Deep Learning for Macroeconomics and Finance
(for solving models with complicated agent distributions)

Jonathan Payne

Princeton

based on work with Goutham Gopalakrishna, Zhouzhou Gu, Mathieu Laurière, Sebastian Merkel, Adam Rebei, and Yucheng Yang

June 5, 2024

PASC 2024: MS6F - Advances of Deep Learning in Economics

Many Economic Questions Involve Distributions Across Agents

- How do tax policies impact wealth inequality?
  (distribution of agent asset holdings)

- How do recessions impact wage and employment dispersion?
  (distribution of labor income across skills, education, and industries)

- How do innovation and trade shocks spread across firms (or countries)?
  (distribution of production technologies across firms)

- How do crisis shocks propagate through financial networks?
  (distribution of risk exposure across banks)

- What determines the distribution of city sizes?
  (distribution of population and firms across cities)

- ... These and many, many other questions require "**heterogeneous agent models**"

Technical Challenges and Solutions

- *Problem:* Hard to solve macro models with heterogeneous agents + aggregate shocks.
  - Infinite dimensional distribution becomes a state variable.
  - Traditional techniques: (linear/quadratic) perturbation, and approximate laws of motion.
- *Our goal:* develop a global solution technique for continuous macro time models:
  - Step 1: derive finite dimensional approximation to the distribution (finite agents, discrete state space, projections onto basis)
  - Step 2: train neural networks to solve the resulting high dimensional PDEs.
- *This talk:* discuss practical lessons from three of my papers:
  - Gu-Lauriere-Merkel-Payne (2024): solves business-cycle style macro models.
  - Payne-Rebei-Yang (2024): solves searching and maching models.
  - Gopalakrishna-Gu-Payne (2024): solves macro-finance models with implicit prices.

Deep learning offers the toolkit for heterogeneous agent macroeconomics

Table of Contents

Environment (Continuous Time Krusell-Smith '98)

- Continuous time, infinite horizon economy.
- Populated by $I = [0, 1]$ households who consume goods, supply labor, and save wealth.
- Representative firm rents capital and labor to produce goods by $Y_t = e^{z_t} K_t^{\alpha} L_t^{1-\alpha}$:
  - K_t is capital hired, L_t is labor hired,
  - z_t is productivity (exogenous aggregate state variable): follows $dz_t = \eta(\bar{z} - z_t)dt + \sigma dB_t^0$
  - B_t^0 is a common Brownian motion process; it generates filtration $\mathcal{F}_t^0$.
- Competitive markets for goods (numeraire), capital (rental rate r_t), labor (wage w_t).

Household Problem

- Household i has idiosyncratic state $x_t^i = (a_t^i, n_t^i)$, where a_t^i is wealth, n_t^i is labor.
- Given belief about price processes, $(\tilde{r}, \tilde{w})$, household chooses consumption $c = \{c_t^i\}_{t \geq 0}$:

$$\begin{aligned} &\max_{\{c_t^i\}_{t\geq 0}} \mathbb{E}_0 \left[\int_0^\infty e^{-\rho t} u(c_t^i) dt \right] \\ &s.t. \quad da_t^i = (\tilde{w}_t n_t^i + \tilde{r}_t a_t^i - c_t^i) dt =: \mu_t^a dt, \quad a_t^i \geq \underline{a} \\ &\qquad\quad n_t^i \in \{n_1, n_2\}, \text{ switches at idiosyncratic Poisson rate } \lambda(n_t^i) \end{aligned} \tag{1}$$

  - $u(c) = c^{1-\gamma}/(1-\gamma)$: utility function, ρ: discount rate, $\underline{a}$: borrowing limit.
  - $(\tilde{r}, \tilde{w}) = \{\tilde{r}_t, \tilde{w}_t\}_{t \geq 0}$ are agent beliefs about prices processes.

- Let $G_t = \mathcal{L}(a_t^i | \mathcal{F}_t^0)$ and g_t be population distribution and density of a_t^i, for history $\mathcal{F}_t^0$
  - Non degenerate since households get uninsurable idiosyncratic labor endowment shocks.

Equilibrium

Definition: Given an initial density g_0, an equilibrium for this economy consists of a collection of $\mathcal{F}_t^0$-adapted stochastic process, $\{c_t^i, g_t, z_t, q_t := [r_t, w_t] : t \geq 0, i \in I\}$, s.t.:

1. Given price process belief $\tilde{q}$, household consumption process, c_t^i, solves problem (1),
2. Given price process belief $\tilde{q}$, firm choose capital and labor optimally:
$$r_t = e^{z_t} \partial_K F(K_t, L) - \delta, \qquad w_t = e^{z_t} \partial_L F(K_t, L)$$
3. The price vector $q_t = [r_t, w_t]$ satisfies market clearing conditions:
$$K_t = \sum_{j \in \{1,2\}} \int a g_t(a, n_j) da, \qquad L = \sum_{j \in \{1,2\}} \int n_j g_t(a, n_j) da$$
4. Agent beliefs about the price process are consistent: $\tilde{q} = q$

Equilibrium (Combining Equations For Prices)

Definition: Given an initial density g_0, an equilibrium for this economy consists of a collection of $\mathcal{F}_t^0$-adapted stochastic process, $\{c_t^i, g_t, q_t := [r_t, w_t], z_t : t \geq 0, i \in I\}$, s.t.:

1. Given price process belief $\tilde{q}$, household consumption process, c_t^i, solves problem (1),

2. The price vector $q_t = [r_t, w_t]$ satisfies:

$$q_t = \begin{bmatrix} r_t \\ w_t \end{bmatrix} = \begin{bmatrix} e^{z_t}\partial_K F(K_t, L) - \delta \\ e^{z_t}\partial_L F(K_t, L) \end{bmatrix} =: Q(z_t, g_t), \text{ where } K_t = \sum_{j\in\{1,2\}} \int a g_t(a, n_j) da,$$

3. Agent beliefs about the price process are consistent: $\tilde{q} = q$.

Having closed form expressions for prices in terms of (z_t, g_t) makes problem very tractable

Recursive Representation of Equilibrium

- Aggregate states: (z, g), individual states: $x = (a, n)$, household value fn: $V(a, n, z, g)$.
- Given a belief $dg_t(x) = \tilde{\mu}_g(z_t, g_t)dt$, household at $x = (a, n)$ choose c to solve HJBE:

$$
\begin{aligned}
0 = \max_c \Big\{ & -\rho V(a,n,z,g) + u(c) + \partial_a V(a,n,z,g)(w(z,g)n + r(z,g)a - c) \\
& + \lambda(n)\left(V(a,\check{n},z,g) - V(a,n,z,g)\right) + \partial_z V(a,n,z,g)\mu^z(z) + 0.5\,(\sigma^z)^2\,\partial_{zz}V(a,n,z,g) \\
& + \int_{\mathcal{X}} \partial_g V(y,z,g)\tilde{\mu}^g(y,z,g)dy \Big\}, \quad s.t. \quad \text{BC: } \partial_a V|_{a=\underline{a}} \geq u'(wn + r\underline{a})
\end{aligned}
$$

where $\check{n}$ is complement of n.

- For optimal policy rule $c^*(a, n, z, g; \tilde{\mu}^g)$ and z_t, population density, g, evolves by KFE:

$$
dg_t(a,n) = \underbrace{\left[-\partial_a[(w(z,g)n + r(z,g)a - c^*)g_t(a,n)] - \lambda(n)g_t(a,n) + \lambda(\check{n})g_t(a,\check{n})\right]}_{=:\mu^g(a_t,n_t,z_t,g_t;\tilde{\mu}^g)}dt
$$

- In equilibrium $\tilde{\mu}^g = \mu^g$.

Recursive Representation of Equilibrium (Soft Borrowing Constraint)

- Aggregate states: (z, g), individual states: $x = (a, n)$, household value fn: $V(a, n, z, g)$.
- Given a belief $dg_t(x) = \tilde{\mu}_g(z_t, g_t)dt$, household at $x = (a, n)$ choose c to solve HJBE:

$$
\begin{aligned}
0 = \max_c \Big\{ & -\rho V(a,n,z,g) + u(c) - \mathbf{1}_{a_t \le \underline{a}} \psi(a_t) + \partial_a V(a,n,z,g)(w(z,g)n + r(z,g)a - c) \\
& + \lambda(n)\left(V(a,\check{n},z,g) - V(a,n,z,g)\right) + \partial_z V(a,n,z,g)\mu^z(z) + 0.5\left(\sigma^z\right)^2 \partial_{zz} V(a,n,z,g) \\
& + \int_{\mathcal{X}} \partial_g V(y,z,g)\tilde{\mu}^g(y,z,g)dy \Big\}, \quad s.t. \quad \cancel{\text{BC: } \partial_a V|_{a=\underline{a}} \ge u'(\cdot)}, \ \psi(a) = -\frac{1}{2}\kappa(a - \underline{a})^2
\end{aligned}
$$

where $\check{n}$ is complement of n.

- For optimal policy rule $c^*(a, n, z, g; \tilde{\mu}^g)$ and z_t, population density, g, evolves by KFE:

$$
dg_t(a,n) = \underbrace{\left[-\partial_a[(w(z,g)n + r(z,g)a - c^*)g_t(a,n)] - \lambda(n)g_t(a,n) + \lambda(\check{n})g_t(a,\check{n})\right]}_{=: \mu^g(c^*_t, a_t, n_t, z_t, g_t; \tilde{\mu}^g)} dt
$$

- In equilibrium $\tilde{\mu}^g = \mu^g$.

"Master Equation" Representation of Equilibrium

- "Master equation" substitutes KFE, market clearing & belief consistency into HJBE.

- Equilibrium value function $V(a, n, z, g)$ characterized by one PDE (if it exists):

$$
\begin{aligned}
0 = & -\rho V(a,n,z,g) + u(c^*(a,n,z,g)) + \mathbf{1}_{a_t \leq \underline{a}} \psi(a_t) \\
& + \partial_a V(a,n,z,g)(w(z,g)n + r(z,g)a - c^*(a,n,z,g)) \\
& + \lambda(x)\left(V(a,\check{n},z,g) - V(a,n,z,g)\right) + \partial_z V(a,n,z,g)\mu^z(z) + 0.5\left(\sigma^z\right)^2 \partial_{zz} V(a,n,z,g) \\
& + \int_{\mathcal{X}} \partial_g V(y,z,g)\mu^g(c^*(y,z,g),y,z,g)dy =: \mathcal{L}V
\end{aligned}
$$

where the optimal control c^* is characterised by: $u'(c^*(a,n,z,g)) = \partial_a V(a,n,z,g)$.

- Theory: studies whether this master equation exists; e.g. [Cardaliaguet et al., 2015]

- Our paper: look for a "global" finite, but high, dimensional approximation to V

Table of Contents

Numerical Approximation Outline

- Goal: characterize approximate solution to Master equation numerically
- Problem: Master equation contains an infinite dimensional derivative.
- Solution: three main ingredients:
  1. High but finite dimensional approximation to distribution and Master equation:
     (i). Replace continuum of agents by a finite population of agents, or
     (ii). Discretize the wealth variable, or
     (ii). Project distribution onto a finite dimensional set of basis functions $b_i(x)$ (e.g. eigenfunctions, Chebyshev polynomials, neural network, ...).
  2. Parameterize V by neural network, and
  3. Train the parameters to minimize the (approximate) master equation residual.

Ingredient 1: Finite Dimensional "Distribution" Approximation

	Finite Population	Discrete State	Projection
Params $\hat{\varphi}$	Agent states $\hat{\varphi}_t = \{(a_t^i, n_t^i)\}_{i \leq N}$	Masses on grid $\hat{\varphi}_{i,t}, \ \forall (a^i, n^i)_{i \leq N}$	Basis coefficients $\hat{\varphi}_{i,t}, \ \forall b_i(a;n)\|_{i \leq N}$
Dist. approx.	$\frac{1}{N}\sum_{i=1}^{N} \delta_{(a_t^i, n_t^i)}$	$\sum_{i=1}^{N} \hat{\varphi}_{i,t} \delta_{(a^i, n^i)}$	$\sum_{i=0}^{N} \hat{\varphi}_{i,t} b_i(a;n)$
KFE approx. ($\mu^{\hat{\varphi}}$)	Evolution of other agents' states	Evolution of mass between grid points (e.g. finite diff.)	Evolution of projection coefficients (least squares)
Dimension (N)	≈ 50	≈ 200	≈ 5

Projections more easily capture shape but have complicated KFE approximation

Ingredient 2: Approximate V by Neural Network

- Let $\boldsymbol{\omega} = (a, n, z, \hat{\varphi})$. Approximate value function $V(\boldsymbol{\omega})$ by neural network with form:

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

- Terminology: a fully connected feed forward NN (finite agent params in blue):
  - H: is the number of *hidden layers*, $(H = 5)$
  - Length of vector $\boldsymbol{h}^{(i)}$: number of *neurons* in hidden layer i, $(Length = 64)$
  - $\phi^{(i)}$: is the *activation function* for hidden layer i, $(\phi^i = tanh)$
  - σ: is the *activation function* for the final layer, (soft-plus)
  - $\boldsymbol{\Theta} = (W^1, \ldots W^{(H)}, b^{(1)}, \ldots, b^{(H)})$ are the *parameters*,
- Discrete state, projection more complicated NN [Sirignano and Spiliopoulos, 2018].

Ingredient 3: Algorithm ("EMINN" or "Economic Deep Galerkin")

1: Initialize the neural networks approximation for $\hat{V}$ with parameters Θ.

2: **while** Loss $>$ tolerance **do**

3: Sample M new training points: $\boldsymbol{S} = (\boldsymbol{S}^m = (a^m, n^m, z^m, (\hat{\varphi}_i^m)_{i \leq N}))_{m=1}^M$.

4: Calculate the weighted average error across sample points, given current Θ:

$$\mathcal{E}(\boldsymbol{S}; \Theta) = \kappa^e \frac{1}{M} \sum_{m \leq M} |\hat{\mathcal{L}}(a^m, n^m, z^m, (\hat{\varphi}_i^m)_{i \leq N}; \Theta)| + \kappa^s \mathcal{E}^s(\boldsymbol{S}; \Theta), \quad \text{where}$$

- $\hat{\mathcal{L}}(a^m, n^m, z^m, (\hat{\varphi}_i^m)_{i \leq N}; \Theta)$ is error in Master equation $\hat{\mathcal{L}}$ at training point $\boldsymbol{S}^m$ (where derivates in $\hat{\mathcal{L}}$ are calculated using automatic differentiation.)
- $\mathcal{E}^s(\boldsymbol{S}; \Theta)$ is penalty for "wrong" shape (e.g. penalty for non-concavity of V)

5: Update parameters Θ by stochastic gradient descent: $\Theta^{new} = \Theta^{old} - \alpha D_\Theta \mathcal{E}(\boldsymbol{S}; \Theta)$

6: **end while**

The Macroeconomic Deep Learning Conundrum

- Algorithm is straightforward to describe and code
- . . . but hard to implement successfully!
- I will discuss some features that we have found helpful.

Sampling Approaches

- Sampling (a, n, z): draw from uniform distribution, then add draws where error high.
- Sampling the parameters in the distribution approximation $(\hat{\varphi}^i)_{i \leq N}$:
  - *Moment sampling:*
    1. Draw samples for selected moments of the distribution (that are important for $\hat{Q}(z, \hat{\varphi})$).
    2. Sample $\hat{\varphi}$ from a distribution that satisfies the moments drawn in the first step.
  - *Mixed steady state sampling:*
    1. Solve for the steady state for a collection of fixed aggregate states z.
    2. Draw random, perturbed mixtures of this collection of steady state distributions.
  - *Ergodic sampling:*
    1. Simulate economy using current value function approximation.
    2. Use simulated distributions as training points.

Need to choose economically relevant subspace on which to sample.

Implementation Details

	Finite Population	Discrete State	Projection
Neural Network			
(i) Structure	Feed-forward	Recurrent + embedding	Recurrent + embedding
(ii) Initialization	$W(a,\cdot) = e^{-a}$	random	random
Sampling			
(i) (a, l)	Active sampling $[\underline{a}, \overline{a}] \times \{y_1, y_2\}$	Uniform sampling $[\underline{a}, \overline{a}] \times \{y_1, y_2\}$	Uniform sampling $[\underline{a}, \overline{a}] \times \{y_1, y_2\}$
(ii) $(\hat{\varphi}_i)_{i \leq N}$	Moment sampling: sample r, then random agents positions that give r	Mixed steady-state sampling then ergodic sampling	Sample K and then orthogonal coefficients from ergodic sampling
(iii) z	$U[z_{min}, z_{max}]$	$U[z_{min}, z_{max}]$	$U[z_{min}, z_{max}]$
Loss Function			
(i) Constraints	$\partial_{aa} V(a,\cdot), \partial_{za} V(a,\cdot) < 0$	$\partial_{aa} V(a,\cdot), \partial_{za} V(a,\cdot) < 0$	$\partial_{aa} V(a,\cdot), \partial_{za} V(a,\cdot) < 0$
(ii) Learning rate	Decay from 10^{-4} to 10^{-6}	Decay from 10^{-4} to 10^{-6}	Decay from 10^{-4} to 10^{-6}

Neural Network Q & A

- ***Q.*** *What are the main differences to discrete time?*
  - Need to calculate derivatives rather than expectations (we use automatic differentiation)
  - Need to choose where to sample (sample more where master equation error is large)
- ***Q.*** *Why do we need shape constraints?*
  - Neural network can find "bad" approximate solutions,
    (E.g. portfolio problem has approximate solution $V \approx 0$ for high γ.) More
  - Option: penalize shape that correspond to known "bad" solutions.
  - Option: train ϕ satisfying $V(a, n, z, g) = \phi(a, n, z, g; \theta)(a - \underline{a})^{1-\gamma}$ instead of training V

Neural Network Q & A

- ***Q.*** *What about slowing down the updating?*
  - For projection methods, we use "Howard improvement algorithm" to slow down the rate of updating (fix policy rule for some iterations and just update V).
  - [Duarte, 2018] and [Gopalakrishna, 2021] suggest introducing a "false" time step but so far we have not found this necessary (or found a way to implement at high scale).
  - We use shape constraints as a replacement.
- ***Q.*** *What about imposing symmetry and/or dimension reduction?*
  - [Han et al., 2021] and [Kahou et al., 2021] suggest feeding the distribution through a preliminary neural network that reduces the dimension and imposes symmetry.
  - We find we can solve the problem with and without this approach.

Table of Contents

Testing the Algorithm

- We test the model with fixed aggregate productivity (Aiyagari (1994)) Plots

	Master equation loss	MSE(NN, FD)
Finite Agent NN	3.135×10^{-5}	4.758×10^{-5}
Discrete State Space NN	9.303×10^{-6}	6.591×10^{-5}

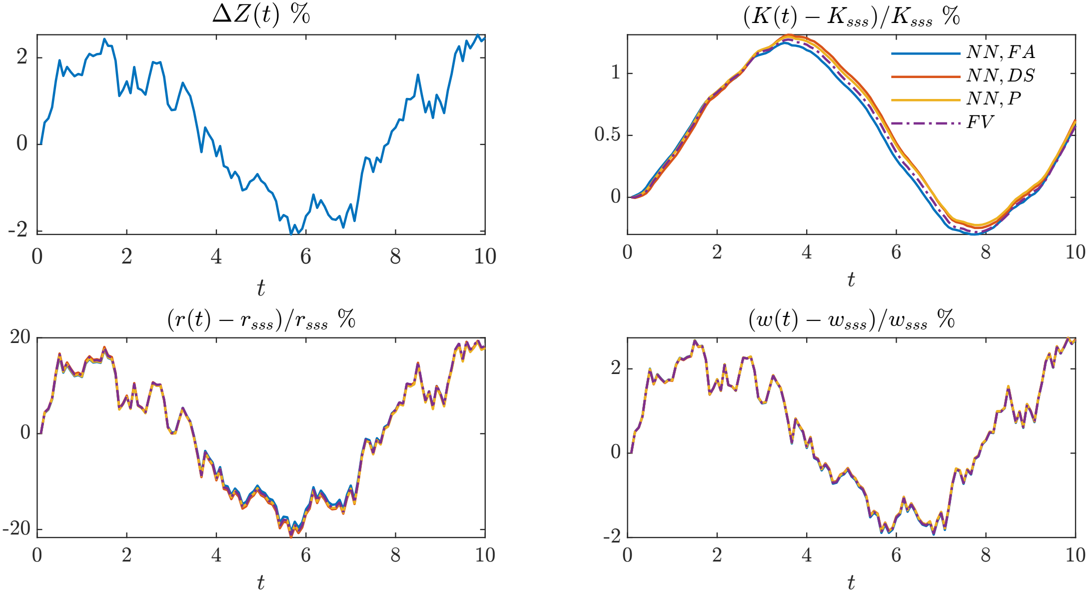
- Solve version with stochastic aggregate productivity (Krusell-Smith (1998)):

	Master equation training loss
Finite Agent NN	3.037×10^{-5}
Discrete State Space NN	9.639×10^{-5}
Projection NN	8.506×10^{-6}

  - Neural network solutions generate similar output to traditional methods.
  - Example plots: comparison to [Fernández-Villaverde et al., 2018]

Krusell-Smith: Sample Time Paths

More Plots



Krusell-Smith: Numerical Results

More Plots

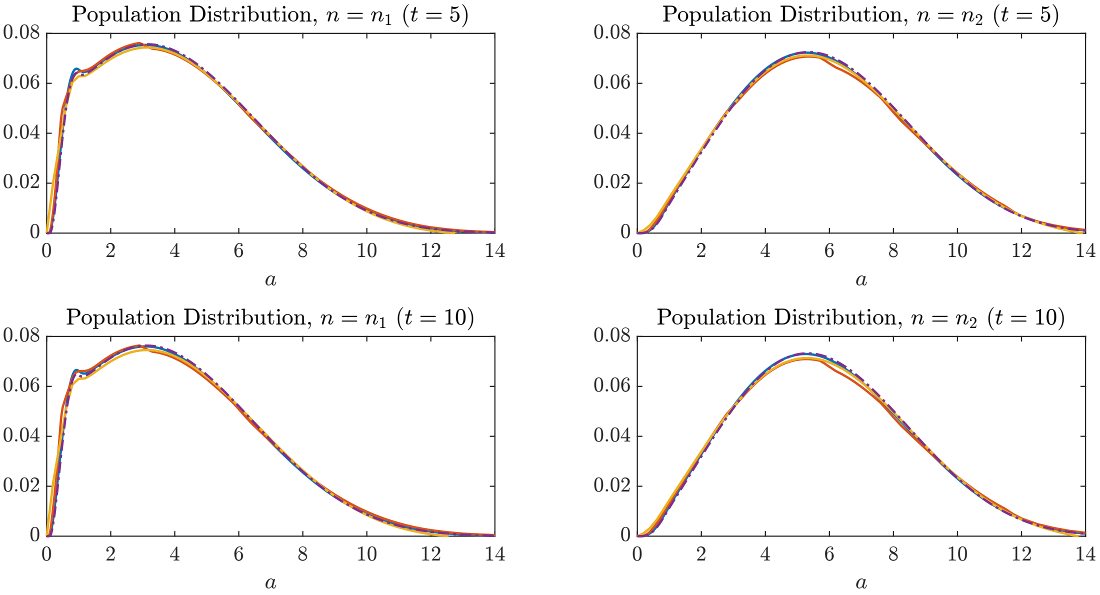


Table of Contents

Two Minor Extensions

1. Heterogeneous firm dynamics (in Gu-Lauriere-Merkel-Payne '24):
   - In KS-98, there is only one firm.
   - We extend our method to models with many firms that each have non-tradable capital and capital adjustment costs (a version of [Khan and Thomas, 2008])
   - Technical difference is that we need to price a firm equity.

2. Population movements (in Gu-Lauriere-Merkel-Payne '24):
   - In the KS-98 model, there was one wage in the entire economy
   - We extend our method to a "spatial" model where there are different wages in different locations and agents can move (a version of [Bilal, 2021]).
   - Technical difference is that the wage in location j only depends on the part of the population distribution at location j.

Two Major Extensions

1. Search and matching (SAM) models (in Payne-Rebei-Yang '24):
   - In KS-98, the mean of distribution entered the master equation through the prices.
   - In search and matching models, the shape of the distribution is more important

	Distribution	Distribution impact on 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

2. Macro finance models with complicated asset pricing (Gopalakrishna-Gu-Payne '24):
   - In the KS-98 model, we can express (r, w) as closed form functions of the state.
   - For pricing long-term assets, the price process is only an implicit function of the state.
   - We also have to handle portfolio choice, which deep learning has found hard.

Table of Contents

Conclusion

1. Deep learning in macroeconomics is a rapidly evolving field.

2. Much progress but still many challenges.

3. Some comments/questions for the computer science literature:
   - Neural networks often seem “too flexible” for macroeconomic models. Other options?
   - How should we rewrite economic models to make them more deep learning “friendly”?
   - How should we dynamically choose weights in the loss function?
   - How do we get the a better stability vs speed tradeoff?

Thank You!

References I

Achdou, Y., Han, J., Lasry, J.-M., Lions, P.-L., and Moll, B. (2022).
Income and wealth distribution in macroeconomics: A continuous-time approach.
The Review of Economic Studies, 89(1):45–86.

Azinovic, M. and Žemlička, J. (2023).
Economics-inspired neural networks with stabilizing homotopies.
arXiv preprint arXiv:2303.14802.

Basak, S. and Cuoco, D. (1998).
An equilibrium model with restricted stock market participation.
The Review of Financial Studies, 11(2):309–341.

Bilal, A. (2021).
Solving heterogeneous agent models with the master equation.
Technical report, University of Chicago.

Brunnermeier, M. K. and Sannikov, Y. (2014).
A Macroeconomic Model with a Financial Sector*.
The American Economic Review, 104(1042):379–421.

Cardaliaguet, P., Delarue, F., Lasry, J.-M., and Lions, P.-L. (2015).
The master equation and the convergence problem in mean field games.
arXiv.

References II

Carmona, R. (2020).
Applications of mean field games in financial engineering and economic theory.
To appear in: Machine Learning and Data Sciences for Financial Markets, Cambridge University Press.

Duarte, V. (2018).
Machine learning for continuous-time economics.
Available at SSRN 3012602.

Fernández-Villaverde, J., Hurtado, S., and Nuño, G. (2018).
Financial Frictions and the Wealth Distribution.
Working Paper, pages 1–51.

Gopalakrishna, G. (2021).
Aliens and continuous time economies.
Swiss Finance Institute Research Paper, (21-34).

Gu, Z., Laurière, M., Merkel, S., and Payne, J. (2023).
Deep learning solutions to master equations for continuous time heterogeneous agent macroeconomic models.

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.

References III

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.

Khan, A. and Thomas, J. K. (2008).
Idiosyncratic shocks and the role of nonconvexities in plant and aggregate investment dynamics.
Econometrica, 76(2):395–436.

Krusell, P. and Smith, A. A. (1998).
Income and Wealth Heterogeneity in the Macroeconomy.
Journal of Political Economy, 106(5):867–896.

Lucas, R. E. (1978).
Asset prices in an exchange economy.
Econometrica, 46(6):1429–1445.

Sirignano, J. and Spiliopoulos, K. (2018).
DGM: A deep learning algorithm for solving partial differential equations.
Journal of computational physics, 375:1339–1364.

Table of Contents

Two Extensions

1. Search and matching (SAM) models (Payne-Rebei-Yang '24):
   - In KS-98, the mean of distribution entered the master equation through the prices.
   - In search and matching models, the shape of the distribution is more important

	Distribution	Distribution impact on 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

2. Macro finance models with complicated asset pricing (Gopalakrishna-Gu-Payne '24):
   - In the KS-98 model, we can express (r, w) as closed form functions of the state.
   - For pricing long-term assets, the price process is only an implicit function of the state.
   - We also have to handle portfolio choice, which deep learning has found hard.

Table of Contents

Shimer-Smith/Mortensen-Pissarides with Two-sided Heterogeneity

- Continuous time, infinite horizon environment.
- Workers $x \in [0,1]$ with exog. density $g_t^w(x)$; Firms $y \in [0,1]$ with density $g_t^f(y)$:
  (We also solve model with and without firm free entry)
  - Unmatched: unemployed workers get benefit b; vacant firms produce nothing.
  - Matched: type x worker and type y firm produce output $z_t f(x,y)$.
  - z_t: follows two-state continuous time Markov Chain (can be generalized).
- Meet randomly at rate $m(\mathcal{U}_t, \mathcal{V}_t)$, $\mathcal{U}_t$ is total unemployment, $\mathcal{V}_t$ is total vacancies.
  (We also solve model with on-the-job search)
- Upon meeting, agents choose whether to match:
  - Match surplus $S_t(x,y)$ divided by generalized Nash bargaining: workers get fraction β.
  - Match acceptance function is $\alpha_t(x,y) = \mathbb{1}\{S_t(x,y) > 0\}$. Matches dissolve rate $\delta(x,y)$.
- Equilibrium object: $g_t(x,y)$ mass function of matches (x,y).

Recursive Equilibrium Part I: Unemployed Workers & KFE

- Idiosyncratic state $= x$, Aggregate states $= (z, g(x,y))$.

- Hamilton-Jacobi-Bellman equation for an unemployed worker's value $V^u(x,z,g)$:

$$\rho V^u(x,z,g) = b + \frac{m(z,g)}{\mathcal{U}(z,g)} \int \alpha(x,\tilde{y},z,g)(V^e(x,\tilde{y},z,g) - V^u(x,z,g)) \frac{g^v(\tilde{y})}{\mathcal{V}(z,g)} d\tilde{y}$$
$$+ \lambda_{z\tilde{z}}(V^u(x,\tilde{z},g) - V^u(x,z,g)) + D_g V^u(x,z,g) \cdot \mu^g$$

  - $\alpha_t(x,\tilde{y},z,g)$ indicates match acceptance
  - $g_t^u(x)$ and $g_t^v(y)$ are mass functions for unemployed workers and vacant firms.
  - $V^e(x,y,z,g)$ is employed worker's value and $D_g V^u(x,z,g)$ is Frechet derivative w.r.t g.

- Other Hamilton-Jacobi-Bellman equations are similar. More

- Kolmogorov forward equation (KFE):

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

Recursive Characterization For Equilibrium Surplus

- Surplus from match $S(x,y,z,g) := V^p(x,y,z,g) - V^v(y,z,g) + V^e(x,y) - V^u(x,z,g)$.
- Characterize equilibrium with master equation for surplus:

$$\begin{aligned}
\rho S(x,y,z,g) &= zf(x,y) - \delta(x,y)S(x,y,z,g) \\
&- (1-\beta)\frac{m(z,g)}{\mathcal{V}(z,g)}\int \alpha(\tilde{x},y,z,g)S(\tilde{x},y,z,g)\frac{g^u(\tilde{x})}{\mathcal{U}(z,g)}d\tilde{x} \\
&- b - \beta\frac{m(z,g)}{\mathcal{U}(z,g)}\int \alpha(x,\tilde{y},z,g)S(x,\tilde{y},z,g)\frac{g^v(\tilde{y})}{\mathcal{V}(z,g)}d\tilde{y} \\
&+ \lambda_{z\tilde{z}}(S(x,y,\tilde{z},g) - S(x,y,z,g)) + D_g S(x,y,z,g)\cdot\mu^g(z,g)
\end{aligned}$$

- Kolmogorov forward equation (KFE):

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

- High-dim PDEs with distribution in state: hard to solve with conventional methods.

Comparison to Other Heterogeneous Agent Search Models

- Lise-Robin '17: sets $\beta = 0$ and has "free" vacancy creation so that:
  (and Postal-Vinay style Bertrand competition for workers searching on-the-job)

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

- Menzio-Shi '11: one-sided heterogeneity, competitive search, and "free" firm entry so:

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

- We look for a solution for S and α in terms of the distribution g.

Modification 1: Discrete State Approximation

- Approximate $g(x, y)$ on finite types: $x \in \mathcal{X} = \{x_1, \ldots, x_{n_x}\}$, $y \in \mathcal{Y} = \{y_1, \ldots, y_{n_y}\}$.
- Finite state approximation $\Rightarrow$ analytical (approximate) KFE: $g \approx \{g_{ij}\}_{i \leq n_x, j \leq n_y}$
- Approximated master equation for surplus:

$$
\begin{aligned}
0 = \mathcal{L}^S S(x, y, z, g) &= -(\rho + \delta) S(x, y, z, g) + z f(x, y) - b \\
&- (1 - \beta) \frac{m(z, g)}{\mathcal{V}(z, g)} \frac{1}{n_x} \sum_{i=1}^{n_x} \alpha(\tilde{x}_i, y, z, g) S(\tilde{x}_i, y, z, g) \frac{g^u(\tilde{x}_i)}{\mathcal{U}(z, g)} \\
&- \beta \frac{m(z, g)}{\mathcal{U}(z, g)} \frac{1}{n_y} \sum_{j=1}^{n_y} \alpha(x, \tilde{y}_j, z, g) S(x, \tilde{y}_j, z, g) \frac{g^v(\tilde{y}_j)}{\mathcal{V}(z, g)} \\
&+ \lambda_{z\tilde{z}} (S(x, y, \tilde{z}, g) - S(x, y, z, g)) + \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \partial_{g_{ij}} S(x, y, z, \{g_{ij}\}_{i,j}) \mu^g(\tilde{x}_i, \tilde{y}_j, z, g)
\end{aligned}
$$

Modification 2: Approximate Discrete Choice

- In the original model,

$$\alpha(x, y, z, g) = \mathbb{1}\{S(x, y, z, g) > 0\}$$

- Discrete choice $\alpha \Rightarrow$ discontinuity of $S(x, y, z, g)$ at some g.

- To ensure master equation well defined & NN algorithm works, we approximate with

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

- Interpretation: logit choice model with utility shocks $\sim$ extreme value distribution. ($\xi \to \infty \Rightarrow$ discrete choice α.)

Feature 3: With and Without Free Entry

- No entry: $\mathcal{V}(z,g)/\mathcal{U}(z,g)$ from exog. population distributions and matches.
- With free entry, depends upon the surplus function:

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

Methodology Q & A

- ***Q.*** *How do we choose where to sample?* Use mixed steady state sampling.
  - We start by drawing distributions near steady states for different fixed z.
  - Can move to ergodic sampling once error is small.

- ***Q.*** *What about dimension reduction?*
  - For competitive markets, Krusell-Smith '98 suggest approximating distribution by mean.
  - We exploited a similar idea when sampling to train a NN to solve Krusell-Smith '98
  - For random search, not clear what moment enables approximation of:

$$\int \alpha(\tilde{x}, y, z, g) S(\tilde{x}, y, z, g) \frac{g^u(\tilde{x})}{\mathcal{U}(z, g)} d\tilde{x}, \quad \text{and} \quad \int \alpha(x, \tilde{y}, z, g) S(x, \tilde{y}, z, g) \frac{g^v(\tilde{y})}{\mathcal{V}(z, g)} d\tilde{y}$$

  - This is why moment sampling is not effective and we need other techniques.

Methodology Q & A

- ***Q.*** *How can we stabilize the algorithm?*
  - Most difficult when $\widehat{S}(x, y, z, g; \boldsymbol{\Theta})$ has sharp curvature. We use "homotopy":
    - Step 1: train NN for parameters that give low curvature in $\widehat{S}^1$
    - Step 2: change parameters closer and retrain NN starting from previous $\widehat{S}^2 = \widehat{S}^1$
    - Step 3+: keep changing parameters and retraining until at desired parameters.
  - Alternative is to introduce false time derivative (e.g. Ahn et al-18, Duarte-18, Gop.-23)
- ***Q.*** *What do we mean when we say this is a global solution?*
  - Algorithm gives a solution across the discretized state space $(x, y, z, \{g_{ij}\}_{i \leq n_x, j \leq n_y})$,
  - ... which is a $3 + n_x \times n_y$ dimensional state space.
  - Not a perturbation in z or g (e.g. Bilal '23).

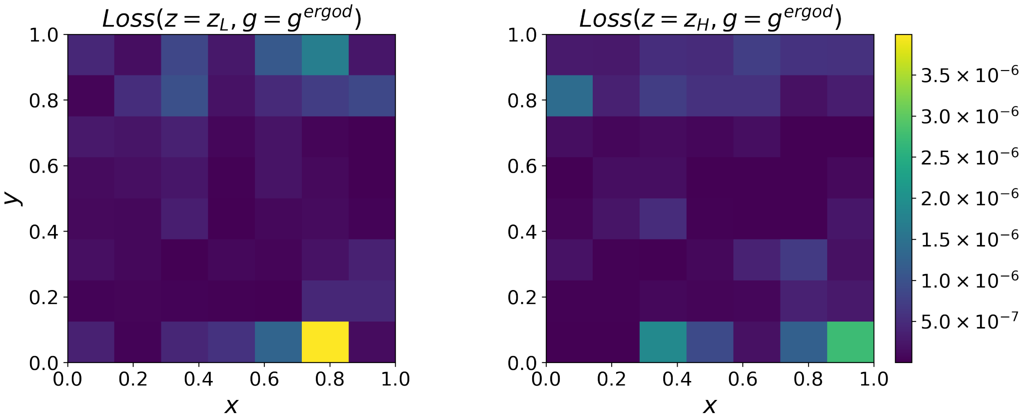
Calibration

Frequency: annual.

Parameter	Interpretation	Value	Target/Source
ρ	Discount rate	0.05	Kaplan, Moll, Violante '18
δ	Job destruction rate	0.2	BLS job tenure 5 years
ξ	Extreme value distribution for α choice	2.0	
$f(x, y)$	Production function for match (x, y)	$0.6 + 0.4\left(\sqrt{x} + \sqrt{y}\right)^2$	Hagedorn et al '17
β	Surplus division factor	0.72	Shimer '05
$z, \tilde{z}$	TFP shocks	1 ± 0.015	Lise Robin '17
$\lambda_z, \lambda_{\tilde{z}}$	Poisson transition probability	0.08	Shimer '05
$\delta, \tilde{\delta}$	Separation shocks	0.2 ± 0.02	Shimer '05
$\lambda_\delta, \lambda_{\tilde{\delta}}$	Poisson transition probability	0.08	Shimer '05
$m(\mathcal{U}, \mathcal{V})$	Matching function	$\kappa \mathcal{U}^\nu \mathcal{V}^{1-\nu}$	Lise Robin '17
ν	Elasticity parameter for meeting function	0.5	Lise-Robin '17
κ	Scale parameter for meeting function	5.4	Unemployment rate
b	Worker unemployment benefit	0.5	Shimer '05
n_x	Discretization of worker types	7	
n_y	Discretization of firm types	8	

Numerical performance: Accuracy I Calibration

- Mean squared loss as a function of type in the master equations of S (at ergodic g).



Numerical performance: Accuracy II Calibration

- Compare steady state solution without aggregate shocks to solution using conventional methods.

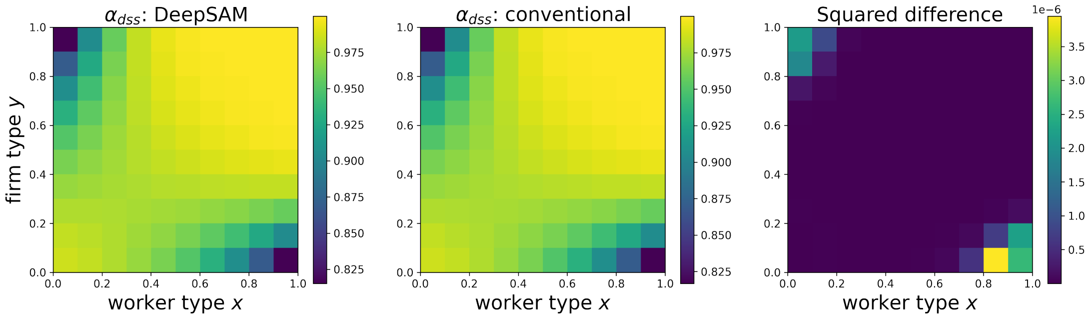


Figure: Comparison with steady-state solution

Comparison for discrete α

DeepSAM vs block recursivity: "depression" shock on g

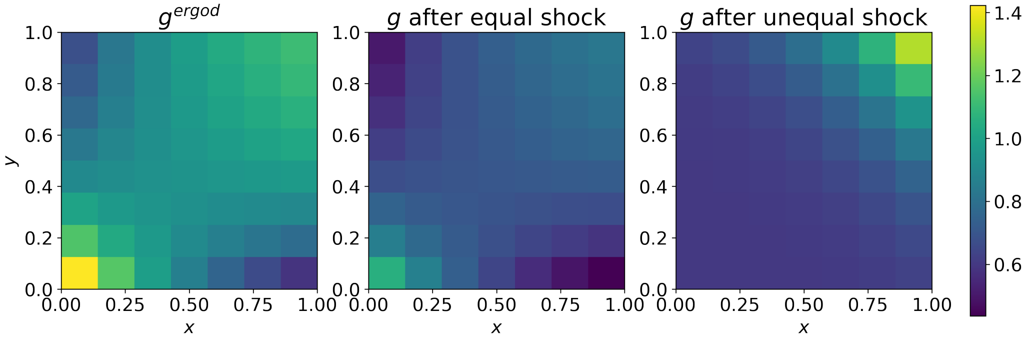


Figure: Ergodic distribution and distribution after the "unequal" and "equal" "depression" shocks

Question: how recovery dynamics differ under full solution (using DeepSAM) vs under "block recursive" solution (where g does not affect decision)?

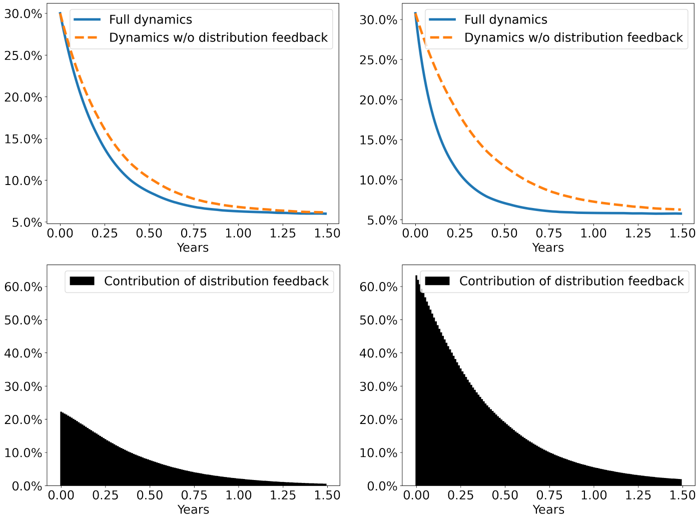
Q. How much does dependence of α on g matter?

- Consider two impulse responses:
  (i) The change in unemployment when acceptance is always evaluated at the long-run ergodic employment distribution but otherwise the distribution follows KFE:
$$\frac{dg_t^{BR}(x,y)}{dt} = -\delta(x,y,z_t)g_t^{BR}(x,y) + \frac{m_t(z,\underline{\boldsymbol{g}}_t)}{\mathcal{U}_t(\underline{\boldsymbol{g}}_t)\mathcal{V}_t(\underline{\boldsymbol{g}}_t)}\alpha(x,y,z_t,\underline{\boldsymbol{g}}^{\text{ergodic}})g_t^{u,BR}(x)g_t^{v,BR}(y)$$
  (ii) The change in unemployment when the acceptance function reacts to the changing employment distribution.

- We interpret the former as the dynamics without the "distribution feedback".

- We define the contribution of g through "distribution feedback" dynamics by:
$$\Delta_t := \frac{|U_t - U_t^{BR}|}{|U_t - U_0|}$$

Unemployment rate IRF after "depression" shock on g



Column 1: "equal" shock; Column 2 "unequal" shock.

Unemployment rate IRF to expansionary TFP shocks

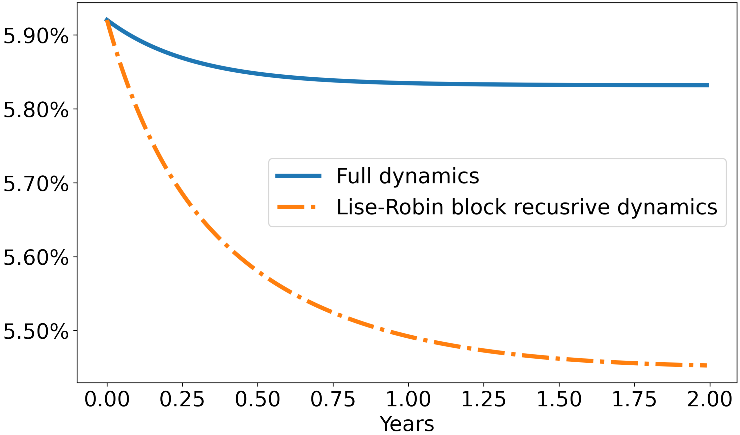
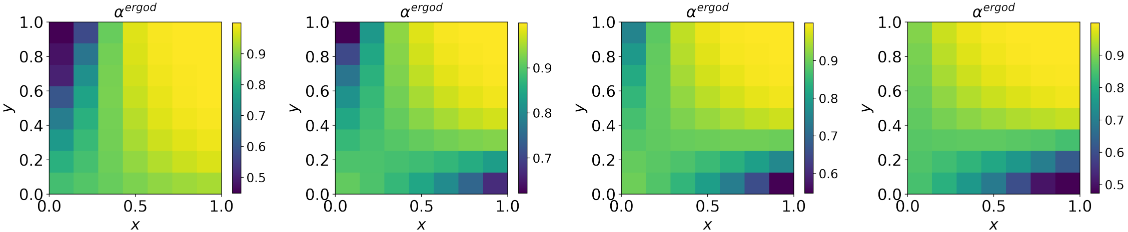


Figure: Comparison: full solution with DeepSAM vs. block-recursive solution à la Lise-Robin

Note: we recalibrate the model to match the unemployment rate at steady state when we adopt the Lise-Robin assumption with $\beta = 0$.

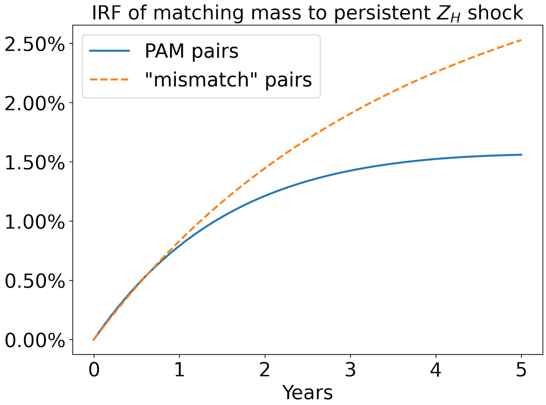
Worker Bargaining Power Influences Assortative Matching

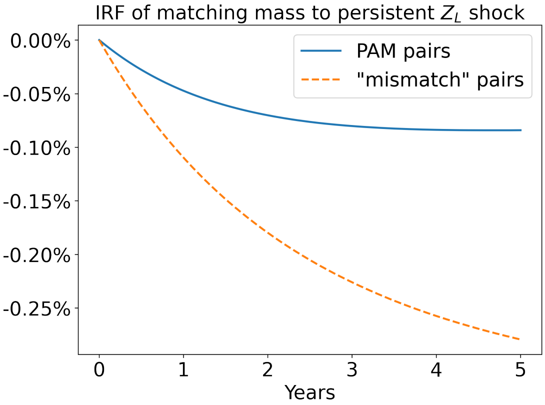


- Sorting at the ergodic distribution for different worker bargaining power β.
  Left to right $\beta = 0$ (Lise-Robin '17), $0.5, 0.72$ (benchmark), 1
- Solved with on-the-job search to compare with Lise-Robin '17.
  Additional parameter calibration is employed worker search intensity: $\phi = 0.2$.

Sorting Over Business Cycles

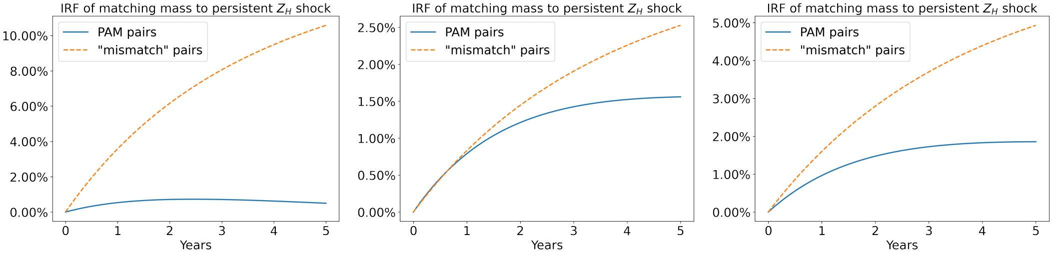
- Study how "mismatch" changes over the business cycle.





Sorting Over Business Cycles

- Countercyclicality of sorting depends on bargaining power.



Left to right $\beta = 0$ (Lise-Robin '17), 0.72 (benchmark), 1.

Table of Contents

Environment

- Continuous time. One good produced by technology $y_t = e^{z_t} k_t$, where:
  - Aggregate productivity follows $dz_t = \zeta(\bar{z} - z_t)dt + \sigma_z dW_t$
  - Capital stock follows $dk_t = (\phi(\iota_t)k_t - \delta k_t)dt$, where ι_t is the investment rate.
- Finite collection of price taking households ($i \leq I - 1$): (see [Gu et al., 2023])
  - Idiosyncratic death shocks at λ_h; new agent gets $1 - \tau$ fraction of dying agent's wealth.
  - Flow utility $u(c_{i,t}) = c_{i,t}^{1-\gamma}/(1-\gamma)$ and effective discount rate $\rho_h := \rho + \lambda_h$
  - Penalty on holding capital: $\Psi_{h,t}(k_{i,t}, a_{i,t})$, $\uparrow$ in capital $k_{i,t}$ and $\downarrow$ in wealth $a_{i,t}$.
- Financial "expert" with $\rho_e > \rho_h$, Epstein-Zin preferences, and no equity raising. More
- Competitive markets for goods, risk-free bonds (at r_t), & capital (with price q_t).

$$\frac{dq_t}{q_t} = \mu_{q,t}dt + \sigma_{q,t}dW_t, \qquad dR_{k,t} := \frac{e^{z_t} - \iota_t k_t}{q_t k_t} + \frac{d(q_t k_t)}{q_t k_t} =: r_{k,t}dt + \sigma_{q,t}dW_t$$

Optimization and Equilibrium

- Given belief about price processes $(\hat{r}, \hat{q})$, household i with wealth $a_{i,t} = b_{i,t} + q_t k_{i,t}$:

$$\max_{c_i, k_i, \iota_i} \left\{ \mathbb{E}_0 \left[\int_0^\infty e^{-\rho_i t} \left(u(c_{i,t}) - \Psi_t(k_{i,t}, a_{i,t}) \right) dt \right] \right\}$$

$$\begin{aligned} s.t. \quad da_{i,t} &= (a_{i,t} - k_{i,t})\hat{r}_{i,t}dt + k_{i,t}d\hat{R}_{k,t}(\iota_t) - c_{i,t}dt + \tau\lambda A_t dt \\ &=: \mu_{a_i} a_{i,t}dt + \sigma_{a,i} a_{i,t} dW_t \end{aligned}$$

- Expert problem similar but without Ψ and with Epstein-Zin preferences More

- Equilibrium:
  1. Given $\hat{r}, \hat{q}$, households and expert optimize.
  2. Prices (q_t, r_t) solves market clearing:
     (i) Goods market $\sum_i c_{i,t} + \sum_i \Phi(\iota_{i,t}) k_{i,t} = y_t$,
     (ii) Capital market $\sum_i k_{i,t} = K_t$ and (iii) Bond market $\sum_i b_{i,t} = 0$.
  3. Agent beliefs are consistent with equilibrium $(\hat{r}, \hat{q}) = (r, q)$.

Recursive Characterization of Equilibrium: (in Wealth Levels)

- Individual state $= a_i$, Aggregate states $= (z, K, \{a_j\}_{j \neq i}) = (\cdot)$.

- Given belief about evolution of other agents, $(\hat{\mu}_{a_j}(\cdot), \hat{\sigma}_{a_j}(\cdot))$, household i solves:

$$\rho V_i(a_i, \cdot) = \max_{c_i, k_i, \iota_i} \Big\{ u(c_i) - \Psi(k_i, a_i, \cdot) + \frac{\partial V_i}{\partial a_i} \mu_{a_i}(a_i, c_i, k_i, \iota, \cdot) + \frac{\partial V_i}{\partial z} \mu_z + \frac{\partial V_i}{\partial K} \hat{\mu}_K(\cdot)$$
$$+ \frac{1}{2} \frac{\partial^2 V_i}{\partial a_i^2} \sigma_{a_i}^2(k_i, \cdot) + \frac{1}{2} \frac{\partial^2 V_i}{\partial z^2} \sigma_z^2 + \frac{\partial^2 V_i}{\partial a_i \partial z} \sigma_{a_i}(k_i, \cdot) \sigma_z$$
$$+ \sum_{j \neq i} \frac{\partial^2 V_i}{\partial a_j \partial z} \hat{\sigma}_{a_j}(\cdot) \sigma_z + \frac{1}{2} \sum_{j \neq i, j' \neq i} \frac{\partial^2 V_i}{\partial a_j \partial a_{j'}} \hat{\sigma}_{a_j}(\cdot) \hat{\sigma}_{a'_j}(\cdot) \Big\}$$

- Expert HJBE is similar but without $\Psi_i(k_i, a_i, \cdot)$ and with Epstein-Zin terms.

- In equilibrium, beliefs are consistent: $(\hat{\mu}_{a_j}(\cdot), \hat{\sigma}_{a_j}(\cdot)) = (\mu_{a_j}(\cdot), \sigma_{a_j}(\cdot))$.

Recursive Characterization of Equilibrium: (in Wealth Shares)

- Change variable to marginal value of wealth: $\xi_i := \partial V_i / \partial a_i$.
- Change distribution to wealth shares $(z, K, \{\eta_i\}_{1 \le i \le I})$, where $\eta_i := a_i/A$ is agent i's share.
- Once equilibrium is imposed, we know ξ_i is a function w.r.t. $(z, K, \{\eta_i\}_{1 \le i \le I})$

  ...so we can write μ_{ξ_i} and σ_{ξ_i} in terms of derivatives of $(z, K, \{\eta_i\}_{1 \le i \le I})$.
- We group the resulting equilibrium equations into three blocks.

Block 1: Optimization

- Given price processes $(r, r_k, q, \mu_q, \sigma_q)$, household optimization implies:

Euler equation:

$$\rho_h \xi_h = r\xi_h + \frac{\partial \xi_h}{\partial z}\mu_z + \frac{\partial \xi_i}{\partial K}\mu_K + \frac{1}{2}\frac{\partial^2 \xi_h}{\partial z^2}\sigma_z^2 + \sum_j \frac{\partial \xi_h}{\partial \eta_j}\eta_j \mu_{\eta_j,t}$$
$$+ \sum_j \frac{\partial^2 \xi_h}{\partial z \partial \eta_j}\eta_j \sigma_{\eta_j,t}\sigma_z + \frac{1}{2}\sum_{j,j'} \frac{\partial^2 \xi_h^2}{\partial \eta_j \partial \eta_{j'}}\eta_j \eta_{j'}\sigma_{\eta_j,t}\sigma_{\eta_{j'},t}$$

Consumption FOC:

$$\xi_h = u'(c_h)$$

Portfolio FOC:

$$\xi_h(r_k - r) = -\Big(\frac{\partial \xi_h}{\partial z}\sigma_z + \sum_j \frac{\partial \xi_h}{\partial \eta_j}\sigma_{j,\eta}\Big)\sigma_q - \frac{\partial \Psi_h}{\partial k_i}$$

- Expert optimization is similar but adjusted for Epstein-Zin More

Block 2: Distribution Evolution

- Given prices $(r, r_k, q, \mu_q, \sigma_q)$ and (c, ξ, k), the law of motion of wealth shares is:

$$\frac{d\eta_{j,t}}{\eta_{j,t}} = \mu_{\eta_j,t}dt + \sigma_{\eta_j,t}dW_t, \qquad \text{where:}$$

$$\mu_{\eta_j,t} = r_t + \theta_{j,t}(r_{k,t} - r_t) - \omega_{j,t} - \mu_{q,t} - \mu_{K,t} + (1 - \theta_{j,t})\sigma_{q,t}^2 + \lambda\tau\left(\frac{\frac{1}{I-1}(1 - \eta_{j,t})}{\eta_{j,t}} - 1\right)$$

$$\sigma_{\eta_j,t} = -(1 - \theta_{j,t})\sigma_{q,t}$$

where

  - $\theta_{k,t} := k_{j,t}/(\eta_{j,t}q_tK_t)$ is agent j's share of wealth in capital,
  - $\omega_{j,t} := c_{j,t}/(\eta_{j,t}q_tK_t)$ is agent j's consumption-to-wealth ratio.

Block 3: Equilibrium Consistency

- Clearing conditions pin down the prices:

$$\sum_i c_{i,t} + \Phi(\iota_t) K_t = y_t \qquad \sum_i (1-\theta_{i,t}) a_{i,t} = 0 \qquad \sum_i \theta_{i,t} a_{i,t} = q_t K_t$$

- But q process is implicit so must impose consistency conditions on q to close model:

$$\begin{aligned} q\mu_{q,t} &= \sum_j \frac{\partial q}{\partial \eta_j} \eta_j \mu_{\eta_j,t} + \frac{\partial q}{\partial z} \mu_{z,t} + \frac{\partial q}{\partial K} \mu_{K,t} + \sum_j \frac{\partial^2 \xi_i}{\partial z \partial \eta_j} \eta_j \sigma_{\eta_j,t} \sigma_z \\ &\quad + \frac{1}{2} \sum_{j,j'} \frac{\partial^2 q}{\partial \eta_j \partial \eta_{j'}} \eta_j \eta_{j'} \sigma_{\eta_j,t} \sigma_{\eta_{j'},t} + \frac{1}{2} \frac{\partial^2 q}{\partial z^2} \sigma_z^2 \\ q\sigma_{q,t} &= \sum_j \frac{\partial q}{\partial \eta_j} \eta_j \sigma_{\eta_j,t} + \frac{\partial q}{\partial z} \sigma_{z,t} \end{aligned}$$

Comparison to Models With Existing Solution Techniques

Models	Non-Trivial Blocks 1	2	3	Method
Representative Agent (à la [Lucas, 1978])	simple	NA	simple	Finite difference
Heterogeneous Agents (à la [Krusell and Smith, 1998])	✓	✓	simple	[Gu et al., 2023]
Long-lived assets (à la [Brunnermeier and Sannikov, 2014])	closed-form	low-dim	✓	[Gopalakrishna, 2021]
HA + Long-lived assets	✓	✓	✓	This paper

The Big Decisions

- **Q.** Which equilibrium functions should we approximate with neural networks?

- **Q.** What should be in the loss functions?

Neural Network Approximation

- Let $\boldsymbol{X} := (z, K, (\eta_i)_{i \le I})$ denote the state vector in the economy
- Approximate $(\{\omega_j := c_j/a_j\}_{j \in \{h,e\}}, \sigma_q)$ by neural nets with params $(\{\Theta_{\omega_j}\}_{j \in h,e}, \Theta_q)$:

$$\hat{\omega}_j(\boldsymbol{X}; \Theta_{\omega_j}), \ \forall j \in \{h, e\}, \qquad \hat{\sigma}_q(\boldsymbol{X}; \Theta_q)$$

Neural Network Structure

- At state $\boldsymbol{X}$, the error (or "loss") in the Neural network approximations is given by: (with $\hat{\xi}_j = u_j'(\hat{\omega}_j(\boldsymbol{X}))$ for $j \in \{h, e\}$ and $\hat{\sigma}_q = \hat{\sigma}_q(\boldsymbol{X})$)

$$\mathcal{L}_{\omega_j}(\boldsymbol{X}) = (r - \rho_j)\hat{\xi}_i + \frac{\partial \hat{\xi}_i}{\partial z}\mu_z + \frac{\partial \hat{\xi}_i}{\partial K}(\phi((\phi')^{-1}(q^{-1}))K_t - \delta K_t) + \sum_j \frac{\partial \hat{\xi}_i}{\partial \eta_j}\eta_j \mu_{\eta_j,t}$$

$$+ \sum_j \frac{\partial^2 \hat{\xi}_i}{\partial z \partial \eta_j}\eta_j \sigma_{\eta_j,t}\sigma_z + \frac{1}{2}\frac{\partial^2 \hat{\xi}_i}{\partial z^2}\sigma_z^2 + \frac{1}{2}\sum_{j,j'} \frac{\partial^2 \hat{\xi}_i^2}{\partial \eta_j \partial \eta_{j'}}\eta_j \eta_{j'} \sigma_{\eta_j,t}\sigma_{\eta_{j'},t}, \quad j \in \{h, e\}$$

$$\mathcal{L}_{\sigma}(\boldsymbol{X}) = -q\hat{\sigma}_q + \sum_j \frac{\partial q}{\partial \eta_j}\eta_j \sigma_{\eta_j} + \frac{\partial q}{\partial z}\sigma_z$$

Algorithm ("EMINN" or "Economic Deep Galerkin")

1: Initialize neural networks $\{\hat{\omega}_h, \hat{\omega}_e, \hat{\sigma}_q\}$ with parameters $\{\Theta_{\omega_h}, \Theta_{\omega_e}, \Theta_q\}$.

2: **while** Loss $>$ tolerance **do**

3: Sample N new training points: $\left(\boldsymbol{X}^n = \left(z^n, K^n, (\eta_i)_{i \leq I}^n\right)\right)_{n=1}^N$.

4: Calculate equilibrium at each training point $\boldsymbol{X}^n$ given current $\{\hat{\omega}_h, \hat{\omega}_e, \hat{\sigma}_q\}$:

(a) Compute $(\hat{\omega}_i^n)_{i \leq I}$ using current approximation $\{\hat{\omega}_h, \hat{\omega}_e\}$ evaluated at $\boldsymbol{X}^n$.

(b) Compute q^n and $(\xi_i^n)_{i \leq I}$ using $(\hat{\omega}_i^n)_{i \leq I}$.

(c) Solve for $(\boldsymbol{\theta}^n, \boldsymbol{\sigma}_{\boldsymbol{\eta}}^n, s^n)$ the current approximations for $\{\hat{\omega}_h, \hat{\omega}_e, \hat{\sigma}_q\}$.

(d) Compute μ_η, μ_q, r.

4: Construct loss as: $\hat{\mathcal{L}}(\boldsymbol{X}) = \frac{1}{N} \sum_n |\hat{\mathcal{L}}_{\omega_h}(\boldsymbol{X}^n)| + \frac{1}{N} \sum_n |\hat{\mathcal{L}}_{\omega_e}(\boldsymbol{X}^n)| + \frac{1}{N} \sum_n |\hat{\mathcal{L}}_{\sigma}(\boldsymbol{X}^n)|$

5: Update $\{\Theta_{\omega_h}, \Theta_{\omega_e}, \Theta_q\}$ using ADAM optimizer (stochastic gradient descent to $\downarrow \hat{\mathcal{L}}$).

6: **end while**

Approach Q & A

- Naive approach: approx. (V, c, θ, r, q) by NN, then put FOC and clearing in loss

  ... This is slow/difficult for the neural network to learn so we make simplifications.

- ***Q.*** *Why do we approximate $\omega = c/a$ not V?*
  - Better to approx. $\xi = \partial_a V$ than V so we can easily impose V concavity.
  - Better to approx. $\omega = c/a$, then reconstruct $\xi_h = (\omega \eta q K)^{-\gamma}$ so NN "less non-linear"
  - Easier to fit neural networks to bounded functions and impose shape explicitly.

- ***Q.*** *Do we also need to fit a neural network to the portfolio constraint Ψ (or to θ)?*
  - No, if Ψ is linear or quadratic since portfolio choice can be solved explicitly in ξ
  - Yes, otherwise.
  - Fit neural networks to min. variables needed to make equilibrium calculation step simple.

Approach Q & A

- ***Q.*** *Why do we work in wealth share space?*
  - Well known deep learning difficulty—adding market clearing loss functions creates instability.
  - So, we need to impose market clearing in the sampling.
  - If we sample in the "a" space, then imposing market clearing means restricting a to an $I-1$ dimensional hyperplane that depends upon equilibrium prices. E.g. $\sum_i a_i = qK$
  - Instead, we solve for the equilibrium ω as a function of $(z, K, (\eta_i)_{i\leq I})$, which means capital market clearing is satisfied by $\sum_i \eta_i = 1$.
  - Better to move market clearing conditions out of the loss function.
  - Similar in spirit to the discrete time approach in [Azinovic and Žemlička, 2023]

Approach Q & A

- ***Q.*** *What type of neural network?*
  - We use fully-connected feed-forward type with 4 hidden layers and 32 neurons per layer.
  - We train using an ADAM optimizer with a learning rate of 0.0005 for 1400 iterations.
- ***Q.*** *How do we sample?*
  - Start with uniform sampling with shifted moments of the distribution.
  - Can then move ergodic sampling, if required.
- ***Q.*** *Does the approach work on standard models?*
  - Test it on [Lucas, 1978], [Basak and Cuoco, 1998], [Brunnermeier and Sannikov, 2014].
  - Difference to finite difference solution approximately 1e-4.
- ***Q.*** *How long does it take?*
  - 10 agents: < 10 minutes on laptop.
  - 25 agents: < 20 minutes on laptop.
  - 50 agents: ≈ 1 hour on cluster

Q. How Does Asset Pricing Impact Inequality?

- Difference between the drift of the wealth share of any two households i and j is:

$$\mu_{\eta_j,t} - \mu_{\eta_i,t} = (\theta_{j,t} - \theta_{i,t})(r_{k,t} - r_t - \sigma_{q,t}^2) - (\omega_j - \omega_i) + \frac{\tau\lambda}{I-1}\left(\frac{1}{\eta_{j,t}} - \frac{1}{\eta_{i,t}}\right)$$

1. Participation constraint: means low wealth agents hold less capital and earn less risk premium. E.g. for log utility and quadratic participation cost ($\psi_{i,t} = 0.5\bar{\psi}\sigma^2\theta_{i,t}^2/\eta_{i,t}$):

$$\theta_{i,t} = \frac{k_{i,t}}{a_{i,t}} \approx \frac{r_{k,t} - r_{f,t}}{\sigma_{q,t}^2 + \bar{\psi}\sigma^2/\eta_{i,t}}, \quad i \in \{1, \ldots, I-1\}$$

2. Differential consumption: low wealth agents save more to escape participation constraint.

3. Redistribution: through death (and wealth taxes)

Equilibrium For Different Participation Constraints

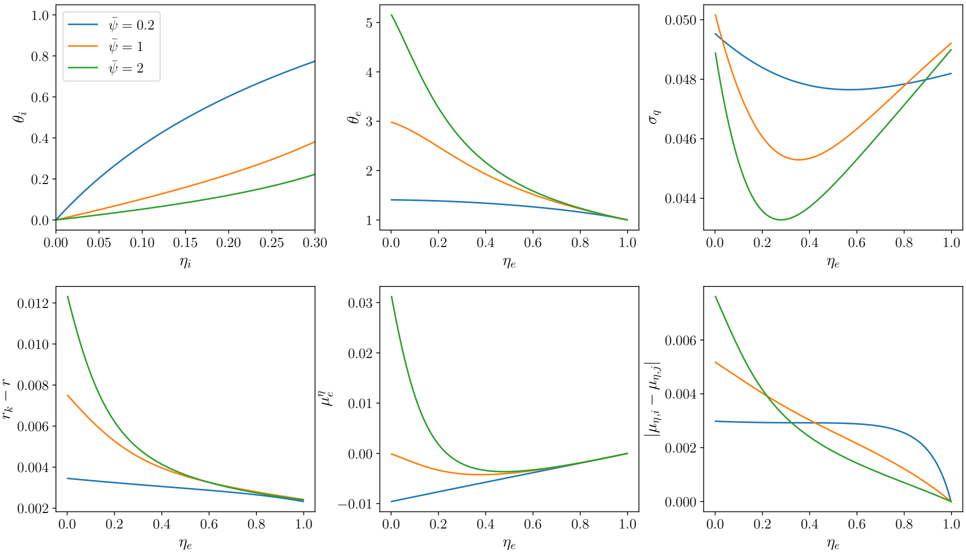


Figure: Equal household wealth distribution. $\rho_e = 0.04, \rho_h = 0.03, \mu = 0.02, \sigma = 0.05.$

Q. How Does Inequality Impact Asset Pricing?

- For log utility and quadratic participation cost, aggregate capital demand is:

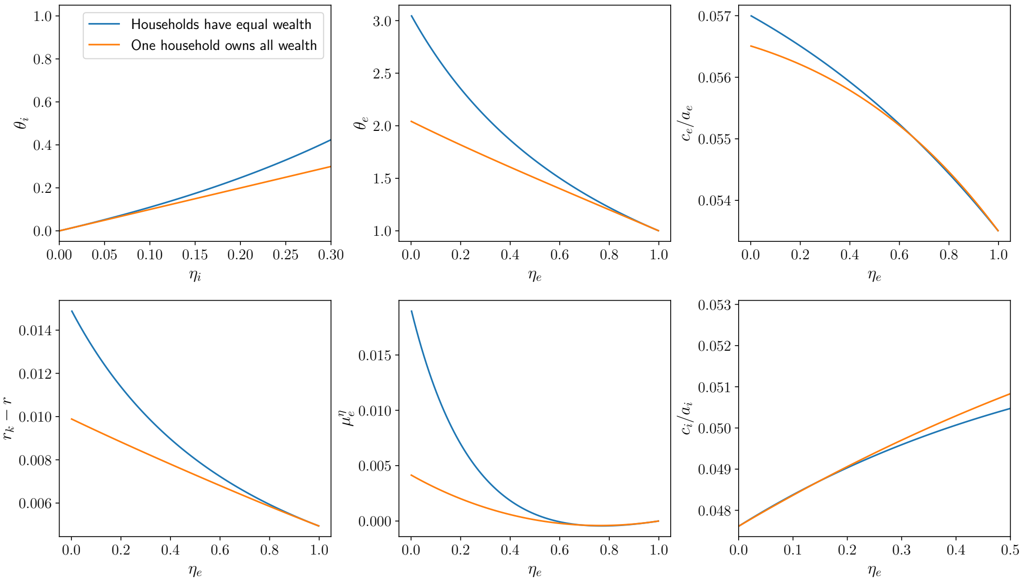
$$\sum_{i=1}^{I-1} \theta_{i,t}\eta_{i,t}A_t + \theta_{e,t}\eta_{e,t}A_t = \left(\sum_{i=1}^{I-1} \frac{\eta_{i,t}^2}{\bar{\psi}\sigma^2 + \sigma_{q,t}^2\eta_i} + \frac{\eta_{I,t}}{\sigma_{q,t}^2} \right) (r_{k,t} - r_{f,t}) q_t K_t$$

- The capital market participation breaks aggregation in household sector.
- More unequal distribution

  $\Rightarrow$ households purchase more capital when expert wealth drops.

  $\Rightarrow$ wealth distribution influences whether household or expects act as "buffer" in recessions.

Equilibrium at Different Wealth Distributions



Conclusion

- *This talk:* showed how we use neural networks to solve continuous time, heterogeneous agent models with search and matching frictions or long-term assets.
- *Practical Lessons:* for continuous time deep learning
    1. Working out the correct sampling approach is very important.
    2. Neural networks have difficulty dealing with inequality constraints.
    3. Enforcing shape constraints and/or rescaling functions is important.
    4. Need tighter tolerance than finite difference.

Approach A: Finite Population

- Replace distribution g_t by finite number of agent $\hat{g}_t := \{x_t^i : i \leq I\}$.
  - Agent $i \leq I$ behaves as if their individual actions do not influence prices.
  - So, their belief is: $\hat{q}_t = \hat{Q}(z_t, \hat{g}_t^{-i})$, where $\hat{g}_t^{-i} := \{x_t^j : j \neq i\}$

- $V(x^i, z, \hat{g})$ solves $(\hat{\mathcal{L}}V)(x^i, z, \hat{g}) = 0$ subject to BCs, where $\hat{\mathcal{L}} := \hat{\mathcal{L}}^h + \hat{\mathcal{L}}^g$

$$
\begin{aligned}
(\hat{\mathcal{L}}^h V)(x^i, z, \hat{g}) &:= (\mathcal{L}^h V)(x^i, z, \hat{g}) \\
(\hat{\mathcal{L}}^g V)(x^i, z, \hat{g}) &:= \sum_{j \leq I} \frac{\partial V}{\partial x^j}(x^i, z, \hat{g}) \mu^x(c^*(x^j, z, \hat{g}), x^j, z, \hat{Q}(z, \hat{g}^{-j})) \\
&\quad + \sum_{j \leq I} \lambda(x^j) \Big(V(x^i, z, \{x^j + \gamma^x(x^j), \hat{g}^{-j}\}) - V(x^i, z, \hat{g}^{-i}) \Big)
\end{aligned}
$$

  - $\hat{\mathcal{L}}^h$ stays the same; $\hat{\mathcal{L}}^g$ changes to capture impact of changes in other agent positions
  - Converges to original model as $I \to \infty$ (see [Carmona, 2020])

Back

Approach B: Projection Onto Basis

- Approximate the distribution $g_t(x)$ by $\sum_{i=1}^{N} \alpha_t^i h^i(x)$, where:
  - α_t^i is a time varying coefficient, $h^i(x)$ is basis function, and
  - Example bases: Indicator Functions, Chebyshev polynomial, Eigenfunctions, ...
  - Distribution characterized by coefficients: $\hat{g}_t := \{\alpha_t^1, ...\alpha_t^N\}$.
  - Substituting $\sum_{i=1}^{N} \alpha_t^i h^i(x)$ into KFE implicitly gives the law of motion for the coefficients:

$$d\alpha_t^i = \hat{\mu}_\alpha^i(z, \hat{g})dt, \quad \text{where } \hat{\mu}_\alpha^i(z, \hat{g}) \text{ solve } \quad \sum_{i=1}^{N} \hat{\mu}_\alpha^i(z, \hat{g}) h^i(x) = \hat{\mathcal{L}}^k \Big[\sum_{i=1}^{N} \alpha_i(t) h^i(x) \Big]$$

- $V(x^i, z, \hat{g})$ solves $(\hat{\mathcal{L}} V)(x^i, z, \hat{g}) = 0$ subject to BCs, where $\hat{\mathcal{L}} := \hat{\mathcal{L}}^h + \hat{\mathcal{L}}^g$:

$$(\hat{\mathcal{L}}^h V)(x, z, \hat{g}) := (\mathcal{L}^h V)(x, z, \hat{g}), \quad (\hat{\mathcal{L}}^g V)(x, z, \hat{g}) := \sum_{i=1}^{N} \hat{\mu}_\alpha^i(z, \hat{g}) \frac{\partial V}{\partial \alpha_i}(x, z, \hat{g})$$

Discrete state space | Eigenvectors

Back

Approach B.1: Project Onto Discrete State Space

- We approximate the distribution by a histogram:
  - Basis is a collection of N^x points: $x_1, \ldots, x_{N^x}$, in $\mathcal{X}$.
  - We approximate g_t by a vector $\alpha_t \in \mathbb{R}^{N^x}$ of mass points at $x_1, \ldots, x_{N^x}$.
- Law of motion of the mass points is the finite difference approximation to the KFE.

Back

Approach B.2: Projection Onto Eigenfunctions

- Let $\{e_i\}_{i\geq 1}$ be eigenfunctions of KFE operator $\hat{\mathcal{L}}^k_{z=0}$ without aggregate shocks:

$$\mathcal{L}^k_{z=0} e_i = \lambda_i e_i, \quad \text{where } \lambda_i \text{ are eigenvalues}$$

  Use finite subset of eigenfunctions of $\hat{\mathcal{L}}^k_{z=0}$ as basis:

$$g_t(x) \approx \sum_{i\leq I} \alpha_t^i e^i(x), \quad \text{so distribution characterized by } \hat{g} = \{\alpha_t^1, \dots, \alpha_t^I\}$$

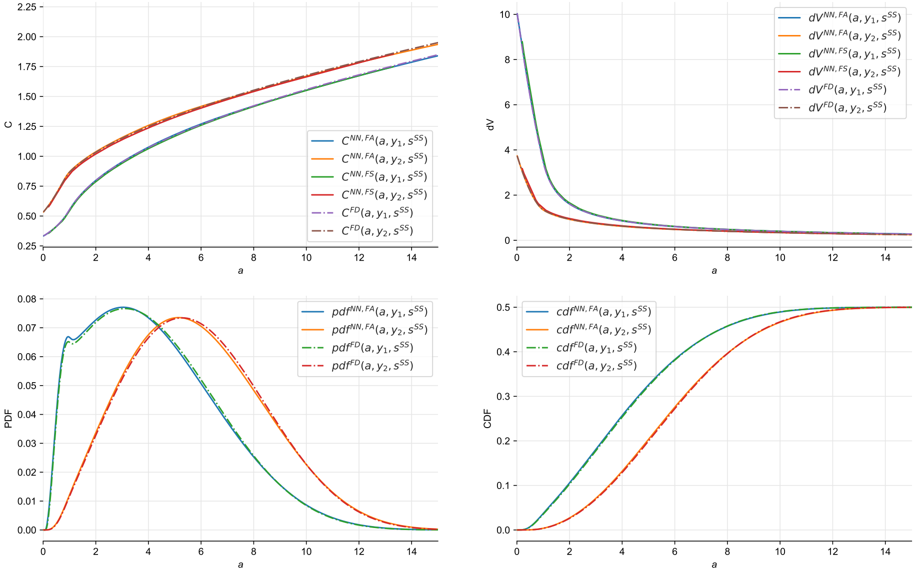
- Drifts of the coefficients $\{\hat{\mu}_\alpha^i\}_{i\leq I}$ satisfy a collection of equations for $i \leq I$:

$$\sum_{i\leq I} \hat{\mu}_\alpha^i \langle e_i, e_j\rangle = \underbrace{\sum_{i\leq I} \alpha_t^i \lambda_i^A \langle e_i, e_j\rangle}_{\text{Weighted } \mathcal{L}^k_{z=0} \text{ eigenvalues}} + \underbrace{\int_{\mathcal{X}} e_j(x) \Big((\mathcal{L}^k_z - \mathcal{L}^k_{z=0}) \Big(\sum_{i\leq I} \alpha_t^i e_i \Big) \Big)(x) dx}_{\text{Weighted difference between } \mathcal{L}^k_z - \mathcal{L}^k_{z=0}}$$

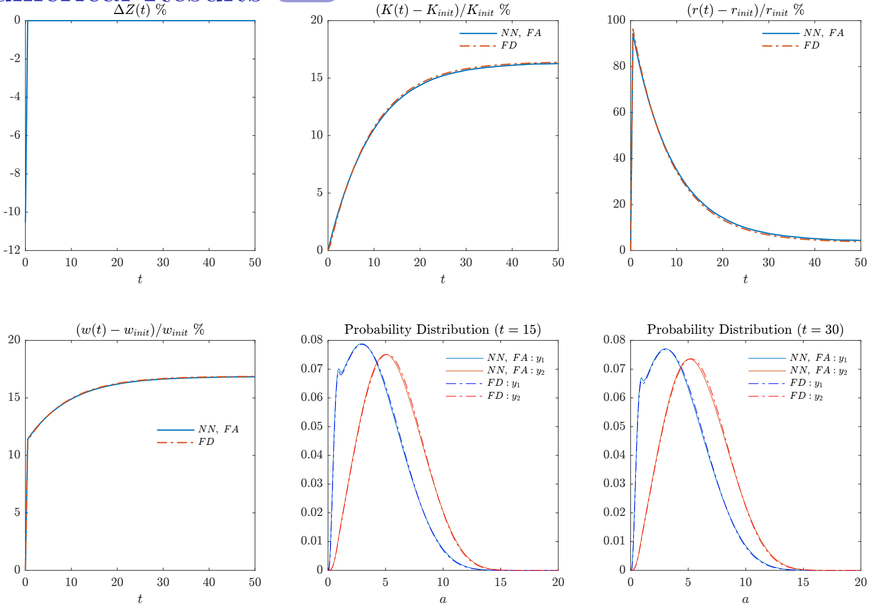
- **Remark:** We approximate operator difference $\hat{\mathcal{L}}^k_z - \hat{\mathcal{L}}^k_{z=0}$:
  - Many papers perturb z or $g(x)$ (e.g. [Cardaliaguet et al., 2015], [Alverez (2023)], [Bilal (2023)])

ABH: Numerical Results

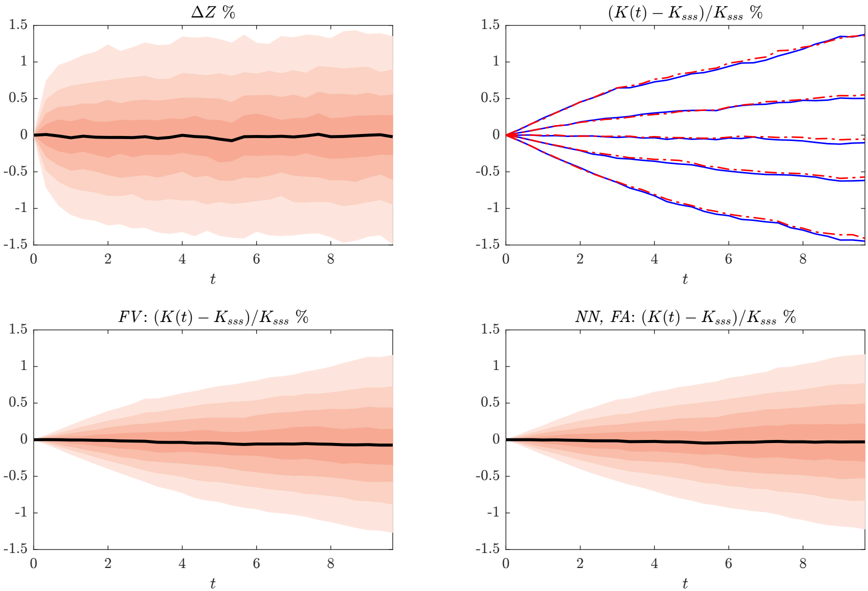
Back



ABH: Numerical Results Back



KS: Numerical Results Back



Approximate Solutions

- Consider HJB equation for the Merton problem (consumption and portfolio choice):

$$\rho V(a) = \max_{c,\theta} u(c) + V'(a)((r + (\bar{R} - r)\theta)a - c) + \frac{1}{2}\sigma^2\theta^2 a^2 V''(a) \tag{2}$$

- Suppose V_0 is the exact solution of Merton's problem, we plug in a scaled solution $k^{-\gamma}V_0$:

$$\rho k^{-\gamma} V_0 = \frac{c^{1-\gamma}k^{1-\gamma}}{1-\gamma} + k^{-\gamma}V_0'((r + (\bar{R} - r)\theta)a - kc) + \frac{1}{2}\sigma^2\theta^2 a^2 k^{-\gamma} V_0''(a) \tag{3}$$

  Which implies that the loss function (with no loss of generality, we use L1 loss here) will be:

$$Loss = \left|\left(k^{1-\gamma} - k^{-\gamma}\right)\right| \cdot \underbrace{\left|-cV_0' + \frac{c^{1-\gamma}}{1-\gamma}\right|}_{\text{Finite value}}$$

  - $\gamma < 1$, no problem because $k^{1-\gamma}$ will explode while $k^{-\gamma}$ vanishes as $k \to \infty$.
  - $\gamma > 1$, a very large k can be problematic because both $k^{1-\gamma}$ and $k^{-\gamma}$ vanish as $k \to \infty$.

- Hence, in the economically relevant case $\gamma > 1$, computer is very good at finding "cheat solution" by simply push value function to be very close to zero.

Back

Example: Projection of Distribution on Chebyshev Polynomials

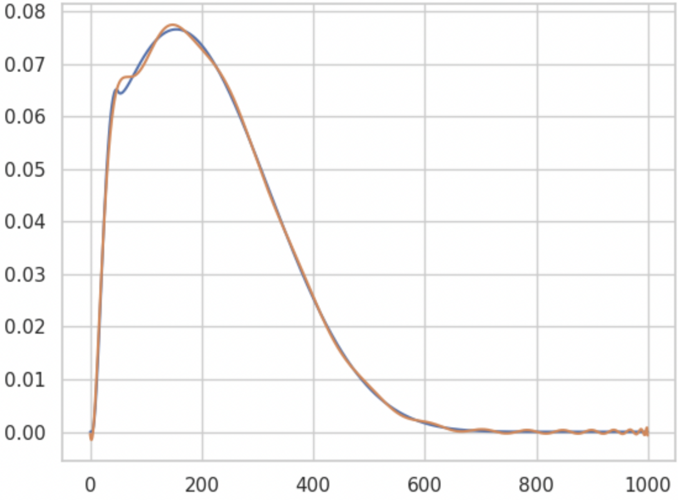


Figure: Capital to Labor Ratio vs borrowing constraint a_{lb}

Example: Projection of Eigenfunctions

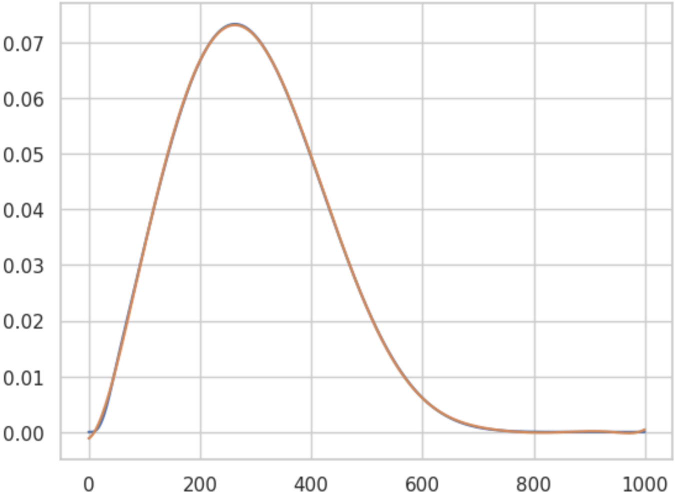


Figure: Capital to Labor Ratio vs borrowing constraint a_{lb}

Recursive Equilibrium Part II: Other Equations

- Hamilton-Jacobi-Bellman equation (HJBE) for employed worker's value $V^e(x, y, z, g)$:

$$\begin{aligned}\rho V^e(x, y, z, g) = w(x, y, z, g) + \delta(x, y)\left(V^u(x, z, g) - V^e(x, y, z, g)\right) \\ + \lambda_{z\tilde{z}}(V^e(x, y, \tilde{z}, g) - V^e(x, y, z, g)) + D_g V^e(x, y, z, g) \cdot \mu^g\end{aligned}$$

- HJBE for a vacant firm's value $V^v(y, z, g)$:

$$\begin{aligned}\rho V^v(y, z, g) = \frac{m(z, g)}{\mathcal{V}(z, g)} \int \alpha(\tilde{x}, y, z, g)(V^p(\tilde{x}, y, z, g) - V^v(y, z, g)) \frac{g^u(\tilde{x})}{\mathcal{U}(z, g)} d\tilde{x} \\ + \lambda_{z\tilde{z}}(V^v(x, \tilde{z}, g) - V^v(x, z, g)) + D_g V^v(y, z, g) \cdot \mu^g\end{aligned}$$

- HJBE for a producing firm's value $V^p(x, y, g)$:

$$\begin{aligned}\rho V^p(x, y, z, g) = zf(x, y) - w(x, \tilde{y}, z, g) + \delta(x, y)(V^v(y, z, g) - V^p(x, y, z, g)) \\ + \lambda_{z\tilde{z}}(V^p(x, y, \tilde{z}, g) - V^p(x, y, z, g)) + D_g V^p(x, y, z, g) \cdot \mu^g\end{aligned}$$

Back

Expert Problem

Given their belief about price processes $(\hat{r}, \hat{q})$, expert i solves:

$$V_{e,t} = \max_{c_i, k_i, \iota_i} \left\{ \mathbb{E}_0 \left[\int_0^\infty e^{-\rho_e t} f(c_t, V_{e,t}) dt \right] \right\}$$

$$s.t. \quad da_{i,t} = (a_{i,t} - k_{i,t})\hat{r}_{i,t} dt + k_{i,t} d\hat{R}_{k,t} - c_{i,t} dt + \tau_t dt$$

where the "felicity" function is:

$$f(c, V) = \frac{1-\gamma}{1-\frac{1}{\varrho}} V \left[\left(\frac{c}{((1-\gamma)V)^{1/(1-\gamma)}} \right)^{1-\frac{1}{\varrho}} - \rho \right]$$

Back

Expert Euler Equations

- Given price processes $(r, r_k, q, \mu_q, \sigma_q)$, expert optimization implies:

Euler equation:
$$\frac{\partial f(c_e, V_e)}{\partial V_e}\xi_h = r\xi_e + \frac{\partial \xi_e}{\partial z}\mu_z + \frac{\partial \xi_e}{\partial K}\mu_K + \frac{1}{2}\frac{\partial^2 \xi_e}{\partial z^2}\sigma_z^2 + \sum_j \frac{\partial \xi_e}{\partial \eta_j}\eta_j \mu_{\eta_j,t}$$
$$+ \sum_j \frac{\partial^2 \xi_e}{\partial z \partial \eta_j}\eta_j \sigma_{\eta_j,t}\sigma_z + \frac{1}{2}\sum_{j,j'} \frac{\partial^2 \xi_e^2}{\partial \eta_j \partial \eta_{j'}}\eta_j \eta_{j'} \sigma_{\eta_j,t}\sigma_{\eta_{j'},t}$$

Consumption FOC:
$$\xi_e = \frac{\partial f(c_e, V_e)}{\partial c_e}$$

Portfolio FOC:
$$\xi_e(r_k - r) = -\left(\frac{\partial \xi_e}{\partial z}\sigma_z + \sum_j \frac{\partial \xi_e}{\partial \eta_j}\sigma_{j,\eta}\right)\sigma_q$$

- where the "felicity" function is:

$$f(c, V) = \frac{1-\gamma}{1-\frac{1}{\varrho}}V\left[\left(\frac{c}{((1-\gamma)V)^{1/(1-\gamma)}}\right)^{1-\frac{1}{\varrho}} - \rho\right]$$

Approximate ω by Neural Network (Feed Forward, Fully Connected)

- Let $\boldsymbol{X} = (z, K, \{\eta_i\}_{i\leq I})$. We approximate surplus $\omega(\boldsymbol{X})$ by neural network with form:

$$
\begin{aligned}
\boldsymbol{h}^{(1)} &= \phi^{(1)}(W^{(1)}\boldsymbol{X} + \boldsymbol{b}^{(1)}) && \ldots\text{Hidden layer 1}\\
\boldsymbol{h}^{(2)} &= \phi^{(2)}(W^{(2)}\boldsymbol{h}^{(1)} + \boldsymbol{b}^{(2)}) && \ldots\text{Hidden layer 2}\\
&\vdots\\
\boldsymbol{h}^{(H)} &= \phi^{(H)}(W^{(H)}\boldsymbol{h}^{(H-1)} + \boldsymbol{b}^{(H)}) && \ldots\text{Hidden layer H}\\
\omega &= \sigma(\boldsymbol{h}^{(H)}) && \ldots\text{Variable}
\end{aligned}
$$

- Terminology (our parameter choices are in blue):
  - H: is the number of *hidden layers*, $(H = 4)$
  - Length of vector $\boldsymbol{h}^{(i)}$: number of *neurons* in hidden layer i, $(Length = 32)$
  - $\phi^{(i)}$: is the *activation function* for hidden layer i, $(\phi^i = tanh)$
  - σ: is the *activation function* for the final layer, $(\sigma = tanh)$
  - $\boldsymbol{\Theta} = (W^1, \ldots W^{(H)}, b^{(1)}, \ldots, b^{(H)})$ are the *parameters*,

Back

Roadmap

Applications to More Difficult Models

- Search and Matching with Heterogeneous Agents and Aggregate Shocks
- Macro-Finance With Portfolio Choice and Long-Lived Assets

Additional Appendices

Solutions to example models

Macroprudential Policy and Inequality

Three Testable Models

Model	Layers	Neurons	Learning Rate	Error
"As-if" Complete Model	4	64	0.001	1.0×10^{-5}
[Basak and Cuoco, 1998]	5	64	0.001	4.9×10^{-4}
[?]	5	32	0.001	7.0×10^{-5}

Representative Agent Model ([Lucas, 1978])

Suppose there are no financial frictions: $\Psi(a_{i,t}, b_{i,t}) = 0$ and no investment.

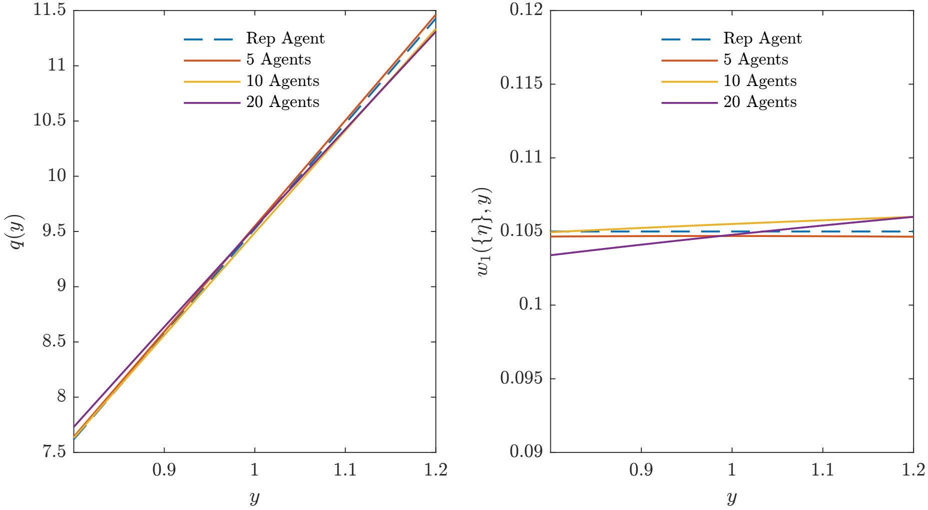
In this case, households are identical so there is a "representative agent".

The model has closed form solution:

$$q(y) = \frac{y}{\rho + (\gamma - 1)\mu - \frac{1}{2}\gamma(\gamma - 1)\sigma^2}$$

$$\omega(y) = \left[\rho + (\gamma - 1)\mu - \frac{1}{2}\gamma(\gamma - 1)\sigma^2\right]$$

[Lucas, 1978] Model solution. MSE: $< 10^{-4}$



As-if Complete Market Model, $\gamma = 5$, $\mu = 2\%$, $\sigma = 5\%$, $\rho = 5\%$.

Limited Participation Model ([Basak and Cuoco, 1998])

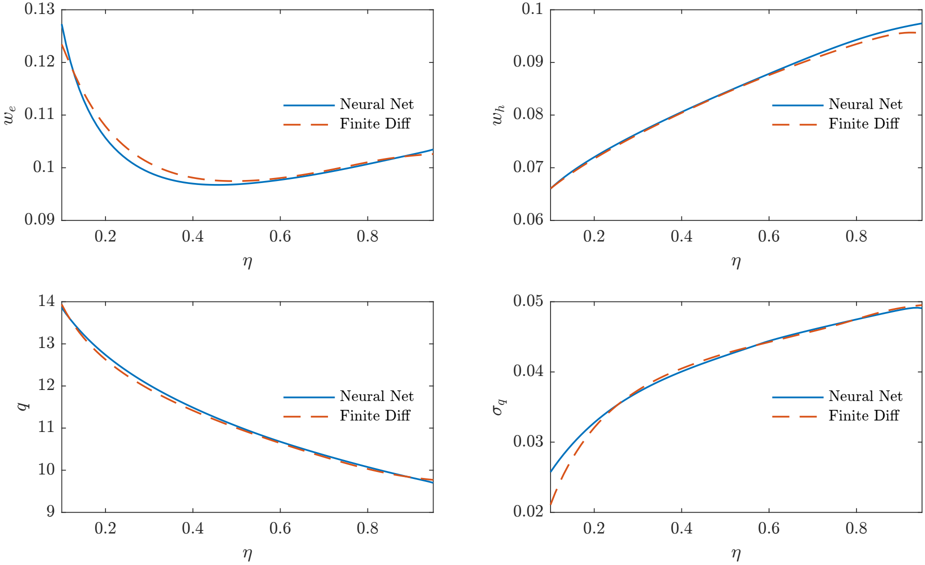
Two types of agents: experts (e) and households (h).

Expert sector can hold stocks and bonds.

Household sector can only hold bonds: $\Psi_h(a_{h,t}, b_{h,t}) = a_{h,t} - b_{h,t} = 0$.

State space is (y, η), where η is expert's wealth share.

[Basak and Cuoco, 1998] Model solution. L2 Loss: $< 10^{-5}$



2 Agents Limited Participation Model, $\gamma = 5, \rho_e = \rho_h = 5\%$, $\mu = 2\%$, $\sigma = 5\%$.

Productivity Gap Model ([?])

Two types of agents: experts (e) and households (h).

We allow households to hold capital but in a less productive way. The productivity of experts and households is z_h, z_e $(z_h < z_e)$ respectively. Their relative risk-aversion are both γ.

Output grows at exogenous drift $\mu_y = y\mu$, volatilty $y\sigma$, and experts cannot issse outside equities.

State space is (y, η), where η is expert's wealth share.

[?] Model solution. L2 Loss: $< 10^{-5}$

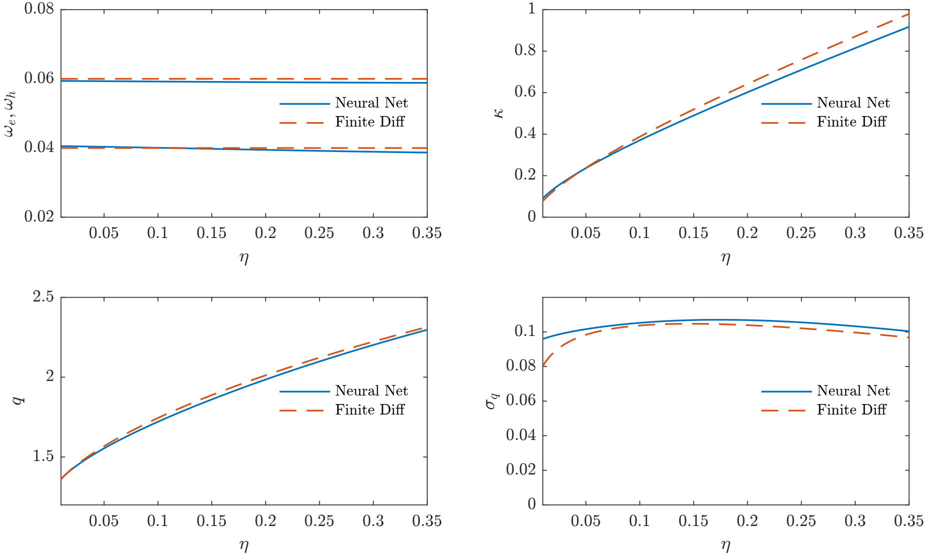
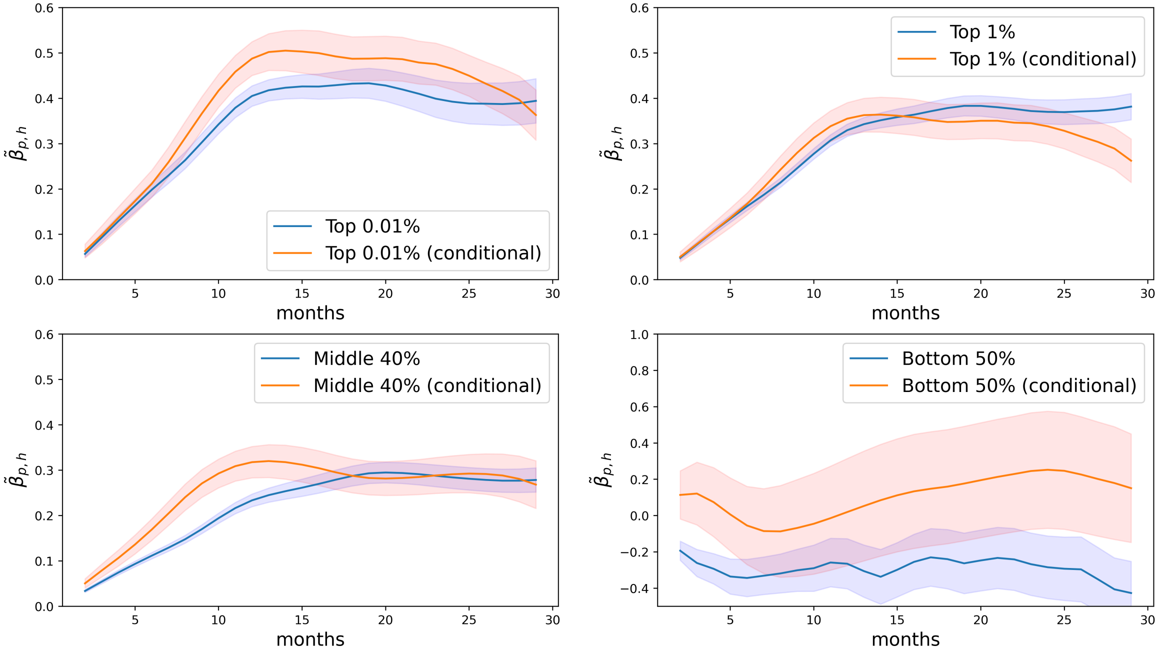


Figure: Solution to the model with productivity gap.



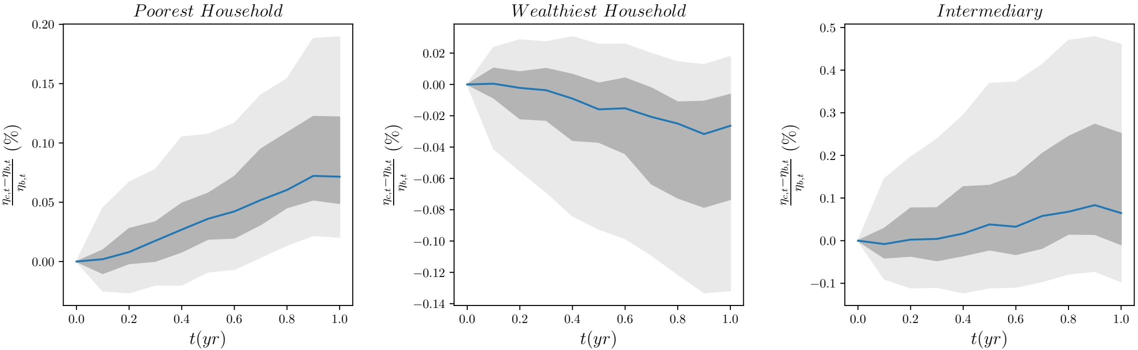
Top 0.01%
Top 0.01% (conditional)
Top 1%
Top 1% (conditional)
Middle 40%
Middle 40% (conditional)
Bottom 50%
Bottom 50% (conditional)
$\tilde{\beta}_{p,h}$
months

Table of Contents

Distributional Impact of Macroprudential Policy

- We introduce an exogenous leverage constraint on financial expert: $\theta_e \leq \bar{\ell}$.
- Then simulate a collection of recession paths and track the distribution evolution.

Looser Leverage Constraint ($\theta \leq 2.0$)



Tighter Leverage Constraint ($\theta \leq 1.5$)