Working Paper Series 802
(ISSN 2788-0443)

Deep Learning in the Sequence Space

Marlon Azinovic-Yang
Jan Žemlička

CERGE-EI
Prague, October 2025

Deep Learning in the Sequence Space*

Marlon Azinovic-Yang†
University of North Carolina at Chapel Hill

Jan Žemlička‡
University of Zurich and Swiss Finance Institute

First version: September 9, 2025
This version: October 10, 2025

Abstract

We develop a deep learning algorithm for approximating functional rational expectations equilibria of dynamic stochastic economies in the sequence space. We use deep neural networks to parameterize equilibrium objects of the economy as a function of truncated histories of exogenous shocks. We train the neural networks to fulfill all equilibrium conditions along simulated paths of the economy. To illustrate the performance of our method, we solve three economies of increasing complexity: the stochastic growth model, a high-dimensional overlapping generations economy with multiple sources of aggregate risk, and finally an economy where households and firms face uninsurable idiosyncratic risk, shocks to aggregate productivity, and shocks to idiosyncratic and aggregate volatility. Furthermore, we show how to design practical neural policy function architectures that guarantee monotonicity of the predicted policies, facilitating the use of the endogenous grid method to simplify parts of our algorithm.

JEL classification: C61, C63, C68, D52, E32.

Keywords: deep learning, heterogeneous firms, heterogeneous households, overlapping generations, deep neural networks, global solution method, life-cycle, occasionally binding constraints.

1 Introduction

In this paper, we bring deep learning solution methods to the sequence space and propose a new global method for computing equilibria in economies with aggregate risk.[1] Exploiting the ergodicity property of a large class of dynamic economies, we approximate the aggregate state vector using a

*This paper has greatly benefited from numerous discussions with Felix Kubler, Yucheng Yang and Simon Scheidegger. We are also grateful for the comments of participants at the Advances in Computational Economics and Finance seminar at the University of Zurich, the Macro Reading Group at UNC Chapel Hill, and the 2025 Torino Conference on Machine Learning for Economics and Finance as well as for helpful discussions with Can Tian, Jaden Chen, Jeppe Druedahl, William Jungerman, Yasutaka Koike-Mori, Jonathan Payne, Stanislav Rabinovich, and Jacob Røpke. Žemlička gratefully acknowledges support from the Swiss National Science Foundation and the kind hospitality of the Department of Economics at Princeton University.

†Email: marlonay@unc.edu

‡Email: jan.zemlicka@df.uzh.ch

[1] We use the expression "sequence space" liberally, referring to the space of truncated sequences of aggregate shocks, not (only) the space of perfect foresight sequences. To be more precise, one could alternatively refer to our approach as moving-average as opposed to sequence-space approach.

truncated sequence of aggregate shocks. We use deep neural networks to parameterize the mapping from truncated histories of aggregate shocks to equilibrium objects of interest, and optimize them using stochastic gradient descent to minimize equilibrium conditions error over the simulated ergodic set of the economy.

We make four distinct contributions to the literature. First, we show, how deep learning can be used to compute global approximations to equilibria of dynamic stochastic economies in the sequence space. Second, we show, how to construct neural network architectures that predict approximate policy functions, which are guaranteed to satisfy ex-ante known monotonicity and concavity properties. Third, based on our monotonicity-preserving architectures, we show, how to obtain supervised learning targets for policy and price functions, using the endogenous gridpoint method of Carroll (2006). Fourth, we illustrate the rich applicability and accuracy of our method by solving a challenging overlapping generations model as well as heterogeneous agent economy with heterogeneous firms and households. To the best of our knowledge, we are the first to obtain a global solution to an economy featuring incomplete markets and household heterogeneity in the spirit of Imrohoroğlu (1989)-Bewley (1977)-Huggett (1993)-Aiyagari (1994)-Krusell and Smith (1998), together with firm heterogeneity in the spirit of Khan and Thomas (2008) and Bloom et al. (2018).

As an example, consider the model by Krusell and Smith (1998) and let z_t denote the realization of TFP in period t, and let $z^t := (z_0, z_1, \dots, z_t)$. The infinite horizon set-up of the Krusell and Smith (1998) economy implies the length of the history sequence is also infinite. The state of the economy in a Markov equilibrium in the Krusell and Smith (1998) economy is the current level of TFP z_t, together with the cross-sectional distribution of wealth and income. Let e_t^i denote the idiosyncratic productivity of the household i in period t and let k_t^i denote their capital holdings. Let $\mu_t = \mu_t(e, k)$ denote the cross-sectional distribution. Since the distribution, even when discretized into a histogram as in Young (2010), is a high-dimensional object, this poses a long standing challenge for traditional numeric methods. The approach of Krusell and Smith (1998) is to approximate the cross-sectional distribution μ_t with a low-dimensional statistic such as aggregate capital K_t. Instead of approximating the state vector using its moments or a histogram, we approximate the aggregate state vector using *sequence-space* objects. In particular, we propose to approximate the aggregate state of the economy by a truncated history of exogenous aggregate shocks. In the context of the Krusell and Smith (1998) model, for example, let $z_T^t = (z_{t-T}, z_{t-T+1}, \dots, z_{t-1}, z_t)$ denote the last T realizations of TFP. Instead of computing the equilibrium function $f^{\text{approx}}(z_t, K_t) \approx f(z_t, \mu_t)$ (or $f^{\text{approx}}(z_t, \mu_t^{\text{histogram}}) \approx f(z_t, \mu_t)$), we propose computing equilibrium functions $f^{\text{approx}}(z_t^T) \approx f(z_t, \mu_t)$. For any $T < \infty$ our approach implies a truncation error. However, in economies featuring ergodic dynamics, this truncation error converges to zero as $T \to \infty$. Hence, we can approximate the information contained in the aggregate state vector arbitrarily well by choosing T sufficiently large. Consequently, our function approximators need to be able to handle large T, implying a high-dimensional input z_t^T, potentially even higher dimensional than a histogram of the wealth distribution. We will refer to our approach, which uses a truncated history of aggregate shocks as input as *sequence-space approach* and to the standard approach (using, for example, the wealth distribution or a lower dimensional sufficient statistic such as standard moments, as input to the equilibrium functions) as *state-space approach*. We use neural

networks to approximate all equilibrium functions that need to be solved for. Like polynomials, neural networks are universal function approximators (Hornik et al., 1989). More importantly however, neural networks have shown great promise in ameliorating the curse of dimensionality (Bellman, 1961) associated high-dimensional function approximation problems, such as solving partial differential equations (see, *e.g.* Grohs et al., 2018; Jentzen et al., 2018) or computing equilibria in dynamic economies (see, *e.g.* Azinovic et al., 2022; Maliar et al., 2021; Kahou et al., 2021; Gu et al., 2023).[2]

The sequence-space formulation has two main advantages in the context of deep learning solution methods. First, the sequence-space formulation is potentially more parsimonious, especially in economies with rich cross-sectional heterogeneity.[3] Second, using truncated shock histories as the only network input mitigates a dangerous feedback loop present in simulation-based deep learning solution methods. While the distribution of shock histories is exogeneous and fixed,[4] the distribution of exogenous states moves during the training as the algorithm updates the approximate policy function. As discussed in Azinovic et al. (2022), a large update policy function might shift the distribution of training states into areas that have been previously only sparsely covered, causing large sample error, and correspondingly large gradients, potentially inducing further jumps in the policy function. While our algorithm still relies on simulating endogenous states an an input for evaluating loss function[5] those do not enter as an input into neural networks parameterizing the core equilibrium objects of the economy.

The second contribution of our paper is that we show how to construct neural network architectures that guarantee monotonicity or even concavity of the resulting approximator with respect to a chosen set of inputs. Rather than directly approximating individual policy function using a neural network that takes the aggregate and idiosyncratic state as input,[6] we build on the operator learning approach of Zhong (2023). A simple operator network maps the sequence of aggregate shocks to a policy functions over idiosyncratic state variables only. Hence, the neural network predicts different idiosyncratic policy functions for different aggregate states. For the Krusell and Smith (1998) economy, for example, the neural network predicts a consumption function based on the realized history of aggregate shocks, which takes only individual productivity and asset holdings as input.

This approach allows us to ensure properties of policy functions, such as monotonicity and concavity in the idiosyncratic asset holdings by construction. Being able to guarantee such properties has several advantages. First, in the spirit of economics-inspired neural networks, we encode known properties of the equilibrium policies directly into the architecture of the neural network, so the network does not have learn something which is known ex-ante, rendering the training procedure more robust. Second, in the context of solving for household policies, an always monotone consumption function enables us to apply the method of endogeneous gridpoints of Carroll (2006), and to train the consumption function in a supervised way, enhancing the stability of the training procedure.

[2] Section 2 provides a more detailed discussion of the related literature in economics.

[3] e.g. The state-space formulation of the workhorse HANK economy features a state vector that includes a cross-sectional distribution of households over idiosyncratic shocks as well as liquid and illiquid assets. A tensor-product discretization of such a distribution might easily generate tens of thousands of aggregate state variables, whereas a highly accurate sequence-space representation could be obtained using a few hundred of shock lags at quarterly frequency.

[4] Because the distribution of fundamental shocks is just a primitive of the economic model at hand.

[5] e.g. for evaluation of marginal product of capital

[6] As in Azinovic et al. (2022) or Maliar et al. (2021).

To illustrate the broad applicability of our method, we proceed by applying it to solve three models of increasing complexity. The models we solve include economies featuring multiple discrete as well as continuous continuous shocks aggregate shocks, Markov-switching regimes as well as uncertainty shocks. Furthermore, we show that the approach is compatible with path-following model transformations and market-clearing layers, as introduced in Azinovic and Zemlicka (2024).

2 Literature

The idea of approximating the endogenous aggregate state by a truncated sequence of aggregate shocks was first introduced by Chien et al. (2011) in the context of an endowment economy with heterogeneous agents and with a single aggregate shock that takes two discrete values. In their setting, they find that considering the history of the past five aggregate shocks is enough to yield a high-accuracy approximation. In the context of production economies with aggregate risk, Lee (2025) relies on truncated histories of aggregate shocks to summarize the endogenous aggregate state. The underlying idea of the repeated transition method (RTM) of Lee (2025) is that in an ergodic economy, all possible states will be realized along a sufficiently long simulation path. Lee (2025) uses this fact to construct the expected continuation value function for the Bellman equations of the agents in the economy.

Our contribution relative to this literature is generality. While those two papers also develop sequence based global solution methods, the applicability of these methods has limitations that does not pose a significant challenge for our method. Namely, the algorithm of Chien et al. (2011) struggles to resolve economies featuring stronger persistence, such as production economies, because strong persistence implies the need to keep track of long history of shocks. The RTM approach of Lee (2025) can handle a rich set of environments, including production economies, however, its main bottleneck lies in handling economies with large number of different aggregate shocks. At each iteration the RTM algorithm has to construct the continuation value function for every *period*. To do so, the algorithm has to split the simulated path into partitions sorted by the realized value of aggregate shock, and then to search through each partition for a realization where the aggregate endogenous state (e.g. wealth distribution) is closest to the *next period* distribution with respect to some norm. This structure implies that the RTM algorithm works the best in environments with a small number of discrete aggregate shock. Complex stochastic processes that can not be approximated using a parsimonious Markov chain pose a challenge for RTM, as the length of simulated path required for ergodic representation grows with the cardinality of shock discretization. In contrast, we demonstrate that our method can efficiently solve rich economies featuring production and complex driving stochastic processes. Besides that, our algorithm can handle discrete as well as continuous shocks. Furthermore, neural networks can learn from a large number of simultaneous trajectories, allowing for significant acceleration on modern massively parallel computing architectures.

For applications where a local method is sufficient, methods based on linearization in the sequence space have recently become the dominant method of choice for heterogeneous agent models in macroeconomics following the game-changing work by Boppart et al. (2018) and Auclert et al. (2021). Boppart et al. (2018) and Auclert et al. (2021) introduce local methods, which linearize

heterogeneous agent models with respect to aggregates, in the sequence space. These methods, together with the accompanying libraries,[7] have substantially increased the tractability of heterogeneous agent models with aggregate risk, rendering a wide set of previously intractable models computationally tractable.[8] Our method complements this literature by providing a global method, which is hence applicable for models with larger shocks and without certainty equivalence with respect to aggregate risk. The cost of having a global method is that the runtime is substantially longer in comparison to local methods, such as the sequence-space Jacobian method by Auclert et al. (2021).

The toolbox that enables us to approximate functions on a very high dimensional domain of histories of possible economic shocks is deep learning. Deep learning as a tool to compute equilibria in economic models dates back to Duffy and McNelis (2001), who use a shallow neural network to solve a stochastic growth model with using a parameterized expectations algorithm. Norets (2012) makes use of neural networks in the context of discrete-state dynamic programming. Closer to this paper Azinovic et al. (2022) and Maliar et al. (2021) use neural networks to approximate equilibrium price and policy functions, and train them to satisfy equilibrium conditions, such as first-order optimality conditions, Bellman equations, and market clearing conditions along the simulated paths of the economy. Azinovic and Zemlicka (2024) extend these methods by introducing market clearing layers[9] and step-wise model transformations to robustly solve models with portfolio choice. Kase et al. (2023) show how to use these methods to compute the solutions of economic models for ranges of economic parameters, which are then estimated with maximum likelihood. Han et al. (2022) and Kahou et al. (2021) introduce symmetry preserving neural network architectures and Valaitis and Villa (2024) use deep learning within a parameterized expectations algorithm. Fernández-Villaverde et al. (2023) use neural networks for a nonlinear forecasting rule within a Krusell and Smith (1998) algorithm. Kahou et al. (2022) investigate the relationship between transversailty conditions and solutions found by a deep-learning based algorithm. Druedahl and Ropke (2025) use deep learning to compute optimal choices in a finite horizon lifecycle models with up to eight durable goods and show how deep learning can be applied to models with discrete as well as continuous choices.

Zhong (2023) introduces operator learning, which we build on in the section 5. While previous deep learning approaches used neural network to directly parameterize mapping from a combination of an aggregate and idiosyncratic state to individual choices, Zhong (2023) uses neural network to approximate the mapping from an aggregate state into a function which then returns individual choices as a function of individual state. We extend the operator learning method of Zhong (2023) by showing how to encode monotonicity or concavity of the predicted functions into the network architecture. Sun (2025a) uses neural networks to approximate continuation values in combination with otherwise standard methods. Neural network based solution methods in continuous time are developed or applied in Duarte (2018), Sauzet (2021), Gopalakrishna (2021), Gu et al. (2023), Gopalakrishna et al. (2024) and Payne et al. (2024).[10]

[7] See `https://github.com/shade-econ/sequence-jacobian` for a user friendly library for the method in Auclert et al. (2021).

[8] See also Reiter (2009) and Bayer and Luetticke (2020) for linearization in aggregate variables in the state space.

[9] Market clearing layers are neural network architectures, designed such that an prediction by the neural network is always consistent with market clearing.

[10] Additional applications of deep learning based solution methods in variety of settings include Folini et al. (2025)

Our contribution relative to the existing literature on deep learning solution methods is twofold. First, we show that deep learning can be efficiently used to construct global approximations to the functional rational expectations equilibria in the sequence space. Second, we construct shape-preserving neural network operator network architectures that allow us to incorporate known shape properties of certain equilibrium functions (e.g. monotonicity and concavity of consumption function in canonical consumption-savings problems) directly into the neural network structure.[11] The ability to guarantee monotonicity of the consumption function allows us to apply the method of endogenous gridpoints of Carroll (2006) and hence simplify the solution algorithm for a large class of models featuring standard consumption-savings household problems.

3 Algorithm and first application

To introduce the algorithm we first apply it to the canonical stochastic growth model of Brock and Mirman (1972). For completeness and clarity the model description is provided section 3.1, and our solution method is introduced in section 3.2.

3.1 Simplest illustrative model

We first explain our proposed algorithm by applying it to the simplest workhorse economy with aggregate risk - the stochastic growth model of Brock and Mirman (1972). This model can be accurately solved using conventional methods, so this section does not aim to claim a computational achievement of our method. Instead, it serves to lay out the proposed algorithm as transparently as possible, which is the easiest to do using a well-known work-horse model with well-known solution. In sections 4 and 5 we then apply our algorithm in settings that pose a substantial, if not prohibitive challenge for presently available numerical methods.

Time is infinite and discrete, $t = 0, 1, \ldots$. We consider an infinitely lived representative household that derives utility $u(C_t)$ from consumption C_t. The representative household owns the capital stock K_t and inelastically supplies a constant amount of efficient units of labor $L = 1$ at the equilibrium wage w_t. The single good in the economy is produced by a representative firm endowed with a Cobb-Douglas production technology

$$Y_t = A_t K_t^{\alpha} L^{1-\alpha}, \tag{1}$$

where A_t denotes stochastic total factor productivity, that evolves according to an AR(1) process

$$\log(A_t) = \rho^A \log(A_{t-1}) + \sigma^A \epsilon_t^A, \tag{2}$$

and Friedl et al. (2023) (climate economics), Carvalho et al. (2025) (production networks), Duarte et al. (2021) (partial equilibrium finite horizon model with rich asset choice), Bretscher et al. (2022) (international business cycle), Sun (2025b) (spatial economics), Jungerman (2023) (monopsony power in the labor market) and Adenbaum et al. (2024) (Bewley (1977) type economy).

[11] The idea of shape-preservation in functional approximation methods for dynamic economies was introduced by Judd and Solnick (1994), and further developed by Cai and Judd (2012).

where $\epsilon_t^A \sim \mathcal{N}(0,1)$. The firm rents capital K_t and efficient units of labor L on competitive spot markets and pays a rental rate $r_t^K = \alpha A_t K_t^{\alpha-1} L^{\alpha}$ on capital and a wage $w_t = (1-\alpha) A_t K_t^{\alpha} L^{-\alpha}$ for efficient units of labor. In each period, the representative household chooses how much to consume and how much to invest in capital. Capital evolves according to $K_{t+1} = (1-\delta)K_t + I_t$, where I_t denotes the investment in capital. The households problem is given by

$$\max_{\{K_t\}_{t=1}^{\infty}} \mathrm{E}\left[\sum_{t=0}^{\infty} \beta^t u(C_t)\right] \tag{3}$$

$$\text{subject to :}$$

$$C_t = L w_t + (1 - \delta + r_t^K) K_t - K_{t+1}.$$

The Euler equation, which characterizes the equilibrium together with a transversality condition, is given by

$$u'(C_t) = \mathrm{E}\left[\beta u'(C_{t+1})(1 - \delta + r_{t+1}^K)\right] \tag{4}$$

$$\Leftrightarrow 0 = \frac{(u')^{-1}\left(\mathrm{E}\left[\beta u'(C_{t+1})(1 - \delta + r_{t+1}^K)\right]\right)}{C_t} - 1. \tag{5}$$

Where the Euler equation (5) is rearranged, such that errors in the optimality condition can be interpreted as relative consumption errors. This allows for an implicit, yet interpretable accuracy measure (see Judd, 1998). Traditionally, the model would be solved using recursive methods, where the state of the economy $\mathbf{X}_t^{\text{state}} := [A_t, K_t] \in \mathbb{R}^2$ is given by the current level of TFP together with the current capital stock. The solution to the model is a policy function $\pi(\mathbf{X}_t^{\text{state}}) = K_{t+1}$, which maps the state of the economy to the endogenous variables of interest, in this case the capital stock for the next period.[12]

3.2 Algorithm

3.2.1 Main idea

Our algorithm uses a deep neural network to approximate the policy function. Let $\mathcal{N}_{\boldsymbol{\rho}}$ denote a neural network with trainable parameters $\boldsymbol{\rho}$. Following the *Deep Equilibrium Nets* algorithm (or similar algorithms often used in the literature) the goal is to approximate the policy function by a neural network. This would mean to find neural network parameters $\boldsymbol{\rho}$, such that for all states

$$\mathcal{N}_{\boldsymbol{\rho}}(\mathbf{X}_t^{\text{state}}) \approx \pi(\mathbf{X}_t^{\text{state}}) = K_{t+1}. \tag{6}$$

The main idea of the algorithm we propose in this paper is to train the neural network to predict policies not based on $\mathbf{X}^{\text{state}} = [A_t, K_t] \in \mathbb{R}^2$ but instead purely based on a truncated history of

[12] There are other equivalent policies, like a policy for investment or consumption, which together the budget constraint and the law of motion for capital lead to the same choice for capital stock in the next period.

shocks, $\mathbf{X}^{\text{seq},T} := [A_{t-T+1}, A_{t-T+2}, \ldots, A_{t-1}, A_t] \in \mathbb{R}^T$, such that

$$\mathcal{N}_{\boldsymbol{\rho}}(\mathbf{X}_t^{\text{seq},T}) \approx \pi(\mathbf{X}_t^{\text{state}}) = K_{t+1}. \tag{7}$$

At first sight, using the history of shock as an input may look like a step in the wrong direction for two reasons: first, for $T < \infty$ the sequence of shocks is only an approximately sufficient statistic for the state of the economy. Hence, by truncating the history, we introduce a limit to the precision that even a perfectly trained neural network could attain. Second, the dimensionality of $\mathbf{X}^{\text{seq},T}$ is sometimes larger than the dimensionality of $\mathbf{X}^{\text{state}}$.[13] Therefore, the sequence space approach would not work well together with approximation methods, for which a low-to-medium dimensional domain is pivotal, such as, for example, (adapative) sparse grids.[14] As we illustrate in this paper, however, the sequence approximation to the endogenous aggregate state is promising when used together with deep neural networks as function approximators.

3.2.2 Sufficiency of the truncated history of shocks

For our method to work, it must be that the truncated sequence of shocks is an approximate sufficient statistic for the endogenous aggregate state of the economy, in this case capital K_t. This can only be the case if the influence of capital at period $t-T$, K_{t-T}, on capital in period t, K_t, is vanishing as $T \to \infty$, *i.e.* our method relies on the ergodicity property of the underlying economy.

To see that this condition indeed holds in our stochastic growth economy, we now consider a special case with full depreciation $\delta = 1$ and logarithmic utility $u(C) := \log(C)$. In this case, the model admits a closed form solution with $K_{t+1} = \alpha\beta A_t K_t^\alpha$. Taking log on both sides, we obtain $\log(K_{t+1}) = \log(\alpha\beta) + \log(A_t) + \alpha \log(K_t)$. Iterating forward, we obtain

$$\begin{aligned}
\log(K_t) &= \log(\alpha\beta) + \log(A_{t-1}) + \alpha \log(K_{t-1}) \\
&= \log(\alpha\beta) + \log(A_{t-1}) + \alpha(\log(\alpha\beta) + \log(A_{t-2}) + \alpha \log(K_{t-2})) \\
&= \log(\alpha\beta) + \log(A_{t-1}) + \alpha(\log(\alpha\beta) + \log(A_{t-2}) + \alpha(\log(\alpha\beta) + \log(A_{t-3}) + \alpha \log(K_{t-3}))) \\
&= \cdots = \underbrace{f(\alpha, \beta, A_{t-1}, A_{t-2}, A_{t-3}, \ldots, A_{t-T})}_{\text{function of } \alpha,\ \beta \text{ and the last } T \text{ productivity values}} + \underbrace{\alpha^T \log(K_{t-T})}_{\text{truncation error}},
\end{aligned} \tag{8}$$

where the function f depends only on the parameters α, β and the truncated history of the last T productivity values. The value of capital at $t-T$ only affects the value of capital with a coefficient of α^T, where $\alpha < 1$ is the share of capital in production, typically around 0.3. The truncation error therefore vanishes exponentially at the rate α.

As an alternative to using lagged values of productivity $A_{t-T}, A_{t-T+1}, \ldots, A_{t-1}$, we can use a

[13] For example in this simple stochastic growth economy.

[14] See Krueger and Kubler (2004) for sparse grids and Brumm and Scheidegger (2017) for adaptive sparse grids.

truncated history of innovations to productivity $\epsilon_{t-T}, \epsilon_{t-T+1}, \ldots, \epsilon_{t-1}$. This is because we have that

$$\begin{aligned}\log(A_t) &= \rho \log(A_{t-1}) + \sigma\epsilon_t = \rho(\rho \log(A_{t-2}) + \sigma\epsilon_{t-1}) + \sigma\epsilon_t \\ &= \underbrace{g(\rho, \sigma, \epsilon_{t-T}, \ldots, \epsilon_t)}_{\text{function of } \rho,\ \sigma \text{ and } T+1 \text{ last innovations}} + \underbrace{\rho^T A_{t-T}}_{\text{truncation error}} .\end{aligned} \tag{9}$$

Hence, for large enough T and $|\rho| < 1$ the truncated history of innovations is an approximate sufficient statistic for the history of productivity values, which in turn are an approximate sufficient statistic for the endogenous aggregate state.

The same argument can also be made in a single step by observing that

$$\begin{bmatrix}\log(A_t) \\ \log(K_t)\end{bmatrix} = \underbrace{\begin{bmatrix}\rho & 0 \\ 1 & \alpha\end{bmatrix}}_{=:M} \begin{bmatrix}\log(A_{t-1}) \\ \log(K_{t-1})\end{bmatrix} + \begin{bmatrix}\sigma\epsilon_t \\ \log(\alpha\beta)\end{bmatrix} \tag{10}$$

The two eiqenvalues of the matrix M are ρ and α and hence the impact of $\log(A_{t-T})$ and $\log(K_{t-T})$ on $\log(K_t)$ and $\log(A_t)$ decays exponentially at rate $\max\{\rho, \alpha\}$. This also implies that in order to accurately predict the policy functions based on a truncated history of innovations in a setting where ρ is close to 1, the history needs to be very long, leading to high dimensional state $\mathbf{X}_t^{\text{seq},T}$.[15]

3.2.3 Remaining parts of the algorithm

The remaining parts of the algorithm follow Azinovic et al. (2022) and Azinovic and Zemlicka (2024) and are summarized here for completeness.

Loss function We train the neural network by minimizing a loss function. Specifically, we use the mean squared error in the household optimality condition, (5), expressed in terms of relative consumption error. By minimizing the (weighted) Euler equation error, we train the neural network policy function to take shape that is consistent with the optimal choices of the representative household.

In order to evaluate the households optimality condition implied by the policy encoded the neural network, $\mathcal{N}_{\boldsymbol{\rho}}(\mathbf{X}_t^{\text{seq},T}) = K_{t+1}$, we need access to the current value of capital and productivity[16] $\mathbf{X}_t^{\text{state}} = [A_t, K_t]$. Hence, even though the neural network's input does only consist of $\mathbf{X}_t^{\text{seq},T}$, we need to also keep track of the associated state vector $\mathbf{X}_t^{\text{state}}$ in order to be able to evaluate the loss function. The distribution over possible $t+1$ histories, $\mathbf{X}_{t+1}^{\text{seq},T}$, is implied by the current history $\mathbf{X}_t^{\text{seq},T}$, together with the stochastic process for the exogenous variable. The distribution over possible $t+1$ states, $\mathbf{X}_{t+1}^{\text{state}} = [A_{t+1}, K_{t+1}] = [A_{t+1}, \mathcal{N}_{\boldsymbol{\rho}}(\mathbf{X}_t^{\text{seq},T})]$, is given by the current state $\mathbf{X}_t^{\text{state}}$, the policy function encoded by the weights of the neural network, and the stochastic processes for the exogenous variables.

[15] With a slight abuse of notation, we use $\mathbf{X}^{\text{seq},T}$ to indicate the approximate state in sequence space based on either sequences of innovations (ϵ) or values (A).

[16] Because we need to evaluate objects like current period output, or marginal product of capital next period.

We construct the relative Euler equation error by plugging the neural network approximation of the history-based policy function into equation (5).

$$ree(\mathbf{X}_t^{\text{seq},T}, \mathbf{X}_t^{\text{state}}, \boldsymbol{\rho}) := \frac{(u')^{-1}\left(\mathrm{E}\left[\beta u'(C_{t+1})(1-\delta+r_{t+1}^K)\right]\right)}{C_t} - 1 \tag{11}$$

where

$$C_t = A_t K_t^\alpha + (1-\delta)K_t - K_{t+1} \tag{12}$$

$$C_{t+1} = A_{t+1} K_{t+1}^\alpha + (1-\delta)K_{t+1} - K_{t+2} \tag{13}$$

$$K_{t+1} = \mathcal{N}_{\boldsymbol{\rho}}(\mathbf{X}_t^{\text{seq},T}) \tag{14}$$

$$K_{t+2} = \mathcal{N}_{\boldsymbol{\rho}}(\mathbf{X}_{t+1}^{\text{seq},T}) \tag{15}$$

$$\mathbf{X}_{t+1}^{\text{seq},T} = [A_{t+1-T}, \ldots, A_{t+1}]. \tag{16}$$

We define the loss function as a simple average of the square of the relative Euler equation error evaluated across a dataset $\mathcal{D}$ of state-histories pairs.

$$\mathcal{D} := \{(\mathbf{X}_1^{\text{seq},T}, \mathbf{X}_1^{\text{state}}), (\mathbf{X}_2^{\text{seq},T}, \mathbf{X}_2^{\text{state}}), \ldots, (\mathbf{X}_N^{\text{seq},T}, \mathbf{X}_N^{\text{state}})\} \tag{17}$$

$$\ell(\mathcal{D}, \boldsymbol{\rho}) := \frac{1}{N} \sum_{(\mathbf{X}_t^{\text{seq},T}, \mathbf{X}_t^{\text{state}}) \in \mathcal{D}} ree(\mathbf{X}_t^{\text{seq},T}, \mathbf{X}_t^{\text{state}}, \boldsymbol{\rho})^2. \tag{18}$$

Updating the neural network parameters We update the parameters using the ADAM optimizer (Kingma and Ba, 2014), which is a variant of gradient descent. The update rule implied by vanilla gradient descent is given by

$$\boldsymbol{\rho}^{\text{new}} = \boldsymbol{\rho}^{\text{old}} + \alpha^{\text{learn}} \nabla_{\boldsymbol{\rho}} \ell(\mathcal{D}, \boldsymbol{\rho}) \tag{19}$$

The ADAM optimizer modifies the equation (19) by introducing a parameter-specific adaptive learning rates rather than using a single constant learning rate for all parameters.

Sampling the most relevant state-history pairs To generate a dataset $\mathcal{D}$ of state-history pairs, we closely follow the ergodic simulation scheme used in the *DEQN* algorithm of Azinovic et al. (2022). We start the simulation with a random sample of initial exogenous and endogenous states, and the simulate exogenous states using their stochastic process, and evolve endogenous states using the current approximation of the policy function. Relative to the original *DEQN* method, our algorithm additionally keeps track of truncated shock histories over the simulation. This is important because we use the truncated histories of shocks as neural network inputs. A key advantage of using truncated shock histories as network input is that their distribution does not change over the training, as it is pinned-down by model primitives, *i.e.* shock distributions, rather than by policy functions.

Parameter	γ	δ	β	α	ρ^A	σ^A
Meaning	relative risk aversion	depreciation of capital	patience	capital share in production	persistence of log TFP	std dev of innovations log TFP
Value	2	0.1	0.95	$\frac{1}{3}$	0.8	0.03

Table 1: Parameter values chosen for the Brock and Mirman (1972) model.

3.3 Parameterization

We set relative risk aversion $\gamma = 2$, depreciation $\delta = 0.1$, and patience $\beta = 0.95$. The capital share in production is set to $\alpha = \frac{1}{3}$, and the process for logarithm of TFP has persistence $\rho^A = 0.8$ and $\sigma^A = 0.03$. The model parameters are summarized in table 1.

3.4 Implementation

We train a densely connected feed-forward neural network to predict the savings rate in period t, s_t as a function of the last T realizations of innovations to aggregate productivity, *i.e.*, $\mathbf{X}_t^{\text{seq},T} = [\epsilon_{t-T}, \ldots, \epsilon_{t-1}, \epsilon_t] \in \mathbb{R}^T$ and $s_t = \mathcal{N}_{\boldsymbol{\rho}}(\mathbf{X}_t^{\text{seq},T})$. Approximating the savings rate as opposed to directly parameterizing the savings function K_{t+1}, has the advantage that we can ensure that savings rate is bounded in $(0, 1)$ by applying a sigmoid activation in the output layer.[17] Because of that, our policy function is guaranteed to satisfy the budget constraint and also guaranteed to ensure non-negativity of consumption. Given the current capital stock and productivity level, we compute the quantity of resources on hand $M_t = A_t K_t^\alpha L^{1-\alpha} + (1-\delta)K_t$. The savings rate and resources on hand then imply the current consumption $C_t = (1 - s_t)M_t$ and the next periods capital $K_{t+1} = s_t M_t$. This structure is a good illustration of why our algorithm still needs to keep track of values of state variables. Even though states are not used as network input, they are still required to construct objects like resources at hand, marginal products, etc.

We organize the training procedure into a sequence of *episodes*. An episode starts by inheriting a neural policy function and a dataset $\mathcal{D}_j$ consisting of state-history pairs from the previous episode. Then we use the approximate policy function and a pseudo-random number generator to simulate a new training dataset of N^{data} state-history pairs $\mathcal{D}_j = \{(\mathbf{X}_{j,i}^{\text{state}}, \mathbf{X}_{j,i}^{\text{seq},T})\}_{i=1}^{N^{\text{data}}}$:

$$\text{obtain savings rate for old state :} \qquad s_{j,i} = \mathcal{N}_{\boldsymbol{\rho}}(\mathbf{X}_{j,i}^{\text{seq},T}) \tag{20}$$

$$\text{compute implied capital in the next period :} \qquad K_{j+1,i} = s_{j,i}(A_{j,i}K_{j,i}^\alpha + (1-\delta)K_{j,i}) \tag{21}$$

$$\text{draw a new random shock :} \qquad \epsilon_{j+1,i} \sim \mathcal{N}(0, 1) \tag{22}$$

$$\text{obtain the new agg. state in seq. space:} \qquad \mathbf{X}_{j+1,i}^{\text{seq},T} = \left[[\mathbf{X}_{j,i}^{\text{seq},T}]_{2:T}, \epsilon_{j+1,i}\right] \tag{23}$$

$$\text{obtain the new agg. state in state. space:} \qquad \mathbf{X}_{j+1,i}^{\text{state}} = [K_{j+1,i}, A_{j+1,i}]. \tag{24}$$

Generating new training data in this way is computationally cheap and does not pose a bottleneck to our algorithm. This has the advantage that we can generate new training dataset after every

[17] The sigmoid function is defined as $\text{sigmoid}(x) := \frac{1}{1+e^{-x}}$.

Parameter	N^{quad}	T	$N^{\text{hidden 1}}$	$N^{\text{hidden 2}}$	$N^{\text{hidden 3}}$	N^{output}
Meaning	Quad. nodes	Length of hist. of shocks	# nodes layer 1 (activation)	# nodes layer 2 (activation)	# nodes layer 3 (activation)	# nodes output layer (activation)
Value	8	100	128 (gelu)	128 (gelu)	128 (gelu)	1 (sigmoid)
Parameter	N^{data}	N^{mb}	N^{episodes}	Optimizer	α^{learn}	
Meaning	States per episode	mini-batch size	# episodes	Optimizer	learning rate	
Value	4096	256	40'000	Adam	10^{-5}	

Table 2: Hyperparameter values chosen for the neural network to solve the Brock and Mirman (1972) model.

episode, such that no datapoint is used twice during the training. Therefore, overfitting is not a concern. A key difference of our approach, relative to previous deep learning based approaches (*e.g.* Azinovic et al., 2022) is that the distribution of neural network inputs, $\mathbf{X}^{\text{seq},T}$, is exogenous and hence remains stationary throughout the training. The state-space representation $\mathbf{X}^{\text{state}}$ on the other hand, is converging to the ergodic set of states as the neural network learns the equilibrium policies.

After obtaining a fresh dataset, we update the policy function by running a short training loop using the Adam optimizer (see Kingma and Ba, 2014) with learning rate α^{learn} and mini-batch size $N^{\text{mini-batch}}$. We evaluate the gradient of the loss function using standard reverse mode automatic differentiation.

3.5 Training

Hyperparameters We use a history of $T = 100$ previous productivity innovations as the approximate sufficient statistic replacing the state vector as a network input. We parameterize the history-based policy function using a densely-connected feed-forward neural network with three hidden layers. Each layer consists of 128 *gelu* activated neurons.[18] The output layer is sigmoid activated, ensuring that the predicted savings rate lies between 0 and 1. To compute the conditional expectation in the Euler equation we use Gauss-Hermite quadrature with $N^{\text{quad}} = 8$ integration nodes.[19] For minimization of the loss function, we rely on the basic ADAM optimizer with a learning rate of 10^{-5} and mini-batches size of 256. In each episode, our algorithm simulates 4096 state-history pairs, implying 16 gradient descent steps per episode. The hyperparameters are summarized in table 2.

Training progress Figure 1 shows the loss function during training. The loss function decreases steadily during the training and reaches a value well below 10^{-8} within a training run.[20]

[18] The gelu activation function is given by $f(x) = x\Phi(x)$, where $\Phi(x)$ denotes the Gaussian cumulative distribution function.

[19] An alternative would be to discretize the AR(1) process, as in Rouwenhorst (1995).

[20] The training run takes roughly 4 minutes on a Tesla T4 GPU.

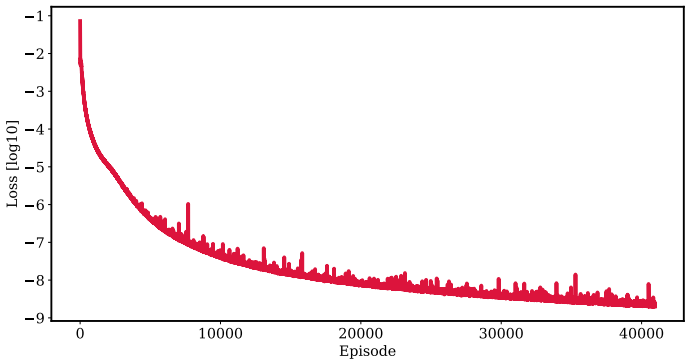

Figure 1: Loss function when training the neural network for 50'000 episodes to solve the Brock and Mirman (1972) model. The loss function is the mean squared error in the equilibrium conditions of the model, each episode consists of 4096 simulated states.

3.6 Accuracy

We now turn to assessing the accuracy of our solution. First, we inspect the error in the equilibrium conditions on the simulated ergodic set of states. The single equilibrium condition that characterizes the solution to the model is the Euler equation (5). The left panel in figure 2 shows the distribution of the absolute value of equilibrium condition error expressed in units of relative consumption errors. The vertical lines show the mean, the 99th percentile as well as the 99.9th percentile of the error distribution. As shown in the figure, the errors in the equilibrium conditions are low, with a mean error below 0.005% and the 99.9th percentile of errors below 0.025%.

The error in the equilibrium conditions is an implicit measure, and depending on the model at hand, there might not be an obvious mapping between the magnitude of this implicit error measure and magnitude of actual policy error. For a complicated model, equilibrium conditions error might be the only available error measure. However, in this simple model, we have an access to high-quality reference solution provided by a standard grid-based solution methods. With a dense enough grid, the present model can be solved to very high accuracy, such that we can regard the obtained solution as a good proxy to the *true* policy function. Exploiting this classical solution, in figure 2, we show a comparison of the policy learned by the neural network, to the policy compute with standard policy time iteration (see, *e.g.*, Judd, 1998).

The comparison shows two things: first, the difference between the two policies is small, the mean relative difference is below 0.005% and the 99.9th percentile is below 0.015%. This illustrates the the neural network is able to approximate the equilibrium policies to very high accuracy. Second, when comparing the *explicit* policy error (right panel) to the *implicit* error in the equilibrium conditions (left panel), we can observe that both errors are of the same magnitude. This indicates that the implicit error in the equilibrium conditions is a good proxy for the policy error. While this is a reassuring finding, it is important to point out that this finding depends on the model, so we can not generalize this conclusion to other models.

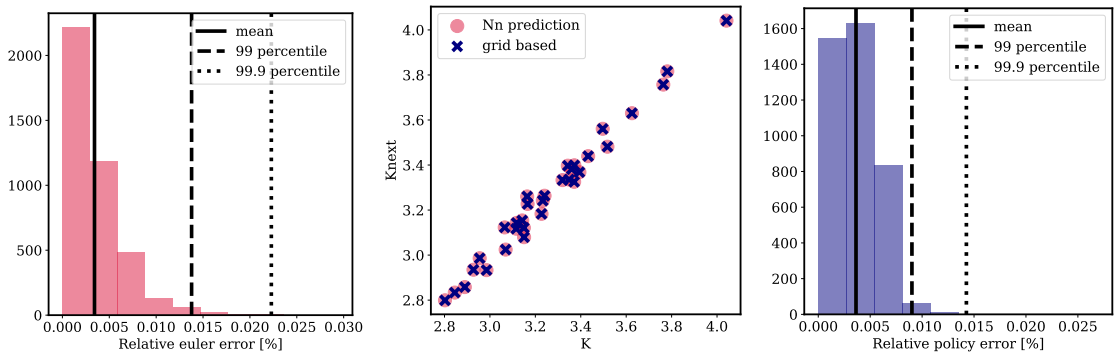

Figure 2: The left panel shows the distribution of errors in the optimality condition (equation (5)) in % on 4096 simulated states after training the neural network. The solid vertical line shows the mean error, the dashed vertical line shows the 99th percentile of errors and the vertical dotted line shows the 99.9th percentile of errors. The middle panel compares the policy learned by the neural network (round dots) to the policy solved for with a conventional grid-based method. The right panel shows the distribution of errors when comparing the policy learned by the neural network to the policy solved with a conventional method.

4 Application to OLG model

In this section, we test the performance of our algorithm in more challenging setup. We compute the approximate equilibrium in an overlapping generations economy featuring portfolio choice, non-trivial market clearing, and more complex stochastic processes. In particular, we model 72 overlapping generations of households, who consume and save. Households allocate their savings into physical capital and risk-free bonds. While bonds are fully liquid, trading capital is subject to convex portfolio adjustment costs, and return to capital is exposed to aggregate risk. There are three sources of uncertainty: as in the Brock and Mirman (1972) model, the logarithm of TFP follows an AR(1) process. Additionally, the economy switches between two regimes: normal times and disaster times. In disaster times, the depreciation to capital is stochastic and on average higher than during normal times. The regime process evolves according to a Markov transition matrix. Since capital is predominantly held by older households, the disaster leads to swings in the intergenerational wealth distribution.

We train the neural network to predict equilibrium prices and policies as a function of truncated histories of aggregate shocks. For this economy, our algorithm keep track of the last T realizations of aggregate shocks, which include T innovations to the logarithm of TFP, T innovations to the depreciation of capital, and the last T regime realizations. The purpose of this section is to demonstrate that our sequence space method is capable of obtaining highly-accurate approximations to equilibria in economies featuring complex stochastic processes. The overlapping generations setup furthermore generates an obvious dependance of current outcomes on shock realizations, reaching far into the past.

4.1 Model

4.1.1 Technology

A representative firm operates a Cobb-Douglas technology and produces a perishable consumption good using labor and capital.

$$Y_t = A_t K_t^{\alpha} L_t^{1-\alpha}. \tag{25}$$

A_t denotes the stochastic total factor productivity (TFP). The logarithm of TFP evolves according to an AR(1) process

$$\log(A_t) = \rho \log(A_{t-1}) + \sigma^A \epsilon_t^A,\ \epsilon_t^A \sim \mathcal{N}(0,1). \tag{26}$$

As in the stochastic growth economy of Brock and Mirman (1972), the firm faces competitive input and output markets. We choose consumption good as numéraire, hence price of consumption is normalized to unity. We denote the wage by w_t and rental rate of capital by r_t^K.

4.1.2 Demographics

Each period a new representative cohort of mass 1 enters the economy and lives for H periods. We denote the age of a cohort by $h \in \{1, \ldots, H\}$. Households are endowed with l^h efficiency units of labor, which they supply inelastically at equilibrium wage. The labor endowment is normalized such that $L = \sum_{h=1}^{H} l^h = 1$. The age dependence of efficient labor supply serves as a device to model a life-cycle profile of household labor income. We abstract from bequest motives, taxes and social security.

4.1.3 Preferences

Households have time separable expected utility over consumption streams. We assume that the felicity function takes the constant relative risk aversion form with coefficient γ. Hence, household rank stochastic consumption stream according to:

$$\mathrm{E}\left[\sum_{i=0}^{H-h} \beta^i u(c_{t+i}^{h+i})\right], \tag{27}$$

where β denotes their patience.

4.1.4 Asset markets

Households are able to re-allocate their consumption across time and states by trading in two assets. Two assets and infinitely many possible shock realizations next period[21] imply incomplete markets environment. Furthermore, there are two additional financial frictions. First, households are subject

[21] We assume innovations to the logarithm TFP and capital depreciation to be gaussian.

to short sale limits on capital and bonds. Second, while the bond is a liquid asset, capital is an illiquid asset and households have to pay adjustment costs when adjusting their capital holdings.

The capital holdings adjustment costs are given by $\psi(k_{t+1}^{h+1}, k_t^h) = \xi^{\text{adj}} \left(k_{t+1}^{h+1} - k_t^h\right)^2$. Short sale constraints on both assets are specified as exogenous limits $b_{t+1}^{h+1} \geq \underline{b}$ and $k_{t+1}^{h+1} \geq \underline{k}$.

The net supply of bonds is fixed at B. Bonds can be purchased or sold at equilibrium price p_t, and they deliver a risk-free payoff of 1 in period $t+1$. Purchasing a unit of capital in period t, promises a risky payout of $1 - \delta_{t+1} + r_{t+1}^K$ in period $t+1$. There are two sources of risk in returns to capital: first, the rental rate of capital r_{t+1}^K, depends on the total factor productivity A_{t+1} in period $t+1$. Second, the depreciation rate of capital is stochastic as well and given by

$$\delta_{t+1} = \delta \frac{2}{1 + D_{t+1}}, \text{ where } \log(D_{t+1}) = \begin{cases} \rho^\delta \log(D_t) & \text{normal times in } t+1 \\ \mu^\delta + \rho^\delta \log(D_t) + \sigma^\delta \epsilon_{t+1}^\delta & \text{disaster in } t+1 \end{cases} \tag{28}$$

with $\mu^\delta < 0$ and $\epsilon_{t+1}^\delta \sim \mathcal{N}(0,1)$. Normal and disaster times are modeled as two discrete Markov regimes with transition matrix $\Pi^{\text{regime}} = \begin{bmatrix} \pi^{n \to n} & 1 - \pi^{n \to n} \\ 1 - \pi^{d \to d} & \pi^{d \to d} \end{bmatrix}$, where $\pi^{n \to n}$ and $\pi^{d \to d}$ denote the persistence of the normal and the disaster regime respectively.

4.1.5 Household problem

Recursively formulated, the households' problem is characterized by the following Bellman equation.

$$V_t^h = \max_{k_{t+1}^{h+1}, b_{t+1}^{h+1}} u(c_t^h) + \beta \mathrm{E}\left[V_{t+1}^{h+1}\right] \tag{29}$$

subject to :

$$\begin{aligned} c_t^h = & \underbrace{l^h w_t}_{\text{lab. inc.}} + \underbrace{b_t^h + (1 - \delta_t + r_t^k) k_t^h}_{\text{payout of assets}} - \underbrace{(p_t b_{t+1}^{h+1} + k_{t+1}^{h+1})}_{\text{savings}} - \underbrace{\psi(k_{t+1}^{h+1}, k_t^h)}_{\text{adj. costs}} \\ & \underline{b} \leq b_{t+1}^{h+1} \\ & \underline{k} \leq k_{t+1}^{h+1}. \end{aligned}$$

In every period, households receive labor and asset income, and decide how much to invest in each of the two assets, subject to adjustment costs and short-sale constraints. The time index and age superscript stand for the dependence on state variables.

The solution to the household problem is characterized by a set of Karush-Kuhn-Tucker (KKT)

conditions for each age-group $h = 1, \ldots, H-1$ and each asset

$$p_t u'(c_t^h) = \beta \mathrm{E}\left[u'(c_{t+1}^{h+1})\right] + \lambda_t^{b,h} \tag{30}$$

$$0 = \lambda_t^{b,h}(b_{t+1}^{h+1} - \underline{b}) \tag{31}$$

$$0 \leq (b_{t+1}^{h+1} - \underline{b}) \tag{32}$$

$$0 \leq \lambda_t^{b,h} \tag{33}$$

$$\left(1 + \frac{\partial}{\partial k_{t+1}^{h+1}}\psi(k_{t+1}^{h+1}, k_t^h)\right) u'(c_t^h) = \beta \mathrm{E}\left[u'(c_{t+1}^{h+1})\left(1 - \delta_{t+1} + r_{t+1}^k - \frac{\partial}{\partial k_{t+1}^{h+1}}\psi(k_{t+2}^{h+2}, k_{t+1}^{h+1})\right)\right] + \lambda_t^{k,h} \tag{34}$$

$$0 = \lambda_t^{k,h}(k_{t+1}^{h+1} - \underline{k}) \tag{35}$$

$$0 \leq (k_{t+1}^{h+1} - \underline{k}) \tag{36}$$

$$0 \leq \lambda_t^{k,h} \tag{37}$$

Making use of the Fischer-Burmeister function (see, Jiang, 1999) $\psi^{FB}(a,b) := \sqrt{a^2 + b^2} - a - b$, a first-order condition and the corresponding complementary slackness condition can be collapsed into a single equation for each age-group and each asset. Analogously to equation (11) for the representative agent model, we rearrange the equations, such that, in equilibrium, the equation has to evaluate to zero, and the deviations from zero are expressed in units of relative consumption error.

4.1.6 Equilibrium

Now, we define the recursive equilibrium of the economy

State space The state of the economy

$$\mathbf{X}_t^{\text{state}} = [\chi_t, A_t, \delta_t, \mu_t] \in \{0,1\} \times \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^{H-1+H-2} \tag{38}$$

is given by the regime indicator, χ_t, total factor productivity A_t, capital depreciation δ_t, and the distribution of assets across the age-groups $\mu_t = \{(b_t^h, k_t^h)\}_{h=1}^H$.[22]

Functional rational expectations equilibrium The functional rational expectations equilibrium of our economy consists of H policy functions $\pi^{k,h}(\mathbf{X}_t^{\text{state}}) = k_{t+1}^{h+1}$ for investment in capital, H policy functions $\pi^{b,h}(\mathbf{X}_t^{\text{state}}) = b_{t+1}^{h+1}$ for investment in the bond, and a price function $\pi^p(\mathbf{X}_t^{\text{state}}) = p_t$, consistent with the KKT conditions encoding household optimal behavior and a market clearing equation.[23] The aim of our algorithm is to compute approximations to these equilibrium functions.

[22] Strictly speaking, because we assume that households enter the economy without assets, and because of market clearing, a sufficient statistic for the asset distribution consists of $H-1$ free parameters for capital and $H-2$ for the bond.

[23] We use the first-order conditions of representative firm directly to compute the equilibrium wage and rental rate as a function of the state of the economy.

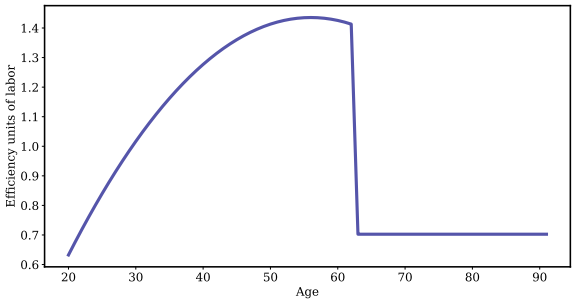

Figure 3: Efficiency units of labor over the lifecycle.

Parameter		Value
H	Number of cohorts	72
γ	Relative risk aversion	2
β	Patience	0.96
B	Bond supply	0.75
$\underline{b}$	Borrowing constr. bond	0
$\underline{k}$	Borrowing constr. capital	0
ξ^{adj}	Adj. cost on capital	0.1
ρ	Persistence of log TFP	0.85
σ^A	Std. dev. innovations to log TFP	0.03
δ	Depreciation of capital in normal times	0.1
ρ^δ	Persistence of depr. in disaster	0
σ^δ	Std. dev. of innovations to depreciation	0.2
μ^δ	Mean depreciation during disasters	-1.10
$\pi^{n\to n}$	Prob. to remain in normal times	0.94
$\pi^{d\to d}$	Prob. to remain in the disaster regime	2/3

Table 3: Parameter values for the OLG model.

4.2 Parameterization

We set the model parameters to standard values in the literature. One model period corresponds to one year, and we model $H = 72$ overlapping generations. The model ages $h = 1$ to 72 corresponds to adult life from 21 years to 92 years. The age dependent efficient units of labor are chosen to generate a life-cycle profile of labor income and are shown in figure 3.

The preferences are set to standard values in the literature. We set the coefficient of relative risk aversion at $\gamma = 2$ and patience to $\beta = 0.96$. We assume strict borrowing constraints with $\underline{b} = \underline{k} = 0$. The adjustment costs for capital are set to $\xi^{\text{adj}} = 0.1$. The persistence of the logarithm of TFP is set to $\rho = 0.85$, and the standard deviation of innovations to the logarithm of TFP is set to $\sigma^A = 0.03$. The long-run depreciation of capital in normal times is given by $\delta = 0.1$. The parameters governing the depreciation of capital in the diaster regime are given by $\rho^\delta = 0$, $\sigma^\delta = 0.2$ and $\mu^\delta = -1.10$, implying 50% higher depreciation in the disaster when $\log(D_t) = \mu^\delta$. The per-period probability to enter into the disaster state is 6% and the per-period probability to exit the disaster state is 33.33%, implying $\pi^{n\to n} = 0.94$ and $\pi^{d\to d} = 2/3$. The parameter values are summarized in Table 3.

4.3 Implementation

We use a densely-connected feedforward neural network to parameterize the key set of functions that allows us to characterize the rest of the equilibrium objects in closed-form. Those key functions consist of the bond price, p_t, as well as the investment in capital, k_{t+1}^{h+1}, and investment in the bond, b_{t+1}^{h+1}, for each age-group.[24] We use the market clearing layers approach (as in Azinovic and Zemlicka, 2024), so the architecture of the neural network architectures ensures that the predicted policies are always consistent with bond market clearing.

Following our sequence space approach, the neural network predicts the policy and price functions as a function of truncated history of shocks. Hence, the input to the neural network is given by

$$\mathbf{X}_t^{\text{seq},T} = [\underbrace{\chi_{t-(T-1)}, \ldots, \chi_{t-1}, \chi_t}_{T \text{ last regime indicators}}, \underbrace{\epsilon^A_{t-(T-1)}, \ldots, \epsilon^A_{t-1}, \epsilon^A_t}_{T \text{ last innovations to TFP}}, \underbrace{\epsilon^\delta_{t-(T-1)}, \ldots, \epsilon^\delta_{t-1}, \epsilon^\delta_t}_{T \text{ last innovations to deprec.}}] \in \mathbb{R}^{3T}. \tag{39}$$

Again, we train the neural network to minimize the errors in all equilibrium conditions. The resulting loss function consists of the $2 \times (H-1)$ optimality conditions for the capital and bond policies for each age group, except the last who consumes everything. As in the previous example, the training data in episode j are given by state-history pairs $\mathcal{D}_j = \{(\mathbf{X}^{\text{state}}_{j,i}, \mathbf{X}^{\text{seq},T}_{j,i})\}_{i=1}^{N^{\text{data}}}$. We again use the Adam optimizer (see Kingma and Ba, 2014) with a learning rate α^{learn} and a mini-batch size $N^{\text{mini-batch}}$. After each training episode we simulate states and histories forward to generate a new training dataset $\mathcal{D}_{j+1}$. We evaluate the expectations over the continuous shocks ϵ^A_{t+1} and ϵ^δ_{t+1} we use Gauss-Hermite quadrature with N^{quad} quadrature nodes.

4.4 Training

Our training procedure follows the stepwise approach to solve models with multiple assets outlined in Azinovic and Zemlicka (2024). We proceed along four steps:

First, we set the net supply of bonds to zero. Given a strict no-short sale constraint and zero net supply market clearing condition implies that all equilibrium bond holdings had to be zero. Hence we can first restrict our attention to solving for capital savings policy functions while keeping bond savings fixed at zero.

Second, we train the neural network price function to learn the bond price implied by consumption allocations of the capital-only economy. We plug the set of consumption functions obtained by solving the capital-only economy into bond Euler equations. We invert the Euler equations to recover implied bond prices. The largest of the prices would be the price of the first ϵ of bond supply introduced into this economy. We train the price network to replicate this price as a function of truncated histories of aggregate shocks.

Third, the previous two steps provide us with a good initial guess for the policy and price functions in an economy with a comparatively small bond supply. We now proceed to slowly increase

[24] To be precise, we only need to predict the asset choice for $H-1$ age groups, since, for our calibration, it is always optimal to consume everything in the last period of life. Similarly to our approach in the Brock and Mirman (1972) model, the neural network predicts the share of cash-at-hand invested in each of the two assets, which we then use to construct the prediction for the asset choices.

Parameter		Value
N^{quad}	Number quadrature nodes TFP / depreciation	4 / 4
T	Length of history of shocks	144
$N^{\text{hidden 1}}$	# neurons in the first hidden layer (activation)	720 (gelu)
$N^{\text{hidden 2}}$	# neurons in the second hidden layer (activation)	720 (gelu)
$N^{\text{hidden 3}}$	# neurons in the third hidden layer (activation)	720 (gelu)
N^{output}	# neurons in the output layer (activation)	214 (None)
N^{data}	States per episode	8192
N^{mb}	mini-batch size	64
N^{episodes}, step 1	Training episodes (capital only)	512
N^{episodes}, step 2	Training episodes (pretrain bond price)	1536
$N^{\text{bond steps}}$, step 3	Number of incremental increases in bond supply	4
N^{episodes}, step 3	Training episodes (for each intermediate bond supply)	16384
N^{episodes}, step 4	Training episodes (final model)	32768
Optimizer	Optimizer	Adam
α^{learn}	Learning rate	10^{-5}

Table 4: Hyper-parameter values for the OLG model.

the bond supply. After every small increase in the bond supply, we retrain the neural network using the previous solution as a starting point. We proceed until the full bond supply is reached.

Fourth, after reaching the full supply, we continue training the neural network on the final model specifications until we reach desired level of accuracy.

The hyperparameters for our implementations are summarized in table 4.

4.5 Accuracy

We assess the accuracy of our neural network solution by investigating the errors in the equilibrium conditions over the simulated ergodic set. Figure 4 shows the errors in the equilibrium conditions for each age group along the simulated paths of the economy. The errors are expressed in units of relative consumption errors. The model is solved accurately, the mean and 90th percentile of errors is below 0.1% and the 99.9th percentile below 0.32% for both assets and all the age groups. The mean error is around 0.03%.

In the middle panel in figure 4 we show the distribution of asset holdings over the age groups. The savings behavior of households is driven by the life-cycle forces. Facing an increasing labor income profile, young households are borrowing constrained. Later in their life, they start to accumulate savings to make sure they are able to finance their later-age consumption after the drop in their labor endowment after the age of 62.[25]

The right panel in figure 4 shows the distribution of consumption across the age groups, conditional on the regime of the economy. The disaster, modeled as an increase in the capital depreciation and uncertainty, is most harmful for households around the age 80.

[25]While we abstract from modeling retirement explicitly, the reduction in labor efficiency units around the age of 60 serves as a proxy for the income decline associated with retirement.

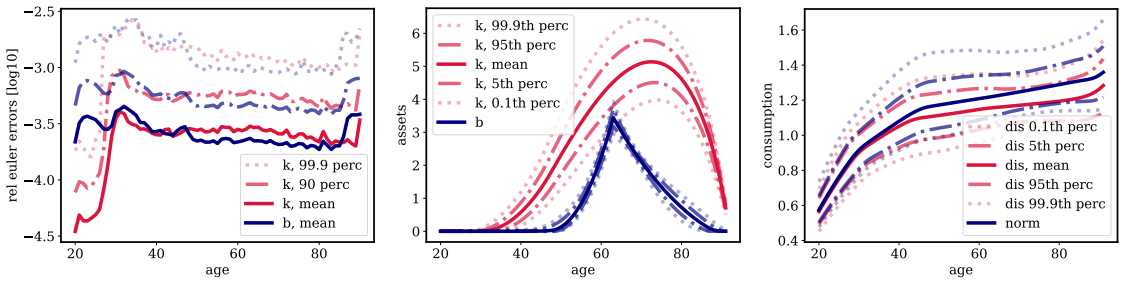

Figure 4: The left panel shows the final level of errors in the equilibrium conditions achieved by our training procedure. The dotted line shows the 99.9th percentile, the dash-dotted line the 90th percentile, and the solid line shows the mean. The red lines show the errors in the optimality conditions for capital for each age group, and the blue lines show the errors in the optimality conditions for the bonds. The middle panel shows the distribution of assets over the simulated ergodic set of states. The right panel shows the resulting consumption by age group. The red lines in the right panel show consumption conditional on the economy being in the normal state, and the red lines show consumption statistics conditional on the economy being in the disaster state.

5 Application to heterogeneous households and heterogeneous firms

Finally, we apply our algorithm to compute an approximate equilibrium in an environment with heterogeneous households and heterogeneous firms. The households and the firms are both subject to uninsurable idiosyncratic risk, as well as to aggregate productivity shocks, and swings in the level of aggregate and idiosyncratic volatility. Firms operate a decreasing returns to scale technology that produces the final good using capital and labor. The firms own and accumulate their capital. Besides that, firms hire workers on a walrasian labor market, and pay dividends to their shareholders. Firms make their decisions to maximize the present discounted value of their dividends. We assume that firms use weighted household marginal rate of substitution to discount future cashflows, where the weighting is proportional to household stock holdings. Households supply labor inelastically at equilibrium wage, consume, and invest into the aggregate stock market index. Besides the fluctuations in wages and stock prices induced by aggregate shocks, households are subject to uninsurable idiosyncratic labor endowment shocks.

Because of that, the state space of the economy includes two endogenously moving distributions: the firm distribution over capital and idiosyncratic firm productivity, as well as the household distribution over stock holdings and idiosyncratic labor endowments. The distribution of firms is relevant to households because it determines the dividends paid on stock holdings, and the wage rate on supplied labor. In turn, the stochastic discount factor used by firms to make their investment decisions depends on the distribution of households.

5.1 Model

Time is discrete and infinite, $t = 0, 1, \ldots$.

5.1.1 Firms and technology

There is a continuum of firms, who face uninsurable idiosyncratic risk, and aggregate risk process featuring stochastic volatility in the spirit of Bloom et al. (2018). At the beginning of each period, there is a continuum of firms of mass 1. As in Azinovic et al. (2023), we assume that only a fraction Γ of the firms survive to the end of the period and produce the consumption good. A fraction $(1-\Gamma)$ of firms exits the economy, and is replaced by an equal measure of start-ups that, conditional on surviving, start to produce in the next period.[26] Each of the firms $i \in [0, \Gamma]$ operates a decreasing returns to scale production technology.

$$y_t^i = A_t z_t^i (k_t^i)^{\alpha} (l_t^i)^{1-\alpha-\zeta}, \tag{40}$$

where A_t denotes stochastic aggregate productivity, which affects all firms in the economy, z_t^i denotes firm specific productivity realization, and α determines the elasticity of output with respect to capital, and ζ parameter controls the degree of returns to scale.[27]

Aggregate productivity follows an AR(1) process

$$\log(A_t) = \rho^A \log(A_{t-1}) + \sigma_{t-1}^A \epsilon_t^A, \tag{41}$$

where $\epsilon_t^A \sim \mathcal{N}(0,1)$ and $\sigma_t^A = \sigma(U_t) \in \{\sigma_L^A, \sigma_H^A\}$ takes two values depending on the uncertainty regime $U_t \in \{L, H\}$. The uncertainty regime follows a two-state Markov process as in Bloom et al. (2018). We denote the transition probabilities between the uncertainty regimes with $\pi_{L,H}^U$, $\pi_{L,L}^U = 1-\pi_{L,U}^U$, $\pi_{H,L}^U$, and $\pi_{H,H}^U = 1-\pi_{H,L}^U$, respectively. The firm-level idiosyncratic productivity follows a three-state Markov process $z_t \in \{z_1, z_2, z_3\}$, with $z_2 = (z_1 + z_3)/2$. The transition probabilities vary with the uncertainty regime: $\Pi_t^z \in \{\Pi_L^z, \Pi_U^z\}$. We are interested in the effect of a pure uncertainty shock, *i.e.* in an increase in the variance of future idiosyncratic productivity without changing the current productivity or the expected future productivity level $\mathrm{E}\left[z_{t+1}^i | z_t\right]$. To do so, assume that the transition matrices in the low and high uncertainty regimes are given by

$$\Pi_L^z = \begin{bmatrix} \pi_{11}^z & \pi_{12}^z & \pi_{13}^z \\ \pi_{21}^z & \pi_{22}^z & \pi_{23}^z \\ \pi_{31}^z & \pi_{32}^z & \pi_{33}^z \end{bmatrix}, \Pi_H^z = \begin{bmatrix} \pi_{11}^z + u & \pi_{12}^z - 2u & \pi_{13}^z + u \\ \pi_{21}^z + u & \pi_{22}^z - 2u & \pi_{23}^z + u \\ \pi_{31}^z + u & \pi_{32}^z - 2u & \pi_{33}^z + u \end{bmatrix}, \tag{42}$$

where $0 \leq u < \min\{\frac{1}{2}\pi_{12}^z, \frac{1}{2}\pi_{22}^z, \frac{1}{2}\pi_{13}^z\}$ parameterizes the increase in the idiosyncratic uncertainty. This formulation allows us to keep the idiosyncratic productivity level z_t^i, as well as the expected future productivity $\mathrm{E}\left[z_{t+1}^i | z_t\right]$ constant, when the economy transitions into the high volatility regime.[28]

[26] We assume that the capital of exiting firms is destroyed in the process.

[27] The capital share is $\alpha/(1-\zeta)$.

[28] To the best of our knowledge, maintaining these two features would not easily be possible using, for example, a Rouwenhorst (1995) or Tauchen (1986) discretization methods for an AR(1) process.

Firms accumulate capital subject to capital adjustment costs of the form

$$\psi(k', k) = \frac{\xi(k', k)}{2} k \left(\frac{k'}{k} - 1 \right)^2, \tag{43}$$

where

$$\xi(k', k) = \begin{cases} \xi^{\text{up}} & \text{for } k' > k \\ \xi^{\text{down}} & \text{for } k' \leq k. \end{cases} \tag{44}$$

We assume that $\xi^{\text{down}} > \xi^{\text{up}}$, so reducing the capital stock is associated with larger adjustment costs than increasing the capital stock by the same percentage. Although the adjustment costs parameter changes discontinuously at $k' = k$, $\psi(k', k)$ is not only continuous but also differentiable. At $k' = k$ we have $\psi(k', k) = \frac{\partial \psi(k', k)}{\partial k'} = \frac{\partial \psi(k', k)}{\partial k} = 0$.[29] Additionally, we smooth out the jump in the adjustment cost parameter such that

$$\xi(k', k) = w^{\text{up}} \xi^{\text{up}} + (1 - w^{\text{up}}) \xi^{\text{down}} \tag{45}$$

$$w^{\text{up}} = \text{sigmoid} \left(s \frac{k' - k}{k} \right), \tag{46}$$

where $s > 0$ denotes a smoothing parameter. For large s, w^{up} converges to a step function, *i.e.* $w^{\text{up}} \approx 1$ for $s \frac{k'-k}{k} >> 0$ and $w^{\text{up}} \approx 0$ for $s \frac{k'-k}{k} << 0$.

Each firm sets its investment and dividend payout policies to maximize its objective function: the present discounted value of its dividends, subject to a budget constraint, and a non-negative dividends constraint. The firm problem is characterized by a following Bellman equation:

$$v_t(z_t^j, k_t^j) = \max_{k_{t+1}^j} d_t^j + \text{E}_t \left[\Lambda_{t+1,t} v_{t+1}(z_{t+1}^j, k_{t+1}^j) \right] \tag{47}$$

subject to:

$$d_t^j = y_t^j - w_t l_t^j - i_t^j - \psi(k_{t+1}^j, k_t^j) \tag{48}$$

$$d_t^j > \underline{d}. \tag{49}$$

The time index of the value function stands for the dependence of the firm problem on the aggregate state of the economy. $\Lambda_{t+1,t}$ denotes the aggregate stochastic discount factor given by the distribution

[29] $\xi(k', k)$ is not continuous at $k' = k$, but the quadratic term $(k'/k - 1)^2$ ensures that $\psi(k', k)$ remains differentiable.

of shareholders. The policy functions of the firms are characterized by a set of KKT conditions.

$$\left(1+\frac{\partial}{\partial k^i_{t+1}}\psi(k^i_{t+1},k^i_t)\right)(1+\lambda^i_t)=\Gamma\mathrm{E}\left[\Lambda_{t+1,t}\left(1+r^{i,k}_{t+1}+\frac{\partial}{\partial k^i_{t+1}}\psi(k^i_{t+2},k^i_{t+1})\right)(1+\lambda^i_{t+1})\right]$$

$$\Leftrightarrow \lambda^i_t=\frac{\Gamma\mathrm{E}\left[\Lambda_{t+1,t}\left(1+r^{i,k}_{t+1}+\frac{\partial}{\partial k^i_{t+1}}\psi(k^i_{t+2},k^i_{t+1})\right)(1+\lambda^i_{t+1})\right]}{\left(1+\frac{\partial}{\partial k^i_{t+1}}\psi(k^i_{t+1},k^i_t)\right)}-1 \tag{50}$$

$$0\leq(d^i_t-\underline{\mathrm{d}}) \tag{51}$$

$$0\leq\lambda^i_t \tag{52}$$

$$0=(d^i_t-\underline{\mathrm{d}})\lambda^i_t, \tag{53}$$

where λ^i_t denotes the Lagrange multiplier on the non-negative dividend constraint and where $r^{i,k}_t$ denotes the marginal product of capital installed in the firm.

$$r^{i,k}_t=A_t z^i_t\alpha(k^i_t)^{\alpha-1}(l^i_t)^{1-\alpha-\zeta}-\delta. \tag{54}$$

Using the Fischer-Burmeister function, we rewrite the KKT conditions of the firm problem as a single equation $\psi^{FB}(\lambda^i_t,d^i_t-\underline{\mathrm{d}})=0.$, where λ^i_t is given in equation (50).

Aggregation of firms Firms hire efficiency units of labor on a competitive spot market, and labor can move freely between the firms. There is a single equilibrium wage w_t per efficiency unit of labor, which is paid by all the firms. The labor demand of an individual firm is given by a static first-order condition

$$w_t=(1-\alpha-\zeta)\frac{y^i_t}{l^i_t}=(1-\alpha-\zeta)A_t z^i_t(k^i_t)^{\alpha}(l^i_t)^{-\alpha-\zeta} \tag{55}$$

$$\Leftrightarrow l^i_t=\left(A_t z^i_t\frac{1-\alpha-\zeta}{w_t}(k^i_t)^{\alpha}\right)^{\frac{1}{\alpha+\zeta}}. \tag{56}$$

Let $\mu^F_t(z^i,k^i)$ denote the distribution of surviving firms over idiosyncratic productivity and capital. The overall mass of surviving firms is given by $\Gamma=\int_i\mu^F_t(z^i,k^i)di$. Since we assume that all households supply their labor inelastically, the aggregate labor supply is constant at $L=1$. This allows us to solve for the equilibrium wage as a function of the productivity level A_t, and the cross-sectional distribution of firms.

$$1=L=\int\left(A_t z^i\frac{(1-\alpha-\zeta)}{w_t}(k^i)^{\alpha}\right)^{\frac{1}{\alpha+\zeta}}\mu^F_t(z^i,k^i)di \tag{57}$$

$$=\left(A_t\frac{(1-\alpha-\zeta)}{w_t}\right)^{\frac{1}{\alpha+\zeta}}\int\left(z^i(k^i)^{\alpha}\right)^{\frac{1}{\alpha+\zeta}}\mu^F_t(z^i,k^i)di \tag{58}$$

$$\Leftrightarrow w_t=A_t(1-\alpha-\zeta)\left(\int\left(z^i(k^i)^{\alpha}\right)^{\frac{1}{\alpha+\zeta}}\mu^F_t(z^i,k^i)di\right)^{\alpha+\zeta}. \tag{59}$$

Given the expression for the wage (59), the labor demand is given in a closed-form, as described in equation (56). Similarly, aggregate dividends are given by

$$D_t = \int d_t(z^i, k^i)\mu_t(z^i, k^i)di. \tag{60}$$

Startups Each period, households trade shares in the aggregate index of existing firms. Owning a share entitles a household to receive a fraction of the aggregate dividend stream. The firms that exist in period t includes a measure Γ of surviving firms that produce in period t, as well as a measure $(1-\Gamma)$ of period t startups, which start producing only in period $t+1$. Holding θ_t^i shares, which have been purchased in period $t-1$, entitles households to a payout of $\theta_t^i(\Gamma p_t + D_t)$. This is because period t startups can only be traded ones they exists. The stocks traded in period $t-1$ hence only make up fraction Γ of the stocks traded at period t.

We assume that a measure $(1-\Gamma)$ startups is replacing exiting firms every period. The value of period t startups is $(1-\Gamma)p_t$. We set the initial capital of startups to be equal to the average capital stock in the economy. The required investment is provided uniformly by all the households who therefore also own the startups. The initial investment is not subject to adjustment costs. The size of startup investment per household is given by

$$I_t^{\text{su}} = (1-\Gamma)\int k_{t+1}^i(z_t^i, k_t^i)\frac{\mu(z_t^i, k_t^i)}{\Gamma}\mathrm{d}i. \tag{61}$$

The aggregate rents from startup creation are given by

$$\Pi_t^{\text{su}} = (1-\Gamma)p_t - I_t^{\text{su}}. \tag{62}$$

The rents from startup creation are equal to the difference between the average capital stock in the economy and the price of an equity share. We assume that the profits from startup creation are equally distributed to all households.

5.1.2 Households

There is a unit-mass continuum of infinitely lived households. Households have time separable expected utility preferences over stochastic consumption streams

$$U\left(\{c_t\}_{t=0}^{\infty}\right) = \mathrm{E}\left[\sum_t \beta^t u(c_t)\right], \text{ where } u(c) = \frac{c^{1-\gamma}}{1-\gamma}, \tag{63}$$

and where β denotes the patience parameter, and γ denotes the coefficient of relative risk aversion. Households' labor endowment is stochastic and follows an AR(1) process

$$\log(e_t^i) = \rho^e \log(e_{t-1}^i) + \sigma^e \epsilon_t^{e,i}, \text{ where } \epsilon_t^{e,i} \sim \mathcal{N}(0,1). \tag{64}$$

The stochasticity of labor endowment makes households subject to uninsurable idiosyncratic risk. We discretize the idiosyncratic labor endowment process into a two-state Markov process using

the Rouwenhorst (1995) algorithm. Households supply their efficient units of labor inelastically at equilibrium wage. While households can save by purchasing equity shares θ_t^i, their borrowing subject to a short-sale limit $\theta_{t+1}^i \geq \underline{\theta}$. The budget constraint of household i reads as

$$c_t^i = e_t^i w_t + \theta_t^i (D_t + \Gamma p_t) - \theta_{t+1}^i p_t + \Pi_t^{\text{SU}}. \tag{65}$$

The solution to the household savings problem is characterized by the following set of KKT conditions

$$p_t u'(c_t^i) = \beta \text{E}\left[u'(c_{t+1}^i)(D_{t+1} + \Gamma p_{t+1})\right] + \lambda_t^i \tag{66}$$

$$0 \leq \lambda_t^i \tag{67}$$

$$0 \leq (\theta_{t+1}^i - \underline{\theta}) \tag{68}$$

$$0 = (\theta_{t+1}^i - \underline{\theta})\lambda_t^i. \tag{69}$$

Households are exposed to idiosyncratic risk because of their labor endowments and to aggregate risk via the wage, dividends, the stock price, and the profits from startup creation. Because of that, aggregate shocks lead to fluctuations in the wealth distribution. We denote the household distribution by $\mu_t^H(e^i, \theta^i)$, where e denotes idiosyncratic labor endowment, and θ share holdings.

5.1.3 Asset markets

Households trade claims on the aggregate of *existing* firms. They do not trade in specific firms but in claims to the dividends stream paid by all the existing firms. Aggregate asset demands of households is given by

$$\Theta_{t+1}^H = \int \theta_{t+1}^i(e^i, \theta^i)\mu^H(e^i, \theta^i)di. \tag{70}$$

Market clearing requires that $\Theta_{t+1}^H = 1$. Firms take the stochastic discount factor as given. We assume that they use the stochastic discount factor of shareholder households weighted by their corresponding share holdings. For a specific realization of aggregate shocks in period $t+1$ and household i with idiosyncratic labor endowment e^i and shares θ^i, we define its stochastic discount factor as

$$\Lambda_{t+1,t}^i(e^i, \theta^i) = \frac{\beta \text{E}_{e_{t+1}^i}\left[u'(c_{t+1}^i)\right]}{u'(c_t^i)}, \tag{71}$$

where the expectations operator is taken over the realizations of idiosyncratic labor endowment, for a specific realization of aggregate shocks in period $t+1$. The aggregate stochastic discount factor is given by

$$\Lambda_{t+1,t}^i = \frac{\int \Lambda_{t+1,t}^i(e^i, \theta^i)\theta_{t+1}(e^i, \theta^i)\mu_t^H(e^i, \theta^i)di}{\int \theta_{t+1}(e^i, \theta^i.)\mu_t^H(e^i, \theta^i)di}. \tag{72}$$

5.1.4 Equilibrium

State space The aggregate state of the economy is given by

$$\mathbf{X}_t^{\text{agg. state}} = [\chi_t, A_t, \mu_t^F, \mu_t^H] \in \{0,1\} \times \mathbb{R}_+ \times \mathbb{R}_+^{N_z \times N_k} \times \mathbb{R}_+^{N_e \times N_s}, \tag{73}$$

where χ_t is an indicator denoting the uncertainty regime, A_t denotes the aggregate productivity level, and μ_t^F and μ_t^H denote the distribution of households and firms. We represent the distributions using a finite histogram approximation (see Young, 2010) with $N_z \times N_k$ and $N_e \times N_\theta$ histogram points for firms and households. N_z and N_e denote the number of histogram points along the exogenous idiosyncratic productivity dimension and N_k and N_θ denote the number of histogram points for the capital and asset holdings respectively.

Functional rational expectations equilibrium The functional rational expectations equilibrium consists of firm policy functions, $\pi^F(\mathbf{X}_t^{\text{agg. state}}, z_t^i, k_t^i) = k_{t+1}^i$, household policy function, $\pi^H(\mathbf{X}_t^{\text{agg. state}}, e_t^i, \theta_t^i) = \theta_{t+1}^i$, as well as a price function $\pi^q(\mathbf{X}_t^{\text{agg. state}}) = p_t$, consistent with optimizing households, optimizing firms, and market clearing.

5.2 Parameterization

One model period corresponds to one calendar year. We set patience to $\beta = 0.95$ and the coefficient of relative risk aversion to $\gamma = 2$. For the idiosyncratic labor process we choose $\rho^e = 0.871$ and $\sigma^e = 0.246$, as in Auclert et al. (2021).[30] We use the Rouwenhorst (1995) method to discretize the labor process into a two state Markov chain with states $e_0 = 0.538, e_1 = 1,462$ and transition matrix

$$\pi^e = \begin{bmatrix} 0.935 & 0.065 \\ 0.065 & 0.935 \end{bmatrix} \tag{74}$$

The short-sale limit is set to $\underline{\theta} = 0$.

Following Bloom et al. (2018), we set the depreciation rate of capital to $\delta = 0.1$ and we choose $\alpha = 0.25$ and $\zeta = 0.25$, consistent with a capital share of $1/3$ in production. The persistence of the logarithm of aggregate productivity is set to $\rho^A = 0.8145$ and the standard deviations of innovations to productivity is set to $\sigma_L^A = 0.0124$ and $\sigma_H^A = 0.0199$, consistent with the quarterly process in Bloom et al. (2018) and Khan and Thomas (2008). As in Azinovic et al. (2023), we set the survival rate of firms to $\Gamma = 0.965$. There are $N_z = 3$ idiosyncratic productivity level for firms corresponding to $z_0 = 0.5$, $z_1 = 1.0$, and $z_2 = 1.5$. The corresponding Markov transition matrices are given by

$$\Pi_L^z = \begin{bmatrix} 0.850 & 0.150 & 0.000 \\ 0.075 & 0.850 & 0.075 \\ 0.000 & 0.150 & 0.850 \end{bmatrix} \text{ and } \Pi_H^z = \begin{bmatrix} 0.925 & 0.000 & 0.075 \\ 0.150 & 0.700 & 0.150 \\ 0.075 & 0.000 & 0.925 \end{bmatrix}. \tag{75}$$

[30] We convert the quarterly process in Auclert et al. (2021) into a yearly process.

Parameter		Value
γ	Relative risk aversion	2
β	Patience	0.95
$\underline{\theta}$	Borrowing constraint	0
ρ^e	Persistence id. income process	0.871
σ^e	Std. dev. id. income process	0.246
ρ^A	Persistence of aggregate TFP	0.8145
σ^A_L, σ^A_H	Std. dev. innovations to TFP	1.24%, 1.99%
δ	Depreciation of capital	0.1
α	Capital share in production	0.25
ζ	Returns to scale	0.25
Γ	Survival rate of firms	0.965
z	Idiosyncratic firm productivity	[0.5, 1.0, 1.5]
Π^L_z, Π^H_z	Transition matrices for id. firm prod.	See text
$\Pi^U_{L,L}$, $\Pi^U_{H,H}$	Persistence of uncertainty regimes	0.90, 0.79
ξ^{up}, ξ^{down}, s	Adjustment costs firms	1.0, 2.5, 400

Table 5: Parameter values for the economy with heterogeneous firms and heterogeneous households.

The transition matrix between the two uncertainty regimes is given by

$$\Pi^U = \begin{bmatrix} 0.90 & 0.10 \\ 0.21 & 0.79 \end{bmatrix}, \tag{76}$$

implying a quarterly persistence of the high uncertainty regime of 0.943 and a quarterly persistence of the low uncertainty regime of 0.974, as in Bloom et al. (2018). The adjustment costs parameters are set to $\xi^{\text{up}} = 1.0$ and $\xi^{\text{down}} = 2.5$.[31] The parameter values are summarized in table 5.

5.3 Implementation

As in the previous examples, we solve the model by training neural networks to approximate a core set of equilibrium functions, and then solve for the remaining equilibrium objects in closed-form. Relative to the previous two applications, this model poses an additional challenge: individual policy functions of firms and households depend not only on aggregate quantities, but also on their corresponding idiosyncratic states. To solve for the firm and household policies, we train neural networks to approximate a mapping from truncated histories of aggregate shocks to *policy functions*, that map the idiosyncratic state variables to equilibrium policies. To do so, we build on, and extend the operator learning approach of Zhong (2023).

5.3.1 Operator learning

Suppose we want to obtain an approximation $\hat{f}(X, x)$ to the function $f(X, x)$, where $X \in \mathbb{R}^{N_1}$, $x \in \mathbb{R}^{N_2}$, and $f(X, x) \in \mathbb{R}$. Let $F(\mathbb{R}^{N_2}, \mathbb{R})$ denote the space of functions mapping $\mathbb{R}^{N_2}$ into $\mathbb{R}$. The

[31] The smoothing parameter in the adjustment cost function is set to $s = 400$, such that $\xi(1.01k, k) = 0.993\xi^{\text{up}} + 0.007\xi^{\text{down}}$ and $\xi(0.99k, k) = 0.007\xi^{\text{up}} + 0.993\xi^{\text{down}}$.

idea of operator learning is to decompose the mapping f as $f(X, x) = g(X)(x)$. Where the *operator* $g : X \to g(X) : \mathbb{R}^{N_1} \to F(\mathbb{R}^{N_2}, \mathbb{R})$, maps the vector $X \in \mathbb{R}^{N_1}$ to a *function* $g(X) \in F(\mathbb{R}^{N_2}, \mathbb{R})$. The function $g(X)$ is then evaluated at x.

For example, let X denote the aggregate state vector, let x denote the idiosyncratic state vector, and let $f(X, x)$ denote a policy function, which depends on aggregate and idiosyncratic state variables. In this example $g : \mathbb{R}^{N_1} \to F(\mathbb{R}^{N_2}, \mathbb{R})$ maps the aggregate state to a policy function. The policy function $g(X)$ in turn, maps the idiosyncratic state x to the choice variable $g(X)(x)$. Let us consider the case of $N_1 = N_2 = 1$ and suppose that conditional on the aggregate state X, the policy function is a linear function of the idiosyncratic state x, *i.e.* $f(X, x) = g(X)(x) = \alpha_X x$. Although the policy function is linear in the idiosyncratic state x, the slope coefficient α_X depends on the aggregate state X in a potentially nonlinear way. We can split the problem of approximating $f(X, x)$ into two parts. First, approximating $g(X)$ by $\hat{g}(X) : X \to \hat{\alpha}_X$.[32] Second, evaluating $\hat{g}(X)(x) = \hat{\alpha}_X x$. As we show in this paper, a key advantage of approximating operators, as opposed to directly parameterizing policy functions, is that it is easy to ensure desirable properties of $\hat{g}(X)(x)$, such as monotonicity or concavity in x for all X.

Shape preserving operators For concreteness, let us consider the household consumption function from the model above: $\pi^c(\mathbf{X}_t^{\text{agg. state}}, e_t^i, \theta_t^i) = c_t^i$. Following the operator learning approach, we decompose the problem of predicting the individual policy into two steps: First we predict a *function* $\pi_t^{c,\text{ ind}}$ for each aggregate state $\mathbf{X}_t^{\text{agg. state}}$. Then we evaluate the resulting function $\pi_t^{c,\text{ ind}}(e_t^i, \theta_t^i) = c_t^i$ for all idiosyncratic states of interest.

Often, some crucial shape properties of the equilibrium functions $\pi_t^{c,\text{ ind}}(e_t^i, \theta_t^i)$ are known ex-ante. In the context of the model we present in this section, or for a large class of other macroeconomic models, we know that the consumption function is strictly increasing and concave in the idiosyncratic asset holding θ^i.[33] In this section, we show a simple way to ensure that all predicted policy functions fulfill these properties. Thereby, we effectively restrict the search space to the space of economically more meaningful functions, increasing the robustness of our algorithm. What is more, a monotone consumption function allows us to apply the endogenous grid method, and to obtain targets to train the policy function with supervised learning, increasing the robustness of our approach.

The first step is to choose a functional form to approximate the function $\pi_t^{c,\text{ ind}}(e_t^i, \theta_t^i)$. We consider the case where e_t^i takes two discrete values $\mathcal{E}^{\text{grid}} = [e_0, e_1]$, where $e_0 < e_1$, and where θ_t^i is continuous. Because e_t^i is a discrete variable, we can split $\pi^{c,\text{ ind}}$ into two univariate functions. Then, we approximate those functions using piecewise linear interpolation on a grid of interpolation nodes $\Theta^{\text{grid}} = [\theta_0, \theta_1, \ldots, \theta_N]$, where $\theta_0 < \theta_1 < \cdots < \theta_N$

$$\mathcal{C}_t^{\text{grid}} = \begin{bmatrix} c_{t,0,0}, c_{t,0,1}, \ldots, c_{t,0,N-1} \\ c_{t,1,0}, c_{t,1,1}, \ldots, c_{t,1,N-1} \end{bmatrix} \in \mathbb{R}^{2 \times N}, \tag{77}$$

[32] A linear function from $\mathbb{R} \to \mathbb{R}$ is fully characterized by its slope coefficient. Therefore, we can essentially predict a function by predicting a single real number.

[33] For example, this is the case in standard incomplete market models in the spirit of Imrohoroğlu (1989); Bewley (1977); Huggett (1993); Aiyagari (1994); Krusell and Smith (1998).

where $c_{t,i,j} = \pi_t^{c,\text{ ind}}(e^i, \theta^j)$ for a given aggregate state $\mathbf{X}_t^{\text{agg. state}}$, and idiosynratic states $e_i \in \mathcal{E}^{\text{grid}}$, and $\theta_j \in \Theta^{\text{grid}}$. For asset holdings θ in between grid points, *i.e.* $\theta_j < \theta < \theta_{j+1}$, we interpolate linearly between $c_{t,i,j}$ and $c_{t,i,j+1}$. Conditional on an aggregate state, and fixed grids $\mathcal{E}^{\text{grid}}$ and Θ^{grid}, the approximate household consumption functions is fully described by the $2 \times N$ values of $\mathcal{C}_t^{\text{grid}}$, that are in turn parameterized by a neural network. Following our sequence space approach, this neural network take as an input the truncate sequence of aggregate shocks $\mathbf{X}_t^{\text{seq},T}$.

Monotonicity: To ensure that the predicted consumption functions are increasing in idiosyncratic asset holdings θ, we need to make sure that the prediction by the neural network always satisfies $c_{t,i,j} < c_{t,i,j+1}\ \forall t, i, j$. To ensure that this condition holds, we follow a two-step procedure. First, instead of predicting $\mathcal{C}_t^{\text{grid}}$ directly, the neural network predicts the boundary consumption value[34] and consumption increments

$$\mathcal{N}_{\boldsymbol{\rho}}^c(\mathbf{X}_t^{\text{seq},T}) \approx d\mathcal{C}_t^{\text{grid}} := \begin{bmatrix} c_{t,0,0}, dc_{t,0,1}, \ldots, dc_{t,0,N-1} \\ c_{t,1,0}, dc_{t,1,1}, \ldots, dc_{t,1,N-1} \end{bmatrix} \in \mathbb{R}_{>0}^{2\times N}, \tag{78}$$

where we ensure that all the $2 \times N$ predicted entries of $d\mathcal{C}_t^{\text{grid}}$ are positive. This is easy to ensure, for example, by using a *softplus* activation function in the output layer. In the second step we construct

$$\mathcal{C}_t^{\text{grid}} = \begin{bmatrix} c_{t,0,0}, c_{t,0,1}, \ldots, c_{t,0,N-1} \\ c_{t,1,0}, c_{t,1,1}, \ldots, c_{t,1,N-1} \end{bmatrix}, \tag{79}$$

where

$$c_{t,i,j} = c_{t,i,0} + \sum_{k=1}^{j} dc_{t,i,k}. \tag{80}$$

Since the architecture of our neural network guarantees that all the predicted increments $dc_{t,i,k}$ are positive, this two step procedure guarantees monotonicity of the predicted consumption at the interpolation nodes: $c_{t,i,j} < c_{t,i,j+1}$. Piecewise linear interpolation guarantees that monotonicity of the grid values translates into monotonicity of the resulting function. A large number of gridpoints, N, and, for example, a log-log spaced grid[35] for the asset holdings Θ^{grid} can ensure that the loss of accuracy coming from the linear interpolation is minimal.

Concavity: In order to jointly guarantee monotonicity and concavity of the consumption function, one needs to ensure that the slope of the consumption function is decreasing in the individual asset holding, while it remains bounded from below by zero. Let $d\theta_j := \theta_j - \theta_{j-1}$ define the distance between two grid points θ_j and θ_{j-1} on the asset grid Θ^{grid}. To ensure concavity and monotonicity of consumption, we follow a three-step procedure. In a first step, we predict:

$$\mathcal{N}_{\boldsymbol{\rho}}^c(\mathbf{X}_t^{\text{seq},T}) \approx dd\mathcal{C}_t^{\text{grid}} := \begin{bmatrix} c_{t,0,0}, ddc_{t,0,1}, ddc_{t,0,2} \ldots, ddc_{t,0,N-1} \\ c_{t,1,0}, ddc_{t,1,1}, ddc_{t,1,2} \ldots, ddc_{t,1,N-1} \end{bmatrix}, \tag{81}$$

[34] Consumption at the lowest gridpoint in the asset grid.

[35] *Log-log* ensures high grid density for low asset holdings, where the short sale limit might bind.

where we ensure that $c_{t,i,0} > 0$ and $ddc_{t,i,j} < 0$ using a suitable activation function. In the second step, we construct the matrix $d\mathcal{C}_t^{\text{grid}}$, using

$$dc_{t,i,j} = d\theta_j \times exp\left(\sum_{k=1}^{j} ddc_{t,i,k}\right). \tag{82}$$

Since all predictions $ddc_{t,i,k} < 0$, our architecture enforces that $dc_{t,i,j}/d\theta_j$ is decreasing in j, ensuring concavity of the consumption function. The exponential function further guarantees that all $dc_{t,i,j}/d\theta_j > 0$, making sure that the consumption function is also monotone. In the third step, we use the cumulative sum to construct the implied consumption grid, as in equation (80).

Borrowing constraints: With a slight modification of the procedure described above, we can additionally encode the borrowing constraint $\theta_{t+1}^i \geq \underline{\theta} = 0$ directly into the network architecture. For a given asset price and given dividends, we can convert the distance between points in the asset grid, $d\theta_j$, to differences in cash-at-hand (cah) by multiplying the asset grid distance with the payout of the asset.[36] In our model: $dcah_j = d\theta_j(D_t + \Gamma p_t)$. We know that the consumption function is increasing and concave in cash-at-hand. We can ensure monotonicity and concavity of the consumption function by making sure that the marginal propensity to consume, i.e. $dc/dcah$, is positive and decreasing. To additionally ensure that the borrowing constraint is always satisfied, it is sufficient to ensure that the marginal propensity to consume, is bounded from above by 1. We can achieve this by following a three step procedure, similar to the one outline in the previous paragraph. The neural network now predicts

$$\mathcal{N}_{\boldsymbol{\rho}}^c(\mathbf{X}_t^{\text{seq},T}) \approx d\mathcal{MPC}_t^{\text{grid}} := \begin{bmatrix} cs_{t,0,0}, dmpc_{t,0,1}, dmpc_{t,0,2} \ldots, dmpc_{t,0,N-1} \\ cs_{t,1,0}, dmpc_{t,1,1}, dmpc_{t,1,2} \ldots, dmpc_{t,1,N-1} \end{bmatrix}, \tag{83}$$

where we use a sigmoid activation function to ensure that $0 < cs_{t,i,0} \leq 1$. Where $cs_{t,i,0}$ denotes the consumption share out of cash-at-hand at the first grid point on the asset grid. We use a softplus activation and multiplication with -1 to ensure that all predicted $dmpc_{t,i,j} < 0$. In the second step we construct the consumption value at the first asset grid point $c_{t,i,0} = cs_{t,i,0} \times cah_{t,i,0}$. Since $cs_{t,i,0}$ always lies between 0 and 1, the borrowing constraint is satisfied. We then obtain the marginal propensity to consume

$$mpc_{t,i,j} = \underbrace{cs_{t,i,0}}_{\in(0,1)} \times \underbrace{e^{\sum_{k=1}^{j} dmpc_{t,i,k}}}_{\in(0,1),\text{ decr in j}}, \tag{84}$$

for $j > 0$ and obtain

$$\mathcal{MPC}_t^{\text{grid}} := \begin{bmatrix} c_{t,0,0}, mpc_{t,0,1}, mpc_{t,0,2}, \ldots, mpc_{t,0,N-1} \\ c_{t,1,0}, mpc_{t,1,1}, mpc_{t,1,2}, \ldots, mpc_{t,1,N-1} \end{bmatrix}. \tag{85}$$

This parameterization ensures that all predicted $mpc_{t,i,j}$ are positive, bounded from above by 1, and

[36] Other income *i.e.* labor income and startup rents is constant across the asset grid, and hence can be omitted in this calculation.

decreasing in the index j. In the third step, we again compute the remaining consumption values for all gridpoints

$$c_{t,i,j} = c_{t,i,0} + \sum_{k=1}^{j} mpc_{t,i,k} \times dcah_{t,i,k}. \tag{86}$$

As in the case of parameterizing *ddc*, the resulting consumption grid is increasing and concave. Because the predicted marginal propensities to consume are bounded from above by 1, this construction additionally ensures, that the borrowing constraint is never violated.[37] Furthermore, it also makes it easy for the neural network to predict the consumption for borrowing constraint households extremely precisely (by predicting the maximum mpc of 1). In models, where the marginal propensity to consume plays a central role, ensuring a precisely satisfied borrowing constraint can be particularly important for the implied model predictions.

Shape preservation in other settings: It is worth noting that our approach of shape-preserving operator learning is not limited to the sequence-space approach or to discrete-time models. Shape-preserving operator learning can be analogously applied in a state-space based methods or in continuous-time models. Lastly, note that our architecture guarantees only the monotonicity and concavity of the predicted values at the interpolation nodes. With linear interpolation in one dimension, monotonicity and concavity of the values at the interpolation nodes is sufficient for the monotonicity and concavity of the interpolating function. Shape-preserving interpolation of multivariate functions is a more complicated problem. Multivariate piecewise linear interpolation is guaranteed to preserve monotonicity, however, concavity of the interpolating function is generally not guaranteed for simple interpolation method in higher dimensions.

5.3.2 Firm policies

As a part of computing an approximate recursive equilibrium, we need to solve for the firms policy functions $\pi^F(\mathbf{X}_t^{\text{agg. state}}, z_t^i, k_t^i) = k_{t+1}^i$ and $\pi^\lambda(\mathbf{X}_t^{\text{agg. state}}, z_t^i, k_t^i) = \lambda_t^i$, such that they are consistent with the optimality conditions given in equations (50) to (53). We approximate both functions using the operator learning approach. Employing the shape-preservation techniques described above, we make sure that the predicted KKT multiplier is non-negative and decreasing in firm capital k. Similarly, we ensure that the firm savings function is increasing in k.

The firm-level productivity follows a three-state Markov chain. We set a grid of N_k points over the firm capital holdings. For each aggregate shocks, we obtain following predictions

$$\mathcal{K}_t^{\text{grid}} = \begin{bmatrix} k_{t+1,0,0} & k_{t+1,0,1} & \dots & k_{t+1,0,N_k-1} \\ k_{t+1,1,0} & k_{t+1,1,1} & \dots & k_{t+1,1,N_k-1} \\ k_{t+1,2,0} & k_{t+1,2,1} & \dots & k_{t+1,2,N_k-1} \end{bmatrix} \tag{87}$$

$$\lambda_t^{\text{grid}} = \begin{bmatrix} \lambda_{t,0,0} & \lambda_{t,0,1} & \dots & \lambda_{t,0,N_k-1} \\ \lambda_{t,1,0} & \lambda_{t,1,1} & \dots & \lambda_{t,1,N_k-1} \\ \lambda_{t,2,0} & \lambda_{t,2,1} & \dots & \lambda_{t,2,N_k-1} \end{bmatrix}, \tag{88}$$

[37] Given the consumption value at the first gridpoint is consistent with the borrowing constraint.

corresponding to the gridded idiosyncratic firm policies for a given aggregate state at the $3 \times N_k$ interpolation nodes. To evaluate the policies for capital values between grid points, we rely on linear interpolation.

We use a shape preserving operator network to ensure that the predicted values for next period's capital, $k_{t+1,i,j}$ are increasing in j (*i.e.* increasing in $k_{t,i,j}$). For the KKT multiplier, $\lambda_{t,i,j}$, we similarly ensure that all the predictions are non-negative and decreasing in j. Following our sequence space approach, the input to the neural networks is given by a truncated sequence of aggregate shocks

$$\mathbf{X}_t^{\text{seq},T} = [\underbrace{\epsilon^A_{t-T+1}, \epsilon^A_{t-T+2}, \ldots, \epsilon^A_t}_{\text{shocks to aggregate TFP}}, \underbrace{\chi_{t-T+1}, \chi_{t-T+2}, \ldots, \chi_t}_{\text{uncertainty regime}}]. \tag{89}$$

In the first step, the neural networks predict

$$\mathcal{N}^k_{\boldsymbol{\rho}}(\mathbf{X}_t^{\text{seq},T}) = \begin{bmatrix} k_{t+1,0,0} & dk_{t+1,0,1} & dk_{t+1,0,2} & \ldots & dk_{t+1,0,N_k-1} \\ k_{t+1,1,0} & dk_{t+1,1,1} & dk_{t+1,1,2} & \ldots & dk_{t+1,1,N_k-1} \\ k_{t+1,2,0} & dk_{t+1,2,1} & dk_{t+1,2,2} & \ldots & dk_{t+1,2,N_k-1} \end{bmatrix} \tag{90}$$

$$\mathcal{N}^\lambda_{\boldsymbol{\rho}}(\mathbf{X}_t^{\text{seq},T}) = \begin{bmatrix} \log(\lambda_{t,0,0}) & d\lambda_{t,0,1} & d\lambda_{t,0,2} & \ldots & d\lambda_{t,0,N_k-1} \\ \log(\lambda_{t,1,0}) & d\lambda_{t,1,1} & d\lambda_{t,1,2} & \ldots & d\lambda_{t,1,N_k-1} \\ \log(\lambda_{t,2,0}) & d\lambda_{t,2,1} & d\lambda_{t,2,2} & \ldots & d\lambda_{t,2,N_k-1}, \end{bmatrix} \tag{91}$$

where we use the softplus activation in the output layer to ensure that $k_{t+1,i,j} \geq 0$, $dk_{t+1,i,j} \geq 0$, and $d\lambda_{t,i,j} < 0$. In the second step, we construct $\mathcal{K}^{\text{grid}}_{t+1}$ and λ^{grid}_t to ensure non-negativity and monotonicity[38] of both policies:

$$k_{t+1,i,j} = k_{t+1,i,0} + \sum_{h=1}^{j} dk_{t+1,i,h} \tag{92}$$

$$\lambda_{t,i,j} = exp\left(\log(\lambda_{t,i,0}) + \sum_{h=1}^{j} d\lambda_{t,i,h}\right). \tag{93}$$

Simulation We represent the cross-sectional distribution of firms over idiosyncratic capital and productivity using the finite histogram method of Young (2010). We base the firm histogram on the same grid as we use for interpolation nodes of firm policy functions. Because of that, we can evaluate the histogram transition using the gridded capital policies predicted by the neural network without the need to resort to interpolation.

Loss function To evaluate the performance of our neural network in producing policies consistent with firm optimality conditions, we need to define a loss function to map equilibrium conditions error into a scalar quantity. As in Azinovic et al. (2022), we choose the standard mean-squared function as our loss. Through the stochastic discount factor the firms' optimality conditions also depend on household policies and on the equity price. Because of that, the loss function also depends on the household network $\mathcal{N}^c_{\boldsymbol{\rho}}$, and on the price network $\mathcal{N}^p_{\boldsymbol{\rho}}$.

[38] The capital policy is increasing in j, while the KKT multiplier policies is decreasing in j.

We use Gauss-Hermite quadrature with N^{GH} quadrature nodes for evaluating the expectation over realizations of shocks to aggregate productivity next period, for each of the two uncertainty regimes. Let $\mathcal{A}_t := \left(\mathbf{X}_t^{\text{state}}, \{\mathbf{X}_{t+1,i}^{\text{state}}\}_{i=1}^{2N_{GH}},\ \mathbf{X}_t^{\text{seq},T}, \mathbf{X}_{t+1,i}^{\text{seq},T}\}_{i=1}^{2N_{GH}}\right)$, denote aggregate quantities in sequence-space and state-space form for a given realization in period t, and at all $2N_{GH}$ integration nodes for the next period $t+1$. Given $\mathcal{A}_t$, the wage can be computed as a closed-form function of the state-vector, as given in equation (59). The outputs of the firm network $\mathcal{N}_{\boldsymbol{\rho}}^k$, the outputs of the household network $\mathcal{N}_{\boldsymbol{\rho}}^c$, and the output of the price network $\mathcal{N}_{\boldsymbol{\rho}}^p$ allows us to construct the aggregate stochastic discount factor $\Lambda_{t+1,t}$, aggregate dividends $\left(D_t, \{D_{t+1,i}\}_{i=1}^{2N_{GH}}\right)$, as well as the stock prices $\left(p_t, \{p_{t+1,i}\}_{i=1}^{2N_{GH}}\right)$. We then sample idiosyncratic states (z_t^i, k_t^i),[39], and evaluate the firm policies on those states to obtain $k_{t+1}^i, \{k_{t+2,j}^i\}_{j=1}^{2N_{GH}}$, and $\{\lambda_{t+1,j}^i\}_{j=1}^{2N_{GH}}$. Then we recover the implied KKT multiplier, $\lambda_t^{i,\text{ implied}}$, from the optimality condition (50):

$$\lambda_t^{i,\text{ implied}} = \frac{\Gamma \mathrm{E}\left[\Lambda_{t+1,t}\left(1 + r_{t+1}^{i,k} + \frac{\partial}{\partial k_{t+1}^i}\psi(k_{t+2}^i, k_{t+1}^i)\right)(1+\lambda_{t+1}^i)\right]}{\left(1 + \frac{\partial}{\partial k_{t+1}^i}\psi(k_{t+1}^i, k_t^i)\right)} - 1. \tag{94}$$

The difference between the predicted and the implied Lagrange multiplier summarizes the error in the firm Euler equation. Besides the Euler equation error, we also need to evaluate the error in the remaining KKT conditions.

$$err_1^{\text{firm}}\left(\mathcal{A}_t, \mathcal{N}_{\boldsymbol{\rho}}^c, \mathcal{N}_{\boldsymbol{\rho}}^p, \mathcal{N}_{\boldsymbol{\rho}}^k, \mathcal{N}_{\boldsymbol{\rho}}^\lambda, z_t^i, k_t^i\right) = \frac{\lambda_t^i - \lambda_t^{i,\text{ implied}}}{1 + \lambda_t^{i,\text{ implied}}} \tag{95}$$

$$err_2^{\text{firm}}\left(\mathcal{A}_t, \mathcal{N}_{\boldsymbol{\rho}}^c, \mathcal{N}_{\boldsymbol{\rho}}^p, \mathcal{N}_{\boldsymbol{\rho}}^k, \mathcal{N}_{\boldsymbol{\rho}}^\lambda, z_t^i, k_t^i\right) = \psi^{FB}\left(\frac{d_t^i - \underline{d}}{k_t^i}, \lambda_t^{i,\text{ implied}}\right), \text{ where} \tag{96}$$

$$\psi^{FB}(a,b) = a + b - \sqrt{a^2 + b^2}, \tag{97}$$

denotes the Fischer-Burmeister function (see, *e.g.* Jiang, 1999). The optimal firm policies are characterized by

$$err_1^{\text{firm}}\left(\mathcal{A}_t, \mathcal{N}_{\boldsymbol{\rho}}^c, \mathcal{N}_{\boldsymbol{\rho}}^p, \mathcal{N}_{\boldsymbol{\rho}}^k, \mathcal{N}_{\boldsymbol{\rho}}^\lambda, z_t^i, k_t^i\right) = 0 \tag{98}$$

$$err_2^{\text{firm}}\left(\mathcal{A}_t, \mathcal{N}_{\boldsymbol{\rho}}^c, \mathcal{N}_{\boldsymbol{\rho}}^p, \mathcal{N}_{\boldsymbol{\rho}}^k, \mathcal{N}_{\boldsymbol{\rho}}^\lambda, z_t^i, k_t^i\right) = 0, \tag{99}$$

for all $\mathcal{A}_t$ and all z_t^i, k_t^i.

5.3.3 Household policies and stock price

The two additional equilibrium functions, which we need to compute are the household policy function, and the equity price function. For households, we choose to parameterize the consumption function $\pi^C(\mathbf{X}_t^{\text{agg. state}}, e_t^i, \theta_t^i) = c_t^i$. For the stock price we need to approximate $\pi^P(\mathbf{X}^{\text{agg. state}}) = p_t$. We parameterize the equity price using a simple densely connected feed-forward neural network that

[39] We sample the idiosyncratic labor endowment states from the ergodic distribution of the corresponding Markov chain. For the idiosyncratic capital, we sample half of the points from the capital grid, and the other half from a uniform distribution.

maps the truncated history of aggregate shocks into a non-negative equity price.[40]

$$\mathcal{N}^{p}_{\boldsymbol{\rho}}(\mathbf{X}^{\text{seq},T}_{t}) = p_t. \tag{100}$$

To parameterize household consumption function, we use our shape-preserving neural network operator architecture, to obtain a matrix $\mathcal{C}^{\text{grid}}_{t}$, that represents the consumption function on a discrete grid over the idiosyncratic assets θ_i and the labor endowment e_i:

$$\mathcal{C}^{\text{grid}}_{t} = \begin{bmatrix} c_{t,0,0}, c_{t,0,1}, \ldots, c_{t,0,N} \\ c_{t,1,0}, c_{t,1,1}, \ldots, c_{t,1,N} \end{bmatrix}, \tag{101}$$

where $c_{t,i,j}$ denote the consumption in period t for each idiosyncratic productivity level e_i in the productivity grid $\mathcal{E}^{\text{grid}} = [e_0, e_1]$, and for each asset holding θ_i in the asset grid $\Theta^{\text{grid}} = [\theta_0, \theta_1, \ldots, \theta_{N_\theta - 1}]$, with $\theta_0 < \theta_1 < \cdots < \theta_{N_\theta - 1}$. We use our shape preserving neural network architecture to ensure that the resulting consumption function satisfies three properties: first, it is increasing in cash-at-hand, second, it is concave in cash-at-hand, and third, it is consistent with the borrowing constraint $\theta^{i}_{t+1} > \underline{\theta} = 0$. We impose these three restrictions by following the three-step procedure, outlined in section 5.3.1. The neural network learns to predict

$$\mathcal{N}^{c}_{\boldsymbol{\rho}}(\mathbf{X}^{\text{seq},T}_{t}) \approx d\mathcal{MPC}^{\text{grid}}_{t} := \begin{bmatrix} cs_{t,0,0}, dmpc_{t,0,1}, dmpc_{t,0,2} \ldots, dmpc_{t,0,N} \\ cs_{t,1,0}, dmpc_{t,1,1}, dmpc_{t,1,2} \ldots, dmpc_{t,1,N} \end{bmatrix}, \tag{102}$$

where $cs_{t,i,0}$ denotes the consumption share out of cash-at-hand at the first grid point on the asset grid. As outlined in section 5.3.1, the neural network architecture ensures that all predicted $dmpc_{t,i,j} < 0$, as well as $0 < cs_{t,i,0} \leq 1$. In a second step we then construct

$$c_{t,i,0} = cs_{t,i,0} \times cah_{t,i,0} \tag{103}$$

$$mpc_{t,i,j} = cs_{t,i,0} \times e^{\sum_{k=1}^{j} dmpc_{t,i,k}}, \tag{104}$$

for $j > 0$ and obtain $\mathcal{MPC}^{\text{grid}}_{t}$. By construction, all predicted $mpc_{t,i,j}$ are positive, bounded from above by 1, and decreasing in the index j. In the third step, we construct the the consumption values for all the remaining grid point

$$c_{t,i,j} = c_{t,i,0} + \sum_{k=1}^{j} mpc_{t,i,k} \times dcah_{t,i,k} \tag{105}$$

$$dcah_{t,i,j} = d\theta_{t,i,j}(D_t + \Gamma p_t) \tag{106}$$

completing the construction of $\mathcal{C}^{\text{grid}}_{t}$.

Loss function While the training objective function we use to encode the firm optimality conditions is relatively standard, *i.e.* we minimize the relative error in the KKT conditions that charac-

[40] To make sure the network predicts non-negative price, we activate the output layer with a *softplus* activation function.

terize the optimization problem of the firm, as in the original *DEQN* algorithm, our treatment of the household and price block is more involved. Instead of backpropagating through the complex computational graph defined by the forward-looking optimality condition, we use the method of endogenous gridpoints (see Carroll, 2006),[41] to derive the period t consumption function implied by the current guess of equilibrium functions in period $t+1$. Exploiting speed and robustness of the endogenous gridpoint method, we are furthermore able to use a simple Newton-Raphson method to solve for market clearing equity price p_t^{solved} together with implied consumption function $\mathcal{C}_t^{\text{grid, solved}}$. Using this procedure, we obtain *training targets* for price and household networks, allowing us to train these networks on a simple supervised learning loss.

From $\mathcal{C}_t^{\text{grid, solved}}$ and p_t^{solved}, we can also obtain target values for the intermediate predictions, *i.e.* $\mathcal{MPC}_t^{\text{grid, solved}}$. The supervised learning loss terms for household and price blocks are summarized in the following equations:

$$err_1^{\text{hh}}\left(\mathcal{A}_t, \mathcal{N}_{\boldsymbol{\rho}}^c, \mathcal{N}_{\boldsymbol{\rho}}^p, \mathcal{N}_{\boldsymbol{\rho}}^k\right) = \left(\mathcal{C}_t^{\text{grid}} - \mathcal{C}_t^{\text{grid, solved}}\right) \oslash \mathcal{C}_t^{\text{grid, solved}} \tag{107}$$

$$err_2^{\text{hh}}\left(\mathcal{A}_t, \mathcal{N}_{\boldsymbol{\rho}}^c, \mathcal{N}_{\boldsymbol{\rho}}^p, \mathcal{N}_{\boldsymbol{\rho}}^k\right) = \left(\mathcal{MPC}_t^{\text{grid}} - \mathcal{MPC}_t^{\text{grid, solved}}\right) \oslash \mathcal{MPC}_t^{\text{grid, solved}} \tag{108}$$

$$err_1^{\text{p}}\left(\mathcal{A}_t, \mathcal{N}_{\boldsymbol{\rho}}^c, \mathcal{N}_{\boldsymbol{\rho}}^p, \mathcal{N}_{\boldsymbol{\rho}}^k\right) = \frac{p_t - p_t^{\text{solved}}}{p_t^{\text{solved}}}, \tag{109}$$

where $\oslash$ denotes element-wise division, and p_t, $\mathcal{C}_t^{\text{grid}}$, and $\mathcal{MPC}_t^{\text{grid}}$ denote predictions by the neural networks.

5.3.4 Loss function

Our overall loss function to train the neural networks is given by

$$\begin{aligned}
\ell(\boldsymbol{\rho}, \mathcal{A}_t) = {} & w_1^{\text{firm}} \left(\frac{1}{N_{\text{sample}}} \sum_{i=1}^{N_{\text{sample}}} err_1^{\text{firm}}\left(\mathcal{A}_t, \mathcal{N}_{\boldsymbol{\rho}}^c, \mathcal{N}_{\boldsymbol{\rho}}^p, \mathcal{N}_{\boldsymbol{\rho}}^k, \mathcal{N}_{\boldsymbol{\rho}}^\lambda, z_t^i, k_t^i\right)^2 \right) \\
& + w_2^{\text{firm}} \left(\frac{1}{N_{\text{sample}}} \sum_{i=1}^{N_{\text{sample}}} err_2^{\text{firm}}\left(\mathcal{A}_t, \mathcal{N}_{\boldsymbol{\rho}}^c, \mathcal{N}_{\boldsymbol{\rho}}^p, \mathcal{N}_{\boldsymbol{\rho}}^k, \mathcal{N}_{\boldsymbol{\rho}}^\lambda, z_t^i, k_t^i\right)^2 \right) \\
& + w_1^{\text{hh}} \left(\frac{1}{2N_\theta} \sum_{i=0}^{1} \sum_{j=0}^{N_\theta - 1} \left[err_1^{\text{hh}}\left(\mathcal{A}_t, \mathcal{N}_{\boldsymbol{\rho}}^c, \mathcal{N}_{\boldsymbol{\rho}}^p, \mathcal{N}_{\boldsymbol{\rho}}^k\right) \right]_{i,j}^2 \right) \\
& + w_2^{\text{hh}} \left(\frac{1}{2N_\theta} \sum_{i=0}^{1} \sum_{j=0}^{N_\theta - 1} \left[err_2^{\text{hh}}\left(\mathcal{A}_t, \mathcal{N}_{\boldsymbol{\rho}}^c, \mathcal{N}_{\boldsymbol{\rho}}^p, \mathcal{N}_{\boldsymbol{\rho}}^k\right) \right]_{i,j}^2 \right) \\
& + w^p \left(err_1^{\text{p}}\left(\mathcal{A}_t, \mathcal{N}_{\boldsymbol{\rho}}^c, \mathcal{N}_{\boldsymbol{\rho}}^p, \mathcal{N}_{\boldsymbol{\rho}}^k\right) \right)^2,
\end{aligned} \tag{110}$$

where w_1^{firm}, w_2^{firm}, w_1^{hh}, w_2^{hh}, and w^p denote the weights of the different components of the loss function and where, with slight abuse of notation, $\boldsymbol{\rho}$ collects the trainable parameters of all four neural networks.

[41] For the endogenous grid method in higher dimensions, see Druedahl and Jørgensen (2017).

5.3.5 Forward simulation

Given the aggregate state-history pair, $\mathcal{A}_t$, we obtain $\mathcal{A}_{t+1}$ by drawing new random shocks for productivity and for the uncertainty regime according to the corresponding probability distributions, and by evolving the endogenous state according to the current guess of equilibrium policy functions. The evolution of the distribution of firms is computed using policy functions obtained by evaluation of the firm network $\mathcal{N}^k_{\boldsymbol{\rho}}$. To make sure that the evolution of the distribution of households satisfies market clearing condition, we use the market clearing policies $\mathcal{C}_t^{\text{grid, solved}}$ and p_t^{solved} computed using Newton-Raphson rootfinding algorithm.[42]

Computing the target values In order to compute the target values for household consumption, and for the stock price, $\mathcal{C}_t^{\text{grid, solved}}$ and p_t^{solved}, that are consistent with household optimality and market clearing, we rely on the endogeneous gridpoint method (EGM). Specifically, given a guess for current stock price p_t^{guess}, and current guess for equilibrium dynamics of the economy,[43] we use the EGM to derive the period t consumption function implied by expectations of $t+1$ marginal utility and asset returns. We denote this consumption function $c_t^{i,\text{ guess}}$ and corresponding savings function as $\theta_{t+1}^{i,\text{ guess}}$. Then we evaluate the excess demand function implied by household policies generated by the current price guess $ED(p_t^{\text{guess}}, \mathcal{A}_t, \mathcal{N}^c_{\boldsymbol{\rho}}, \mathcal{N}^p_{\boldsymbol{\rho}}, \mathcal{N}^k_{\boldsymbol{\rho}}) = \Theta_{t+1}^{\text{guess}} - 1$. Then we update the price guess using a Newton-Raphson algorithm

$$p_t^{\text{new}} = p_t^{\text{old}} - \frac{ED(p_t^{\text{old}}, \mathcal{A}_t, \mathcal{N}^c_{\boldsymbol{\rho}}, \mathcal{N}^p_{\boldsymbol{\rho}}, \mathcal{N}^k_{\boldsymbol{\rho}})}{\frac{\partial}{\partial p_t^{\text{old}}} ED(p_t^{\text{old}}, \mathcal{A}_t, \mathcal{N}^c_{\boldsymbol{\rho}}, \mathcal{N}^p_{\boldsymbol{\rho}}, \mathcal{N}^k_{\boldsymbol{\rho}})}. \tag{111}$$

Our shape preserving neural network architecture ensures that the excess demand function is well behaved in the price guess. Therefore, a small number of Newton-Raphson steps is sufficient to compute the market clearing price.[44] Furthermore, the prediction of the neural network $\mathcal{N}^p_{\boldsymbol{\rho}}$ provides us with an excellent starting guess for the market clearing price, once the initial training stage is completed.[45] When the market clearing price p_t^{solved} is found, we save the price and the corresponding household policies $\mathcal{C}_t^{\text{grid, solved}}$ as target for supervised training.

5.4 Training

5.4.1 Outline of the step-wise training procedure

We again follow a step-wise training procedure. Our step-wise procedure ensures that our algorithm learns reasonable firm policies, and hence generates sensible wages and dividends before it starts

[42] Note that the consumption policies, together with the stock price, pin down the asset policies via the households' budget constraint.

[43] *i.e.* we evaluate the evolution of the firm distribution using the firm policy functions encoded in the firm network, and then we evaluate the evolution of the household distribution using household network as an input to solve for market clearing price and policy.

[44] For GPU efficiency reasons, we use a predetermined number of Newton-Raphson steps, so that the same number of steps is used for all the states. The derivative of the excess demand function with respect to the guessed price can be computed efficiently using automatic differentiation.

[45] Hence, the algorithm might start with the initial number of Newton-Raphson iterations set to 10, which can be reduced to 3 once the algorithm converges to sufficiently low level of price prediction errors.

solving the household block.

Step 1: Firm side only: In the first step we only train the firm side of the model. To do so, we set the stochastic discount factor of the firm to be equal to the patience parameter $\Lambda_{t+1,t} = \beta$, and set the weights in the loss function to $w_1^{\text{firm}} = w_2^{\text{firm}} = 1$ and $w_1^{\text{hh}} = w_2^{\text{hh}} = w^p = 0$. Imposing an exogenous stochastic discount factor and setting weights on household and price error to zero allows us to isolate and solve the firm problem, so we can then start solving the household problem with a good guess for firm policies already at hand. Furthermore, we start the training procedure with lower values for the adjustment cost parameters $\xi^{\text{up}} = 0.1$ and $\xi^{\text{down}} = 0.25$. The policies for the simplified parameterization can be learned quickly by the neural network.

Step 2, Pre-training price and household policies In the second step, we ensure that the neural networks parameterizing household policy and the equity price function also start with a good initial guess. To facilitate initial convergence, we again start the training procedure in a simplified economy: we introduce an artificial parameter τ. τ modifies the payout of equity to $p_{t+1}\Gamma(1-\tau)+D_{t+1}$. With $\tau = 0$, we recover the original economy. Setting $\tau = 1$ makes equity effectively a claim on $t+1$ dividends, reducing it into a short-lived asset. Given $\tau = 0$, we use $p_t^{\text{pre-train}} = D_t$ as a pre-training target for the price network $\mathcal{N}_{\boldsymbol{\rho}}^p$. For the household policy function, we use $\theta_{t+1}^{\text{i}} = \max(0, 0.7\theta_t^i - 0.1)$ for $e_t^i = e_0$ and $\theta_{t+1}^{\text{i}} = 0.7\theta_t^i + 0.6$ for $e_t^i = e_1$ as pre-training targets. This provides us with a good starting guess for the price function (in the short-lived asset calibration, $\tau = 1$), and the household policy functions.

Step 3, Training the firm and household side together: Retaining the simplified parameterization of the model, we now train all price and policy functions on the full equilibrium loss function, given in equation (110) with weights set to $w_1^{\text{firm}} = w_2^{\text{firm}} = w_1^{\text{hh}} = w_2^{\text{hh}} = w^p = 1$.

Step 4, Step-wise model transformation to full parameterization: We then gradually change the parameters to the desired parameterization of the full model. We linearly increase the adjustment cost parameters to the final values of $\xi^{\text{up}} = 1.0$ and $\xi^{\text{up}} = 2.5$. At the same time, we increase τ from $\tau = 1$ to $\tau = 0$ following a quadratic schedule. Additionally, we gradually change the stochastic discount factor used by firms from β to $\Lambda_{t+1,t}$. At every step, firms use a weighted average between β and $\Lambda_{t+1,t}$ as a stochastic discount factor. We start with a full weight of 1 on β. Then we gradually decrease it to 0 while we increase the weight on $\Lambda_{t+1,t}$ from 0 to 1.

Step 5, Training with the final model parameters: We train the neural networks on the full loss function with the final model parameterization, until we reach sufficiently low level of remaining errors in equilibrium conditions.

5.4.2 Hyperparameters

We summarize the hyperparameters that we used to solve the model in table 6. We choose all four neural networks, $\mathcal{N}_{\boldsymbol{\rho}}^k$, $\mathcal{N}_{\boldsymbol{\rho}}^\lambda$, $\mathcal{N}_{\boldsymbol{\rho}}^c$, and $\mathcal{N}_{\boldsymbol{\rho}}^p$, to be densely connected feed-forward neural networks with three hidden layers and gelu activation functions. We truncate the history of shocks after 300 periods. The input layers consists of 600 neurons, corresponding to the truncated history of uncertainty regimes and the innovations to productivity. Each hidden layer consists of 1024 gelu activated neurons.

Parameter		Value
T	Length of history of shocks (per shock)	300
N^{input}	# nodes input layer	600
$N^{\text{hidden 1}}$	# neurons in the first hidden layer (activation)	1024 (gelu)
$N^{\text{hidden 2}}$	# neurons in the second hidden layer (activation)	1024 (gelu)
$N^{\text{hidden 3}}$	# neurons in the third hidden layer (activation)	1024 (gelu)
N^{output}	# neurons in the output layer	$\mathcal{N}^k_{\boldsymbol{\rho}}$: 600 $\mathcal{N}^\lambda_{\boldsymbol{\rho}}$: 600 $\mathcal{N}^c_{\boldsymbol{\rho}}$: 200 $\mathcal{N}^p_{\boldsymbol{\rho}}$: 1
N^k	# grid points for capital grid (log spaced)	200
N^θ	# grid points for asset grid (log spaced)	200
N^{quad}	# quadrature nodes TFP	5
N^{data}	States per episode	131072
N^{mb}	mini-batch size	128
N^{episodes}, step 1	Training episodes	100
N^{episodes}, step 2	Training episodes	100
N^{episodes}, step 3	Training episodes	100
N^{episodes}, step 4	Training episodes	500
N^{episodes}, step 5	Training episodes	16300
Optimizer	Optimizer	Adam
α^{learn}	Learning rate	10^{-6}

Table 6: Hyperparameter values for the heterogeneous firms and households model.

5.5 Accuracy

In this section, we demonstrate the accuracy of our solution. Since a closed-form or a high-quality numerical reference solution is not available, we investigate the errors in the equilibrium conditions, implied by the price and policy functions encoded in the learned parameters of our neural networks. We report the accuracy measures in economically interpretable units. In addition, we provide a comprehensive set of statistics on the distribution of errors across the state space.

5.5.1 Firm policies

The optimal firm policies are characterized by the set of firms' KKT conditions, given in equations (50) to (53). Rearranging equation (50) we obtain:

$$0 = \frac{\Gamma \mathrm{E}\left[\Lambda_{t+1,t}\left(1 + r^{i,k}_{t+1} + \frac{\partial}{\partial k^i_{t+1}}\psi(k^i_{t+2}, k^i_{t+1})\right)(1+\lambda^i_{t+1})\right]}{\left(1 + \frac{\partial}{\partial k^i_{t+1}}\psi(k^i_{t+1}, k^i_t)\right)(1+\lambda^i_t)} - 1, \tag{112}$$

where the numerator in the first term denotes the expected marginal benefit of additional capital in the next period and the denominator gives the marginal cost in the current period. The errors in equation (112) are therefore interpretable as relative marginal cost errors.[46] The left panel of figure 5 shows statistics of the distribution of errors in equation (112) for different idiosyncratic productivity

[46] An error of 0.01, for example, would indicate that the marginal benefit is 1% higher than the marginal cost.

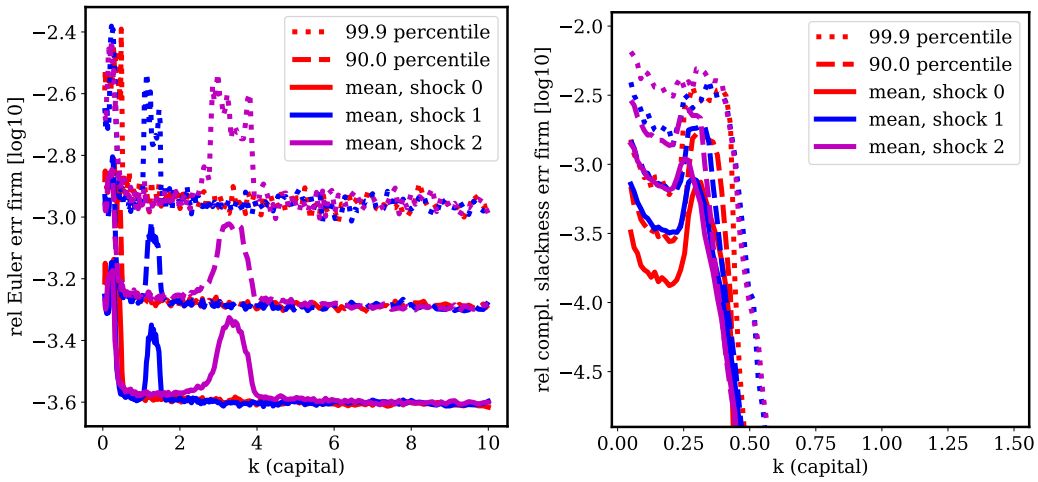

Figure 5: Remaining errors in the equilibrium conditions for the firm problem. The left panel shows the errors in the firms' Euler equations and the right panel shows the errors in the remaining KKT conditions.

shocks, idiosyncratic levels of capital, and 1024 aggregate states drawn from the ergodic distribution of the economy. The mean marginal cost error remain below 0.07%, the 90th percentile below 0.16% and the 99.9th percentile below 0.4% for all levels of firm capital and the three productivity levels. Even though the errors are very low everywhere, they are slightly larger in the areas of the state space where the firm policies feature a kink due to the constraint on dividends and due to asymmetric adjustment costs.[47]

The right panel of figure 5 shows statistics of the distribution of errors in the remaining KKT condition of firms

$$0 = \psi^{\text{FB}}\left(\frac{\lambda_t^i}{1+\lambda_t^i}, \frac{d_t^i - \underline{d}}{k_t^i}\right), \tag{113}$$

where $\psi^{\text{FB}}(x, y)$ denotes the Fischer-Burmeister function. Errors in equation (113) summarize violations of the all remaining KKT conditions associated with the firm problem.[48] Except for very low levels of capital, the error is virtually zero. Throughout, the mean error remains below 0.2% and even the 99.9th percentile of errors remains below 0.8%. We conclude that the firm policies are computed with high accuracy across the aggregate and idiosyncratic state space.

5.5.2 Household policies

As described above, the endogenous gridpoints method combined with a Newton-Raphson loop allows us to jointly solve for the market clearing equity price and the corresponding consumption policies of households. Hence, we can assess the accuracy of the consumption policies learned by the

[47] For a fixed idiosyncratic level of capital, and a fixed idiosyncratic productivity level, the mean and the percentiles are computed across 1024 aggregate states drawn from the ergodic distribution of the economy.

[48] Because the Fischer-Burmeister function satisfies $\psi^{\text{FB}}(x, y) = 0 \Leftrightarrow x \geq 0, y \geq 0, xy = 0$.

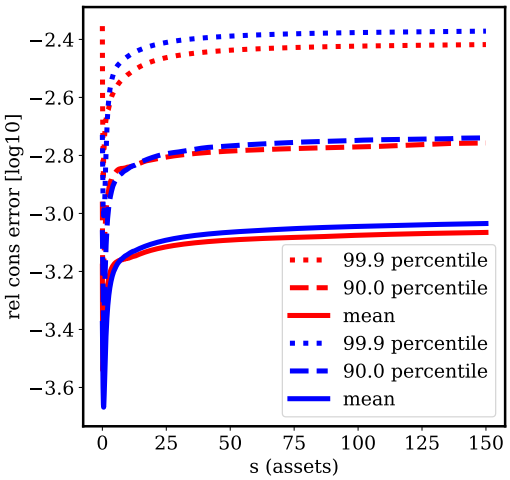

Figure 6: Remaining errors in the equilibrium conditions for the households' optimality conditions, expressed in units of relative consumption errors.

neural network by comparing the neural network predictions for period t to the period t consumption policies implied by household optimality and market clearing prices.[49] Figure 6 shows the accuracy of the consumption function learned by the neural network across the aggregate and idiosyncratic state space. For all the idiosyncratic states, the mean error remains below 0.08%, the 90th percentile below 0.17%, and the 99.9th percentile below 0.4%.[50] We conclude that the household policies are approximated to a high accuracy.

5.5.3 Stock price

The left panel of figure 7 shows the distribution of discrepancies between the equity price predicted by the neural network relative and the market clearing equity price obtained using the EGM-Newton-Raphson algorithm. The mean error of the price prediction is 0.17%, the 90th percentile 0.35% and the 99.9th percentile is 0.80%. The right panel of figure 7 shows the price predictions by the neural network together with the computed market clearing prices in a scatter plot against aggregate dividends. While the accuracy of the price function is slightly lower than the accuracy we achieved for the policy functions of firms and households, it remains high. The error in the asset demand, as implied by the predicted price and the predicted policy functions is lower and below 0.1% over the whole simulated ergodic set.

[49] Where we evaluate the expectations over $t+1$ objects using policies and prices encoded by neural networks.

[50] For a fixed idiosyncratic asset holding and a fixed idiosyncratic productivity level, the mean and the percentiles are computed across 1024 aggregate states drawn from the ergodic distribution of the economy.

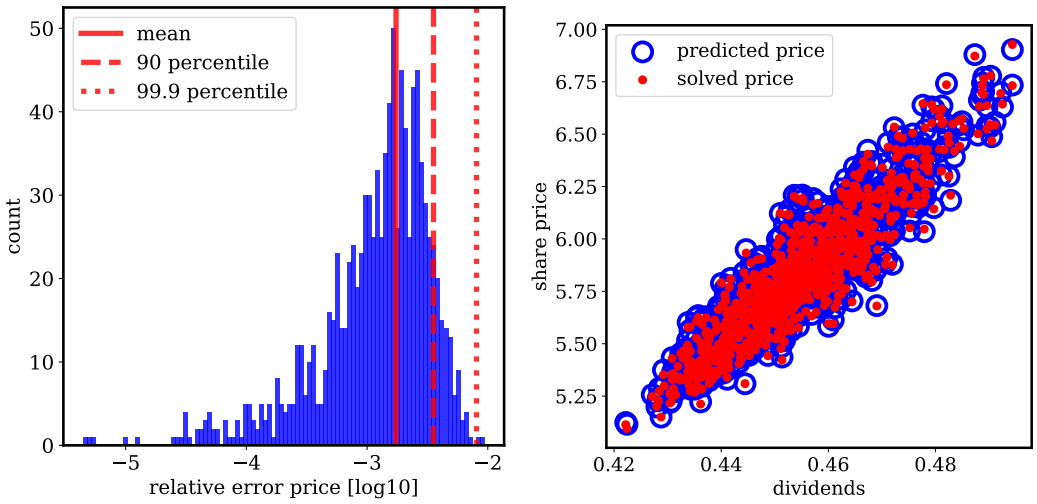

Figure 7: Prediction errors for the price network. The left panel shows the distribution of relative prediction errors across 1024 aggregate states drawn from the ergodic distribution of the economy. The right panel shows the network predictions (blue circles) and the solved market clearing prices (red dots).

5.6 Inspecting the learned equilibrium policies

5.6.1 Firms

We turn to inspecting the equilibrium policies. The left panel of figure 8 shows the density of firms across the capital grid for different simulated aggregate states. We can see that the distribution of firms across capital varies substantially. Due to the decreasing returns to scale production function, together with capital adjustment costs, this variation directly affects aggregate wages and dividends (see, *e.g.*, equation (59)), such that the households problem depends crucially on the firm distribution.

The middle panel in figure 9 shows the dividend policies of firms for different capital levels (horizontal axis) and different idiosyncratic productivity levels (represented by different colors). The shaded lines show the policies for several randomly selected aggregate states, and the solid lines show the mean taken over aggregate states. Each of the policy functions shows two kinks: at the first kink, the non-negativity constraint on dividends stops to bind. At the second kink, firms switch from adjusting capital upward to adjusting it downward. The exact location of the kinks depends on the productivity level and the firms capital as well as on the aggregate state of the economy.

Similarly, the right panel shows the policy functions for the KKT multiplier on the dividend constraint. Perhaps surprisingly, the multiplier for very low values of capital is larger for the low productivity values than for the high productivity values. This is the case because, while the firms are constraint for all three productivity levels, the high productivity firms are able to choose a higher capital level for the next period.

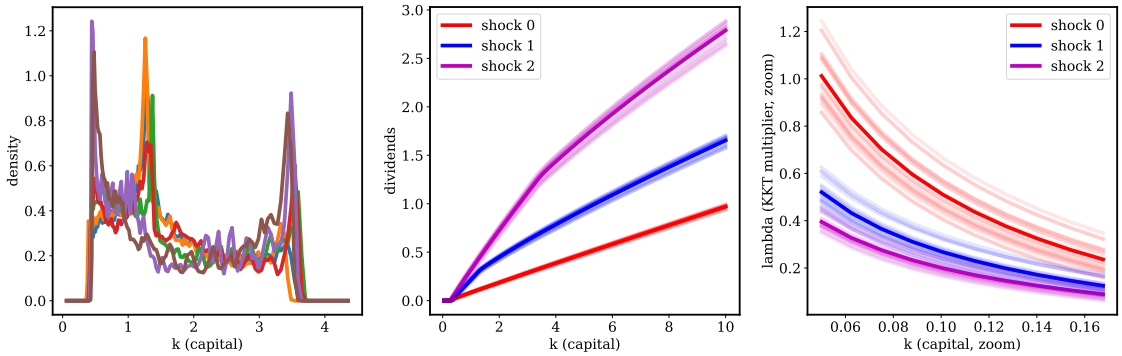

Figure 8: Left panel: density of firms across the capital grid for different simulated aggregate states. Middle panel: dividends policy for firms over the capital grid (horizontal axis) and across different productivity levels (different colors). The shaded lines show the policies for different aggregate states, and the solid lines show the average taken across the sample of aggregate states. Right panel: firms' KKT multiplier associated with the non-negative dividend constraint over the different capital and productivity levels, as well as across different aggregate states.

5.6.2 Households

Figure 9 shows the household policies predicted by the neural network. Each of the shaded lines corresponds to a different aggregate state, and the solid lines correspond to the mean, averaged over aggregate states, for a specific idiosyncratic productivity and wealth level.

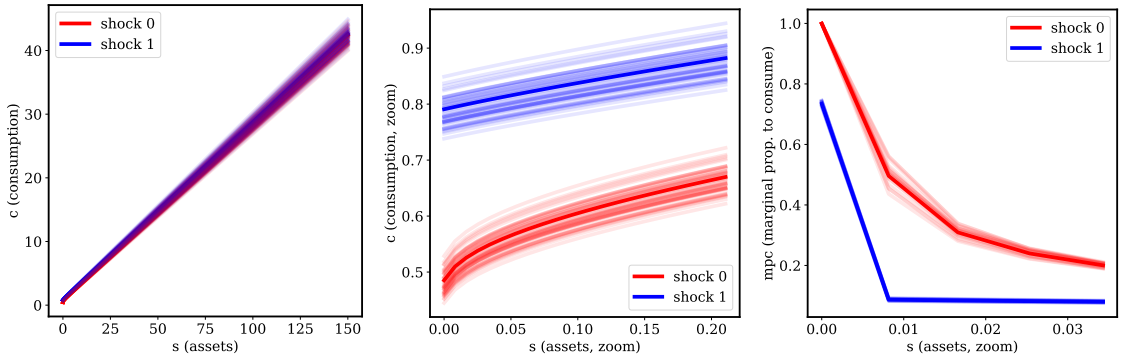

Figure 9: Left and middle panel: consumption function of households for their idiosyncratic productivity level (in different colors) and for different idiosyncratic asset holdings (horizontal axis). The shaded lines show the consumption policies for several different aggregate states, and the solid lines show the mean across aggregate states. Right panel: marginal propensity to consume for different productivity and wealth levels, as well as for several aggregate states.

As figure 9 shows, consumption varies substantially between different aggregate states. We can also see that, for each of the aggregate states, consumption is increasing and concave in the wealth of households. Our shape preserving neural network architecture ensures that this economically important feature of the consumption functions is guaranteed. We achieve this using the three step procedure described above. The households' marginal propensities to consume are constructed from

the predictions of the neural network, such that the marginal propensity to consume, for a given aggregate state and idiosyncratic productivity level, is always decreasing and convex in cash-at-hand, as well as bounded between 1 and 0 (see the right panel in figure 9). This ensures on the one hand the monotonicity and concavity of the consumption function, and on the other hand facilitates the precise prediction of the marginal propensity to consume (and hence consumption) of borrowing constraint households.

6 Conclusion

We introduce a new algorithm for computing global solutions of dynamic stochastic general equilibrium models. Exploiting the ergodic property of a large class of dynamic economies, we rely on the truncated history of aggregate shocks as an approximate sufficient statistic for the aggregate state. Based on this approximation, we use deep neural networks to parameterize the mapping from truncated aggregate shock histories to equilibrium objects of interest using deep neural networks. Finally, we train our neural networks to satisfy all the equilibrium conditions along simulated paths of the economy.

We illustrate the accuracy and wide applicability of our algorithm by solving three example economies. As a first example and proof of concept, we solve a simple stochastic growth model. This allows us to compare our solution against a high-accuracy solution that can be obtained using conventional grid-based methods. Second, we solve an overlapping generations model with 72 age groups, portfolio choice, and multiple sources of aggregate risk. The overlapping generations model shows that the sequence space approach is applicable, also when there is a clear dependence of equilibrium objects on long histories of shocks: the shocks experienced during an agent's lifetime affect their subsequent equilibrium outcomes. The model includes stochastic shocks to productivity and depreciation, which are drawn from normal distributions, together with a Markov regime switching model with two discrete states. We thereby illustrate that our method is capable of solving models with several aggregate shocks of different types. As a third example, we consider an economy where a continuum of heterogeneous households trade in a long-lived financial asset, which constitutes a claim to the dividends paid by a continuum of heterogeneous firms, who operate a decreasing returns to scale production technology and face asymmetric adjustment costs on capital. Additionally, the economy is subject to stochastic fluctuations in aggregate productivity and in the level of aggregate and idiosyncratic uncertainty. The state hence includes two endogenously moving distributions: the distribution over wealth and productivity on the household side and the distribution of firms over productivity and capital. We show that we solve all three models accurately, with mean relative errors in equilibrium conditions below 0.1%.

To solve models with aggregate and idiosyncratic risk, we introduce shape preserving operator learning: we train deep neural networks to predict idiosyncratic policy *functions* as a function of the truncated history of aggregate shocks, such that we can guarantee the predicted policy functions to be monotone or concave in key idiosyncratic state variables, such as household wealth. These guaranties allow us to use the method of endogenous gridpoints and a simple, yet-reliable, Newton-Raphson algorithm to obtain high-quality training targets for supervised learning of household policies and

the market clearing stock price, further increasing the robustness of our deep learning algorithm.

References

Adenbaum, J., Babalievsky, F., and Jungerman, W. (2024). Deep reinforcement learning for economics. Technical report, Working Paper.

Aiyagari, S. R. (1994). Uninsured idiosyncratic risk and aggregate saving. *The Quarterly Journal of Economics*, 109(3):659–684.

Auclert, A., Bardóczy, B., Rognlie, M., and Straub, L. (2021). Using the sequence-space jacobian to solve and estimate heterogeneous-agent models. *Econometrica*, 89(5):2375–2408.

Azinovic, M., Cole, H. L., and Kubler, F. (2023). Asset pricing in a low rate environment. Working Paper 31832, National Bureau of Economic Research.

Azinovic, M., Gaegauf, L., and Scheidegger, S. (2022). Deep equilibrium nets. *International Economic Review*, 63(4):1471–1525.

Azinovic, M. and Zemlicka, J. (2024). Intergenerational consequences of rare disasters.

Bayer, C. and Luetticke, R. (2020). Solving discrete time heterogeneous agent models with aggregate risk and many idiosyncratic states by perturbation. *Quantitative Economics*, 11(4):1253–1288.

Bellman, R. (1961). *Adaptive Control Processes: A Guided Tour*. 'Rand Corporation. Research studies. Princeton University Press.

Bewley, T. (1977). The permanent income hypothesis: A theoretical formulation. *Journal of Economic Theory*, 16(2):252–292.

Bloom, N., Floetotto, M., Jaimovich, N., Saporta-Eksten, I., and Terry, S. J. (2018). Really uncertain business cycles. *Econometrica*, 86(3):1031–1065.

Boppart, T., Krusell, P., and Mitman, K. (2018). Exploiting mit shocks in heterogeneous-agent economies: the impulse response as a numerical derivative. *Journal of Economic Dynamics and Control*, 89:68–92.

Bretscher, L., Fernández-Villaverde, J., and Scheidegger, S. (2022). Ricardian business cycles. *Available at SSRN 4278274*.

Brock, W. A. and Mirman, L. J. (1972). Optimal economic growth and uncertainty: the discounted case. *Journal of Economic Theory*, 4(3):479–513.

Brumm, J. and Scheidegger, S. (2017). Using adaptive sparse grids to solve high-dimensional dynamic models. *Econometrica*, 85(5):1575–1612.

Cai, Y. and Judd, K. L. (2012). Dynamic programming with shape-preserving rational spline hermite interpolation. *Economics Letters*, 117(1):161–164.

Carroll, C. D. (2006). The method of endogenous gridpoints for solving dynamic stochastic optimization problems. *Economics letters*, 91(3):312–320.

Carvalho, V. M., Covarrubias, M., and Nuno, G. (2025). Planning against disasters in dynamic production networks. *Working Paper*.

Chien, Y., Cole, H., and Lustig, H. (2011). A multiplier approach to understanding the macro implications of household finance. *The Review of Economic Studies*, 78(1):199–234.

Druedahl, J. and Jørgensen, T. H. (2017). A general endogenous grid method for multi-dimensional models with non-convexities and constraints. *Journal of Economic Dynamics and Control*, 74:87–107.

Druedahl, J. and Ropke, J. (2025). Deep learning algorithms for solving finite-horizon models. Technical report, University of Copenhagen.

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

Duarte, V., Fonseca, J., Goodman, A. S., and Parker, J. A. (2021). Simple allocation rules and optimal portfolio choice over the lifecycle. Working Paper 29559, National Bureau of Economic Research.

Duffy, J. and McNelis, P. D. (2001). Approximating and simulating the stochastic growth model: Parameterized expectations, neural networks, and the genetic algorithm. *Journal of Economic Dynamics and Control*, 25(9):1273–1303.

Fernández-Villaverde, J., Hurtado, S., and Nuno, G. (2023). Financial frictions and the wealth distribution. *Econometrica*, 91(3):869–901.

Folini, D., Friedl, A., Kübler, F., and Scheidegger, S. (2025). The climate in climate economics. *Review of Economic Studies*, 92(1):299–338.

Friedl, A., Kübler, F., Scheidegger, S., and Usui, T. (2023). Deep uncertainty quantification: With an application to integrated assessment models. Technical report, Working paper, University of Lausanne.

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

Gopalakrishna, G., Gu, Z., and Payne, J. (2024). Asset pricing, participation constraints, and inequality. Technical report, Princeton Working Paper.

Grohs, P., Hornung, F., Jentzen, A., and Von Wurstemberger, P. (2018). A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of black-scholes partial differential equations. *arXiv preprint arXiv:1809.02362*.

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

Han, J., Yang, Y., et al. (2022). Deepham: A global solution method for heterogeneous agent models with aggregate shocks. *arXiv preprint arXiv:2112.14377*.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

Huggett, M. (1993). The risk-free rate in heterogeneous-agent incomplete-insurance economies. *Journal of economic Dynamics and Control*, 17(5-6):953–969.

Imrohoroğlu, A. (1989). Cost of business cycles with indivisibilities and liquidity constraints. *Journal of Political economy*, 97(6):1364–1383.

Jentzen, A., Salimova, D., and Welti, T. (2018). A proof that deep artificial neural networks overcome the curse of dimensionality in the numerical approximation of kolmogorov partial differential equations with constant diffusion and nonlinear drift coefficients. *arXiv preprint arXiv:1809.07321*.

Jiang, H. (1999). Global convergence analysis of the generalized newton and gauss-newton methods of the fischer-burmeister equation for the complementarity problem. *Mathematics of Operations Research*, 24(3):529–543.

Judd, K. L. (1998). *Numerical methods in economics*. MIT press.

Judd, K. L. and Solnick, A. (1994). Numerical dynamic programming with shape-preserving splines. *Report.[657]*.

Jungerman, W. (2023). Dynamic monopsony and human capital. Technical report, mimeo.

Kahou, M. E., Fernández-Villaverde, J., Gómez-Cardona, S., Perla, J., and Rosa, J. (2022). Spooky boundaries at a distance: Exploring transversality and stability with deep learning.

Kahou, M. E., Fernández-Villaverde, J., Perla, J., and Sood, A. (2021). Exploiting symmetry in high-dimensional dynamic programming. Working Paper 28981, National Bureau of Economic Research.

Kase, H., Melosi, L., and Rottner, M. (2023). Estimating nonlinear heterogeneous agents models with neural networks. *CEPR Discussion Paper No. DP17391*.

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.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krueger, D. and Kubler, F. (2004). Computing equilibrium in olg models with stochastic production. *Journal of Economic Dynamics and Control*, 28(7):1411–1436.

Krusell, P. and Smith, Jr, A. A. (1998). Income and wealth heterogeneity in the macroeconomy. *Journal of political Economy*, 106(5):867–896.

Lee, H. (2025). Global nonlinear solutions in sequence space and the generalized transition function. *Working paper*.

Maliar, L., Maliar, S., and Winant, P. (2021). Deep learning for solving dynamic economic models. *Journal of Monetary Economics*, 122:76–101.

Norets, A. (2012). Estimation of dynamic discrete choice models using artificial neural network approximations. *Econometric Reviews*, 31(1):84–106.

Payne, J., Rebei, A., and Yang, Y. (2024). Deep learning for search and matching models. *Available at SSRN*.

Reiter, M. (2009). Solving heterogeneous-agent models by projection and perturbation. *Journal of Economic Dynamics and Control*, 33(3):649–665.

Rouwenhorst, K. G. (1995). Asset pricing implications of equilibrium business cycle models. *Frontiers of business cycle research*, 1:294–330.

Sauzet, M. (2021). Projection methods via neural networks for continuous-time models. *Available at SSRN 3981838*.

Sun, J. E. (2025a). Continuation value is all you need: "drop-in" deep learning for heterogeneous-agent models with aggregate shocks.

Sun, J. E. (2025b). The distributional consequences of climate change: Housing wealth, expectations, and uncertainty.

Tauchen, G. (1986). Statistical properties of generalized method-of-moments estimators of structural parameters obtained from financial market data. *Journal of Business & Economic Statistics*, 4(4):397–416.

Valaitis, V. and Villa, A. T. (2024). A machine learning projection method for macro-finance models. *Quantitative Economics*, 15(1):145–173.

Young, E. R. (2010). Solving the incomplete markets model with aggregate uncertainty using the krusell–smith algorithm and non-stochastic simulations. *Journal of Economic Dynamics and Control*, 34(1):36–41.

Zhong, Y. (2023). Operator learning in macroeconomics.

Abstrakt

V tomto článku prezentujeme nový algoritmus hlubokého učení pro aproximaci funkcionálních rovnovážných stavů v dynamických stochastických ekonomikách s racionálními očekáváními. K parametrizaci rovnovážných objektů ekonomiky používáme hluboké neuronové sítě, a to jako funkci zkrácených historií agregátních exogenních šoků. Neuronové sítě trénujeme tak, aby splňovaly všechny rovnovážné podmínky podél simulovaných trajektorií ekonomiky. Pro ilustraci efektivity naší metody řešíme tři ekonomiky rostoucí složitosti: stochastický růstový model, vícerozměrnou ekonomiku překrývajících se generací s více zdroji agregátního rizika a nakonec ekonomiku, ve které jak domácnosti, tak firmy čelí nepojistitelnému idiosynkratickému riziku, šokům do agregátní produktivity a šokům do idiosynkratické i agregátní volatility. Dále ukazujeme, jak sestavit praktické architektury neuronových reakčních funkcí, které zaručují monotonicitu predikovaných reakčních funkcí, což usnadňuje využití metody endogenní mřížky ke zjednodušení částí našeho algoritmu.

Working Paper Series
ISSN 2788-0443

Individual researchers, as well as the on-line version of the CERGE-EI Working Papers (including their dissemination) were supported from institutional support RVO 67985998 from Economics Institute of the CAS, v. v. i.

Specific research support and/or other grants the researchers/publications benefited from are acknowledged at the beginning of the Paper.

Published by
Charles University, Center for Economic Research and Graduate Education (CERGE)
and
Economics Institute of the CAS, v. v. i. (EI)
CERGE-EI, Politických vězňů 7, 111 21 Prague 1, tel.: +420 224 005 153, Czech Republic.
Phone: + 420 224 005 153
Email: office@cerge-ei.cz
Web: https://www.cerge-ei.cz/

Editor: Byeongju Jeong

The paper is available online at https://www.cerge-ei.cz/working-papers/.

ISBN 978-80-7343-609-4 (Univerzita Karlova, Centrum pro ekonomický výzkum a doktorské studium)