separation rate shocks	0.18, 0.22	Shimer (2005)
$\lambda_{\delta,LH}, \lambda_{\delta,HL}$	Poisson transition probability	0.1, 0.1	Shimer (2005)

Table 3: Economic Parameters.

this parameter to target an average unemployment rate across the economy of 6% in the ergodic mean.

Neural network parameters: We describe the details of the neural network approximation and sampling in Table 4. We use a fully connected feed-forward network with 4 layers, 50 neurons per layer, and a $\tanh(\cdot)$ activation function. We sample data points from random combinations of the stationary distribution distribution for each TFP z .

3.3 Accuracy and Verification

We start by testing the accuracy of our solution approach. We first consider a version of the model without aggregate shocks since we can compare that to solutions from existing fixed point solution methods such as that in Hagedorn et al. (2017). We then examine the numerical error and stability when we use DeepSAM to solve the model with aggregate shocks.

Parameter	Value
Number of layers	4
Neurons per layer	50
Activation function	$\tanh(\cdot)$
Initial learning rate	10^{-4}
Final learning rate	10^{-5}
Sampling distribution	
Initial sample size per epoch	256
Final sample size per epoch	512
Convergence threshold for target calibration	10^{-6}

Table 4: Neural network parameters

3.3.1 Solution Without Aggregate Shocks

To check the numerical accuracy of the DeepSAM method, we first use it to solve the model without aggregate shocks, namely $z_t \equiv \bar{z}, \delta_t \equiv \bar{\delta}$. The master equation for this special case is a 58-dimensional PDE for $S(x, y, \underline{\mathbf{g}})$ under our choice of discretization, and the detailed setup is presented in Appendix B. We perform a number of checks to verify the accuracy of our solution technique. We discuss the tuning process to achieve this level of accuracy in the neural network training process in Appendix B.2.1.

Numeric performance metric	Value
PDE training loss	3.9×10^{-6}
MSE to deterministic steady state	5×10^{-6}

Table 5: Model without aggregate shock: loss and comparison to conventional method.

We show the results in Table 5. The PDE training loss is the mean square error reached an average of 3.9×10^{-6} over the 58-dimensional state space $(x, y, \underline{\mathbf{g}})$ after 57 minutes of training. For this problem without aggregate shocks, we can use a conventional method, such as the fixed point algorithm such as that in [Shimer and Smith \(2000\)](#); [Hagedorn et al. \(2017\)](#), to solve for the deterministic steady state. Then we can compare our solution at the deterministic steady state with the benchmark solution using conventional methods $S_{\text{Conventional}}^{\text{DSS}}(x, y)$. By setting $\underline{\mathbf{g}} = \underline{\mathbf{g}}^{\text{DSS}}$, we get our solution at DSS as

$$S_{\text{DeepSAM}}^{\text{DSS}}(x, y) = S(x, y, \underline{\mathbf{g}} = \underline{\mathbf{g}}^{\text{DSS}})$$

We define the squared difference of the two methods for each (x, y) pair as $\|S_{\text{DeepSAM}}^{\text{DSS}}(x, y) -$

$S_{\text{Conventional}}^{\text{DSS}}(x, y)\|^2$. The mean squared difference takes the average of the squared difference across all (x, y) pairs. We also depict the results visually. The upper right panel of Figure 1 shows the mean squared loss in the surplus Master equation at the steady state distribution. Evidently, the training error is in the order of magnitude of 10^{-6} and not biased in a particular part of the state space. The lower right panel of Figure 1 plots the squared difference between the DeepSAM solution and conventional solution at the DSS.⁵ Evidently, the difference is in the order of magnitude of 10^{-5} , which we interpret as high accuracy.

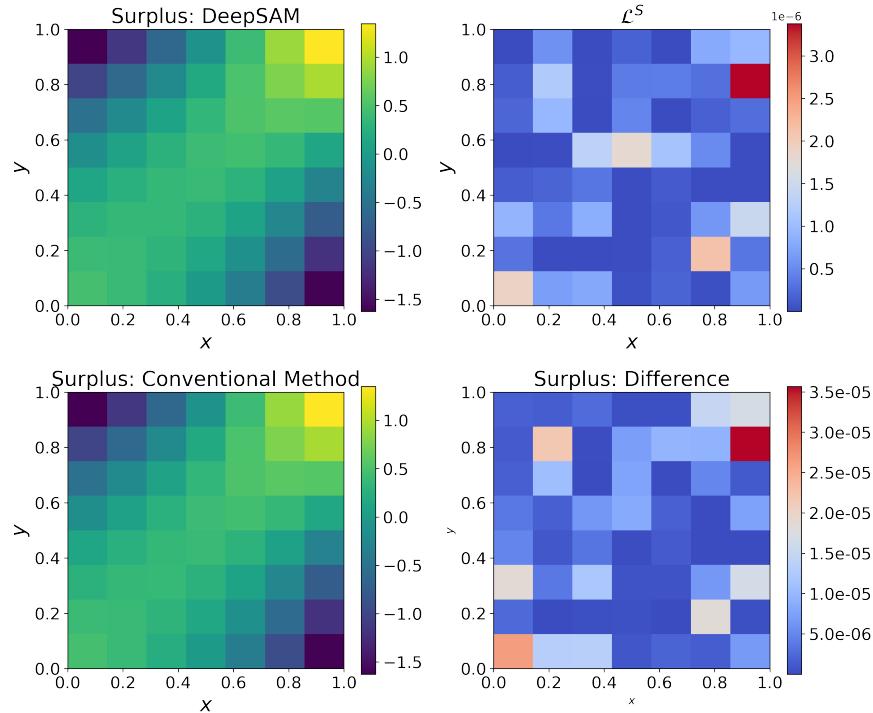


Figure 1: Surplus solution for the labor search model without aggregate shocks, using DeepSAM and conventional method. Figure note: The upper and lower left plots show the surplus solution from the DeepSAM method and conventional method, respectively. The upper right plot shows the value of loss for the master equation at each grid point of the state space. The lower right plot computes the squared difference of the surplus value at each grid point of the state space using the two different methods.

⁵It takes about 57 minutes to solve the problem as a 58-dimensional PDE on an A100 GPU on Google Colab, which is easily accessible to all researchers at <https://colab.research.google.com/signup>. The computation time varies with different calibrations. For example, if we set a relatively small $\kappa = 0.4$, it only takes less than 30 minutes to solve the 58 dimensional PDE. That's because the high dimensional function that is approximated by neural network is flatter in the curvature, which makes it easier to “learn”.

3.3.2 Solution With Aggregate Shocks

For the labor search model with aggregate TFP and separation shocks, we cannot compare it to an existing solution technique because none exists. Instead, we study the error and stability of our neural network approximation.

Table 6 summarizes the average training loss across the training sample. This tells us that the error is small ($\mathcal{O}(10^{-7})$) across the breadth of our sample. Figure 2 focuses on the mean squared error at the ergodic mean. The left and right panels plot the loss as a function of worker and firm types, at $z = z_L$ and $z = z_H$ respectively. We can see the loss is uniformly small ($\mathcal{O}(10^{-7})$ to $\mathcal{O}(10^{-6})$) on all state space of x, y, z .

Table 6: Solve the model with aggregate shock: loss.

Numerical Performance Metric	Value
PDE training loss	9×10^{-7}

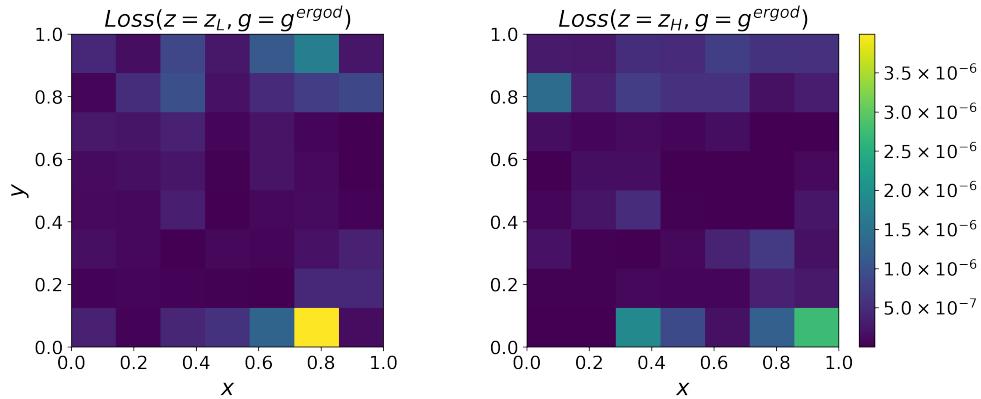


Figure 2: Mean squared loss as a function of worker and firm type.

Figure 3 presents the mean and standard deviation of α^{ergodic} in 15 independent runs of the DeepSAM algorithm. As is shown in the right panel, the standard deviation of α^{ergodic} is of the order of $O(10^{-4})$, which is much smaller than the level of the mean, as well as the level of differences or impact we will discuss below. Low stability is a common difficulty for sampling based PDE training algorithms (they often need to take the average over many runs of the algorithm) so we view these results as very impressive.

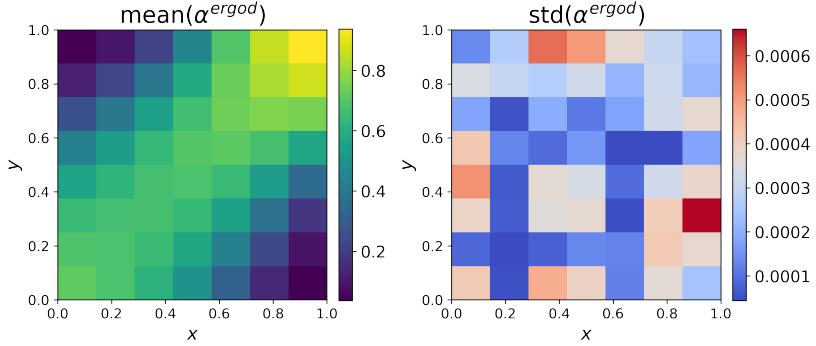


Figure 3: Average and standard deviation of α^{ergodic} across 15 independent runs.

3.4 Distribution Feedback to Aggregate Dynamics

An important advantage of our solution technique is that we have solved for α explicitly as a function of the distribution. In this subsection, we study this relationship and compare model dynamics with and without this distribution feedback.

3.4.1 Relationship Between α and g

Figure 4 shows how α varies as the match distribution varies. The top plots show three employment distributions (from left to right): the ergodic distribution, a distribution with equal employment across types, and a distribution with relatively less employment for low types and relatively more employment for high types. The left bottom plot shows the acceptance function at the ergodic distribution. The other bottom plots show the percentage change in the acceptance function when the distribution changes.

The change in the acceptance function can be understood by considering how the distribution impacts the opportunity cost of waiting for a better match. Moving to a distribution where high types have higher employment rates weakens assortative matching in the acceptance function. This is because the opportunity cost of waiting for a high type agent increases when most high type agents are already employed and so high type workers and firms become less picky. By contrast, moving to a distribution with more equal employment across types leads to more positive assorting in the acceptance function. This is because the more equal distribution increases the likelihood that a weighted will yield a better match.

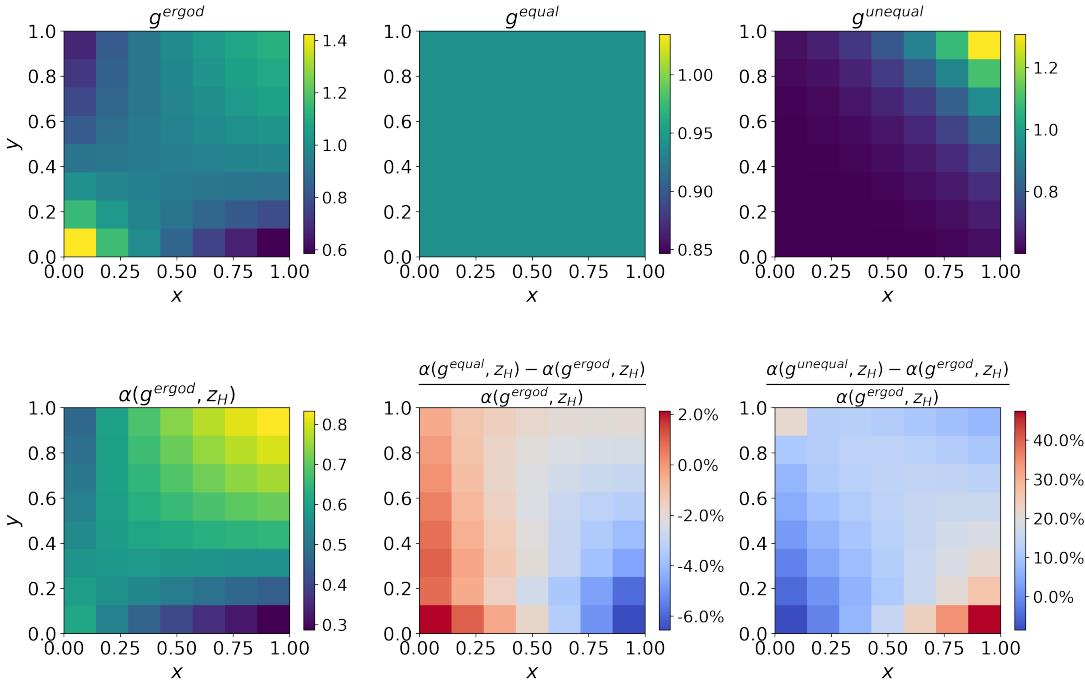


Figure 4: Variation in α as g Varies.

3.4.2 Response to a Labor Market Crisis

We now focus on how the dependence of the acceptance on the distribution changes transition paths in the economy. In order to do this we extend our model to include an additional “depression” state, which occurs with rate 0.01 (once every 100 years). We consider two examples of a “depression” state: a “symmetric” depression where δ increases to 2 for all types and an “asymmetric” depression where δ increases more for low types of workers and firms. In both cases, the model is calibrated so that the model has a 25% unemployment rate after staying in the depression state for 1 year, which is consistent with the US unemployment rate during the Great Depression. Figure 5 shows the ergodic distribution of matches, the distribution of matches that emerges if the economy moves permanently to the symmetric depression, and the distribution of matches that emerges if the economy moves permanently to the asymmetric depression. As can be seen, the symmetric depression state keeps a similar positive assortative pattern to the ergodic steady state but has a lower proportion of matched agents. By contrast, the asymmetric depression state leads to a very different matching pattern where unemployment is concentrated with low types of workers and firms.

For each type of depression, we compute impulse responses for a shock sequence

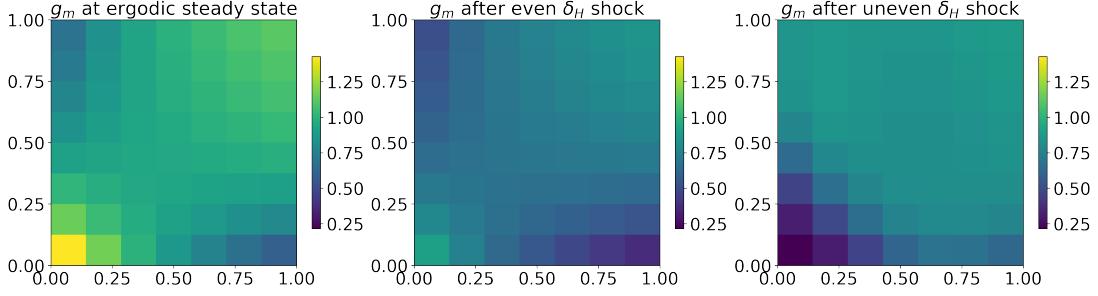


Figure 5: Ergodic distribution and distribution after the “unequal” and “equal” “depression” shocks

where the economy enters the depression state for a year and then reverts to the good state. For each type, we decompose the impulse response in the unemployment rate into two channels:

- (i) the change in unemployment when acceptance is always evaluated at the long-run ergodic employment distribution but otherwise the distribution follows KFE, and
- (ii) the additional change in unemployment when the acceptance function reacts to the changing employment distribution.

We interpret the former as the “restricted” dynamics for the experiment without the additional feedback from the distribution to the acceptance function. Mathematically, under the restricted dynamics, the distribution g_t^R evolves according to:

$$\begin{aligned} \frac{dg_t^R(x, y)}{dt} = & -\delta(x, y, z_t)g_t^R(x, y) \\ & + \frac{m_t(z, \underline{\mathbf{g}}_t^R)}{\mathcal{U}_t(\underline{\mathbf{g}}_t^R)\mathcal{V}_t(\underline{\mathbf{g}}_t^R)}\alpha(x, y, z_t, \underline{\mathbf{g}}^{\text{ergodic}})g_t^{u,R}(x)g_t^{v,R}(y) \end{aligned} \quad (3.1)$$

while, under the full dynamics, the distribution g_t satisfies equation (2.9).

Figure 6 plots the unemployment transition paths for both example shocks. Subplot 6a shows the impulse response for aggregate unemployment following a symmetric shock, and subplot 6b shows the impulse response for aggregate unemployment following an asymmetric shock. On each subplot, the orange dashed line shows the dynamics without the feedback from distribution to agents’ decision, while the solid blue depicts the full dynamics. In both subplots 6a and 6b, the average unemployment rate in the full model increases to 25%. However, there are differences in how closely the model without distribution feedback dynamics can approximate the full model. For the case

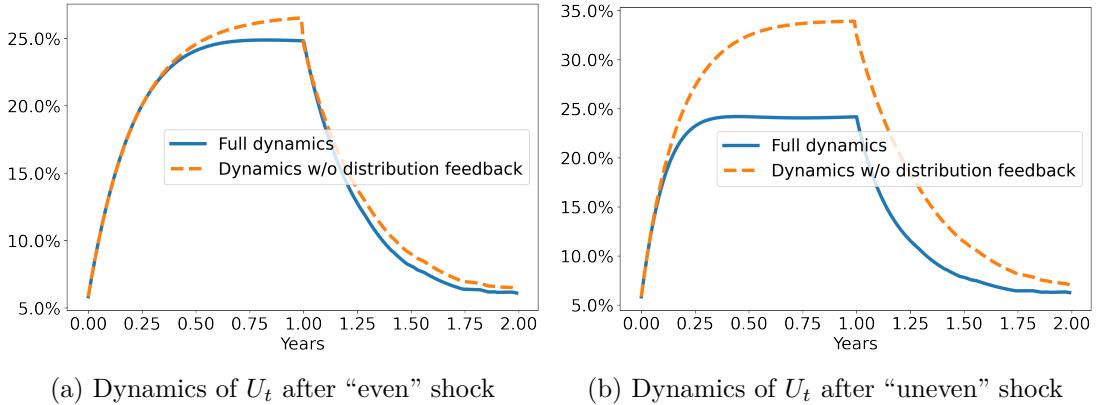


Figure 6: Dynamics of U_t after high separation shocks: full dynamics vs dynamics in which the decision of workers and firms does not rely on the distribution g_t directly as in Equation (3.1). The left panel shows the dynamics after the “even” shock, while the right one corresponds to the “uneven” shock.

of a symmetric unemployment increase, there is little difference between the block recursive dynamics and the full solution. By contrast, for the case of an asymmetric employment shock, the lines diverge and the model without feedback dynamics overshoots peak unemployment by 10 percentage points. The difference arises because the asymmetric shock increases the proportion of unemployed workers who have low type and so increases the opportunity cost of waiting for a high type. This leads workers in the full model to be less “picky” and accept more low type matches compared to workers in the model without distribution feedback. Taken together, our results suggest that the feedback from the distribution to aggregate dynamics in our calibrated labor market model is much stronger when the distribution experiences asymmetric shocks to employment.

3.5 Revisiting “Block Recursivity”

In this subsection, we bring all the discussion from this section together and examine how imposing a block recursive model structure restricts the model dynamics. We do this by comparing the impulse responses to negative TFP shocks in the following two models:

1. Our model solved using the DeepSAM method, and
2. The counterpart “block-recursive” model, which retains all our structural assumptions except that we adopt the [Lise and Robin \(2017\)](#) restriction that worker’s

bargaining power is $\beta = 0$.

The counterpart model is “block recursive” in the sense that agents’ decisions are independent of the endogenous distribution of matches over time. In order to make the comparison consistent, we recalibrate the matching efficiency to $\kappa = 5.7$ for the “block-recursive” model so that it also matches the unemployment rate at the steady state.

The impulse response comparison is presented in Figure 7. We study the dynamics of the unemployment rate following a negative productivity shock that lasts for one year and then return to the stochastic process for aggregate TFP governed by a finite state Markov chain. Evidently, the increase in the unemployment rate in the block recursive model is significantly larger. The intuition for this is that when $\beta = 0$ and firms take all the surplus from meetings, then vacancy creation and so unemployment are more responsive to TFP shocks.

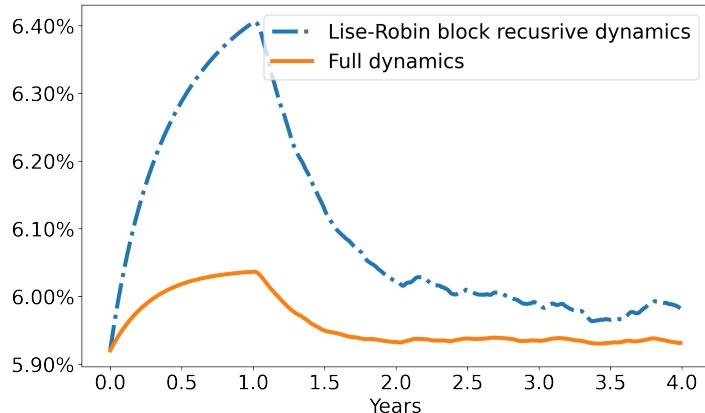


Figure 7: Unemployment following negative TFP Shock: our model solution with DeepSAM vs. block-recursive solution à la [Lise and Robin \(2017\)](#).

4 On-The-Job Search and Cyclical Sorting

In this section, we introduce endogenous job-to-job and job-to-unemployment transitions into the model from Section 3. We use this model to study asymmetric labor market dynamics over the business cycles.

4.1 Environment Changes

We make the following changes to the environment from subsection 3.1.

Search and Matching: All workers now engage in random search. The matching function becomes $m(\mathcal{U}_t + \phi\mathcal{E}_t, \mathcal{V}_t)$ with the interpretation that ϕ is the exogenous relative intensity at which employed workers search. Let $\mathcal{W}_t := \mathcal{U}_t + \phi\mathcal{E}_t$ denote the total mass of searchers. The probabilities that an unemployed or an employed worker meets a potential employer are given by:

$$\mathcal{M}_t^u = \frac{m(\mathcal{W}_t, \mathcal{V}_t)}{\mathcal{W}_t}, \quad \mathcal{M}_t^e = \phi \frac{m(\mathcal{W}_t, \mathcal{V}_t)}{\mathcal{W}_t}$$

while the probability that a vacant firm meets a potential hire is:

$$\mathcal{M}_t^v = \frac{m(\mathcal{W}_t, \mathcal{V}_t)}{\mathcal{V}_t}.$$

Conditional on meeting a worker, we define the probabilities that the worker is unemployed or employed by $\mathcal{C}_u = \frac{\mathcal{U}}{\mathcal{U}+\phi\mathcal{E}}$ and $\mathcal{C}_e = \frac{\phi\mathcal{E}}{\mathcal{U}+\phi\mathcal{E}}$ respectively. The probability for a firm to meet an unemployed worker $x \in X \subset [0, 1]$ equals $\mathcal{M}_v \mathcal{C}_u \int_X g^u(x) dx$. The probability for a firm to meet an employed worker $x \in X \subset [0, 1]$ equals $\mathcal{M}_v \mathcal{C}_e \int_X g^e(x) dx$.

Bargaining: Let $V_t^u(x)$ denote the value of unemployment for a worker of type x . Let $V_t^e(x, y)$ denote the value of worker x employed at a firm of type y . Let $V_t^v(y)$ denote the value of a vacancy for firm y . Let $V_t^p(x, y)$ denote the value of firm y employing an unemployed worker of type x .

As before, the surplus of a match between an unemployed worker and an vacant firm is defined as:

$$S_t^u(x, y) := V_t^p(x, y) - V_t^v(y) + V_t^e(x, y) - V_t^u(x)$$

and the division of surplus is given by:

$$\begin{aligned} \beta S_t^u(x, y) &= V_t^e(x, y) - V_t^u(x) \\ (1 - \beta) S_t^u(x, y) &= V_t^p(x, y) - V_t^v(y) \end{aligned}$$

When a worker employed at some firm of type y meets a firm of type \tilde{y} , the two firms engage in Bertrand competition, as described in Postel-Vinay and Robin (2002). If $S(x, \tilde{y}) > S(x, y)$, then the worker moves to firm \tilde{y} and receives the incumbent em-

ployer's reservation value $(1 - \beta)S_t(x, y)$ (in addition to whatever they had already been promised). If the worker does not move, then their continuation value updates with the surplus of new matches for (x, y) .

Endogenous exit: In addition to allowing workers to search, we also allow matches to breakup. If $S_t(x, y) < 0$, the worker returns to unemployment and the firm returns to posting vacancies.

4.2 Recursive Characterization of Equilibrium

Surplus Differential Equation: In Appendix C we show that the differential equation for the surplus becomes the following:

$$\begin{aligned} \rho S_t(x, y) = & f_t(x, y) - \delta S_t(x, y) - \alpha_t^b(x, y)S_t(x, y) - b \\ & - (1 - \beta) \frac{m(\mathcal{W}_t, \mathcal{V}_t)}{\mathcal{W}_t \mathcal{V}_t} \int \alpha(\tilde{x}, y)S_t(\tilde{x}, y)g_t^u(\tilde{x})d\tilde{x} \\ & - (1 - \beta)\phi \frac{m(\mathcal{W}_t, \mathcal{V}_t)}{\mathcal{W}_t \mathcal{V}_t} \int \alpha_t^p(y, \tilde{x}, \tilde{y})g_t^m(\tilde{x}, \tilde{y})(S_t(\tilde{x}, y) - S_t(\tilde{x}, \tilde{y}))d\tilde{x}d\tilde{y} \\ & + (1 - \beta)\phi \frac{m(\mathcal{W}_t, \mathcal{V}_t)}{\mathcal{W}_t \mathcal{V}_t} \int \alpha_t^e(x, y, \tilde{y})S_t(x, y)g_t^v(\tilde{y})d\tilde{y} \\ & - \beta \frac{m(\mathcal{W}_t, \mathcal{V}_t)}{\mathcal{W}_t \mathcal{V}_t} \int \alpha_t(x, \tilde{y})S_t(x, \tilde{y})g_t^v(\tilde{y})d\tilde{y} \\ & + \lambda(z)(S_t(x, y, \tilde{z}) - S_t(x, y, z)) + \partial_t S_t(y) \end{aligned}$$

where:

$$\begin{aligned} \alpha_t(x, \tilde{y}) &:= \begin{cases} 1, & \text{if } S_t(x, \tilde{y}) > 0 \\ 0, & \text{otherwise} \end{cases} \\ \alpha_t^b(x, \tilde{y}) &:= \begin{cases} 1, & \text{if } S_t(x, \tilde{y}) < 0 \\ 0, & \text{otherwise} \end{cases} \\ \alpha_t^e(x, y, \tilde{y}) &:= \begin{cases} 1, & \text{if } S_t(x, \tilde{y}) \geq S_t(x, y) \geq 0 \\ 0, & \text{otherwise} \end{cases} \\ \alpha_t^p(y, \tilde{x}, \tilde{y}) &:= \begin{cases} 1, & \text{if } S_t(\tilde{x}, y) \geq S_t(\tilde{x}, \tilde{y}) \geq 0 \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

Evidently, the main difference to the differential equation for surplus in Section 3 is that now we have to take into account the possibility of movements from job to job. These jobs potentially have a different acceptance function than the unemployment to

job transitions.

Kolmogorov Forward Equation: Likewise, in Appendix C we show that the measure of matches evolves according to following differential equation:

$$\begin{aligned} dg_t^m(x, y) = & -\delta g_t^m(x, y)dt - \alpha_t^b(x, y)g_t(x, y) \\ & - \phi \frac{m(\mathcal{W}_t, \mathcal{V}_t)}{\mathcal{W}_t \mathcal{V}_t} g_t^m(x, y) \int \alpha_t^e(x, y, \tilde{y}) g_t^v(\tilde{y}) d\tilde{y} dt \\ & + \frac{m(\mathcal{W}_t, \mathcal{V}_t)}{\mathcal{W}_t \mathcal{V}_t} \alpha_t(x, y) g_t^u(x) g_t^v(y) dt \\ & + \phi \frac{m(\mathcal{W}_t, \mathcal{V}_t)}{\mathcal{W}_t \mathcal{V}_t} \int \alpha_t^e(\tilde{x}, \tilde{y}, y) g_t^v(y) \frac{g_t^m(\tilde{x}, \tilde{y})}{\mathcal{E}_t} d\tilde{x} d\tilde{y} dt \end{aligned}$$

If we know measure of matches, then we can recover the other distribution:

$$\begin{aligned} g_t^e(x) &= \int g_t^m(x, y) dy, & g_t^u(x) &= g_t^w(x) - \int g_t^m(x, y) dy, \\ g_t^p(y) &= \int g_t^m(x, y) dx, & g_t^v(y) &= g_t^f(y) - \int g_t^m(x, y) dx, \\ \mathcal{U}_t &= \int g_t^u(x) dx, & \mathcal{V}_t &= \int g_t^v(y) dy \\ \mathcal{E}_t &= \int g_t^e(x) dx, & \mathcal{P}_t &= \int g_t^p(y) dy \end{aligned}$$

4.3 Working Bargaining Power Influences Assortative Matching

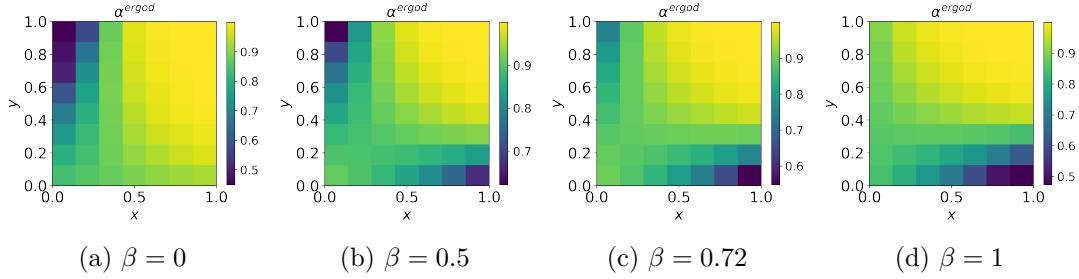


Figure 8: Sorting at the ergodic distribution for different worker bargaining power β . This figure presents the acceptance decision $\alpha(x, y)$ for worker x and firm y at the ergodic distribution of the economies with different β .

Unlike Lise and Robin (2017) and subsequent papers, our solution relaxes the assumption that firms have all the bargaining power in the market between unemployed workers and firms (i.e. they assume $\beta = 0$). This allows us to study the interaction between bargaining power and sorting. Figure 8 shows how the ergodic acceptance

function varies with worker bargaining power. Evidently, firm bargaining power significantly skews the assortative matching dynamics with high firm bargaining power making high type firms less picky and high worker bargaining power making high type workers less picky. This illustrates that with a “standard production” function we cannot get the empirical assortative matching patterns with the $\beta = 0$ imposed in the “block-recursive” random search models. Instead, we would need to customize the production function to offset the skew in the acceptance and get dynamics similar to cases with $\beta \approx 0.5$.

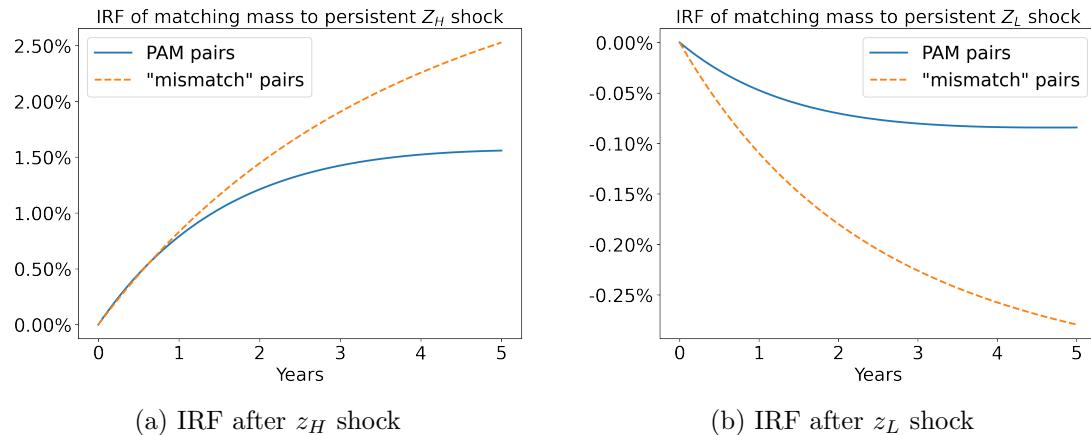


Figure 9: IRF of the total mass of different pairs after a persistent productivity shock in the on-the-job search model with aggregate shocks. Figure note: The blue solid line shows the relative change of the total mass of “PAM pairs” compared to that in the ergodic distribution. The yellow dotted line shows the relative change of the total mass of “mismatch pairs”.

4.4 Revisiting Cyclical Sorting with On-the-job Search

In this section, we study cyclical sorting in our model with on-the-job search and aggregate shocks. We call a pair of match (x, y) as a “PAM pair” if $|x - y| \leq \frac{1}{2}$, and otherwise as a “mismatch pair”. Figure 9 shows relative changes (compared to that in the ergodic distribution) in the total mass of “PAM pairs” and “mismatch pairs” over time when persistent positive or negative productivity shocks hit the economy. There is a strong pattern of countercyclical sorting over the business cycles. In the expansion, the mass of “mismatch pairs” grows faster than that of “PAM pairs”, which means workers and firms with very different types may accept each other, weakening the positive assortative matching. In the recession, the mass of “mismatch pairs” drops faster than that of “PAM pairs”, which means workers and firms become more “picky”,

amplifying the positive assortative matching. This is consistent with the cleansing effect of business cycles (Caballero and Hammour, 1994). We discuss further sorting patterns in the ergodic distribution in Appendix B.3.

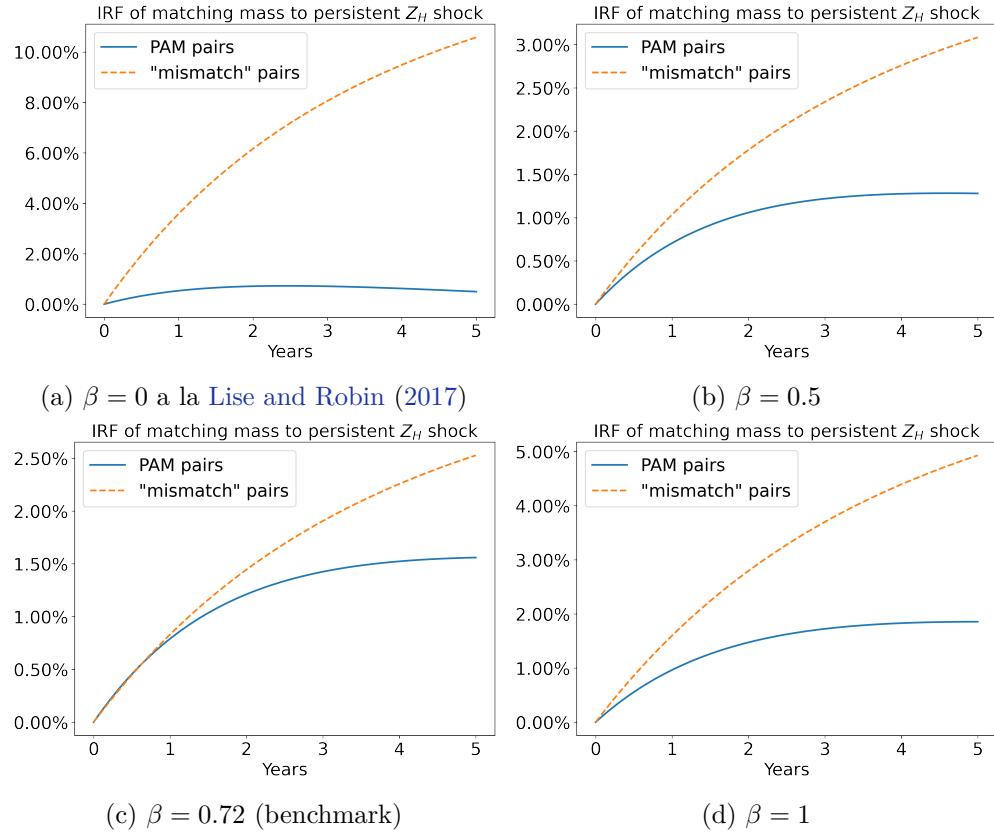


Figure 10: IRF of the total mass of different pairs after a persistent productivity shock in the on-the-job search model with aggregate shocks for different β .

Figure 10 shows how cyclical assortative matching varies with the worker bargaining power β . The cyclical sorting dynamics are amplified when β is closer to the extreme values of 0 or 1. When $\beta = 0.72$, as calibrated in the business cycle literature such as Shimer (2005), the cyclical sorting dynamics are approximately 75% smaller than in an economy with $\beta = 0$, as assumed by Lise and Robin (2017). The intuition is that when one side of the market has all the bargaining power their decisions end up being more responsive to changes in the opportunity cost of waiting. This means the classic assumption about take-it-or-leave-it offers following Postel-Vinay and Robin (2002) has important implications on the model dynamics.

5 Conclusion

In this paper, we developed a new method for characterizing global solutions to search and matching models with aggregate shocks and heterogeneous agents. This allowed us to study dynamics in models where agent decisions depend upon the distribution and so the model is not “block-recursive”. We showed block recursivity is a reasonable assumption when shocks are symmetric across the distribution but becomes inaccurate when shocks have an asymmetric impact. We then used our model to study how business cycles have a “cleansing effect” by increasing the assortative matching in the economy. We believe our methodology is a major breakthrough in our understanding of search and matching dynamics with potential applications in the labor, finance, and spatial literature.

References

- Ahn, S., Kaplan, G., Moll, B., Winberry, T., and Wolf, C. (2018). When inequality matters for macro and macro matters for inequality. *NBER macroeconomics annual*, 32(1):1–75.
- Alvarez, F., Lippi, F., and Souganidis, P. (2023). Price setting with strategic complementarities as a mean field game. *Econometrica*, 91(6):2005–2039.
- Alves, F. (2022). Job ladder and business cycles. Technical report, Bank of Canada.
- Azinovic, M., Gaegau, L., and Scheidegger, S. (2022). Deep equilibrium nets. *International Economic Review*, 63(4):1471–1525.
- Bilal, A. (2023). Solving heterogeneous agent models with the master equation. Technical report, National Bureau of Economic Research.
- Caballero, R. J. and Hammour, M. L. (1994). The cleansing effect of recessions. *American Economic Review*, 84(5):1350–1368.
- Cahuc, P., Postel-Vinay, F., and Robin, J.-M. (2006). Wage bargaining with on-the-job search: Theory and evidence. *Econometrica*, 74(2):323–364.
- Cardaliaguet, P., Delarue, F., Lasry, J.-M., and Lions, P.-L. (2015). The master equation and the convergence problem in mean field games. *arXiv*.
- Duarte, V. (2018). Machine learning for continuous-time economics. Available at SSRN 3012602.
- Engbom, N. (2021). Contagious unemployment. Technical report, National Bureau of Economic Research.
- Fernández-Villaverde, J., Hurtado, S., and Nuno, G. (2023). Financial frictions and the wealth distribution. *Econometrica*, 91(3):869–901.
- Gopalakrishna, G. (2021). Aliens and continuous time economies. *Swiss Finance Institute Research Paper*, (21-34).
- Gu, Z., Lauriere, M., Merkel, S., and Payne, J. (2023). Deep learning solutions to master equations for continuous time heterogeneous agent macroeconomic models. Technical report.
- Hagedorn, M., Law, T. H., and Manovskii, I. (2017). Identifying equilibrium models of labor market sorting. *Econometrica*, 85(1):29–65.
- Han, J., Yang, Y., and E, W. (2021). DeepHAM: A global solution method for heterogeneous agent models with aggregate shocks. *arXiv preprint arXiv:2112.14377*.

- Huang, J. (2022). A probabilistic solution to high-dimensional continuous-time macrofinance models. *Available at SSRN 4122454*.
- Kahou, M. E., Fernández-Villaverde, J., Perla, J., and Sood, A. (2021). Exploiting symmetry in high-dimensional dynamic programming. Technical report, National Bureau of Economic Research.
- Kaplan, G., Moll, B., and Violante, G. (2018). Monetary Policy According to HANK. *American Economic Review*, 108(3):697–743.
- Krusell, P., Mukoyama, T., Rogerson, R., and Sahin, A. (2017). Gross worker flows over the business cycle. *American Economic Review*, 107(11):3447–3476.
- Krusell, P. and Smith, A. A. (1998). Income and Wealth Heterogeneity in the Macroeconomy. *Journal of Political Economy*, 106(5):867–896.
- Lise, J. and Robin, J.-M. (2017). The macrodynamics of sorting between workers and firms. *American Economic Review*, 107(4):1104–1135.
- Maliar, L., Maliar, S., and Winant, P. (2021). Deep learning for solving dynamic economic models. *Journal of Monetary Economics*, 122:76–101.
- Menzio, G. and Shi, S. (2010). Block recursive equilibria for stochastic models of search on the job. *Journal of Economic Theory*, 1.
- Menzio, G. and Shi, S. (2011). Efficient Search on the Job and the Business Cycle. *Journal of Political Economy*, 119(3):468–510.
- Moen, E. R. (1997). Competitive Search Equilibrium. *Journal of Political Economy*, 105(2):385–411.
- Mortensen, D. T. and Pissarides, C. A. (1994). Job creation and job destruction in the theory of unemployment. *The review of economic studies*, 61(3):397–415.
- Moscarini, G. and Postel-Vinay, F. (2018). The cyclical job ladder. *Annual Review of Economics*, 10:165–188.
- Petrosky-Nadeau, N. and Zhang, L. (2017). Solving the diamond–mortensen–pissarides model accurately. *Quantitative Economics*, 8(2):611–650.
- Postel-Vinay, F. and Robin, J.-M. (2002). Equilibrium wage dispersion with worker and employer heterogeneity. *Econometrica*, 70(6):2295–2350.
- Qiu, X. (2023). Vacant jobs.
- Sauzet, M. (2021). Projection methods via neural networks for continuous-time models. *Available at SSRN 3981838*.

- Schaal, E. (2017). Uncertainty, Productivity and Unemployment in the Great Recession. *Econometrica*, 85(6):250–274.
- Shimer, R. (2005). The cyclical behavior of equilibrium unemployment and vacancies. *American economic review*, 95(1):25–49.
- Shimer, R. and Smith, L. (2000). Assortative matching and search. *Econometrica*, 68(2):343–369.

A Master Equation with the Free Entry Condition

As in Section 2.2.6, the free entry condition is

$$c = \int V_t^v(\tilde{y}) d\tilde{y}$$

Recall from (2.2) the HJB equation for a vacant institution with productivity y is

$$\rho V_t^v(y) = \mathcal{M}_t^v \int \alpha(\tilde{x}, y) \frac{g_t^u(\tilde{x})}{\mathcal{U}_t} (1 - \beta) S_t(\tilde{x}, y) d\tilde{x} + \partial_t V_t^v(y)$$

Integrating and combining these equations, we have that:

$$\begin{aligned} \rho \int V_t^v(\tilde{y}) d\tilde{y} &= \mathcal{M}_t^v \int \int \alpha(\tilde{x}, \tilde{y}) \frac{g_t^u(\tilde{x})}{\mathcal{U}_t} (1 - \beta) S_t(\tilde{x}, \tilde{y}) d\tilde{x} d\tilde{y} + \partial_t \int V_t^v(\tilde{y}) d\tilde{y} \\ \Rightarrow \rho c &= \frac{m(\mathcal{U}_t, \mathcal{V}_t)}{\mathcal{V}_t} \int \int \alpha(\tilde{x}, \tilde{y}) \frac{g_t^u(\tilde{x})}{\mathcal{U}_t} (1 - \beta) S_t(\tilde{x}, \tilde{y}) d\tilde{x} d\tilde{y} \\ \Rightarrow \frac{m(\mathcal{U}_t, \mathcal{V}_t)}{\mathcal{V}_t} &= \frac{\rho c}{\int \int \alpha(\tilde{x}, \tilde{y}) \frac{g_t^u(\tilde{x})}{\mathcal{U}_t} (1 - \beta) S_t(\tilde{x}, \tilde{y}) d\tilde{x} d\tilde{y}} \end{aligned}$$

Assuming that $m(\mathcal{U}_t, \mathcal{V}_t)/(\mathcal{V}_t) = \hat{m}(\mathcal{V}_t/\mathcal{U}_t)$, we have that:

$$\mathcal{V}_t = \mathcal{U}_t \hat{m}^{-1} \left(\frac{\rho c}{\int \int \alpha(\tilde{x}, \tilde{y}) \frac{g_t^u(\tilde{x})}{\mathcal{U}_t} (1 - \beta) S_t(\tilde{x}, \tilde{y}) d\tilde{x} d\tilde{y}} \right)$$

where $g_t^u = g_t^w - \int g_t^m(x, y) dy$ and so the RHS can be computed from g_t^m and S_t . (For example, if $m(\mathcal{U}, \mathcal{V}) = \kappa \mathcal{U}^\nu \mathcal{V}^{1-\nu}$, then $\mathcal{M}_t^v = m(\mathcal{U}, \mathcal{V})/\mathcal{V} = \hat{m}(\mathcal{V}_t/\mathcal{U}_t) = \kappa(\mathcal{U}/\mathcal{V})^\nu$ and $\mathcal{M}_t^u = m(\mathcal{U}, \mathcal{V})/\mathcal{U} = \kappa(\mathcal{V}/\mathcal{U})^{1-\nu}$). Since firm y draws are uniformly distributed, we have that g_t^f is given by:

$$\begin{aligned} g_t^f &= \mathcal{V}_t + \mathcal{P}_t \\ &= \mathcal{U}_t \hat{m}^{-1} \left(\frac{\rho c}{\int \int \alpha(\tilde{x}, \tilde{y}) (g_t^u(\tilde{x})/\mathcal{U}_t) (1 - \beta) S_t(\tilde{x}, \tilde{y}) d\tilde{x} d\tilde{y}} \right) + \int \int g_t^m(x, y) dy dx \end{aligned}$$

where \mathcal{V}_t is from (2.7), $\mathcal{P}_t = \int \int g_t^m(x, y) dy dx$, and $\mathcal{U}_t = \int (g_t^w(x) - \int g_t^m(x, y) dy)$. This means that g_t^f can be computed from g_t^m and S_t . Finally, this means that

$$\begin{aligned} g_t^v(y) &= g_t^f(y) - g_t^p(y) \\ &= \mathcal{U}_t \hat{m}^{-1} \left(\frac{\rho c}{\int \int \alpha(\tilde{x}, \tilde{y})(g_t^u(\tilde{x})/\mathcal{U}_t)(1-\beta)S_t(\tilde{x}, \tilde{y}) d\tilde{x} d\tilde{y}} \right) + \int \int g_t^m(x, y) dy dx \\ &\quad - \int g_t^m(x, y) dx \end{aligned}$$

We can now calculate the differential equation for surplus:

$$\begin{aligned} \rho S_t(x, y) &= \rho(V_t^p(x, y) - V_t^v(y) + V_t^e(x, y) - V_t^u(x)) \\ &= f_t(x, y) - w_t(x, y) - \delta(1-\beta)S_t(x, y) + \partial_t V_t^p(x, y) \\ &\quad - \left(\mathcal{M}_t^v \int \alpha(\tilde{x}, y) \frac{g_t^u(\tilde{x})}{U_t} (1-\beta)S_t(\tilde{x}, y) d\tilde{x} + \partial_t V_t^v(y) \right) \\ &\quad + w_t(x, y) - \beta\delta S_t(x, y) + \partial_t V_t^e(x, y) \\ &\quad - \left(b + \mathcal{M}_t^u \int \alpha_t(x, \tilde{y}) \frac{g_t^v(\tilde{y})}{V_t} \beta S_t(x, \tilde{y}) d\tilde{y} + \partial_t V_t^u(x) \right) \\ &= f_t(x, y) - \delta S_t(x, y) - \mathcal{M}_t^v \int \alpha(\tilde{x}, y) \frac{g_t^u(\tilde{x})}{U_t} (1-\beta)S_t(\tilde{x}, y) d\tilde{x} \\ &\quad - b - \mathcal{M}_t^u \int \alpha_t(x, \tilde{y}) \frac{g_t^v(\tilde{y})}{V_t} \beta S_t(x, \tilde{y}) d\tilde{y} + \partial_t S_t(x, y) \end{aligned}$$

where:

$$\begin{aligned} \mathcal{V}_t &= \mathcal{U}_t \hat{m}^{-1} \left(\frac{\rho c}{\int \int \alpha(\tilde{x}, \tilde{y}) \frac{g_t^u(\tilde{x})}{U_t} (1-\beta)S_t(\tilde{x}, \tilde{y}) d\tilde{x} d\tilde{y}} \right) \\ \mathcal{M}_t^v &= m(\mathcal{U}, \mathcal{V})/\mathcal{V} = \hat{m}(\mathcal{V}_t/\mathcal{U}_t) = \kappa(\mathcal{U}/\mathcal{V})^\nu, \\ \mathcal{M}_t^u &= m(\mathcal{U}, \mathcal{V})/\mathcal{U} = \kappa(\mathcal{V}/\mathcal{U})^{1-\nu} \\ g_t^v(y) &= g_t^f(y) - g_t^p(y) \\ &= \mathcal{V}_t + \mathcal{P}_t - g_t^p(y) \\ &= \mathcal{U}_t \hat{m}^{-1} \left(\frac{\rho c}{\int \int \alpha(\tilde{x}, \tilde{y}) (g_t^u(\tilde{x})/\mathcal{U}_t)(1-\beta)S_t(\tilde{x}, \tilde{y}) d\tilde{x} d\tilde{y}} \right) + \int \int g_t^m(x, y) dy dx - \int g_t^m(x, y) dx \end{aligned}$$

and the KFE is in the same form as (2.4) with different definitions of $g^f(y)$ and \mathcal{V} .

B Appendix For the Labor Search Model in Section 3

B.1 Master equation and loss function for the model without aggregate shocks

To verify the accuracy of the DeepSAM method, we apply it to solve a labor search model without aggregate shocks, which can also be solved with a conventional numerical method such as that in Hagedorn et al. (2017). The master equation and loss function for the Surplus is given by:

$$\begin{aligned} 0 = \mathcal{L}^S S &= -(\rho + \delta(x, y))S(x, y, \underline{\mathbf{g}}) + F(x, y) - b \\ &\quad - (1 - \beta) \frac{m(\underline{\mathbf{g}})}{\mathcal{U}(\underline{\mathbf{g}})\mathcal{V}(\underline{\mathbf{g}})} \frac{1}{n_x} \sum_{i=1}^{n_x} \alpha(x_i, y, \underline{\mathbf{g}})S(x_i, y, g)\underline{\mathbf{g}}^u(x_i) \\ &\quad - \beta \frac{m(\underline{\mathbf{g}})}{\mathcal{U}(\underline{\mathbf{g}})\mathcal{V}(\underline{\mathbf{g}})} \frac{1}{n_y} \sum_{j=1}^{n_y} \alpha(x, y_j, \underline{\mathbf{g}})S(x, y_j, \underline{\mathbf{g}})\underline{\mathbf{g}}^v(y_j) \\ &\quad + \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \partial_{g_{ij}} S(x, y, \underline{\mathbf{g}}) \mu^g(x_i, y_j, \underline{\mathbf{g}}) \end{aligned}$$

in which

$$\begin{aligned} dg_t(x, y)/dt &= \mu^g(x, y, \underline{\mathbf{g}}_t) = -\delta(x, y)g_t(x, y) + \frac{m(\underline{\mathbf{g}}_t)}{\mathcal{U}_t\mathcal{V}_t} \alpha(x, y, \underline{\mathbf{g}}_t) \\ &\quad \times \left(g^w(x) - \frac{1}{n_y} \sum_{j=1}^{n_y} \underline{\mathbf{g}}_t(x, y_j) \right) \left(g^f(y) - \frac{1}{n_x} \sum_{i=1}^{n_x} \underline{\mathbf{g}}_t(x_i, y) \right) \end{aligned}$$

and $\alpha(x, y, g)$ is given by:

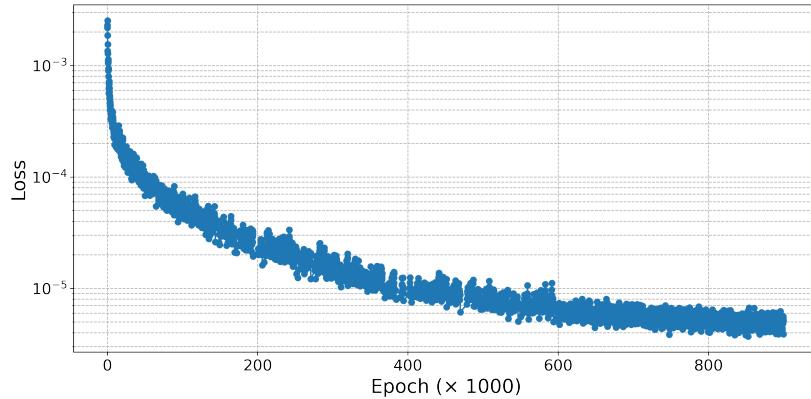
$$\alpha(x, y, \underline{\mathbf{g}}) = \left(1 + e^{-\xi S(x, y, \underline{\mathbf{g}})} \right)^{-1}.$$

B.2 Numerical method and performance

B.2.1 Hyperparameters for the neural networks

Training loss, learning rate, and sample size. Figure 11 presents the value of the loss function (2.10) along the training process. It takes 1.5 hours on an A100 GPU for the neural network to converge to a stable solution. The learning rate is 10^{-4} for the first 400,000 epoch, is 10^{-5} after that. Sample size: 256 in first 400k, 512 from after that. We use a cosine scheme for to adjust the learning rate over time.

Figure 11: Loss function along training epochs



B.3 Additional results

Figure 12 compares the ergodic distribution of the problem with aggregate shocks with the deterministic steady state of the problem in the absence of aggregate shocks. We find on average, the aggregate shocks destroy matches between low productivity workers (firms) and high productivity firms (workers) and amplify the positive assortative matching. Using α as an example, the differences in the upper right panel are an order of magnitude larger than the standard deviation of α^{ergodic} in Figure 3, which means our results are credible in a statistical sense.

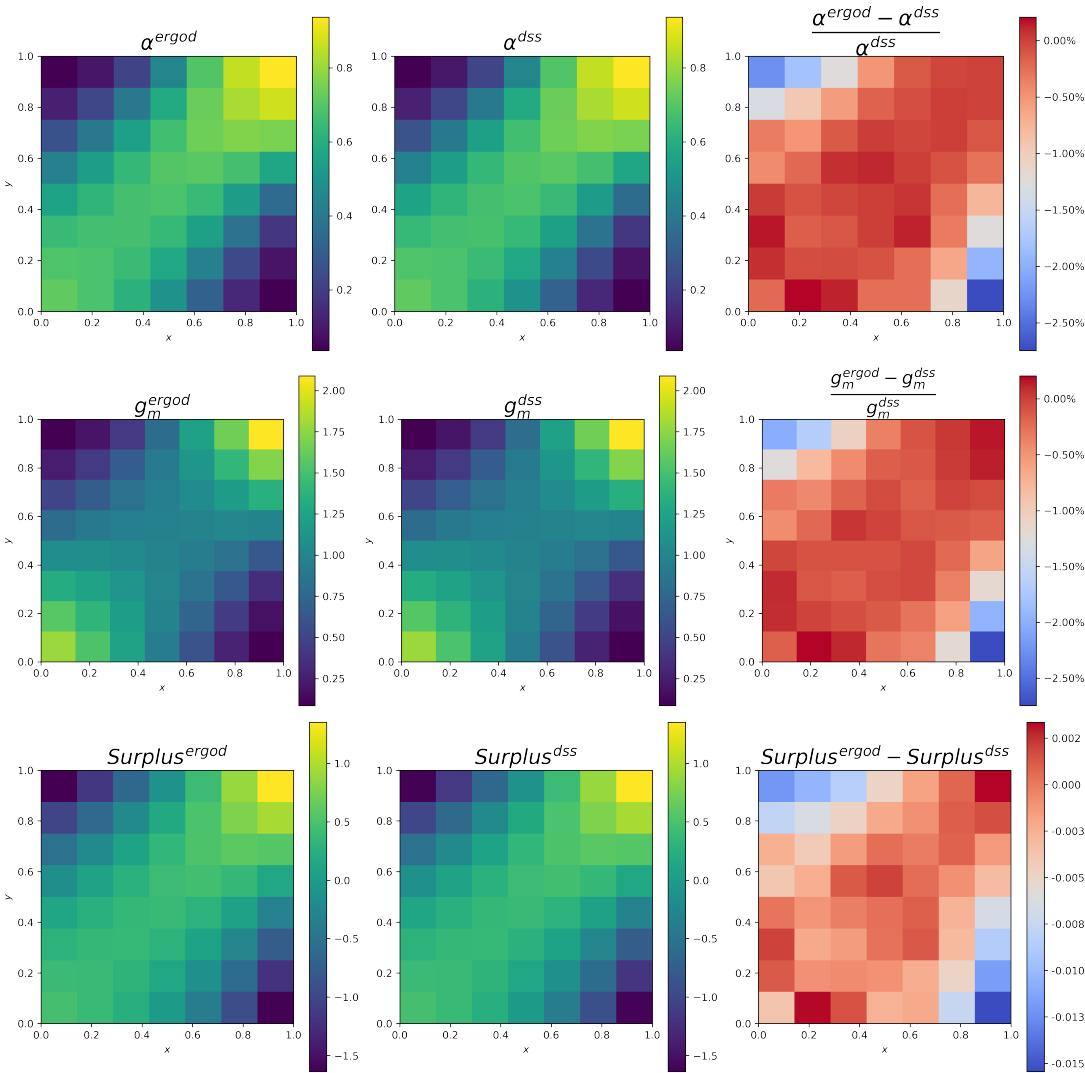


Figure 12: Ergodic mean vs deterministic steady state

Figure 13 provides a more closer look into the first row of Figure 12.

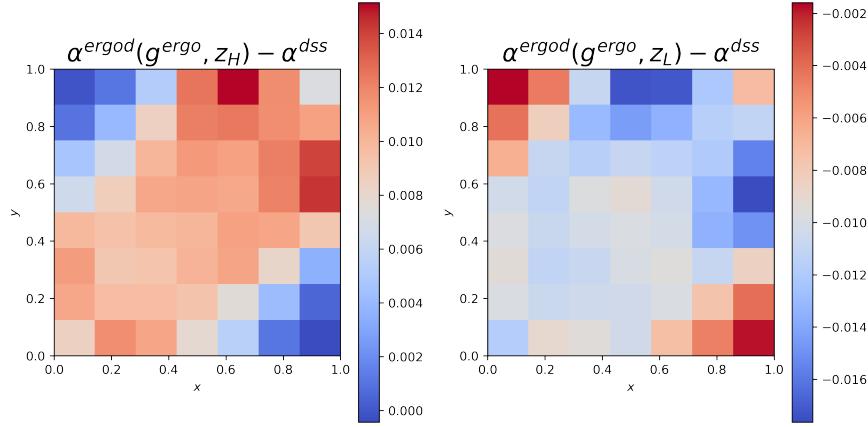


Figure 13: Ergodic α vs deterministic steady state

C Appendix For the On-the-job Search Model

C.1 Agent Problems

Unemployed Workers: The HJB equation for an unemployed worker of type x is the same as in the model without on-the-job search:

$$\begin{aligned} \rho V_t^u(x) &= b + \mathcal{M}_t^u \int \alpha_t(x, \tilde{y})(V_t^e(x, \tilde{y}) - V_t^u(x)) \frac{g_t^v(\tilde{y})}{\mathcal{V}_t} d\tilde{y} + \partial_t V_t^u(x) \\ &= b + \mathcal{M}_t^u \int \alpha_t(x, \tilde{y}) \beta S_t(x, \tilde{y}) \frac{g_t^v(\tilde{y})}{\mathcal{V}_t} d\tilde{y} + \partial_t V_t^u(x) \end{aligned}$$

where:

$$\alpha_t(x, \tilde{y}) := \begin{cases} 1, & \text{if } S_t(x, \tilde{y}) > 0 \\ 0, & \text{otherwise} \end{cases}$$

Employed Workers: The HJB equation for an employed worker of type x matched with type y becomes:

$$\begin{aligned} \rho V_t^e(x, y) &= w_t(x, y) + \mathcal{M}_t^e \int \alpha_t^e(x, y, \tilde{y})(1 - \beta) S_t(x, y) \frac{g_t^v(\tilde{y})}{\mathcal{V}_t} d\tilde{y} \\ &\quad + \delta(V_t^u(x) - V_t^e(x, y)) + \partial_t V_t^e(x, y) \\ &= w_t(x, y) + \mathcal{M}_t^e \int \alpha_t^e(x, y, \tilde{y})(1 - \beta) S_t(x, y) \frac{g_t^v(\tilde{y})}{\mathcal{V}_t} d\tilde{y} \\ &\quad - \beta \delta S_t(x, y) + \partial_t V_t^e(x, y) \end{aligned}$$

where we have used that $S_t(x, y) = V_t^u(x) - V_t^e(x, y)$ because the surplus is indexed to that of new hires and where:

$$\alpha_t^e(x, y, \tilde{y}) := \begin{cases} 1, & \text{if } S_t(x, \tilde{y}) \geq S_t(x, y) \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

Vacant Firms: The HJB equation for a vacant firm is:

$$\begin{aligned} \rho V_t^v(y) &= \mathcal{M}_v \mathcal{C}_u \int \alpha(\tilde{x}, y)(V_t^p(\tilde{x}, y) - V_t^v(x)) \frac{g_t^u(\tilde{x})}{\mathcal{U}_t} d\tilde{x} \\ &\quad + \mathcal{M}_v \mathcal{C}_e \int \alpha_t^p(y, \tilde{x}, \tilde{y}) \frac{g_m(\tilde{x}, \tilde{y})}{\mathcal{E}_t} (1 - \beta)(S_t(\tilde{x}, y) - S_t(\tilde{x}, \tilde{y})) d\tilde{x} d\tilde{y} \\ &\quad + \partial_t V_t^v(y) \\ &= \mathcal{M}_v \mathcal{C}_u \int \alpha(\tilde{x}, y)(1 - \beta)S_t(\tilde{x}, y) \frac{g_t^u(\tilde{x})}{\mathcal{U}_t} d\tilde{x} \\ &\quad + \mathcal{M}_v \mathcal{C}_e \int \alpha_t^p(y, \tilde{x}, \tilde{y}) \frac{g_m(\tilde{x}, \tilde{y})}{\mathcal{E}_t} (1 - \beta)(S_t(\tilde{x}, y) - S_t(\tilde{x}, \tilde{y})) d\tilde{x} d\tilde{y} \\ &\quad + \partial_t V_t^v(y) \end{aligned}$$

where

$$\alpha_t^p(y, \tilde{x}, \tilde{y}) := \begin{cases} 1, & \text{if } S_t(\tilde{x}, y) \geq S_t(\tilde{x}, \tilde{y}) \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

Producing Firms: The HJB equation for a producing firm becomes:

$$\begin{aligned} \rho V_t^p(x, y) &= f_t(x, y) - w_t(x, y) + \delta(V_t^v(x) - V_t^p(x, y)) + \partial_t V_t^p(x, y) \\ &= f_t(x, y) - w_t(x, y) - \delta(1 - \beta)S_t(x, y) + \partial_t V_t^p(x, y) \end{aligned}$$

if the division of future surplus from continuing the match is the same as the division of surplus for new matches.

Master Equation For Surplus: we show that the differential equation for the surplus

becomes the following:

$$\begin{aligned}
\rho S_t(x, y) = & f_t(x, y) - \delta S_t(x, y) - \alpha_t^b(x, y) S_t(x, y) - b \\
& - \mathcal{M}_t^v \mathcal{C}_t^u \int \alpha(\tilde{x}, y)(1 - \beta) S_t(\tilde{x}, y) \frac{g_t^u(\tilde{x})}{\mathcal{U}_t} d\tilde{x} \\
& - \mathcal{M}_t^v \mathcal{C}_t^e \int \alpha_t^p(y, \tilde{x}, \tilde{y}) \frac{g_t^m(\tilde{x}, \tilde{y})}{\mathcal{E}_t} (1 - \beta) (S_t(\tilde{x}, y) - S_t(\tilde{x}, \tilde{y})) d\tilde{x} d\tilde{y} \\
& + \mathcal{M}_t^e \int \alpha_t^e(x, y, \tilde{y}) (1 - \beta) S_t(x, y) \frac{g_t^v(\tilde{y})}{\mathcal{V}_t} d\tilde{y} \\
& - \mathcal{M}_t^u \int \alpha_t(x, \tilde{y}) \beta S_t(x, \tilde{y}) \frac{g_t^v(\tilde{y})}{\mathcal{V}_t} d\tilde{y} \\
& + \lambda(z) (S_t(x, y, \tilde{z}) - S_t(x, y, z)) + \partial_t S_t(y)
\end{aligned}$$

where:

$$\begin{aligned}
\alpha_t(x, \tilde{y}) &:= \begin{cases} 1, & \text{if } S_t(x, \tilde{y}) > 0 \\ 0, & \text{otherwise} \end{cases} \\
\alpha_t^b(x, \tilde{y}) &:= \begin{cases} 1, & \text{if } S_t(x, \tilde{y}) < 0 \\ 0, & \text{otherwise} \end{cases} \\
\alpha_t^e(x, y, \tilde{y}) &:= \begin{cases} 1, & \text{if } S_t(x, \tilde{y}) \geq S_t(x, y) \geq 0 \\ 0, & \text{otherwise} \end{cases} \\
\alpha_t^p(y, \tilde{x}, \tilde{y}) &:= \begin{cases} 1, & \text{if } S_t(\tilde{x}, y) \geq S_t(\tilde{x}, \tilde{y}) \geq 0 \\ 0, & \text{otherwise} \end{cases}
\end{aligned}$$

Observe that:

$$\begin{aligned}
\frac{\mathcal{M}_t^v \mathcal{C}_t^u}{\mathcal{U}_t} &= \frac{m(\mathcal{W}_t, \mathcal{V}_t)}{\mathcal{W}_t \mathcal{V}_t}, & \frac{\mathcal{M}_t^v \mathcal{C}_t^e}{\mathcal{E}_t} &= \phi \frac{m(\mathcal{W}_t, \mathcal{V}_t)}{\mathcal{W}_t \mathcal{V}_t}, \\
\frac{\mathcal{M}_t^e}{\mathcal{V}_t} &= \phi \frac{m(\mathcal{W}_t, \mathcal{V}_t)}{\mathcal{W}_t \mathcal{V}_t}, & \frac{\mathcal{M}_t^u}{\mathcal{V}_t} &= \frac{m(\mathcal{W}_t, \mathcal{V}_t)}{\mathcal{W}_t \mathcal{V}_t}
\end{aligned}$$

and so the surplus equation becomes:

$$\begin{aligned}
\rho S_t(x, y) = & f_t(x, y) - \delta S_t(x, y) - \alpha_t^b(x, y) S_t(x, y) - b \\
& - (1 - \beta) \frac{m(\mathcal{W}_t, \mathcal{V}_t)}{\mathcal{W}_t \mathcal{V}_t} \int \alpha(\tilde{x}, y) S_t(\tilde{x}, y) g_t^u(\tilde{x}) d\tilde{x} \\
& - (1 - \beta) \phi \frac{m(\mathcal{W}_t, \mathcal{V}_t)}{\mathcal{W}_t \mathcal{V}_t} \int \alpha_t^p(y, \tilde{x}, \tilde{y}) g_t^m(\tilde{x}, \tilde{y}) (S_t(\tilde{x}, y) - S_t(\tilde{x}, \tilde{y})) d\tilde{x} d\tilde{y} \\
& + (1 - \beta) \phi \frac{m(\mathcal{W}_t, \mathcal{V}_t)}{\mathcal{W}_t \mathcal{V}_t} \int \alpha_t^e(x, y, \tilde{y}) S_t(x, y) g_t^v(\tilde{y}) d\tilde{y} \\
& - \beta \frac{m(\mathcal{W}_t, \mathcal{V}_t)}{\mathcal{W}_t \mathcal{V}_t} \int \alpha_t(x, \tilde{y}) S_t(x, \tilde{y}) g_t^v(\tilde{y}) d\tilde{y} \\
& + \lambda(z) (S_t(x, y, \tilde{z}) - S_t(x, y, z)) + \partial_t S_t(y)
\end{aligned}$$

Kolmogorov Forward Equation: The measure of matches evolves according to following differential equation:

$$\begin{aligned}
dg_t^m(x, y) = & - \delta g_t^m(x, y) dt - \alpha_t^b(x, y) S_t(x, y) - \mathcal{M}_t^e g_t^m(x, y) \int \alpha_t^e(x, y, \tilde{y}) \frac{g_t^v(\tilde{y})}{\mathcal{V}_t} d\tilde{y} dt \\
& + \mathcal{M}_t^u g_t^u(x) \alpha_t(x, y) \frac{g_t^v(y)}{\mathcal{V}_t} dt + \mathcal{M}_t^e \int \alpha_t^e(\tilde{x}, \tilde{y}, y) \frac{g_t^v(y)}{\mathcal{V}_t} \frac{g_t^m(\tilde{x}, \tilde{y})}{\mathcal{E}_t} d\tilde{x} d\tilde{y} dt
\end{aligned}$$

where this KFE has been written from the perspective of the workers (as discussed in earlier sections, it could equivalently be written from the point of view of the firms). The first term on the RHS is the exit rate due to exogenous separations, the second term is the exit rate due to workers finding better matches, the third term is new matches from unemployed workers matching, and the final term is employed workers moving to (x, y) . Observe that:

$$\begin{aligned}
\frac{\mathcal{M}_t^e}{\mathcal{V}_t} &= \phi \frac{m(\mathcal{W}_t, \mathcal{V}_t)}{\mathcal{W}_t \mathcal{V}_t} \\
\frac{\mathcal{M}_t^u}{\mathcal{V}_t} &= \frac{m(\mathcal{W}_t, \mathcal{V}_t)}{\mathcal{W}_t \mathcal{V}_t}
\end{aligned}$$

So, the KFE becomes:

$$\begin{aligned}
dg_t^m(x, y) = & -\delta g_t^m(x, y)dt - \alpha_t^b(x, y)g_t(x, y) \\
& - \phi \frac{m(\mathcal{W}_t, \mathcal{V}_t)}{\mathcal{W}_t \mathcal{V}_t} g_t^m(x, y) \int \alpha_t^e(x, y, \tilde{y}) g_t^v(\tilde{y}) d\tilde{y} dt \\
& + \frac{m(\mathcal{W}_t, \mathcal{V}_t)}{\mathcal{W}_t \mathcal{V}_t} \alpha_t(x, y) g_t^u(x) g_t^v(y) dt \\
& + \phi \frac{m(\mathcal{W}_t, \mathcal{V}_t)}{\mathcal{W}_t \mathcal{V}_t} \int \alpha_t^e(\tilde{x}, \tilde{y}, y) g_t^v(y) \frac{g_t^m(\tilde{x}, \tilde{y})}{\mathcal{E}_t} d\tilde{x} d\tilde{y} dt
\end{aligned}$$

If we know measure of matches, then we can recover the other distribution:

$$\begin{aligned}
g_t^e(x) &= \int g_t^m(x, y) dy \\
g_t^u(x) &= g_t^w(x) - g_t^e(x) = g_t^w(x) - \int g_t^m(x, y) dy \\
g_t^p(y) &= \int g_t^m(x, y) dx \\
g_t^v(y) &= g_t^f(y) - g_t^p(y) = g_t^f(y) - \int g_t^m(x, y) dx \\
\mathcal{U}_t &= \int g_t^u(x) dx \\
\mathcal{V}_t &= \int g_t^v(y) dy \\
\mathcal{E}_t &= \int g_t^e(x) dx \\
\mathcal{P}_t &= \int g_t^p(y) dy
\end{aligned}$$