Structural Reinforcement Learning for Heterogeneous Agent Macroeconomics

Yucheng Yang*,[1] Chiyuan Wang*,[2] Andreas Schaab[3] Benjamin Moll[4]

First version: December 2025
This version: March 2026
[latest version]

Abstract

We present a new approach to formulating and solving heterogeneous agent models with aggregate risk. We replace the cross-sectional distribution with low-dimensional prices as state variables and let agents learn equilibrium price dynamics directly from simulated paths. To do so, we introduce a *structural reinforcement learning* (SRL) method which treats prices via simulation while exploiting agents' structural knowledge of their own individual dynamics. Our SRL method yields a general and highly efficient global solution method for heterogeneous agent models that sidesteps the Master equation and handles models traditional methods struggle with, like those with nontrivial market-clearing conditions. We illustrate the approach in the Krusell-Smith model, the Huggett model with aggregate shocks, and a HANK model with a forward-looking Phillips curve, all of which we solve globally within minutes.

We thank Jesús Fernández-Villaverde, Jakob Foerster, Felix Kübler, Dima Mukhin, Richard Rogerson, Tom Sargent, Bo Li, Sebastian Towers, Gianluca Violante, Clarisse Wibault, Tiphaine Wibault, Zhuoran Yang, and many seminar participants for helpful comments. Yang acknowledges support from the Swiss NSF (#10003091), Wang acknowledges support from the NSFC (#72450002), Moll acknowledges support from the Leverhulme Trust and the European Research Council (#101200645). We will release codes implementing our computational experiments as soon as possible.

*Equal contribution.

[1]University of Zurich and SFI. Email: yucheng.yang@uzh.ch

[2]Peking University, School of Computer Science and BIGAI. Email: wang2021@stu.pku.edu.cn

[3]UC Berkeley. Email: schaab@berkeley.edu

[4]London School of Economics, corresponding author. Email: b.moll@lse.ac.uk

1 Introduction

Many of the most important questions in macroeconomics call for studying models with heterogeneous agents and aggregate risk. A well-known difficulty is that, in standard recursive rational-expectations formulations of such models, the cross-sectional distribution of agents becomes a state variable in the Bellman equation characterizing agents' decision problems – the "Master equation."[1] This difficulty arises even though agents do not directly "care about" the distribution, i.e. it does not enter their objective functions; instead, as is standard in competitive equilibrium models, they only care about prices. Intuitively, low-dimensional equilibrium prices do not follow a Markov process but the extremely high-dimensional distribution does. Therefore, agents with rational expectations forecast prices by forecasting distributions.

While the recent literature has made some impressive advances, the extreme curse of dimensionality inherent in the Master equation remains a central computational bottleneck for *global* solutions to heterogeneous agent models with aggregate risk.[2] Even seemingly simple model environments like a Huggett (1993) model with aggregate risk (in which there is a non-trivial market clearing condition) or a one-asset HANK model with a forward-looking Phillips curve lead to exceedingly difficult computational problems. This lack of a general and efficient global solution method limits the applicability of heterogeneous-agent macroeconomics, in particular to questions in which aggregate non-linearities play a key role.[3]

The contribution of this paper is to develop an approach that sidesteps the Master equation by using ideas from reinforcement learning (RL). RL means learning value or policy functions of incompletely-known Markov decision processes via some form of Monte Carlo simulation (Sutton and Barto, 2018). We apply this idea to equilibrium prices and let agents compute price expectations directly from simulated paths. However, our approach differs from standard RL in that we assume that agents have *structural knowledge* about the dynamics of their own individual states (e.g. their budget constraint and idiosyncratic income process). We term this hybrid approach *structural reinforcement learning* (SRL) and our specific algorithm a *structural policy gradient* (SPG) algorithm.[4] By imposing that policy functions depend only on current prices (or a short price history) we keep the state space low-dimensional so that we can work with a grid-based (tabular) approach rather than deep neural networks. Finally, we provide an efficient implementation in JAX (Bradbury et al., 2018; Sargent and Stachurski, 2025) that can be run on Google Colab.[5]

SRL delivers a new and highly efficient global solution method for heterogeneous agent models with aggregate risk. Importantly, it solves problems traditional methods struggle with. We demonstrate this with two example applications. First, a model environment with a non-

[1] See, e.g., Den Haan (1996), Krusell and Smith (1998), Schaab (2020), and Bilal (2023). The name "Master equation" comes from the mathematics literature on Mean Field Games (Cardaliaguet et al., 2019).

[2] There is, of course, also the global solution method of Krusell and Smith (1998) and Den Haan (1996) which assumes that agents forecast prices by forecasting *moments* of cross-sectional distributions rather than the distributions themselves. We discuss similarities and differences to our approach further below.

[3] While we do not consider such models in the present draft, we think that one particularly promising application of our global solution method will be modeling infrequent but large boom-bust cycles like financial crises.

[4] The "structural" in SRL is analogous to that in structural vector autoregressions (SVARs).

[5] Google Colab is a cloud computing platform that is easily accessible to all researchers.

trivial market-clearing conditions, namely a Huggett (1993) model with aggregate risk. Second, a HANK model with a forward-looking Phillips curve. Both model environments have proven notoriously difficult for traditional methods.[6] We instead solve the Huggett model in around one minute and the HANK model in around three minutes.[7] For completeness, we also solve the easier Krusell and Smith (1998) model in around 55 seconds.

Almost all existing global solution methods for heterogeneous agent models use dynamic programming. They either use dynamic programming to directly tackle the high-dimensional Master equation with the distribution as state variable; or, as in Krusell and Smith (1998), they use model-generated data to estimate a low-dimensional Markovian "perceived law of motion" (PLM) for moments of the distribution and then apply dynamic programming to this lower-dimensional approximate problem.[8] Our SPG algorithm instead works directly with the sequential formulation of the problem and never attempts to force it into the standard Markovian structure required for applying dynamic programming. While we reduce the dimensionality of the state space in a fashion reminiscent of moment-based methods, we never estimate a PLM. Instead, we use the simple idea at the core of all RL methods that value functions are expected values – here, expected discounted lifetime utilities – and can therefore be approximated by averaging across simulated trajectories. We then find the low-dimensional policy function that maximizes expected lifetime utility using stochastic gradient ascent.

The key elements of our SPG method delivering fast computations even in challenging model environments are as follows. Most importantly, as already mentioned, we replace the cross-sectional distribution with low-dimensional prices as state variables. We do this in two steps. The first step is to impose that agents observe the history of equilibrium prices but not the cross-sectional distribution. By the Wold representation theorem, this assumption does not, by itself, imply any departure from full information rational expectations. In the second step, we restrict the agents' state space: we assume that their policy functions depend only on current prices or, perhaps, a short price history. This second assumption means that we do not solve for the model's rational-expectations equilibrium; instead, we solve for a restricted perceptions equilibrium (RPE) in the sense of Sargent (1991), Evans and Honkapohja (2001), or Branch (2006): while agents' expectations are restricted, they are nevertheless statistically consistent with actual equilibrium outcomes, thereby fulfilling one of the desiderata of rational expectations.

The next key element delivering fast computations is the defining assumption of our SPG approach that agents have structural knowledge about the dynamics of their own individual

[6]There is an interesting historical note regarding the computational difficulty of the Huggett model with aggregate risk. According to Maliar and Maliar (2020), in the influential JEDC special issue on numerical solution methods for HA models with aggregate risk (Den Haan et al., 2010), all participants were initially asked to solve two benchmark models globally – the Krusell and Smith (1998) model and the Huggett (1993) model with aggregate shocks (Maliar and Maliar call this "the HANC model with savings through bonds"). However, as they report, no single team was able to successfully solve the latter model, and it was ultimately dropped from the JEDC project.

[7]All experiments are implemented in JAX and are run on a single NVIDIA A100 GPU on Google Colab. We will discuss the precise convergence criteria in Section 3.7.

[8]The PLM is simply an approximate Markov process for the low-dimensional vector of moments. The PLM may have a simple parametric (e.g. linear) functional form as in Krusell and Smith (1998) or it may be a general, non-linear function, e.g. parameterized by a neural network as in Fernández-Villaverde et al. (2023).

states. In contrast to standard policy gradient methods which estimate approximate policy gradients, this assumption allows us to compute exact policy gradients by differentiating through these individual dynamics. The only part of the environment that is treated as unknown is the process for general-equilibrium prices and aggregate shocks, which is learned from simulated data. Specifically, our SPG method discretizes the individual state space so that agents' individual states evolve according to a known transition matrix $\mathbf{A}_{\boldsymbol{\pi}}$ where $\boldsymbol{\pi}$ denotes the vector of individual policies. When computing policy gradients of lifetime utilities with respect to $\boldsymbol{\pi}$, we exploit this structural knowledge and differentiate through $\mathbf{A}_{\boldsymbol{\pi}}$ while using simulation only for prices and aggregate shocks.

Finally, our low-dimensional grid- and price-based policy functions allow us to (globally) simulate the economy forward in time in a very efficient manner. For any particular trajectory of aggregate shocks, we update the distribution using the "histogram method" of Young (2010). The fact that policy functions depend on current prices allows us to efficiently handle non-trivial market clearing conditions like in Huggett (1993). Intuitively, *policy functions double as individual demand schedules*. Integrating across the distribution to obtain the corresponding aggregate demand schedule, it is then straightforward to solve for equilibrium prices at each point in time along a simulation path. Our treatment of market clearing mirrors the RL literature's distinction between agents and environment: agents can interact with their environment under any given policy including suboptimal ones; finding optimal policies is conceptually separate. In line with this dichotomy, we treat market clearing as part of the environment and find equilibrium prices also for suboptimal policies. This approach differs from standard practice in macroeconomics which first finds optimal policies given prices in an inner loop and then finds equilibrium prices in an outer loop.

In summary, SRL enables efficient *global* solutions of heterogeneous-agent models with full cross-sectional distributions. Importantly, while our restricted state space (and use of an RPE) simplifies agents' decision problems *inside* the model, it does *not* diminish the rich dynamics of the *economy* which still evolves stochastically and non-linearly, driven by the policy functions of forward-looking heterogeneous agents.

We illustrate our method in three benchmark environments: a Huggett (1993) economy, the classic Krusell and Smith (1998) model, and a one-account heterogeneous agent New Keynesian (HANK) model with sticky prices. In all three cases, our price-based SRL algorithm converges quickly on modern hardware: solving the Krusell-Smith model takes 55 seconds while the Huggett and HANK models – typically viewed as more challenging because of their non-trivial market-clearing conditions and, in the HANK case, a forward-looking Phillips curve – take only modestly longer (about 75 seconds and roughly 3 minutes, respectively). We present extensive tests to verify the accuracy of our solutions. For the Krusell-Smith model, our method produces equilibrium dynamics that closely match the rational expectations obtained using deep-learning-based methods. As a complementary exercise, we extend the information set to allow agents to condition on lagged prices and show that this richer price history does not meaningfully change the solution, suggesting that current prices already contain most of the information that matters for behavior. Finally, in the HANK application we show how to

use our approach to solve the household and firm problems jointly, using the same SPG algorithm not only for consumption-saving decisions but also for the forward-looking price-setting problem of firms.

To be clear, we do not view SRL as an empirically realistic model of human learning.[9] Instead, it is an efficient computational method for finding RPEs in heterogeneous agent models with aggregate risk. Nevertheless, as we note in the conclusion, the core idea of our approach – agents forming expectations about equilibrium prices by sampling – could, in principle, serve as a building block for an empirical theory of expectations formation in macroeconomics.

Relation to Economics Literature. A huge theoretical and quantitative literature studies environments in which heterogeneous households are subject to uninsurable idiosyncratic shocks. See Krusell and Smith (2006), Heathcote et al. (2009), Quadrini and Ríos-Rull (2015), Krueger et al. (2016), Sargent (2023), and Auclert et al. (2025a) for surveys.

Within this literature, a sizable subliterature is concerned with the *global* solution of heterogeneous agent models with aggregate risk. A first strand of the literature tackles the Master equation directly, typically using deep neural networks – see e.g. Han et al. (2021), Maliar et al. (2021), Azinovic et al. (2022), Kahou et al. (2021), Duarte et al. (2024), Huang (2023), Kase et al. (2024), Gu et al. (2024), Payne et al. (2024), and Gopalakrishna et al. (2024). Our approach differs from all of these papers in that we *sidestep* the Master equation rather than attempting to "tame the curse of dimensionality" inherent in it. Among these papers, Han et al. (2021) is most related to and influential for our work: like them, we learn value and policy functions of heterogeneous agents via simulation; also like them, we take advantage of agents' structural knowledge of their own individual dynamics – what we have termed a "structural" RL approach. The key difference is that Han et al. (2021) include the high-dimensional cross-sectional distribution in the agents' state space whereas we replace this distribution with low-dimensional prices as state variables.

A second strand of the literature estimates a low-dimensional Markovian PLM for moments of the distribution and then applies dynamic programming as in Krusell and Smith (1998) and Den Haan (1996). Similar to us, this approach sidesteps the Master equation by working with a low-dimensional state space that does not include the distribution. One variant of this approach formulates PLMs directly in terms of equilibrium prices so that – like in our approach – this low-dimensional state space includes prices.[10] However, this approach introduces an additional fixed-point loop in which the perceived law of motion is repeatedly updated and the associated dynamic programming problem re-solved, a procedure that is computationally slow in many economically interesting environments, particularly those with non-trivial market-clearing conditions. By contrast, we work directly with the problem's sequential formulation and never estimate a PLM.

[9] Moll (2025) proposed three criteria for alternatives to rational expectation in heterogeneous agent models: (1) computational tractability, (2) consistency with empirical evidence, and (3) (some) immunity to the Lucas critique. Our SRL approach delivers on (1) and (3) but not (2).

[10] See for example Lee and Wolpin (2006), Storesletten et al. (2007), Gomes and Michaelides (2008), Favilukis et al. (2017), Llull (2018), Kaplan et al. (2020), and Fernández-Villaverde et al. (2024b).

Our approach is also related to the adaptive learning literature (e.g. Bray, 1982; Marcet and Sargent, 1989; Evans and Honkapohja, 2001). Jacobson (2025) and Giusto (2014) apply adaptive learning in heterogeneous agent models with aggregate risk.[11] Similar to the learning literature, our approach computes an equilibrium that is "self-confirming" (Sargent, 1999; Cho and Sargent, 2016).[12] But because we restrict the individual state space to not include the distribution, our self-confirming equilibrium is an RPE (Sargent, 1991; Branch, 2006). In the language of Guarda (2025), our RPE features expectations that are both "narrow" and "short": narrow because they do not condition on the distribution and short because they do not condition on a long price history.[13]

Our focus on a low-dimensional vector of equilibrium variables links SRL to the "sequence space" approach of Auclert et al. (2021) and Auclert et al. (2025b). The difference is that SRL handles stochastic price sequences outside steady-state neighborhoods. By using Monte Carlo simulation, it yields a global rather than local solution method in sequence space.[14]

Relation to RL Literature. Our approach connects to several central ideas in the RL literature. As already mentioned, in RL, agents learn optimal policies in Markov decision processes using sampled trajectories (Sutton and Barto, 2018). RL is at the core of some impressive advances in artificial intelligence, e.g. RL agents learning to play Go and Atari games better than humans (Mnih et al., 2015; Silver et al., 2016, 2017) or post-training of "reasoning" large language models (LLMs OpenAI, 2024; DeepSeek-AI, 2025).[15]

Conceptually, our SRL method adopts the RL principle of optimizing policies using Monte Carlo estimates of expected returns. But it exploits full structural knowledge of each agent's reward function and individual transition dynamics to compute exact end-to-end policy gradients. Only equilibrium objects are treated as unknown parts of the environment. In this sense, our SRL approach is a hybrid between dynamic programming and model-free RL: we differentiate exactly through the known micro-level transition probabilities while learning the induced macro environment from simulation. In a companion paper (Wibault et al., 2026), we provide a more detailed taxonomy and term such approaches "Hybrid Structural Methods" (HSMs). We also show that, when available, such hybrid approaches outperform model-free RL.

While classical RL demonstrates considerable power in solving complex environments, its general-purpose nature implies suboptimal performance in specialized domains. For instance,

[11] RL and adaptive learning are linked because both are special cases of a more general set of stochastic approximation methods (Robbins and Monro, 1951).

[12] Agents form price expectations from data generated by the economy in which they live. Their expectations are therefore statistically consistent with actual equilibrium outcomes but may be incorrect for events that are infrequently observed (e.g. events off the equilibrium path).

[13] Guarda works with what he calls a "nonparametric restricted perceptions equilibrium" (NRPE). The difference is that Guarda defines his NRPE recursively and uses dynamic programming whereas our RPE is sequential.

[14] Recent work has also explored global solution methods in sequence space using high-dimensional approximation techniques, e.g., Azinovic-Yang and Žemlička (2025).

[15] RL ideas have also been applied in economics. Besides Han et al. (2021), see e.g. Barberis and Jin (2023), Chen et al. (2023), de la Barrera and de Silva (2024), Calvano et al. (2020), Dou et al. (2025), and the references in Fernández-Villaverde et al. (2024a) for applications in finance and macroeconomics. RL ideas have also been applied in game theory (e.g. Erev and Roth, 1995, 1998; Fudenberg and Levine, 2016) but using a different formulation that does not work with value functions and instead directly reinforces the "propensities" of choosing strategies.

when applied to the computationally intensive post-training phase of LLMs, classical RL algorithms introduce substantial overhead. To address this, Rafailov et al. (2024) reformulated RL from human feedback (RLHF) as a deep learning problem and proposed the Direct Preference Optimization (DPO) algorithm, which significantly reduces computational cost and enhances training stability. Similarly, Hu et al. (2020) and Freeman et al. (2021) introduced differentiable physics engines – DiffTaichi and Brax (the latter also implemented using JAX) – that enable gradient computation directly from built-in physical equations.[16] This allows policy updates via exact gradient methods, eliminating the need for sampling-based approximations and thereby improving both accuracy and efficiency. In Brax, this method is termed "analytical policy gradient," which shares conceptual foundations with our SRL framework.

Heterogeneous-agent models in macroeconomics can be viewed as a special case of the more general Mean Field Games (MFGs) studied in the mathematics literature (e.g. Lasry and Lions, 2007; Cardaliaguet et al., 2019). In the RL literature, the work closest to ours is therefore the line of research that applies RL methods to solve MFGs (e.g. Yang et al., 2018; Laurière et al., 2022, 2024; Xu et al., 2023; Wu et al., 2025). With the exception of Wu et al. (2025), none of these papers study the case with aggregate risk (or "common noise" in MFG terminology) considered here. More importantly, none of the papers applying RL to MFGs exploit the structural knowledge of agents' individual dynamics. Specifically, they do not exploit the transition matrix (or infinitesimal generator) for individual states $\mathbf{A}_{\boldsymbol{\pi}}$ and its known dependence on the policy $\boldsymbol{\pi}$, even though this matrix (operator) is typically available and even used for updating the distribution.[17]

In order to sidestep the Master equation and keep the state space manageable, we assume a form of *partial observability*: agents do not observe the high-dimensional distribution which is the system's underlying Markov state and instead observe low-dimensional prices. Wibault et al. (2026) consider general partially observable MFGs and extend our SPG method to a "recurrent SPG" method that keeps track of full price histories using recurrent neural networks as in recurrent RL (Hausknecht and Stone, 2017; Ni et al., 2022).

Roadmap. Section 2 starts by describing the setup our SRL method applies to starting with a simple example. Section 3 describes the SRL method and Section 4 reports our computational experiments. Section 5 concludes.

2 Setup

To explain the logic of our approach in the simplest possible fashion, we present it in a context that is very familiar to most economists: a general equilibrium model with incomplete markets and uninsured idiosyncratic labor income risk. We first do this in an economy in which indi-

[16] More generally, our approach of exploiting the problem's economic structure is akin to the use of physics-informed neural networks (Raissi et al., 2019). In this sense, SRL is "economics-informed."

[17] Gabriele et al. (2025) uses various multi-agent RL (MARL) algorithms to solve the Krusell and Smith (1998) model. These algorithms also do not exploit this structural knowledge. Another difference is that Gabriele et al. (2025) consider a relatively small finite number (e.g. 529) of agents rather than the MFG continuum limit.

viduals save in unproductive bonds that are in zero net supply as in Huggett (1993) but with aggregate risk. This economy is ideal for illustrating our method because it features a non-trivial market clearing condition. We later consider different ways of closing the model, for example a Krusell and Smith (1998) economy in which individuals save in productive capital.

2.1 A Huggett Model with Aggregate Risk

Individuals. There is a continuum of individuals that are indexed by i. Each individual has an income which evolves stochastically over time. Specifically, income $y_{i,t}z_t$ is an endowment of the economy's final good and consists of an idiosyncratic component $y_{i,t}$ (idiosyncratic risk) and an aggregate component z_t (aggregate risk). Individuals are heterogeneous in the idiosyncratic income component $y_{i,t}$ and their wealth $b_{i,t}$. The states of the economy are (i) the joint distribution of income and wealth which we denote by $G_t(b, y)$ and (ii) the aggregate shock z_t.

Individuals have standard preferences over utility from future consumption c_t:

$$\mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t u(c_t). \tag{1}$$

The wealth of an individual takes the form of bonds and evolves according to

$$c_{i,t} + q_t b_{i,t+1} = b_{i,t} + y_{i,t}z_t, \tag{2}$$

where q_t is the bond price. Equivalently, individuals can borrow and save at a real interest rate $1 + r_{t+1} = 1/q_t$ between time periods t and $t+1$. Individuals also face a borrowing limit

$$b_{i,t} \geq \underline{b}, \tag{3}$$

with $-\infty < \underline{b} \leq 0$ and which we assume to be tighter than the "natural borrowing constraint" (Aiyagari, 1994). The two income components follow Markov processes:

$$y_{i,t+1} \sim \mathcal{T}_y(\cdot|y_{i,t}) \quad \text{and} \quad z_{t+1} \sim \mathcal{T}_z(\cdot|z_t), \tag{4}$$

where $\mathcal{T}_y$ and $\mathcal{T}_z$ summarize the respective transition probabilities. We typically assume that the idiosyncratic component $y_{i,t}$ lives on a finite grid $\{y_1, ..., y_{J_y}\}$ whereas the aggregate component z_t is continuous, for example, the logarithm of z_t may follow an AR(1) process.

Individuals maximize (1) subject to (2), (3) and (4), taking as given the evolution of the equilibrium bond price q_t for $t \geq 0$. The individuals' optimization problem gives rise to consumption and saving policy functions which denote by $c_t(b, y, z)$ and $b'_t(b, y, z)$ where the prime superscript indexes the next period, i.e. $b_{i,t+1} = b'_t(b_{i,t}, y_{i,t}, z_t)$. For future reference, we denote the collection of both policy functions by

$$\pi_t(b, y, z) = \{c_t(b, y, z), b'_t(b, y, z)\}. \tag{5}$$

Equilibrium. The economy can be closed in a variety of ways. We here follow Huggett (1993) and assume that the bond price q_t (which is the only price in the economy) is determined by the requirement that, in equilibrium, bonds must be in zero net supply:

$$\int b'_t(b, y, z_t)\, dG_t(b, y) = 0, \quad \text{all } t \geq 0, \tag{6}$$

where $b'_t(b, y, z)$ is saving of an individual with states (b, y, z) as just discussed. We later consider alternative ways of closing the economy. For instance Section 4.2 assumes that wealth takes the form of productive capital hired by a representative firm so that the interest rate equals the aggregate marginal product of capital as in Krusell and Smith (1998).

A competitive equilibrium is defined in the usual way: quantities and prices $\{q_t\}_{t=0}^{\infty}$ such that

1. Individuals maximize (1) subject to (2), (3) and (4), taking as given $\{q_t\}_{t=0}^{\infty}$.
2. Markets clear: (6) holds for all $t \geq 0$.

Importantly, in this competitive equilibrium, individuals do *not* "care about" the cross-sectional distribution G_t, i.e. it does not enter their objective functions. Instead they only care about *prices*, here the bond price q_t.

Compact Notation. To ease notation going forward and also with an eye toward other, more complex heterogeneous agent models, we introduce some additional notation. First, we denote the vector of individual states by

$$s = (b, y).$$

and summarize the budget constraint (2) and $y_{i,t}$-process in (4) in terms of transition probabilities for this vector s:

$$s_{i,t+1} \sim \mathcal{T}_s(\cdot|s_{i,t}, c_{i,t}, z_t, p_t). \tag{7}$$

Second, we denote the vector of prices by p_t. Of course, in the Huggett model above, there is only one price $p_t = q_t$. But this notation will be useful in other applications, e.g. in Krusell and Smith (1998) there is a wage w_t and an interest rate r_t so that $p_t = (r_t, w_t)$.

Finally, in equilibrium, prices depend on the economy's state variables, the distribution $G_t(s)$ and the aggregate shock z_t. We therefore denote by

$$p_t = P^*(G_t, z_t) \tag{8}$$

the "equilibrium price functional" that summarizes this dependence. In the Huggett model above, this equilibrium price functional is implicitly determined by the market clearing condition (6).

Generalizations. The tools developed in this paper apply to a large class of heterogeneous agent models. We consider a few of these in our computational experiments in Section 4. The

most general setup is as follows: agents solve

$$v_{i,0} = \max_{\{a_{i,t}\}} \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t R(s_{i,t}, a_{i,t}, z_t, p_t) \quad \text{subject to} \quad s_{i,t+1} \sim \mathcal{T}_s(\cdot|s_{i,t}, a_{i,t}, z_t, p_t),$$

where $R(s, a, z, p)$ is a reward function that depends on individual states $s \in \mathcal{S} \subseteq \mathbb{R}^{n_s}$, actions $a \in \mathcal{A} \subseteq \mathbb{R}^{n_a}$, aggregate states $z \in \mathcal{Z} \subseteq \mathbb{R}^{n_z}$, and prices $p \in \mathcal{P} \subseteq \mathbb{R}^{n_p}$, and where $\mathcal{T}_s$ summarizes the transitions of individual states s. Prices p_t are determined in equilibrium from a set of market clearing conditions which results in a mapping $p_t = P^*(G_t, z_t)$ just like in (8). While our computational method applies to such general setups, going forward, we will explain it in terms of the simple Huggett model with two-dimensional individual state s, one-dimensional aggregate state z, and one-dimensional price p for concreteness.

2.2 Discretized Representation

To illustrate and implement our method it is convenient to discretize the individual state space. While idiosyncratic income realizations y are already on a finite grid, wealth b is continuous. We therefore place the individual state $s = (b, y)$ on a finite grid $s \in \{s_1, \ldots, s_J\}$ with $J = J_y \times J_b$. Objects such as the value function or cross-sectional distribution become J-dimensional vectors:

$$\boldsymbol{v}_t = \begin{bmatrix} v_t(s_1) \\ \vdots \\ v_t(s_J) \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{g}_t = \begin{bmatrix} g_t(s_1) \\ \vdots \\ g_t(s_J) \end{bmatrix},$$

where we use boldfaced notation to denote vectors. Note that the vector $\boldsymbol{g}_t$ is simply the "histogram" which collects the fraction of agents at each point of the state space.

Similarly, the consumption-saving policy $\pi_t(s, z)$ becomes a vector $\boldsymbol{\pi}_t(z)$ defined on the J-dimensional grid. Given a policy $\pi_t(\cdot)$, the induced one-step transitions for s can be collected in a $J \times J$ transition matrix,

$$\mathbf{A}_{\boldsymbol{\pi}_t(z_t)} \qquad \text{with entries} \qquad \Pr(s_{i,t+1} = s_{j'} \,|\, s_{i,t} = s_j) = \mathcal{T}_s(s_{j'} \,|\, s_j, \pi_t(s_j, z_t), p_t).$$

Entry jj' of this matrix represents the probability that an individual in state j transitions to state j' next period, with rows summing to 1. These probabilities are encoded in $\mathcal{T}_s$ and depend on the policy $\pi_t(s, z)$ as well as the period-t realization of prices p_t.

The cross-sectional distribution then evolves according to the discrete-time Chapman-Kolmogorov equation

$$\boldsymbol{g}_{t+1} = \mathbf{A}^{\mathrm{T}}_{\boldsymbol{\pi}_t(z_t)} \boldsymbol{g}_t, \tag{9}$$

where $\mathbf{A}^{\mathrm{T}}_{\boldsymbol{\pi}_t(z_t)}$ denotes the transpose of the transition matrix. Intuitively, think of probability mass at each grid point s_j *flowing out* across the row j of $\mathbf{A}_{\boldsymbol{\pi}_t(z_t)}$. The transpose in (9) simply accumulates these inflows at destinations $s_{j'}$. The use of the "histogram" $\boldsymbol{g}_t$ on the discretized state space to track the cross-sectional distribution is as in Young (2010) and Achdou et al. (2021).

2.3 Key Difficulty: Equilibrium Prices Are Not Markov

Two immediate implications follow. First, the high-dimensional aggregate state $(\boldsymbol{g}_t, z_t)$ is Markov by construction. The transition probabilities for $(\boldsymbol{g}_{t+1}, z_{t+1})$ depend on the current realizations of $(\boldsymbol{g}_t, z_t)$ only. Second, equilibrium prices are not Markov on the other hand (Moll, 2025). They are determined as a function of the aggregate state by the price functional $p_t = P^*(\boldsymbol{g}_t, z_t)$. The transition probabilities for p_{t+1} therefore depend on the high-dimensional state $(\boldsymbol{g}_t, z_t)$. Concretely, taking as given a policy function $\pi_t(s, z)$, the law of motion of prices in equilibrium is given by the equations

$$\begin{aligned} p_t &= P^*(\boldsymbol{g}_t, z_t) \\ \boldsymbol{g}_{t+1} &= \mathbf{A}^{\mathrm{T}}_{\boldsymbol{\pi}_t(z_t)} \boldsymbol{g}_t \\ z_{t+1} &\sim \mathcal{T}_z(\cdot \mid z_t). \end{aligned} \tag{10}$$

The transition probabilities for p_{t+1} cannot be determined as a function of the current price realization p_t alone. We have $p_{t+1} = P^*(\boldsymbol{g}_{t+1}, z_{t+1})$ so the conditional distribution of p_{t+1} depends on $(\boldsymbol{g}_t, z_t)$ and not just on p_t. Thus, $(\boldsymbol{g}_t, z_t)$ is a Markov state but p_t is not.

The Master equation. Dynamic programming requires Markov state variables. Since p_t alone is not Markov, the Bellman equation cannot be written only in terms of p_t. The key idea of dynamic programming — splitting the sequence problem into a current flow payoff and a continuation value term — fails when transitions are not Markov, since the current state does not provide sufficient information to determine the continuation value. The recursive formulation must instead include the true aggregate state $(\boldsymbol{g}, z)$:

$$\begin{aligned} V(s, \boldsymbol{g}, z) = \max_{c} \quad & u(c) + \beta \mathbb{E}[V(s', \boldsymbol{g}', z') \mid s, \boldsymbol{g}, z] \\ \text{s.t.} \quad & s' \sim \mathcal{T}_s(\cdot \mid s, c, z, p), \\ & p = P^*(\boldsymbol{g}, z), \\ & \boldsymbol{g}' = \mathbf{A}^{\mathrm{T}}_{\boldsymbol{\pi}(\mathbf{g}, z)} \boldsymbol{g}, \end{aligned} \tag{11}$$

where $\boldsymbol{\pi}(\mathbf{g}, z)$ in $\mathbf{A}_{\boldsymbol{\pi}(\mathbf{g}, z)}$ is the optimal policy associated with $V(s, \boldsymbol{g}, z)$. Equation (11) is often referred to as the "Master equation" — a Bellman equation on a state space that includes the cross-sectional distribution $\boldsymbol{g}$ (Cardaliaguet et al., 2019; Ahn et al., 2018; Schaab, 2020; Bilal, 2023; Gu et al., 2024).

Why this matters. Households care directly about prices because p enters their budget sets; they do not care directly about $\boldsymbol{g}$. Yet under rational expectations the only way to forecast p_{t+1} is to forecast $(\boldsymbol{g}_{t+1}, z_{t+1})$, so the high-dimensional $\boldsymbol{g}$ becomes a state variable and the associated Bellman equation inherits an extreme curse of dimensionality. But what if there was a way to approximate value and policy functions directly in terms of the current prices p_t, for which there are no Markov transition probabilities? This is the approach we develop in the next section.

3 Sidestepping the Master Equation via Structural Reinforcement Learning

This section describes our SRL method. Rather than solving the Master equation (11) with the full cross-sectional distribution as a state variable, we work with a low-dimensional state consisting of prices p_t (and the aggregate shock z_t). We adapt RL ideas to let agents learn optimal behavior from simulated equilibrium data, taking (s_t, z_t, p_t) as their state. In contrast to standard RL, our SRL method exploits agents' structural knowledge of their own individual dynamics.

3.1 The Key Idea of Reinforcement Learning: Monte Carlo instead of Bellman Equations

Before proceeding, we briefly summarize the basic ideas of RL.[18] RL means learning value or policy functions of incompletely-known Markov decision processes via some form of Monte Carlo simulation. The key problem addressed by RL is: what to do in dynamic optimization problems in which the agent does not know the exact environment she is operating in, specifically the stochastic process for the underlying state variables? The key insight of RL is that, in such environments, one can still approximate optimal value and policy functions *as long as one can simulate*.

A simple analogy is how to compute the expected value $\mathbb{E}[x]$ of a random variable x. The standard way is to compute $\mathbb{E}[x] = \int x f(x) dx$ for a known probability distribution $f(x)$. But what if f is unknown? In such cases, one can often still sample from f and approximate the expected value $\mathbb{E}[x]$ with the sample mean $\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_n$.

Building on this intuition, consider the question of how to calculate the following value function:

$$v_0 = \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t u(p_t)\right],$$

where u is a utility function and p_t is some exogenous stochastic process. The standard approach is to use dynamic programming: assume that p_t is Markov with known transition probabilities $f(p'|p)$; then write and solve the Bellman equation

$$v(p) = u(p) + \beta \int v(p') f(p'|p) dp'.$$

An alternative approach is to use Monte Carlo simulation: simply sample N trajectories $\{p_t^n\}_{t=0}^{\mathrm{T}}$ for $n = 1, ..., N$ and approximate the expected value v_0 as

$$v_0 \approx \widehat{v}_0 = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=0}^{T} \beta^t u(p_t^n).$$

This basic idea – to compute expected values via simulation – lies at the heart of all RL algo-

[18] See Sutton and Barto (2018) and Zhao (2025) for brilliant textbook treatments. Also see Murphy (2025) and Silver (2015).

rithms. Crucially this simulation-based approach does not require knowledge of the transition probabilities f. It also works directly with the sequential formulation of the problem. In particular, it is unnecessary to force the problem into the standard Markovian structure required for applying dynamic programming, e.g. by estimating a perceived law of motion for prices p_t. As we explain next, our SRL approach to heterogeneous agent macroeconomics uses this same approach to compute expectations about equilibrium prices.

3.2 Revisiting the Agents' Decision Problem

Recall from Section 2 that agents choose $\{c_{i,t}\}$ to solve

$$v_{i,0} = \max_{\{c_{i,t}\}} \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t u(c_{i,t})\right] \quad \text{s.t.} \quad s_{i,t+1} \sim \mathcal{T}_s(\cdot \mid s_{i,t}, c_{i,t}, z_t, p_t), \quad p_t = P^*(G_t, z_t). \tag{12}$$

We begin by specifying what individuals observe and on what they condition their decisions.

Assumption 1 (Information)**.** *At date t, an individual observes the entire history of aggregate prices $\{p_t, p_{t-1}, \ldots\}$, but not the cross-sectional distribution $\boldsymbol{g}_t$.*

Assumption 1 rules out direct conditioning on the distribution but does not, by itself, imply any departure from full information rational expectations. Under standard regularity conditions, a stationary price process $\{p_t\}$ admits a Wold representation and can be written as an infinite-order vector moving average process

$$p_t = \sum_{j=0}^{\infty} \kappa_j \varepsilon_{t-j},$$

for some coefficients $\{\kappa_j\}$ and white-noise innovations ε_t. Under additional restrictions (invertibility), the stationary price process also has an equivalent VAR(∞) representation,

$$p_{t+1} \sim \mathcal{T}_p(\,\cdot \mid p_t, p_{t-1}, p_{t-2}, \ldots),$$

so that the infinite history of prices is sufficient for forecasting under rational expectations. In this sense, the price history is rich enough to recover all information that is relevant to the agents, even though they never observe the cross-sectional distribution directly. Note that the Wold representation theorem effectively converts the recursive formulation for the price process (10) into a sequential stochastic process. It is also worth pointing out that, in the language of the RL literature, Assumption 1 is a "partial observability" assumption.

Of course, working with the infinite price history is infeasible in practice. Instead, starting from the rational expectations benchmark in Assumption 1, we restrict attention to low-dimensional summaries of this history. We begin with the most restrictive case, in which agents condition only on current prices.

Assumption 2 (Restricted state space)**.** *Agents' decision rules (policy functions) take the form*

$$\pi(s_t, z_t, p_t),$$

so that policies do not depend on lagged prices.[19]

Assumption 2 imposes a particular restriction on perceptions: agents treat the current price vector as a sufficient statistic for decision making, even though in the true equilibrium the price process is not Markov in p_t alone. Within this class of policies we then solve problem (12) by optimizing over many simulated equilibrium paths.

In Section 4.1 we relax Assumption 2 and allow policies to depend on a short history of past prices. In a companion paper (Wibault et al., 2026), we alternatively keep track of full price histories using recurrent neural networks and show that, in the Krusell-Smith model, this does not materially affect equilibrium policies and dynamics.

Given any (possibly suboptimal) policy $\pi(s, z, p)$, the discretization of individual states on a finite grid $s \in \{s_1, \dots, s_J\}$ allows us to write the policy in vector form

$$\boldsymbol{\pi}(z, p) = \begin{bmatrix} \pi(s_1, z, p) \\ \vdots \\ \pi(s_J, z, p) \end{bmatrix}.$$

As in Section 2.2, we represent the cross-sectional distribution as a vector $\boldsymbol{g}_t$, and the individual transitions induced by the policy are encoded in a sparse transition matrix $\mathbf{A}_{\boldsymbol{\pi}(z,p)}$.

3.3 Sequential Restricted Perceptions Equilibrium

Under Assumptions 1 and 2, an individual's relevant state is simply (s_t, z_t, p_t). Given this restricted state space, we define equilibrium as a variant of a restricted perceptions equilibrium (RPE) in the sense of Sargent (1991) and Branch (2006). Because we work directly with the sequential formulation of the agents' problem, we term it a "sequential RPE".

Definition 1 (Sequential restricted perceptions equilibrium)**.** *A sequential restricted perceptions equilibrium consists of a pair of mappings $(\pi^*(s, z, p), P^*(\boldsymbol{g}, z))$ such that:*

1. ***Optimality.*** *Given a stochastic process $\{z_t\}_{t\geq 0}$ as well as a price process $\{p_t\}_{t\geq 0}$ generated by $p_t = P^*(\boldsymbol{g}_t, z_t)$, the policy $\pi^* = \{c^*, b'^*\} \in \Pi$ solves*

$$\max_{\pi\in\Pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t u(c(s_t, z_t, p_t))\right] \quad s.t. \quad s_{t+1} \sim \mathcal{T}_s(\cdot \mid s_t, \pi(s_t, z_t, p_t), z_t, p_t),$$

where $c(s, z, p)$ is the consumption component of $\pi(s, z, p)$. Here Π denotes the set of measurable policies $\pi : \mathcal{S} \times \mathcal{Z} \times \mathcal{P} \to \mathcal{C} \times \mathcal{B}$ that satisfy the budget and borrowing constraints.

[19] Whether z_t is payoff-relevant depends on the particular application. If, conditional on (s_t, p_t), z_t affects neither current payoffs nor individual transitions, then it can be dropped from the state vector for the individual's decision problem. For example, z_t is directly payoff-relevant in the Huggett model of Section 2.1 because it scales current income $y_{i,t}z_t$. By contrast, in our Krusell-Smith and HANK applications in Sections 4.2 and 4.3, z_t matters for individuals only through equilibrium prices, so policies could equivalently be written as $\pi(s_t, p_t)$. For notational uniformity we keep z_t in $\pi(s_t, z_t, p_t)$, but in these cases it is redundant from the household's point of view.

2. ***Market clearing.*** *For every t, the market clearing conditions hold: in the Huggett model*

$$\int b'(s, z_t, p_t)\, dG_t(s) = 0, \tag{13}$$

where $b'(s, z, p)$ is the saving component of $\pi(s, z, p)$. The solution is a mapping $p_t = P^*(\boldsymbol{g}_t, z_t)$.

3. ***Consistency.*** *When all agents follow π^*, the cross-sectional distribution evolves according to*

$$\boldsymbol{g}_{t+1} = \mathbf{A}^{\mathrm{T}}_{\boldsymbol{\pi}^*(z_t, p_t)} \boldsymbol{g}_t,$$

where $\mathbf{A}_{\boldsymbol{\pi}(z,p)}$ is the transition matrix induced by $\mathcal{T}_s$ and π and prices are given by $p_t = P^*(\boldsymbol{g}_t, z_t)$.

This equilibrium notion has three features that are important for our purposes. First, it is a *sequential* equilibrium: We work directly with time paths of states and prices rather than with a recursive formulation in terms of a Markov state. Second, it is *self-confirming* in the sense that, given their information and policy, agents' beliefs about price dynamics are consistent with the equilibrium price process along the realized paths. Third, it features *restricted perceptions* because agents condition only on prices, not on the full distribution. This both reflects a realistic informational environment and greatly reduces the dimensionality of the state space. As mentioned in the introduction, in the language of Guarda (2025), agents' beliefs are both "narrow" and "short".

These features make it natural to use RL. In practice, we parameterize the low-dimensional policy function $\pi(s, z, p)$ on the discretized state space and approximate the maximization problem (12) by evaluating it along many simulated equilibrium paths. In the next subsection, we describe how to simulate these trajectories efficiently under a given candidate policy.

3.4 Simulating the Economy for Given Policy Functions

Starting from an initial pair $(\boldsymbol{g}_0, z_0)$, the simulated economy evolves according to

$$\begin{aligned} z_{t+1} &\sim \mathcal{T}_z(\cdot \mid z_t) \\ \boldsymbol{g}_{t+1} &= \mathbf{A}^{\mathrm{T}}_{\boldsymbol{\pi}(z_t, p_t)} \boldsymbol{g}_t \\ p_t &= P^*(\boldsymbol{g}_t, z_t). \end{aligned}$$

Updating the aggregate shock z_{t+1} is straightforward. For the distribution $\boldsymbol{g}_{t+1}$, we adopt Young (2010)'s non-stochastic simulation method and extend it to a full matrix formulation: the policy function induces a sparse transition matrix over the grid, and the distribution evolves deterministically via matrix-vector multiplication. The main computational difficulty lies in the last step. In some models, such as Krusell and Smith (1998), the price functional $P^*(\boldsymbol{g}_t, z_t)$ is available in closed form. In others, such as the Huggett model in Section 2.1, prices are defined only implicitly by market-clearing conditions. Computing such implicit prices represents a significant challenge. Most existing numerical methods solve a non-linear root-finding problem in every period of the simulation, typically making this step the slowest part of the

algorithm; see for example Krusell and Smith (1997), Schaab (2020), as well as the historical note in footnote 6 in the introduction. Our method delivers an efficient way to compute these implicit prices along simulated paths. We show next how this works in the Huggett economy of Section 2.1.

Efficient handling of non-trivial market clearing conditions. In the Huggett model, the equilibrium bond price $p_t = q_t$ is pinned down by the requirement that bonds must be in zero net supply, see (6). The key insight that allows us to find equilibrium prices efficiently is that, due to Assumption 1, individual policy functions $\pi(s, z, p)$ depend on current prices p rather than the cross-sectional distribution $G_t(s)$. In addition to the much lower dimensionality of p as compared to $G_t(s)$, this has another key payoff: *policy functions double as individual supply schedules* which can easily be aggregated to obtain *aggregate* supply curves at each point in time which also depend on the current price p. Given an aggregate supply curve as a function of p, it is then straightforward to solve for the equilibrium price $p_t = P^*(G_t, z_t)$.

To see this in more detail, consider the market clearing condition in our restricted perceptions equilibrium (13). Equivalently,

$$D_t(p_t, z_t) = 0 \quad \text{where} \quad D_t(p, z) = \int b'(s, z, p) dG_t(s)$$

is aggregate bond demand implied by the individual policy function $b'(s, z, p)$. The key observation is that the individual policy function depends on the bond price p. This means that varying p traces out an entire individual demand *schedule* $p \mapsto b'(s, z, p)$. Aggregating yields the analogous aggregate demand schedule $p \mapsto D_t(p, z)$. The equilibrium price can then be computed time-period by time-period along each simulated path. Our approach of including current prices in the state space to clear markets at each point along a simulation is similar to Krusell and Smith (2006, Section 3.5).

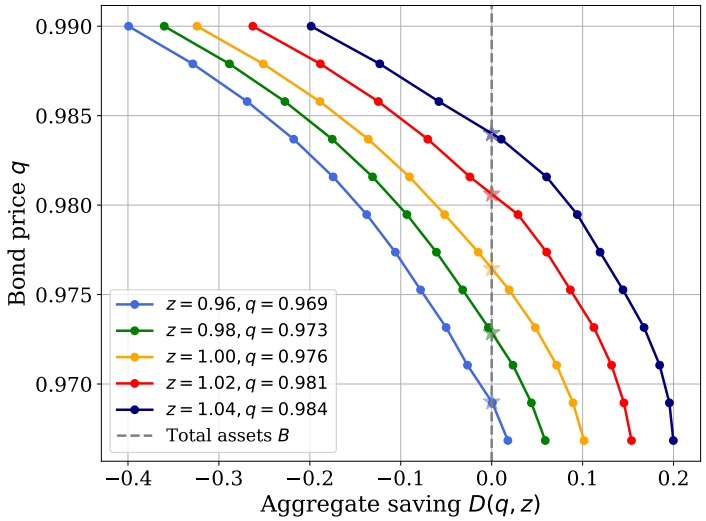


Figure 1: Aggregate bond demand $D(q, z)$ in the Huggett economy

Finding the equilibrium bond price can be numerically implemented in a variety of ways.

In our experiments, we found that the simplest way of doing this is to use our discretized representation. Once we know the vectors $\boldsymbol{g}_t$ and $\boldsymbol{b}'(z,p)$ for the discretized cross-sectional distribution and saving policy function, we can compute the entire bond demand schedule at time t for all grid values of (z,p) as

$$D_t(p,z) = \boldsymbol{b}'(z,p)^{\mathrm{T}}\boldsymbol{g}_t. \tag{14}$$

Figure 1 plots the aggregate bond demand $p \mapsto D(p,z)$ in the Huggett economy at some date t, with each line corresponding to a grid value of z. Given the realized aggregate state z_t, we select the corresponding demand curve $D_t(\cdot, z_t)$ and determine the bond price p_t so that $D_t(p_t, z_t) = 0$. In our numerical experiments the function $D_t(p,z)$ is weakly decreasing in p under the optimal policy $\boldsymbol{b}'(z,p)$ and in a neighborhood of the market-clearing price, so the solution is unique.[20] However, market clearing is part of the agents' environment and needs to hold for all candidate policy functions, not just at the optimal policy. As we explain in Section 3.7 below, we solve for optimal policies $\pi(s,z,p) = \{c(s,z,p), b'(s,p,z)\}$ iteratively starting from some initial guess. This means that it is important to have a reasonable initial guess for $b'(s,z,p)$ that delivers a downward-sloping aggregate bond demand under this initial guess.

Both the dot product in (14) and the interpolation step are simple vector operations and are very fast when implemented with JAX on GPUs. It is therefore feasible to compute the market clearing price p_t in each period and *along each simulated trajectory*, including for suboptimal policies. This contrasts with existing methods which instead call a separate non-linear solver inside an outer fixed-point iteration (Krusell and Smith, 1997; Schaab, 2020).

3.5 Exploiting Agents' Structural Knowledge of their Individual Dynamics

Standard RL applications, such as board games or Atari environments, typically treat both rewards and state transitions as unknown and learn them entirely from simulated data. Economic models are different. Agents know their preferences and they know how their choices affect their own individual state next period (Han et al., 2021). We make explicit use of this structure and refer to our approach as *structural reinforcement learning* (SRL).

The value for an individual i starting from state (s,z,p) and following a given policy $\pi(s,z,p) = \{c(s,z,p), b'(s,z,p)\}$ is

$$v_\pi(s,z,p) = \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t u(c(s_{i,t}, z_t, p_t)) \,\middle|\, s_{i,0} = s,\ z_0 = z,\ p_0 = p\right]. \tag{15}$$

Here $v_\pi(s,z,p)$ is the value associated with a *fixed* policy π and a given stochastic process for

[20] On the bounded price grid $\{p_k\}_{k=1}^K$ we compute $\{D_t(p_k,z_t)\}_{k=1}^K$ and find the two adjacent points that bracket the root. If

$$D_t(p_k,z_t) \le 0 \le D_t(p_{k+1},z_t),$$

we set the market-clearing price to the linear interpolant

$$p_t = p_k - \frac{D_t(p_k,z_t)}{D_t(p_{k+1},z_t) - D_t(p_k,z_t)}(p_{k+1} - p_k).$$

the aggregate variables (p_t, z_t). The evolution of the individual state $s_{i,t}$ is governed by the known transition kernel $\mathcal{T}_s$ or, when individual states are discretized, by the known transition matrices $\mathbf{A}_{\boldsymbol{\pi}(z_t,p_t)}$. Agents know this mapping from current states and actions into next-period states because it is implied by their budget constraint and the exogenous process for idiosyncratic income, which we assume agents know, i.e. agents have rational expectations about their idiosyncratic income process. By contrast, agents *do not* know the law of motion for the aggregate variables (p_t, z_t) which are determined in general equilibrium from everyone's decisions and market clearing. Under rational expectations, one would require agents to know the stochastic process for (p_t, z_t) exactly and to treat $(\boldsymbol{g}_t, z_t)$ as a Markov state so that prices become a function $p_t = P^*(\boldsymbol{g}_t, z_t)$. In our setup, agents instead take the process for (p_t, z_t) as an object to be learned from simulated data.

SRL takes advantage of this separation between individual and aggregate dynamics by *partitioning the state space* (s, z, p) into individual states s and aggregate states (z, p) and treating these differently. We use the structural model to simulate individual transitions and payoffs exactly, i.e., conditional on a policy $\boldsymbol{\pi}(z, p)$, we treat the transition matrix $\mathbf{A}_{\boldsymbol{\pi}(z,p)}$ and hence the mapping $(s_t, z_t, p_t) \mapsto s_{t+1}$ as known by the agent and use this matrix to compute the value function the agent maximizes – see (16) below. We then use RL only to deal with the low-dimensional but non-Markov aggregate state (z_t, p_t).

In particular, our algorithm updates the policy π so as to increase $v_\pi(s, z, p)$ based on simulated histories of (z_t, p_t) rather than using a Bellman equation on the high-dimensional state space $(\boldsymbol{g}_t, z_t)$. This keeps the learning problem low-dimensional while preserving the full heterogeneous-agent structure of the economy. While the simulations feature the full cross-sectional distribution which evolves stochastically and non-linearly over time, we never approximate that distribution nor any mapping from it. Instead only low-dimensional prices enters the agent's state.

3.6 Problem To Be Solved

The value $v_\pi(s, z, p)$ associated with a given policy $\pi(s, z, p)$ was defined in (15). The agent's problem is to choose a policy $\pi = (c, b')$ to maximize her value v_π given an initial state (s, z, p). It is convenient to use discretized notation and rewrite the value function $v_\pi(s, z, p)$ in vector form as

$$\boldsymbol{v}_\pi(z, p) = \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t \, \mathbf{A}_{\boldsymbol{\pi},0\to t} \, \boldsymbol{u}(\boldsymbol{c}(z_t, p_t)) \,\middle|\, z_0 = z,\ p_0 = p\right], \tag{16}$$

where $\boldsymbol{\pi}(z, p) = \{\boldsymbol{c}(z, p), \boldsymbol{b}'(z, p)\}$ is the discretized policy vector and

$$\mathbf{A}_{\boldsymbol{\pi},0\to t} = \mathbf{A}_{\boldsymbol{\pi}(z_0,p_0)} \times \cdots \times \mathbf{A}_{\boldsymbol{\pi}(z_{t-1},p_{t-1})} \tag{17}$$

denotes the transition matrix of individual states between time 0 and time t under a particular trajectory $\{z_\tau, p_\tau\}_{\tau=0}^{t-1}$. Note again our partitioning of the state space into s and (z, p): the transition matrices $\mathbf{A}_{\boldsymbol{\pi}(z_t,p_t)}$ keep track of all s-transitions while the expectation in (16) is taken

only over (z, p)-trajectories.[21] Importantly, the transition matrix $\mathbf{A}_{\boldsymbol{\pi}(z_t,p_t)}$ encodes all relevant structural knowledge about the dynamics of agents' own individual states s. The presence of this transition matrix in the optimization objective (16) is why we refer to our approach as *structural* RL.

When agents choose policies $\boldsymbol{\pi}$ they *take as given* the evolution of equilibrium prices which evolve according to the true general-equilibrium dynamics,

$$p_t = P^*(\boldsymbol{g}_t, z_t), \qquad \boldsymbol{g}_{t+1} = \mathbf{A}^{\mathrm{T}}_{\boldsymbol{\pi}(z_t,p_t)}\, \boldsymbol{g}_t, \qquad z_{t+1} \sim \mathcal{T}_z(\cdot \mid z_t), \tag{18}$$

with $(\boldsymbol{g}_0, z_0)$ given.

The agents' objective is to find a policy vector $\boldsymbol{\pi}(z, p)$ that maximizes $\boldsymbol{v}_\pi(z, p)$ for all initial values (z, p) taking as given the evolution of equilibrium prices in (18). Note that maximizing agents take into account the dependence of the transition matrix $\mathbf{A}_{\boldsymbol{\pi},0\to t}$ in (16) and (17) on the policy $\boldsymbol{\pi}$; in contrast, because they take prices as given, they *do not* take into account how price dynamics depend on the policy $\boldsymbol{\pi}$ via the term $\mathbf{A}^{\mathrm{T}}_{\boldsymbol{\pi}(z_t,p_t)}$ in the Chapman-Kolmogorov equation for $\boldsymbol{g}_t$ in (18).[22]

An important feature of this problem is that, while the true state of the economy $(\boldsymbol{g}_t, z_t)$ is extremely high-dimensional (because $\boldsymbol{g}_t$ is a full cross-sectional distribution), the state that enters agents' policy and value functions (s_t, z_t, p_t) is low-dimensional. The agent does not work with a perceived law of motion for prices. Instead, she treats the process for (p_t, z_t) as given, observes realized sequences along simulated paths, and bases decisions on these realizations. As a result, there is no inner-outer fixed-point loop over perceived price laws of motion as in Krusell and Smith (1998). Our algorithm therefore operates directly on the low-dimensional state (s_t, z_t, p_t) from the agent's perspective, while the high-dimensional state $(\boldsymbol{g}_t, z_t)$ only appears in the background through the law of motion for prices.

3.7 Implementation: Structural Policy Gradient Algorithm

To evaluate a candidate policy π, we approximate the value vector using Monte Carlo simulation as explained in Section 3.1. We simulate N trajectories of the economy under this policy and form the sample analog of the value vector

$$\widehat{\boldsymbol{v}}_\pi = \frac{1}{N}\sum_{n=1}^{N}\left[\sum_{t=0}^{T}\beta^t \mathbf{A}^n_{\boldsymbol{\pi},0\to t} u(\boldsymbol{c}(z_t^n, p_t^n))\right], \tag{19}$$

where T is a large truncation horizon and the simulated paths are generated from

$$p_t^n = P^*(\boldsymbol{g}_t^n, z_t^n), \qquad \boldsymbol{g}_{t+1}^n = \mathbf{A}^{\mathrm{T}}_{\boldsymbol{\pi}(z_t^n,p_t^n)}\boldsymbol{g}_t^n, \qquad z_{t+1}^n \sim \mathcal{T}_z(\cdot \mid z_t^n), \tag{20}$$

[21] Tracking value vectors using the transition matrix $\mathbf{A}_{\boldsymbol{\pi}}$ in this way is analogous to the use of such matrices in finite-difference methods for continuous-time HJB equations and HA models (Achdou et al., 2021).

[22] As we explain below, in our SPG algorithm which maximizes the Monte Carlo counterpart to (16) with respect to $\boldsymbol{\pi}$, we apply a stop-gradient to simulated prices to ensure that agents take prices as given.

starting from initial conditions $\boldsymbol{g}_0^n \sim \psi_g$ and $z_0^n \sim \psi_z$. Thus, $\widehat{\boldsymbol{v}}_\pi$ in (19) is the sample analog of $\boldsymbol{v}_\pi(z,p)$ in (16) averaged over the initial distribution of (z,p) induced by the initial distribution of $(\boldsymbol{g},z)$, i.e. $\widehat{\boldsymbol{v}}_\pi \approx \mathbb{E}_{z_0\sim\psi_z,\boldsymbol{g}_0\sim\psi_g}[\boldsymbol{v}_\pi(z_0,p_0)]$ with $p_0 = P^*(\boldsymbol{g}_0,z_0)$.

In practice, we work with a scalar objective that averages over initial individual states. Let $\boldsymbol{d}_0$ denote the uniform distribution on the individual-state grid, so that $d_0(s_j) = 1/J$. We then maximize

$$\mathcal{L}(\boldsymbol{\theta}) = \boldsymbol{d}_0^{\mathrm{T}}\widehat{\boldsymbol{v}}_\pi. \tag{21}$$

Because our state space is low-dimensional, we can work with a grid-based (tabular) approach and we parameterize the policy as

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\pi}(z_1,p_1) \\ \vdots \\ \boldsymbol{\pi}(z_K,p_L) \end{bmatrix} = \begin{bmatrix} \pi(s_1,z_1,p_1) \\ \vdots \\ \pi(s_J,z_K,p_L) \end{bmatrix},$$

where J, K, and L are the numbers of grid points on the s, z, and p grids. That is, the parameter vector $\boldsymbol{\theta}$ is simply the $J\times K\times L$-dimensional vector of values of the policy on the (s,z,p) grid.[23]

Furthermore, in practice it is costly to compute the high-dimensional matrix multiplications $\mathbf{A}^n_{\boldsymbol{\pi},0\to t} = \mathbf{A}_{\boldsymbol{\pi}(z_0^n,p_0^n)} \times \cdots \times \mathbf{A}_{\boldsymbol{\pi}(z_{t-1}^n,p_{t-1}^n)}$ in (19). After substituting in (21), we therefore rewrite the optimization objective as

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=0}^{T}\beta^t(\boldsymbol{d}^n_{\boldsymbol{\pi},t})^{\mathrm{T}}u(\boldsymbol{c}(z_t^n,p_t^n)),$$

where $\boldsymbol{d}^n_{\boldsymbol{\pi},t} = (\mathbf{A}^n_{\boldsymbol{\pi},0\to t})^{\mathrm{T}}\boldsymbol{d}_0$ is the cross-sectional distribution of s at time t under policy $\boldsymbol{\pi}$ starting from the initial uniform distribution $\boldsymbol{d}_0$. We compute this distribution iteratively by solving the Chapman-Kolmogorov equation $\boldsymbol{d}^n_{\boldsymbol{\pi},t+1} = (\mathbf{A}^n_{\boldsymbol{\pi}(z_t,p_t)})^{\mathrm{T}}\boldsymbol{d}^n_{\boldsymbol{\pi},t}$ forward in time using the Young (2010) method. [24]

Figure 2 presents a computational graph that illustrates the algorithm. Since the mapping from parameters $\boldsymbol{\theta}$ to the objective $\mathcal{L}(\boldsymbol{\theta})$ is differentiable, we use stochastic gradient ascent (or variants) to update

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + \eta_k\nabla_\theta\mathcal{L}(\boldsymbol{\theta}^k),$$

where η_k is the learning rate at iteration k. A stop-gradient is applied to the price update $p_t^n = P^*(\boldsymbol{g}_t^n,z_t^n)$ in (20) so that, when computing gradients, prices are treated as exogenous. This is consistent with the standard notion of competitive equilibrium, in which agents take the process for (p_t,z_t) as given and do not internalize how their behavior affects price formation. Finally, we assess convergence of the algorithm by tracking the change in the policy vector $\boldsymbol{\theta}$ over the (s,z,p) grid. Convergence is achieved when the L^∞ (sup) norm of the policy update

[23] One can instead parameterize the policy as a neural network $\pi(s,z,p;\boldsymbol{\theta})$. For the low-dimensional policy functions in this paper, a grid-based parameterization is sufficient.

[24] Azinovic et al. (2022) use the Young (2010) method in a related fashion to construct an optimization objective involving a discretized cross-sectional distribution. However, they use neural networks to minimize Euler equation errors whereas we use a grid-based approach to maximize a value vector.

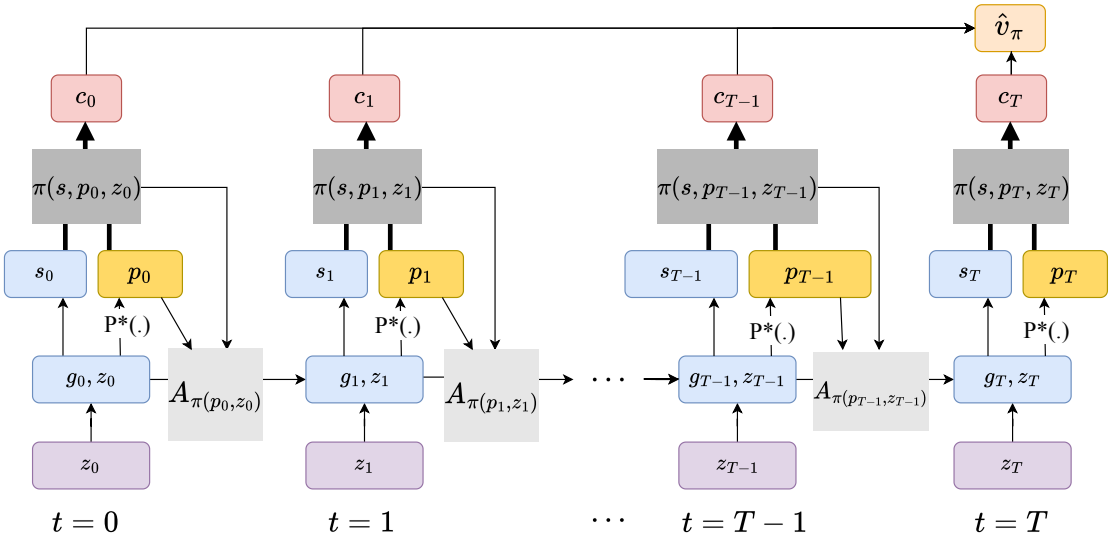


Figure 2: Computational graph

falls below a tolerance,

$$\|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\|_\infty < \epsilon_{\text{converge}} \qquad \text{where} \qquad \|\boldsymbol{x}\|_\infty \equiv \max_{j,k,l} |x_{jkl}|.$$

We report our choice of convergence tolerance $\epsilon_{\text{converge}}$ for each application in Appendix A. Algorithm 1 summarizes the implementation.

Algorithm 1: Structural Policy Gradient Algorithm

Input : Initial policy parameters $\boldsymbol{\theta}^0$; step size sequence $\{\eta_k\}$; number of simulated trajectories N; horizon T.

Output: Approximate optimal policy parameters θ^*.

1. Initialize parameters $\boldsymbol{\theta}^0$.
2. **For each iteration** $k = 0, 1, 2, \ldots$:
   (a) Simulate N trajectories

$$\{(z_t^n, p_t^n, \mathbf{g}_t^n)\}_{t=0}^T, \qquad n = 1, \ldots, N,$$

   using policy $\boldsymbol{\pi}(\cdot; \boldsymbol{\theta}^k)$ and market clearing conditions.

   (b) Compute the sample objective

$$\mathcal{L}(\boldsymbol{\theta}^k) = \boldsymbol{d}_0^{\mathrm{T}} \widehat{\boldsymbol{v}}_\pi \quad \text{where} \quad \widehat{\boldsymbol{v}}_\pi = \frac{1}{N} \sum_{n=1}^{N} \left[\sum_{t=0}^{T} \beta^t \, \mathbf{A}_{\boldsymbol{\pi}, 0 \to t} \, u(\boldsymbol{c}(z_t^n, p_t^n)) \right].$$

   (c) Update parameters by stochastic gradient ascent (or variants):

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + \eta_k \nabla_\theta \mathcal{L}(\boldsymbol{\theta}^k).$$

   (d) Stop when convergence criteria are met.

---

4 Computational Experiments

In this section we report computational experiments for three benchmark economies: the Huggett model from Section 2, the classic Krusell and Smith (1998) model, and a one-account heterogeneous agent New Keynesian (HANK) model with nominal rigidities.

We implement all three models in JAX and run them on a single NVIDIA A100 GPU on Google Colab. Table 1 summarizes performance. Since our algorithm is stochastic and uses Monte Carlo simulations, we run our algorithm 10 times for each specification and report averages across runs. The first column shows the average number of epochs until convergence, and the last column the corresponding average time for a single run.

Model	Average converge epoch	# Runs	Average Runtime (sec)
Krusell-Smith	462.3	10	36.77
Huggett with agg. shocks	573.0	10	45.91
HANK with agg. shocks	707.5	10	246.49
Partial Equilibrium (Huggett)	355.8	10	24.50

Table 1: Runtimes

Solving the Krusell-Smith model takes about 37 seconds, in line with other fast global solution methods in the literature. By contrast, the Huggett and HANK models are typically viewed as more challenging because they feature non-trivial market clearing conditions: standard approaches nest an inner loop that repeatedly solves for prices until markets clear. Nevertheless, our method solves the Huggett model in 46 seconds and the HANK model in about 4 minutes.

Finally, we also compare the cost of computing the model's general equilibrium (GE) to that of computing the corresponding partial equilibrium (PE) problem (see below). We find that, while computing the GE problem takes longer as expected, the difference in runtime is modest. In the Huggett model, for example, moving from partial to general equilibrium increases runtime from 25 seconds to 46 seconds. This is because we do not solve general equilibrium with a nested inner-outer loop that alternates between solving optimal policies and updating price functions or perceived laws of motion. Instead, prices are learned in an online fashion: along each simulated path we compute the market-clearing price implied by current policies, and the policy update uses these realized prices directly.

4.1 Huggett Model with Aggregate Risk

We start our computational experiments with the Huggett economy described in Section 2.[25]

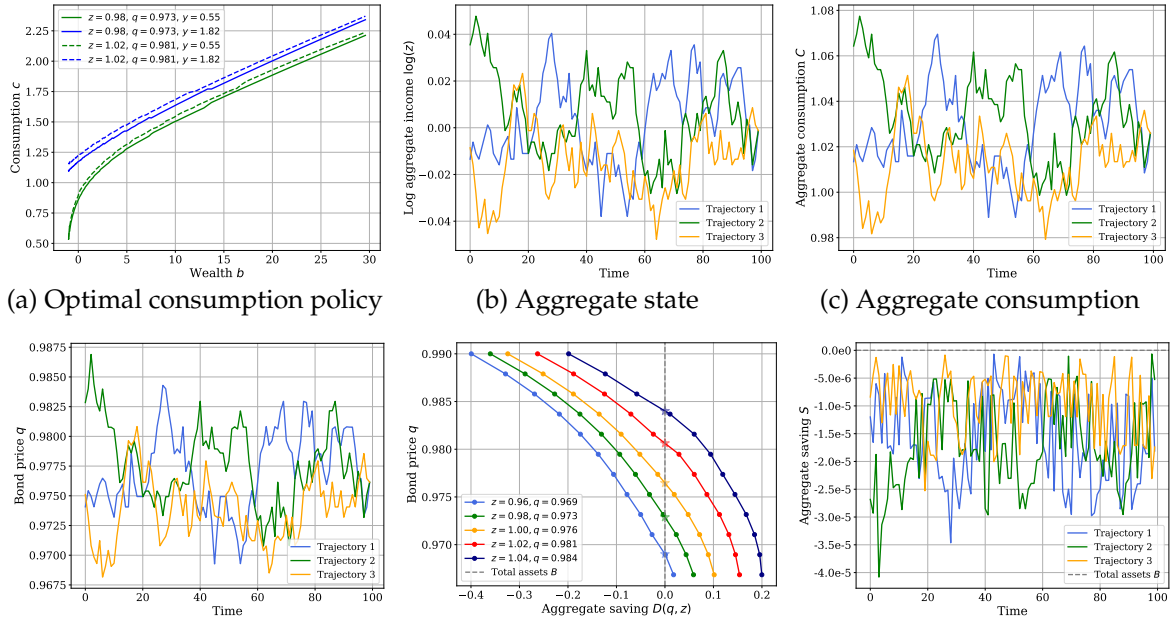
Calibration. We interpret one period as a year and set the household discount factor to $\beta = 0.96$. Preferences are isoelastic $u(c) = \frac{c^{1-\sigma}}{1-\sigma}$ and we set $\sigma = 2$. In the Huggett model, both the

[25] Our comparison of runtimes in Table 1 and Figure 4 below make reference to the partial equilibrium problem of the Huggett economy. We present the details in Appendix A.1.

idiosyncratic and the aggregate income components follow log AR(1) processes. We set the persistence parameters to $\rho_y = 0.6$ and $\rho_z = 0.9$ and the standard deviations of the innovations to $\nu_y = 0.2$ and $\nu_z = 0.02$. We discretize these processes on finite grids using a standard Tauchen procedure (details in the Appendix A). Finally, we impose a borrowing limit $\underline{b} = -1$ and fix aggregate bond supply at $B = 0$, so bonds are in zero net supply. The full calibration table is presented in Appendix A.

Hyperparameters. We discuss and report hyperparameter choices for all our experiments in the Appendix A.

Numerical Results. Figure 3 reports the numerical solution and a simulation for the Huggett economy.



(a) Optimal consumption policy (b) Aggregate state (c) Aggregate consumption

(d) Bond price (e) Bond market equilibrium (f) Market clearing residual

Figure 3: Simulation Results

Panel (a) plots the optimal consumption policy as a function of wealth b on the horizontal axis. Each line corresponds to one combination of individual income, aggregate income, and prices. We choose the realizations of (y_t, z_t, p_t) that occur frequently in our simulations. The policy is monotonically increasing and concave in b, as expected from standard theory: richer households consume more, but at a decreasing marginal propensity.

Panels (b)-(d) display simulated time series for the aggregate state, aggregate consumption and equilibrium prices. Panel (b) shows the exogenous AR(1) process for log aggregate income z_t. Panel (c) plots aggregate consumption C_t which equals aggregate income in equilibrium. Panel (d) shows the resulting bond price q_t, which adjusts endogenously to clear the bond market in the presence of incomplete markets and zero net bond supply.

Panel (e) revisits the equilibrium bond demand schedule $D(p, z)$ discussed in Section 3.4, evaluated at the trained policy. For different realizations of the aggregate state z, the figure

shows how the aggregate demand for bonds varies with the bond price. Market clearing corresponds to the intersection of $D(p, z)$ with zero.

Finally, Panel (f) plots the bond-market clearing residual along the simulated path, i.e. the difference between aggregate bond holdings implied by households' policies and the fixed supply of zero. The residual remains very close to zero throughout the simulation. The small deviations that do arise are due to numerical interpolation in prices rather than to a failure of the algorithm to enforce equilibrium. In practice, these deviations are negligible both in absolute terms. The average gap in bond market clearing for a single run is about 1.4×10^{-5}.

Partial equilibrium problem. To gauge the accuracy of our SRL approach, we first consider a partial equilibrium (PE) version of the Huggett economy. In PE, households take as given an exogenous Markov process for interest rates and solve their individual dynamic program using either our SRL method or a standard value function iteration (VFI) algorithm. Because the PE environment dispenses with the fixed point over prices and distributions, it is a setting in which there is broad agreement on the correctness of conventional VFI solutions. This makes it a natural benchmark against which to compare the policies implied by our method. We describe the details of the PE specification and calibration in Appendix A.1.

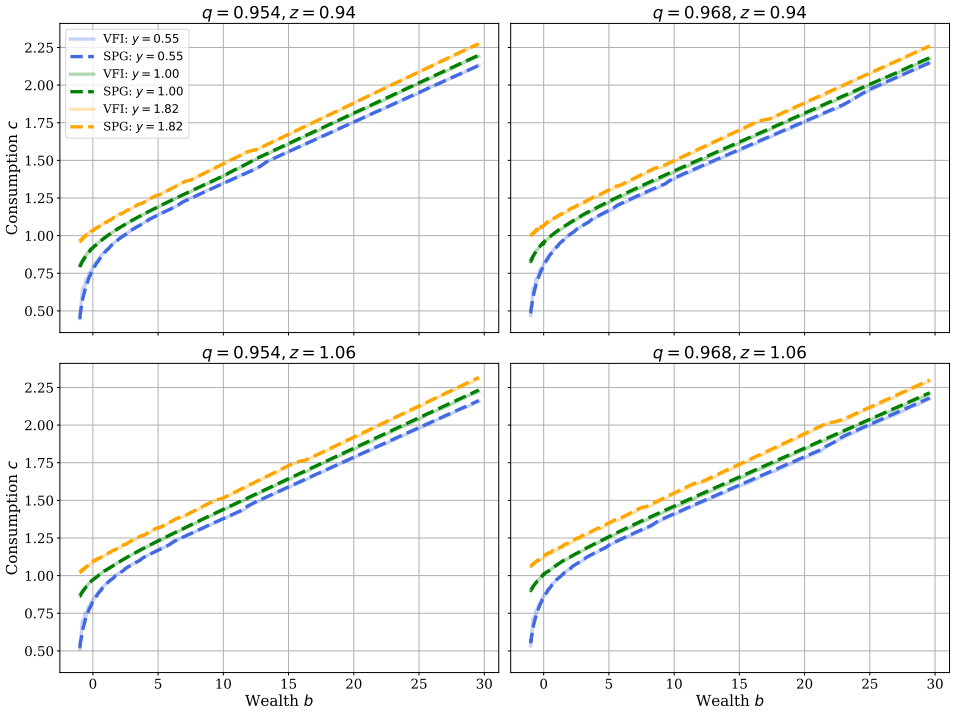


Figure 4: Solution comparison for the PE problem: SRL vs VFI

Figure 4 reports the comparison. Each panel plots the optimal consumption policy as a function of wealth b for four combinations of individual income y and interest rates r. The dashed line shows the policy obtained from our SRL algorithm, while the solid line shows the corresponding VFI solution in the PE environment. Across all four panels, the two sets of policy functions are almost indistinguishable. This comparison is a first reassuring test of the accuracy of our SRL method. It shows that, in a setting where a trusted VFI benchmark is

available, our SRL approach replicates the rational expectation solution closely.

Solutions with Lagged Price History. A complementary way to assess the restrictiveness of conditioning only on p_t is to enlarge the observable state with lagged prices. Conceptually, this moves us part of the way toward the full MA(∞) representation of the agent's problem, which in the limit would reproduce the rational expectations solution. Concretely, we now re-solve the Huggett model allowing households to keep track of one lagged price, so that p_{t-1} becomes an additional state variable.

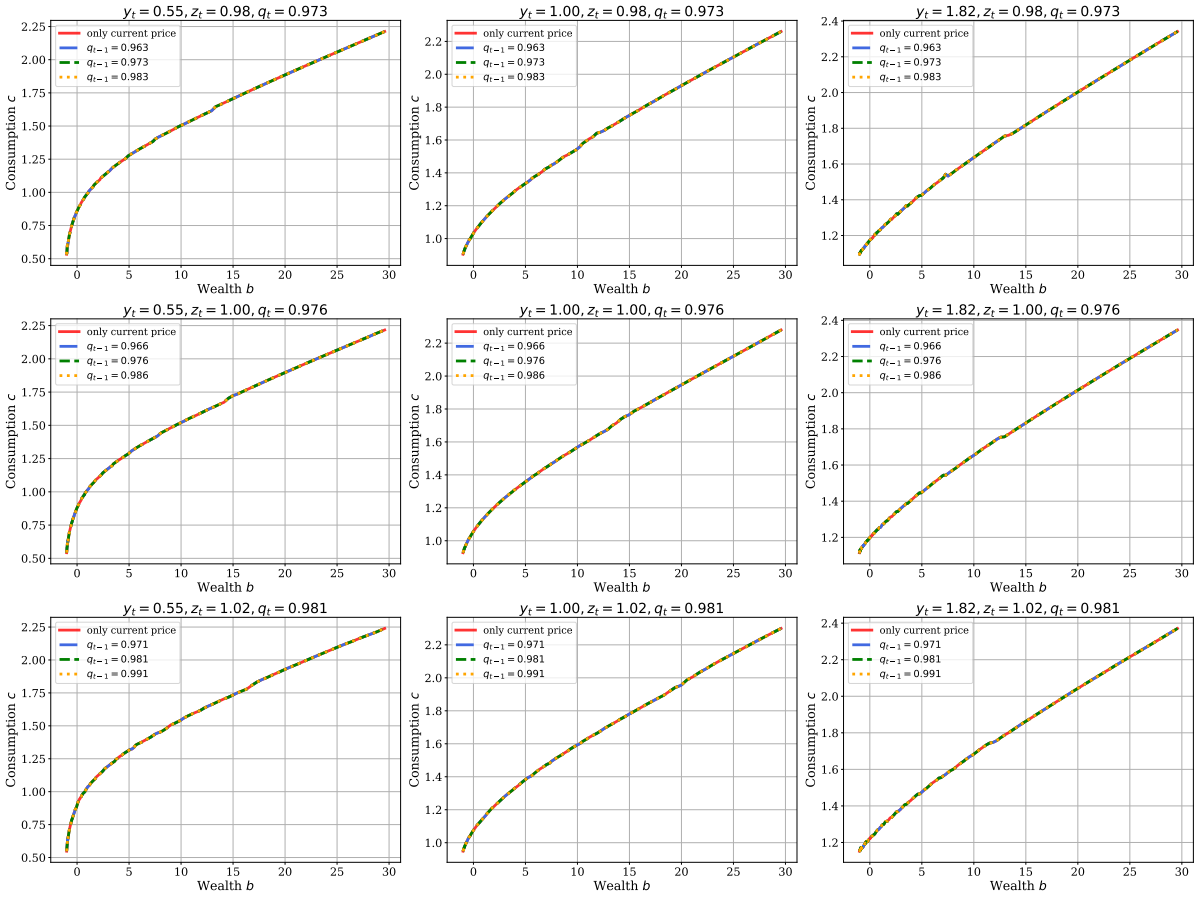


Figure 5: Consumption Policy Function with Price Lags p_{t-1} as State Variable

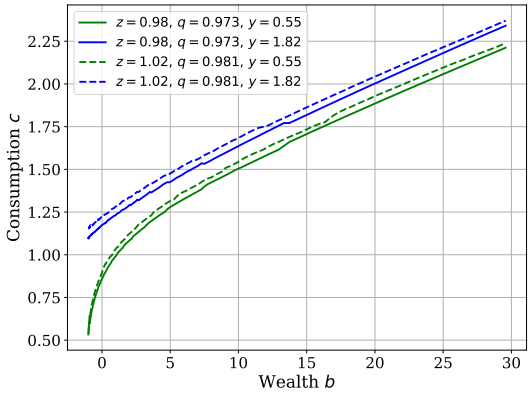
From a computational standpoint, this extension is straightforward: our method can accommodate a small number of lagged observables without reintroducing the curse of dimensionality. From an economic standpoint, it should, in principle, help agents forecast: because prices are not Markov in p_t alone, a longer price history ought to contain incremental information about future prices.

Figure 5 shows that, in practice, the effect of this additional information is minimal. The figure compares solutions to the Huggett model when agents condition (i) only on the current price p_t (solid blue line) and (ii) on both p_t and its lag p_{t-1} (dashed lines). Each panel plots the consumption policy across wealth b for fixed values of the individual income state y, the aggregate income state z, and the current bond price q. Different dashed lines correspond to different realizations of *past* prices.

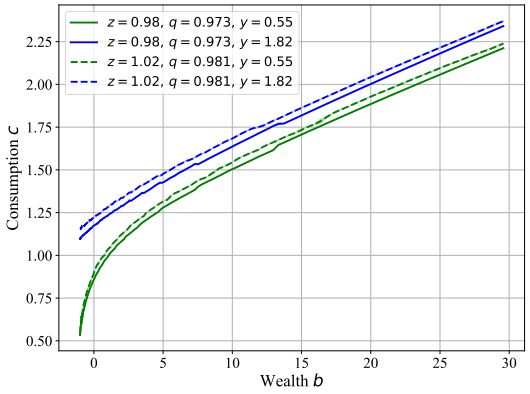
Across all panels, the dashed lines lie almost on top of the solid line. That is, once we fix the current state (y_t, z_t, p_t), optimal consumption is almost insensitive to the additional information contained in p_{t-1}. This suggests that, at least in the Huggett environment, current prices already summarize the relevant aspects of the history for household decisions, and that extending the observable state to include one lag has only a negligible impact on behavior.

Dependence on the number of trajectories (sample size). Next, we study how the quality and stability of the learned policy depend on the number of simulated trajectories used for training. Figure 6 summarizes these results.

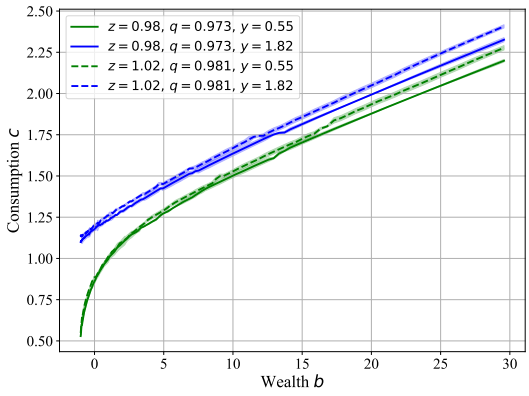
Panel (a) reports the consumption policy obtained from a single training run with 128 simulated trajectories. This number of trajectories is a key hyperparameter in our algorithm: it controls how many distinct state-price paths agents observe and learn from. The resulting policy is monotone and concave in wealth, as theory would suggest.



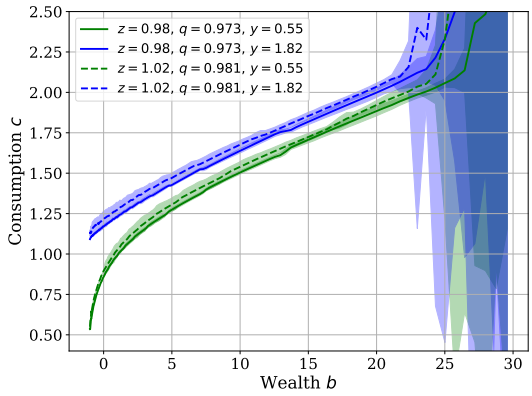
(a) Policy with $N_{\text{sample}} = 128$



(b) Policy CI with $N_{\text{sample}} = 128$



(c) Policy CI with $N_{\text{sample}} = 64$



(d) Policy CI with $N_{\text{sample}} = 8$

Figure 6: Dependence of Policies on Sample Size in the Huggett model

Panel (b) turns to sampling uncertainty. Here we compute pointwise 95% confidence intervals (CIs) for the policy based on 10 independent training runs, each again using 128 trajectories. The figure shows that the confidence bands are quite tight across the wealth distribution.

Panels (c) and (d) vary the number of training trajectories to examine the solution confidence bands. In Panel (c), we reduce the number of trajectories to 64. The point estimates of the policy change very little, but the confidence bands widen modestly. In other words, feeding

the algorithm fewer data primarily increases uncertainty; it does not systematically move the policy itself. Panel (d) shows the case with even fewer trajectories. Here the confidence bands become much wider, especially at high wealth levels. This pattern is natural: states with high wealth are visited only rarely in the simulations — indeed, beyond roughly $b > 10$ there is essentially no mass in the stationary distribution, so the algorithm has fewer observations from which to learn. Regions of the state space that are visited infrequently thus come with more sampling noise in the estimated policy. Put differently, when agents have learned from only a small number of simulations, agents who happen to find themselves in the same high-wealth state can end up taking quite different actions across independent runs.

These results suggest a useful way to think about our stochastic solution method. If we could associate to each point in the state space a simple measure of "confidence" — for example, based on the width of the confidence band or on how often that state is visited in simulation — it would reveal that agents are very certain about their behavior in frequently visited states, but much less certain in rare states. Our experiments indicate that, for economically relevant regions of the state space (low and medium wealth), the learned policies are both stable and precise, while the main residual uncertainty is confined to tails that households almost never reach in practice.

4.2 Krusell-Smith Model

Setup. The household side of the Krusell-Smith economy is as in the Huggett model of Section 2, except that financial wealth is now productive capital owned by households and rented to a representative firm. The firm uses capital and labor to produce according to

$$Y_t = z_t K_t^{\alpha} L_t^{1-\alpha}.$$

Under perfect competition, factor prices equal marginal products, $w_t = (1-\alpha)\frac{Y_t}{L_t}$ and $r_t^K = \alpha \frac{Y_t}{K_t}$, where w_t is the real wage rate and r_t^K the rental rate of capital. Since households own the capital and pay for depreciation, their net rate of return on capital is $r_t = r_t^K - \delta$. The market clearing condition for capital is

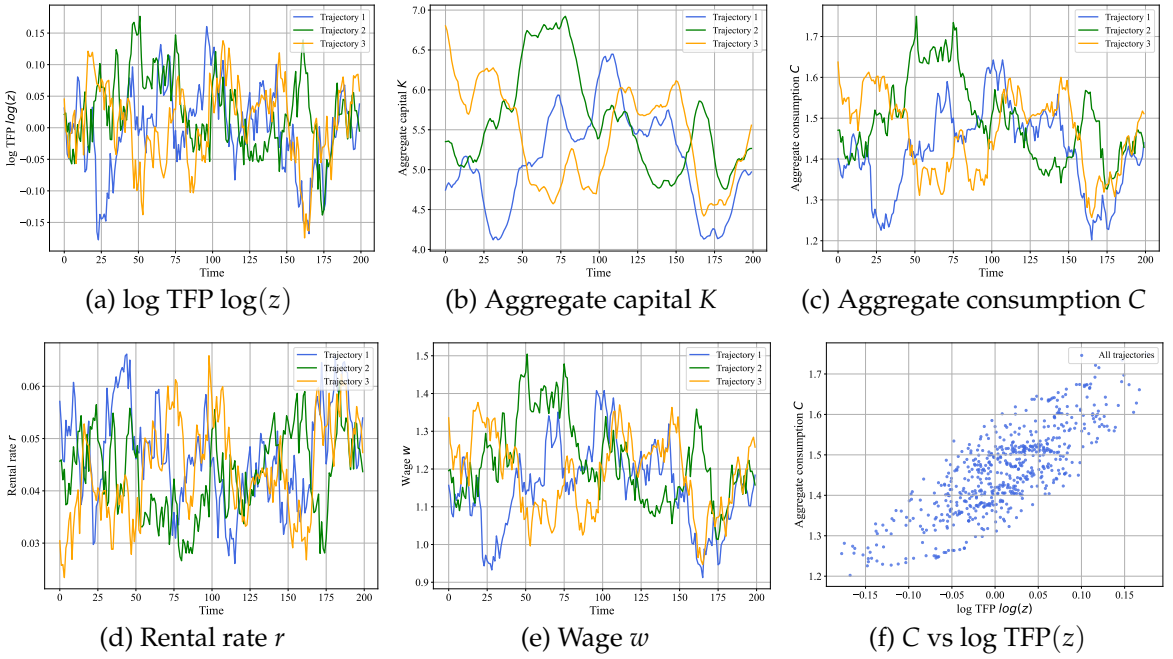
$$\int b \, dG_t(b, y) = K_t,$$

and labor market clearing condition is given by $L_t = 1$ because each household supplies one unit of labor inelastically.

Calibration. One period corresponds to a year. On the preference side, we set $\beta = 0.95$ and use CRRA preferences with coefficient of relative risk aversion $\sigma = 3$. On the production side, we set the capital share to $\alpha = 0.36$ and the depreciation rate to $\delta = 0.08$, standard in the quantitative macro literature. For idiosyncratic income, we retain the same AR(1) specification and parameters as in the Huggett model, so that the cross-sectional heterogeneity is directly comparable across the two experiments. Aggregate productivity z_t also follows a log AR(1) process with persistence $\rho_z = 0.9$ and innovation volatility $\nu_z = 0.03$. All log AR(1) processes

are discretized on finite grids using a standard Tauchen procedure; the details of the grids are reported in Appendix A.

Numerical Results. Figure 7 plots simulation results for our solution of the Krusell-Smith model. We start in Panel (a) with the exogenous aggregate productivity process z_t, and plot in Panel (b) aggregate capital K_t, in Panel (c) aggregate consumption C_t, in Panel (d) the aggregate rental rate r_t^K and in Panel (e) the aggregate wage w_t. As in standard neoclassical models, K_t and C_t comove strongly with z_t, while r_t^K and w_t move in opposite directions.



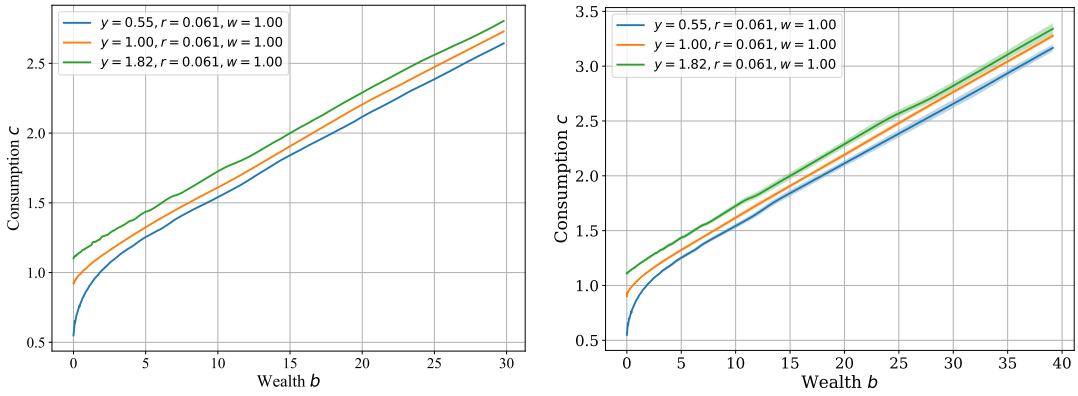
(a) log TFP $\log(z)$ (b) Aggregate capital K (c) Aggregate consumption C

(d) Rental rate r (e) Wage w (f) C vs log TFP(z)

Figure 7: Simulation Results

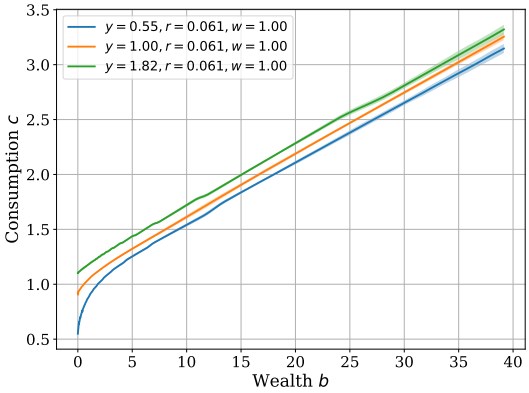
The final panel (f) presents a scatter plot of aggregate consumption C_t against the exogenous productivity shock z_t. We find substantial *vertical* dispersion: for a given realization of z_t, different periods in the simulation exhibit quite different levels of aggregate consumption. If all points lay on a single curve, then the same aggregate productivity level would always be associated with the same C_t. Instead, the cross-sectional wealth distribution shifts over time in ways that matter for aggregates, so the mapping $z_t \mapsto C_t$ is history-dependent. The blue dots in Panel (f) represent states actually visited along the simulated path, so this vertical spread measures the quantitative importance of distributional dynamics for aggregate outcomes. For values of z_t very close to zero, simulated realizations of C_t range roughly from 1.3 to 1.6, i.e. a difference on the order of 20% of steady-state consumption. This illustrates that, even in this relatively simple heterogeneous-agent model, the cross-sectional distribution has a non-trivial impact on the aggregate response.

Dependence on the number of trajectories (sample size). We now illustrate how the learned policy depends on the number of simulated trajectories used for training, similar to our discussion of Figure 6 for the Huggett model. Panels (a) and (b) report the consumption policy

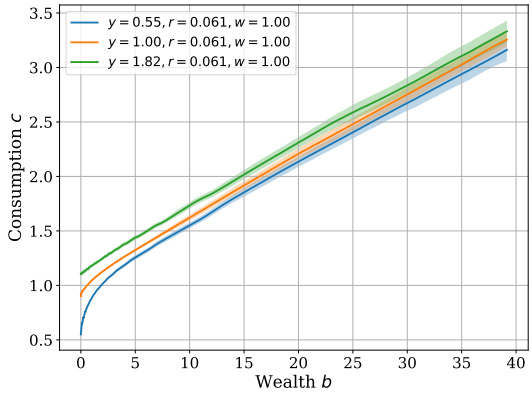


(a) Policy with $N_{\text{sample}} = 128$

(b) Policy CI with $N_{\text{sample}} = 128$



(c) Policy CI with $N_{\text{sample}} = 512$



(d) Policy CI with $N_{\text{sample}} = 32$

Figure 8: Dependence of Policies on Sample Size in the Krusell and Smith (1998) model

from a single training run and the associated 95% confidence intervals based on 10 independent training runs, both using 128 trajectories, respectively. The resulting policy is monotone and concave in wealth and visually very similar to the policies we documented in the Huggett experiment.There is only mild sampling uncertainty. Confidence bands are tight, especially for low wealth levels.

Panel (c) increases the number of trajectories to 512 and illustrates that confidence bands shrink even for the highest wealth levels. Panel (d) instead reduces the number of trajectories to 32 and illustrates that sampling uncertainty across runs increases substantially, especially at high wealth levels which are visited more rarely during the simulation.

Comparison to rational expectation solution. The partial equilibrium comparison we used in Section 4.1 isolates the individual dynamic programming problem, where there is broad agreement on the accuracy of conventional VFI solutions. In general equilibrium, by contrast, obtaining the rational expectations (RE) solution requires treating the entire cross-sectional distribution as a state variable and solving the Master equation, a problem of much higher computational complexity. A growing literature proposes global solution methods for such RE equilibria. One recent example is the DeepHAM approach of Han et al. (2021), which uses deep neural networks to approximate high-dimensional policy and value functions.

To benchmark our SRL method in this environment, we compare it directly to DeepHAM.

Because the RE policy functions in DeepHAM conditions on the full cross-sectional distribution, whereas our approach conditions only on prices, the policy functions are not directly comparable. We therefore focus on equilibrium dynamics. Specifically, we initialize both economies from the same cross-sectional distribution and expose them to identical sequences of aggregate shocks. Figure 9 reports the resulting paths of aggregate consumption and capital. Across both panels, the two methods generate nearly indistinguishable aggregate dynamics.

This comparison indicates that, at least in the Krusell-Smith environment, our SRL approach can replicate the rational expectations solution while retaining the flexibility and scalability of a reinforcement-learning implementation.

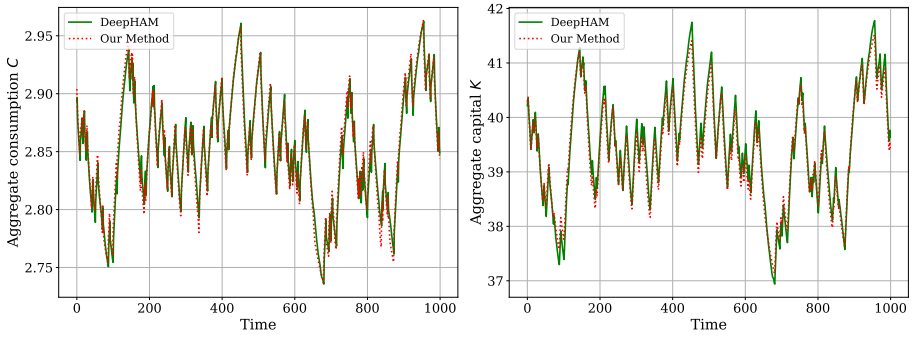


Figure 9: Comparison to RE Solution with the DeepHAM method in Han et al. (2021)

4.3 HANK Model

Our final benchmark economy is a one-account HANK model with sticky prices. This environment adds nominal rigidities and a richer firm block to the incomplete-markets structure from Section 2.

Setup. The problem of household i is similar to that in Section 2, except that bonds are nominal rather than real and that labor supply is now endogenous. Denoting by $b_{i,t}$ bond holdings at the end of period $t-1$ relative to the price level, the households' problem is:[26]

$$v_{i,0} = \max_{\{c_{i,t}, n_{i,t}\}} \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t u(c_{i,t}, n_{i,t})$$
$$\text{s.t.} \quad c_{i,t} + q_t b_{i,t+1} = \frac{b_{i,t}}{1+\Pi_t} + w_t y_{i,t} n_{i,t} + d_t - T_t, \qquad b_{i,t+1} \geq 0.$$

Here $q_t = 1/(1+i_{t+1})$ is the price of the nominal bond with i_{t+1} set by nominal policy and Π_t is inflation (see below). Further, w_t is the real wage, d_t denotes dividend payouts, and T_t is a lump-sum tax. Households are the ultimate owners of firms but equity shares are not traded. The idiosyncratic income process $y_{i,t}$ is as in Section 2.

[26] In nominal terms, the budget constraint is $P_t c_{i,t} + q_t \tilde{b}_{i,t+1} = \tilde{b}_{i,t} + \tilde{w}_t y_{i,t} n_{i,t} + \tilde{d}_t - \tilde{T}_t$ where $\tilde{b}_{i,t}$ are nominal bond holdings, P_t is the price level, $\tilde{w}_t$ is the nominal wage, and so on. Dividing by P_t and defining $b_{i,t} = \tilde{b}_{i,t}/P_{t-1}$ yields the budget constraint in the text.

On the production side, a competitive final-good firm aggregates a continuum of intermediate inputs using a CES production technology with elasticity of substitution ε. Denoting by Y_t aggregate output of the final good, cost minimization implies that demand for intermediate input j is

$$Y_{j,t} = \left(\frac{P_{j,t}}{P_t}\right)^{-\varepsilon} Y_t, \tag{22}$$

where $P_{j,t}$ is the price of good j and

$$P_t = \left(\int_0^1 P_{j,t}^{1-\varepsilon} dj\right)^{\frac{1}{1-\varepsilon}}$$

is the aggregate price index.

Each intermediate good j is produced by a monopolistically competitive firm with technology $Y_{j,t} = z_t L_{j,t}$. The productivity term z_t is common to all firms and follows a Markov process $z_{t+1} \sim \mathcal{T}_z(\cdot \mid z_t)$. Firm j chooses its price $\{P_{j,t}\}$ to maximize the discounted value of nominal profits discounted at the nominal risk-free rate $1 + i_{t+1} = 1/q_t$ subject to a quadratic adjustment cost as in Rotemberg (1982). Written in real terms, firm j's problem is:[27]

$$J_{j,0} = \max_{\{P_{j,t}\}} \mathbb{E}_0 \sum_{t=0}^{\infty} \Lambda_{0\to t}\left[\left(\frac{P_{j,t}}{P_t} - \frac{w_t}{z_t}\right) Y_{j,t} - \frac{\theta}{2}\left(\frac{P_{j,t} - P_{j,t-1}}{P_{j,t-1}}\right)^2 Y_t\right] \tag{23}$$

where $\Lambda_{0\to t} = \prod_{s=0}^{t-1} \frac{1+\Pi_{s+1}}{1+i_{s+1}}$ with $\Lambda_{0\to 0} = 1$ and subject to the demand function (22) and taking as given an initial price $P_{j,-1}$.

We focus on a symmetric distribution of initial prices, $P_{j,-1} = P_{j',-1}$, which implies symmetry ex post. In equilibrium we therefore have $P_{j,t} = P_t$ and $Y_{j,t} = Y_t$ for all j. Denoting inflation by

$$\Pi_t = \frac{P_t - P_{t-1}}{P_{t-1}}, \tag{24}$$

the firm's problem gives rise to the New Keynesian Phillips curve

$$\Pi_t(1 + \Pi_t) = \frac{\varepsilon}{\theta}\left(\frac{w_t}{z_t} - \frac{\varepsilon - 1}{\varepsilon}\right) + \mathbb{E}_t\left[\Lambda_{t\to t+1}\frac{Y_{t+1}}{Y_t}\Pi_{t+1}(1 + \Pi_{t+1})\right]. \tag{25}$$

The first term captures the gap between real marginal cost $\frac{w_t}{z_t}$ and the desired markup $\frac{\varepsilon}{\varepsilon-1}$, while the second term reflects expected future inflation, discounted by the real interest rate and scaled by output growth. Given inflation, firm dividend payments are

$$d_t = \left(1 - \frac{w_t}{z_t}\right) Y_t - \frac{\theta}{2}\Pi_t^2 Y_t.$$

[27] The firm's original problem in nominal terms is

$$J_{j,0}^{\text{nominal}} = \max_{\{P_{j,t}\}} \mathbb{E}_0 \sum_{t=0}^{\infty} Q_{0\to t}\left[\left(P_{j,t} - \frac{\tilde{w}_t}{z_t}\right) Y_{j,t} - \frac{\theta}{2}\left(\frac{P_{j,t} - P_{j,t-1}}{P_{j,t-1}}\right)^2 P_t Y_t\right],$$

where $Q_{0\to t} = \prod_{s=0}^{t-1} q_s = \prod_{s=0}^{t-1} \frac{1}{1+i_{s+1}}$ and $\tilde{w}_t$ is the nominal wage. Dividing by P_t and defining $\Lambda_{0\to t} = Q_{0\to t}\frac{P_t}{P_0}$, and $w_t = \tilde{w}_t / P_t$ yields the firm's problem in real terms (23).

Monetary policy – which determines the bond price $q_t = 1/(1+i_{t+1})$ – follows a Taylor rule

$$1 + i_{t+1} = \bar{R}(1+\Pi_t)^{\phi} e^{\epsilon_t},$$

where $\phi > 1$ and the monetary policy shock $\epsilon_{t+1} \sim \mathcal{T}_\epsilon(\cdot \mid \epsilon_t)$ follows a Markov process.

The government budget constraint is $q_t B_{t+1} + T_t = \frac{B_t}{1+\Pi_t}$ where B_t denotes bonds outstanding at the end of period $t-1$ relative to the price level (as in the definition of $b_{i,t}$ for households). We assume that the government has a fiscal rule that holds bonds constant over time at $B_t = B$ so that

$$T_t = \left(\frac{1}{1+\Pi_t} - q_t\right) B.$$

Finally, three markets must clear in equilibrium. Goods market clearing requires

$$Y_t = \int c_t(b,y) dG_t(b,y) + \frac{\theta}{2}\Pi_t^2 Y_t.$$

so aggregate output is absorbed by consumption and price-adjustment costs. Labor market clearing implies

$$L_t = \int n_t(b,y) y dG_t(b,y),$$

and the bond market clears when

$$B = \int b_t'(b,y)\, dG_t(b,y),$$

The definition of competitive equilibrium is standard.

Firm Policy Gradient Method for the Phillips Curve. The price-setting block introduces an additional difficulty relative to the Huggett and Krusell-Smith models: Firm optimality gives rise to the forward-looking Phillips curve (25). Standard approaches typically parameterize the conditional expectation term in Equation (25) and solve a non-trivial fixed point for the law of motion of inflation; see for example Kase et al. (2024); Fernández-Villaverde et al. (2024b).

By contrast, we treat the firm problem (23) exactly as we treat the household problem and solve it using the same SPG method. Appendix A.3 explains this in detail. The key idea is to rewrite the firm problem in a way that makes inflation the firm's policy choice. We then parameterize the symmetric equilibrium inflation policy as a function of payoff-relevant prices and aggregate shocks, $\Pi_t = \Pi(z_t, Y_t, w_t)$, and update it jointly with household policies using SPG. Firms and households learn from the *same* simulated equilibrium trajectories, and all policy functions are updated *simultaneously*. Since we sidestep the expectation term in (25), there is no separate fixed-point iteration for equilibrium expectations.

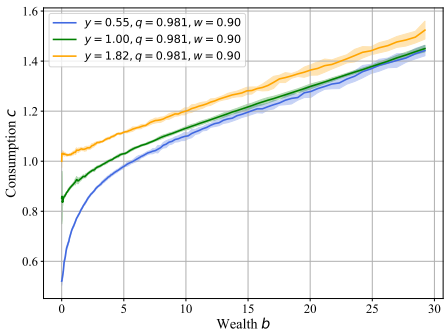
In our experiments, this symmetric treatment of households and firms has good convergence properties. As Table 1 shows, solving the HANK model with the forward-looking Phillips curve is fast and the computational cost remains similar to that for the baseline Huggett model, despite the added forward-looking structure in the firm block.

Calibration. A time period corresponds to a year, with $\beta = 0.95$. Preferences are CRRA and separable in consumption and labor, $u(c, n) = \frac{1}{1-\sigma}c^{1-\sigma} - \frac{1}{1+\eta}n^{1+\eta}$, with coefficient of relative risk aversion $\sigma = 1$ implying log utility over consumption, and inverse Frisch elasticity $\eta = 1$.

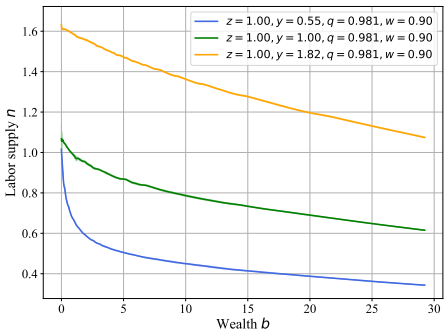
On the production side, we set the elasticity of substitution across intermediate goods to $\varepsilon = 10$ and the Rotemberg adjustment cost parameter to $\theta = 100$. We set the Taylor rule coefficient to $\phi = 1.5$, and we set the fixed government bond supply to $B = 5$.

This economy features three shocks: one idiosyncratic and two aggregate. For households' idiosyncratic income process, we use the same AR(1) specification and parameterization as in the Huggett model so that cross-sectional heterogeneity is directly comparable across applications. Aggregate risk comprises the TFP process z_t and the monetary policy shock ϵ_t, both following AR(1) processes. We set their persistence to $\rho_z = \rho_\epsilon = 0.9$ and their innovation volatilities to $\nu_z = 0.07$ and $\nu_\epsilon = 0.002$, respectively. We use a standard Tauchen procedure to discretize all three processes on finite grids and report all remaining details in Appendix A.

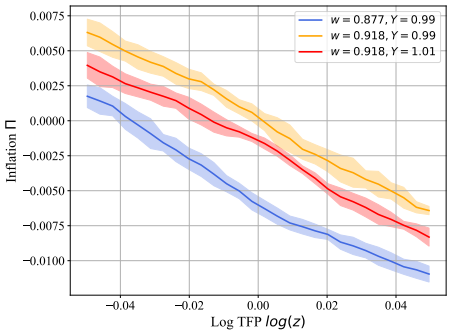
Numerical Results. Figure 10 reports the optimal policy functions for households and firms in the HANK model. Panels (a) and (b) show household consumption and labor supply policies, while Panel (c) displays the firm's inflation policy. For each policy, we plot the point estimate together with confidence bands obtained from 10 independent training runs with $N = 256$ simulated trajectories each.



(a) Consumption policy



(b) Labor supply policy



(c) Inflation policy

Figure 10: Household and Firm Policy Functions in HANK

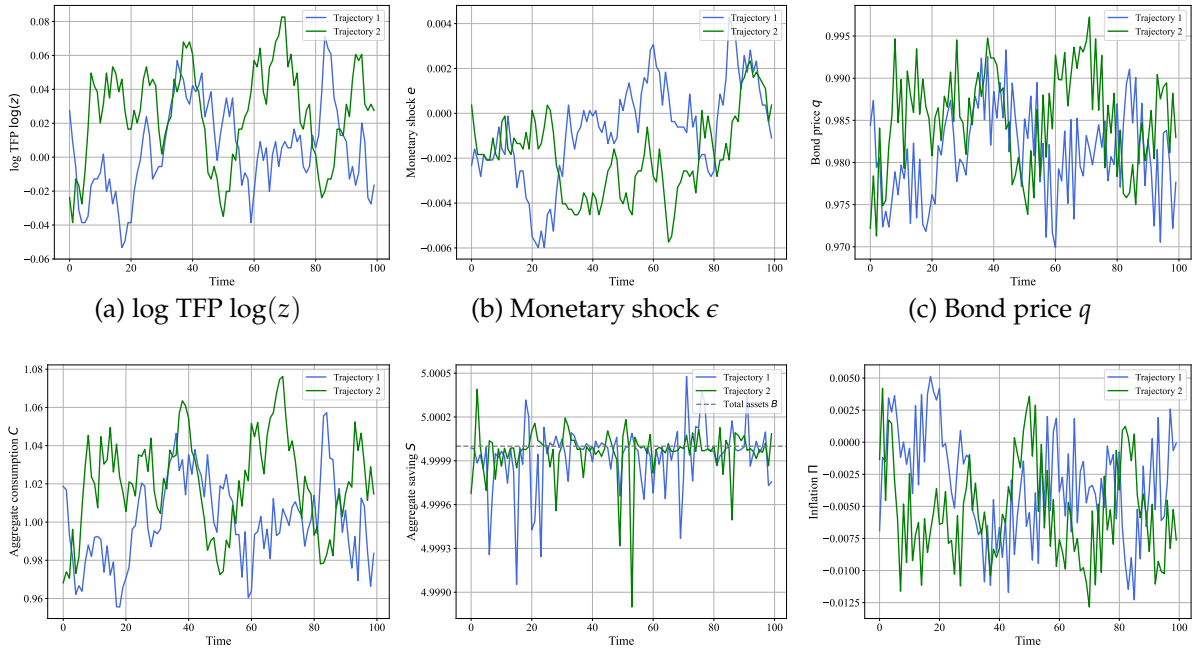
Panel (a) shows that consumption is increasing and concave in wealth and increasing in idiosyncratic labor productivity, as standard theory would suggest. Panel (b) shows the corresponding labor supply policy, which is decreasing and convex in wealth and increasing in

labor productivity: richer households work less at the margin, while high-productivity households supply more labor.

Panel (c) displays the firm's inflation policy function. It is decreasing and almost linear in log aggregate productivity and increasing in marginal cost. In particular, positive supply (TFP) shocks lower marginal cost and induce lower inflation, consistent with the New Keynesian Phillips curve (25).

Across all three panels, the confidence bands are tight over the bulk of the wealth distribution and widen somewhat only in the far tails, where states are rarely visited in simulation. The firm's policy is the most challenging to learn — reflected in somewhat wider confidence bands — but remains well behaved and economically sensible. Overall, Figure 10 suggests that our SPG method recovers accurate policy functions in this richer HANK environment.

Figure 11 reports simulated time series for the HANK model. Panels (a) and (b) display the two aggregate shocks: log TFP z_t and the monetary policy shock ϵ_t. Panels (c)-(f) then show the induced time series for the nominal bond price, aggregate consumption, aggregate savings, and inflation.



(a) log TFP $\log(z)$ (b) Monetary shock ϵ (c) Bond price q

(d) Aggregate consumption C (e) Aggregate saving S (f) Inflation Π

Figure 11: Simulated Trajectories in HANK

Aggregate consumption in Panel (d) moves procyclically with productivity: positive TFP shocks raise Y_t and, through higher labor income and lower marginal costs, also increase C_t. Inflation in Panel (f) is countercyclical with respect to TFP, consistent with the Phillips curve (25): favorable supply shocks reduce marginal costs and put downward pressure on inflation. In our calibration macro aggregates appear more sensitive to TFP innovations than to monetary shocks.

Panel (e) plots households' aggregate demand for bonds, together with the fixed supply $B = 5$. Asset demand fluctuates tightly around the supply level, and deviations from the

horizontal supply line measure residuals in the bond market clearing condition. On average, the relative absolute deviation of aggregate demand from supply is 0.003%. These residuals primarily reflect the use of linear interpolation in solving for market-clearing prices.

5 Conclusion

We develop a new *structural reinforcement learning* (SRL) approach to formulating and globally solving heterogeneous agent models with aggregate risk. We replace the cross-sectional distribution with low-dimensional prices as state variables and let agents compute price expectations directly from simulated paths. Our approach differs from standard RL in that we assume that agents have structural knowledge about the dynamics of their own individual states (e.g. their budget constraint and idiosyncratic income process). Our *structural policy gradient* (SPG) algorithm sidesteps the Master equation and efficiently handles heterogeneous agent models traditional methods struggle with, like those with nontrivial market-clearing conditions. By imposing that policy functions depend only on current prices (or a short price history) we keep the state space low-dimensional so that we can work with a grid-based (tabular) approach rather than deep neural networks.

We implement our method in JAX and conduct computational experiments for three benchmark models in macroeconomics – the Huggett (1993) model, the Krusell and Smith (1998) model, and a one-asset HANK model with sticky prices. In all three cases, the SPG algorithm converges in only a few minutes. In the Krusell-Smith model, the resulting policies are close to those obtained from alternative global solutions of the rational expectations equilibrium. Allowing agents to condition on a longer history of lagged prices hardly moves the solution, indicating that much of the relevant information for forecasting future prices is already contained in current prices. In the HANK model, we show how the same approach can be used symmetrically on the household and firm sides to globally solve the forward-looking New Keynesian Phillips curve.

The core idea of our approach – that agents form price expectations by sampling – could, in principle, serve as a building block for an empirically realistic theory of expectations formation in macroeconomics (which the algorithm in this paper is not). The plausibility of RL-based approaches is supported by evidence that reinforcement learning underpins a substantial share of human and animal learning.[28] To develop such an RL-based theory of expectations formation would require several modifications to our SRL method. First, the algorithm would need to be converted to a fully online, incremental RL algorithm, with agents updating policies and value estimates continuously while interacting with their environment, rather than only after observing N price trajectories of length T. Second, the assumption that agents sample equilibrium prices in an unbiased way would likely need to be relaxed. Empirical evidence suggests that people disproportionately weight certain personal experiences (e.g. Malmendier and Nagel, 2011, 2015), so incorporating biased or experience-weighted sampling would be

[28] See, for example, Niv (2009), Glimcher (2011), Caplin and Dean (2008), Glimcher et al. (2013), Gershman and Daw (2017), and Barberis and Jin (2023).

more realistic. Third, our SRL agents form price expectations in a model-free manner; this may be too simplistic and one could instead treat prices using a model-based RL approach.

References

Achdou, Yves, Jiequn Han, Jean-Michel Lasry, Pierre-Louis Lions, and Benjamin Moll, "Income and Wealth Distribution in Macroeconomics: A Continuous-Time Approach," *The Review of Economic Studies*, 04 2021, *89* (1), 45–86.

Ahn, SeHyoun, Greg Kaplan, Benjamin Moll, Thomas Winberry, and Christian Wolf, "When Inequality Matters for Macro and Macro Matters for Inequality," *NBER Macroeconomics Annual*, 2018, *32* (1), 1–75.

Aiyagari, S. Rao, "Uninsured Idiosyncratic Risk and Aggregate Saving," *The Quarterly Journal of Economics*, August 1994, *109* (3), 659–84.

Auclert, Adrien, Bence Bardóczy, Matthew Rognlie, and Ludwig Straub, "Using the Sequence-Space Jacobian to Solve and Estimate Heterogeneous-Agent Models," *Econometrica*, September 2021, *89* (5), 2375–2408.

_ , Matthew Rognlie, and Ludwig Straub, "Fiscal and Monetary Policy with Heterogeneous Agents," *Annual Review of Economics*, 2025, *17* (Volume 17, 2025), 539–562.

_ , Rodolfo Rigato, Matthew Rognlie, and Ludwig Straub, "Beyond Certainty Equivalence: Second-order Dynamics in the Sequence Space," Presentation slides 2025.

Azinovic, Marlon, Luca Gaegauf, and Simon Scheidegger, "Deep Equilibrium Nets," *International Economic Review*, 2022, *63* (4), 1471–1525.

Azinovic-Yang, Marlon and Jan Žemlička, "Deep Learning in the Sequence Space," *arXiv preprint arXiv:2509.13623*, 2025.

Barberis, Nicholas and Lawrence Jin, "Model-free and Model-based Learning as Joint Drivers of Investor Behavior," NBER Working Papers 31081 March 2023.

Bilal, Adrien, "Solving Heterogeneous Agent Models with the Master Equation," NBER Working Papers 31103, National Bureau of Economic Research 2023.

Bradbury, James, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang, "JAX: composable transformations of Python+NumPy programs," 2018.

Branch, William A., "Restricted Perceptions Equilibria and Learning in Macroeconomics," *Post Walrasian Macroeconomics: Beyond the Dynamic Stochastic General Equilibrium Model. Cambridge University Press, New York*, 2006, pp. 135–160.

Bray, Margaret, "Learning, Estimation, and the Stability of Rational Expectations," *Journal of Economic Theory*, April 1982, *26* (2), 318–339.

Calvano, Emilio, Giacomo Calzolari, Vincenzo Denicolo, and Sergio Pastorello, "Artificial intelligence, algorithmic pricing, and collusion," *American Economic Review*, 2020, *110* (10), 3267–3297.

Caplin, Andrew and Mark Dean, "Dopamine, Reward Prediction Error, and Economics," *The Quarterly Journal of Economics*, 2008, *123* (2), 663–701.

Cardaliaguet, Pierre, François Delarue, Jean-Michel Lasry, and Pierre-Louis Lions, *The Master Equation and the Convergence Problem in Mean Field Games*, Vol. 201 of *Annals of Mathematics Studies*, Princeton University Press, 2019.

Chen, Mingli, Andreas Joseph, Michael Kumhof, Xinlei Pan, and Xuan Zhou, "Deep Reinforcement Learning in a Monetary Model," 2023.

Cho, In-Koo and Thomas J. Sargent, *Self-confirming Equilibria*, Palgrave Macmillan UK, December 2016.

de la Barrera, Marc and Tim de Silva, "Model-Agnostic Dynamic Programming," Working Paper, Stanford University 2024.

DeepSeek-AI, "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning," 2025.

Den Haan, Wouter J., "Heterogeneity, Aggregate Uncertainty, and the Short-Term Interest Rate," *Journal of Business & Economic Statistics*, October 1996, *14* (4), 399–411.

_ , Kenneth L. Judd, and Michel Juillard, "Computational suite of models with heterogeneous agents: Incomplete markets and aggregate uncertainty," *Journal of Economic Dynamics and Control*, 2010, *34* (1), 1–3.

Dou, Winston W., Itay Goldstein, and Yan Ji, "AI-Powered Trading, Algorithmic Collusion, and Price Efficiency," Working Paper 4452704, SSRN 2025.

Duarte, Victor, Diogo Duarte, and Dejanir H Silva, "Machine Learning for Continuous-Time Finance," *The Review of Financial Studies*, 2024, *37* (11), 3217–3271.

Erev, Ido and Alvin E. Roth, "Learning in Extensive-Form Games: Experimental Data and Simple Dynamic Models in the Intermediate Term," *Games and Economic Behavior*, 1995, *8* (1), 164–212.

_ and _ , "Predicting How People Play Games: Reinforcement Learning in Experimental Games with Unique, Mixed Strategy Equilibria," *American Economic Review*, 1998, pp. 848–881.

Evans, George W. and Seppo Honkapohja, *Learning and Expectations in Macroeconomics*, Princeton University Press, 2001.

Farmer, Leland E. and Alexis Akira Toda, "Discretizing nonlinear, non-Gaussian Markov processes with exact conditional moments," *Quantitative Economics*, 2017, *8* (2), 651–683.

Favilukis, Jack, Sydney C. Ludvigson, and Stijn Van Nieuwerburgh, "The Macroeconomic Effects of Housing Wealth, Housing Finance, and Limited Risk Sharing in General Equilibrium," *Journal of Political Economy*, 2017, *125* (1), 140–223.

Fernández-Villaverde, Jesús, Galo Nuño, and Jesse Perla, "Taming the Curse of Dimensionality: Quantitative Economics with Deep Learning," NBER Working Papers 33117, National Bureau of Economic Research November 2024.

_ , _ , and Samuel Hurtado, "Financial Frictions and the Wealth Distribution," *Econometrica*, May 2023, *91* (3), 869–901.

_ , _ , Joël Marbet, and Omar Rachedi, "Inequality and the Zero Lower Bound," *Journal of Econometrics*, 2024, p. 105819.

Freeman, C. Daniel, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem, "Brax – A Differentiable Physics Engine for Large Scale Rigid Body Simulation," 2021.

Fudenberg, Drew and David K. Levine, "Whither Game Theory? Towards a Theory of Learning in Games," *Journal of Economic Perspectives*, 2016, *30* (4), 151–170.

Gabriele, Federico, Aldo Glielmo, and Marco Taboga, "Heterogeneous RBCs via deep multi-agent reinforcement learning," 2025.

Gershman, Samuel J. and Nathaniel D. Daw, "Reinforcement Learning and Episodic Memory in Humans and Animals: An Integrative Framework," *Annual Review of Psychology*, 2017, *68* (Volume 68, 2017), 101–128.

Giusto, Andrea, "Adaptive Learning and Distributional Dynamics in an Incomplete Markets Model," *Journal of Economic Dynamics and Control*, 2014, *40*, 317–333.

Glimcher, Paul W., "Understanding Dopamine and Reinforcement Learning: The Dopamine Reward Prediction Error Hypothesis," *Proceedings of the National Academy of Sciences*, 2011, *108* (supplement_3), 15647–15654.

_ , Colin Camerer, Ernst Fehr, and Russell Poldrack, *Neuroeconomics: Decision Making and the Brain* Academic Press, Academic Press, 2013.

Gomes, Francisco and Alexander Michaelides, "Asset Pricing with Limited Risk Sharing and Heterogeneous Agents," *Review of Financial Studies*, January 2008, *21* (1), 415–448.

Gopalakrishna, Goutham, Zhouzhou Gu, and Jonathan Payne, "Asset Pricing, Participation Constraints, and Inequality," Working Paper 2024.

Gu, Zhouzhou, Mathieu Laurière, Sebastian Merkel, and Jonathan Payne, "Global Solutions to Master Equations for Continuous Time Heterogeneous Agent Macroeconomic Models," *arXiv preprint arXiv:2406.13726*, 2024.

Guarda, Sebastian, "Narrow and Short Beliefs in Macroeconomics with Heterogeneous Agents," Technical Report, Princeton University November 2025. Working paper, available at https://files.sebastianguarda.com/JMP.pdf.

Han, Jiequn, Yucheng Yang, and Weinan E, "DeepHAM: A Global Solution Method For Heterogeneous Agent Models With Aggregate Shocks," *arXiv preprint arXiv:2112.14377*, 2021.

Hausknecht, Matthew and Peter Stone, "Deep Recurrent Q-Learning for Partially Observable MDPs," 2017.

Heathcote, Jonathan, Kjetil Storesletten, and Giovanni L. Violante, "Quantitative Macroeconomics with Heterogeneous Households," *Annual Review of Economics*, 05 2009, *1* (1), 319–354.

Hu, Yuanming, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Frédo Durand, "DiffTaichi: Differentiable Programming for Physical Simulation," 2020.

Huang, Ji, "Breaking the Curse of Dimensionality in Heterogeneous-Agent Models: A Deep Learning-Based Probabilistic Approach," 2023.

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

Jacobson, Margaret M., "Beliefs, Aggregate Risk, and the U.S. Housing Boom," Technical Report 2025.

Kahou, Mahdi Ebrahimi, Jesús Fernández-Villaverde, Jesse Perla, and Arnav Sood, "Exploiting symmetry in high-dimensional dynamic programming," Technical Report 2021.

Kaplan, Greg, Kurt Mitman, and Giovanni L. Violante, "The Housing Boom and Bust: Model Meets Evidence," *Journal of Political Economy*, 2020, *128* (9), 3285–3345.

Kase, Hanno, Leonardo Melosi, and Matthias Rottner, "Estimating Nonlinear Heterogeneous Agent Models with Neural Networks," Working Paper, FRB Chicago 2024.

Krueger, D., K. Mitman, and F. Perri, "Chapter 11 - Macroeconomics and Household Heterogeneity," in John B. Taylor and Harald Uhlig, eds., *Handbook of Macroeconomics*, Vol. 2, Elsevier, 2016, pp. 843–921.

Krusell, Per and Anthony A. Smith, "Income and wealth heterogeneity, portfolio choice, and equilibrium asset returns," *Macroeconomic dynamics*, 1997, *1* (2), 387–422.

_ and _ , "Income and Wealth Heterogeneity in the Macroeconomy," *Journal of Political Economy*, October 1998, *106* (5), 867–896.

_ and _ , "Quantitative Macroeconomic Models with Heterogeneous Agents," in "Advances in Economics and Econometrics: Theory and Applications, Ninth World Congress," Cambridge University Press, 2006.

Lasry, Jean-Michel and Pierre-Louis Lions, "Mean field games," *Japanese Journal of Mathematics*, 2007, *2*, 229–260.

Laurière, Mathieu, Sarah Perrin, Julien Pérolat, Sertan Girgin, Paul Muller, Romuald Élie, Matthieu Geist, and Olivier Pietquin, "Learning in Mean Field Games: A Survey," 2024.

_ , _ , Sertan Girgin, Paul Muller, Ayush Jain, Theophile Cabannes, Georgios Piliouras, Julien Perolat, Romuald Elie, Olivier Pietquin, and Matthieu Geist, "Scalable Deep Reinforcement Learning Algorithms for Mean Field Games," in Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, eds., *Proceedings of the 39th International Conference on Machine Learning*, Vol. 162 of *Proceedings of Machine Learning Research* PMLR 17–23 Jul 2022, pp. 12078–12095.

Lee, Donghoon and Kenneth I. Wolpin, "Intersectoral Labor Mobility and the Growth of the Service Sector," *Econometrica*, 2006, *74* (1), 1–46.

Llull, Joan, "Immigration, Wages, and Education: A Labour Market Equilibrium Structural Model," *The Review of Economic Studies*, 2018, *85* (3), 1852–1896.

Maliar, Lilia and Serguei Maliar, "Deep learning: Solving HANC and HANK models in the absence of Krusell-Smith aggregation," *Available at SSRN 3758315*, 2020.

_ , _ , and Pablo Winant, "Deep Learning for Solving Dynamic Economic Models," *Journal of Monetary Economics*, 2021, *122* (C), 76–101.

Malmendier, Ulrike and Stefan Nagel, "Depression Babies: Do Macroeconomic Experiences Affect Risk Taking?," *The Quarterly Journal of Economics*, 2011, *126* (1), 373–416.

_ and _ , "Learning from Inflation Experiences," *The Quarterly Journal of Economics*, October 2015, *131* (1), 53–87.

Marcet, Albert and Thomas J. Sargent, "Convergence of Least Squares Learning Mechanisms in Self-referential Linear Stochastic Models," *Journal of Economic Theory*, August 1989, *48* (2), 337–368.

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis, "Human-level control through deep reinforcement learning," *Nature*, Feb 2015, *518* (7540), 529–533.

Moll, Benjamin, "The Trouble with Rational Expectations in Heterogeneous Agent Models: A Challenge for Macroeconomics," 2025.

Murphy, Kevin, "Reinforcement Learning: An Overview," 2025.

Ni, Tianwei, Benjamin Eysenbach, and Ruslan Salakhutdinov, "Recurrent Model-Free RL Can Be a Strong Baseline for Many POMDPs," 2022.

Niv, Yael, "Reinforcement Learning in the Brain," *Journal of Mathematical Psychology*, 2009, *53* (3), 139–154. Special Issue: Dynamic Decision Making.

OpenAI, "OpenAI o1 System Card," 2024.

Payne, Jonathan, Adam Rebei, and Yucheng Yang, "Deep Learning for Search and Matching Models," 2024.

Quadrini, Vincenzo and José-Víctor Ríos-Rull, "Inequality in Macroeconomics," *Handbook of Income Distribution*, 2015, *2*, 1229 – 1302.

Rafailov, Rafael, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn, "Direct Preference Optimization: Your Language Model is Secretly a Reward Model," 2024.

Raissi, M., P. Perdikaris, and G.E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, 2019, *378*, 686–707.

Robbins, Herbert and Sutton Monro, "A Stochastic Approximation Method," *The Annals of Mathematical Statistics*, 1951, *22* (3), 400–407.

Rotemberg, Julio J, "Sticky prices in the United States," *Journal of political economy*, 1982, *90* (6), 1187–1211.

Sargent, Thomas J., "Equilibrium with Signal Extraction from Endogenous Variables," *Journal of Economic Dynamics and Control*, 1991, *15* (2), 245–273.

_ , *The Conquest of American Inflation*, Princeton University Press, 1999.

_ , "HAOK and HANK Models," *Unpublished manuscript*, 2023. Available at http://www.tomsargent.com/research/HAOK_HANK.pdf.

_ **and John Stachurski**, "Quantitative Economics with JAX," 2025. Available at https://jax.quantecon.org/intro.html.

Schaab, Andreas, "Micro and Macro Uncertainty," Working Paper, UC Berkeley 2020.

Silver, David, *Introduction to Reinforcement Learning*, DeepMind x UCL, Video recordings available at https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ, 2015.

_ , **Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis**, "Mastering the game of Go with deep neural networks and tree search," *Nature*, Jan 2016, *529* (7587), 484–489.

_ , **Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis**, "Mastering the Game of Go without Human Knowledge," *Nature*, Oct 2017, *550* (7676), 354–359.

Storesletten, Kjetil, Chris Telmer, and Amir Yaron, "Asset Pricing with Idiosyncratic Risk and Overlapping Generations," *Review of Economic Dynamics*, October 2007, *10* (4), 519–548.

Sutton, Richard S. and Andrew G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 2018.

Wibault, Clarisse, Sebastian Towers, Tiphaine Wibault, Juan Duque, Johannes Forkel, George Whittle, Andreas Schaab, Yucheng Yang, Chiyuan Wang, Michael Osborne, Benjamin Moll, and Jakob Foerster, "Recurrent Structural Policy Gradient for Partially Observable Mean Field Games," 2026.

Wu, Zida, Mathieu Lauriere, Matthieu Geist, Olivier Pietquin, and Ankur Mehta, "Population-aware Online Mirror Descent for Mean-Field Games with Common Noise by Deep Reinforcement Learning," 2025.

Xu, Ruitu, Yifei Min, Tianhao Wang, Michael I. Jordan, Zhaoran Wang, and Zhuoran Yang, "Finding Regularized Competitive Equilibria of Heterogeneous Agent Macroeconomic Models via Reinforcement Learning," in Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, eds., *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, Vol. 206 of *Proceedings of Machine Learning Research* PMLR 25–27 Apr 2023, pp. 375–407.

Yang, Yaodong, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang, "Mean Field Multi-Agent Reinforcement Learning," in Jennifer Dy and Andreas Krause, eds., *Proceedings of the 35th International Conference on Machine Learning*, Vol. 80 of *Proceedings of Machine Learning Research* PMLR 10–15 Jul 2018, pp. 5571–5580.

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

Zhao, Shiyu, *Mathematical Foundations of Reinforcement Learning*, Springer Nature, available at https://github.com/MathFoundationRL/Book-Mathematical-Foundation-of-Reinforcement-Learning, 2025.

Online Appendix

A Model Calibration and Hyperparameters

This appendix presents additional model, calibration and implementation details for the applications we solve in Section 4 using our SRL approach. We start with the Huggett (1993) application in Appendix A.1, move to the Krusell and Smith (1998) application in Appendix A.2, and conclude with the HANK application in Appendix A.3.

A.1 Appendix for Huggett Application

We summarize the calibration we use in Section 4.1 in Table 2. Table 3 summarizes the hyperparameters used to solve the Huggett model with our SRL algorithm. We briefly discuss the main choices and their rationale.

Partial equilibrium specification. In the general equilibrium Huggett model, the interest rate is a complicated function of the aggregate state and the cross-sectional distribution. It is not Markov. For the partial equilibrium (PE) exercise discussed in Section 4.1, we instead take as given an exogenous Markov law of motion for the interest rate $r_t = 1/q_{t-1} - 1$ and let households solve their individual problem taking as given this process.

We model the interest rate as a mean-reverting process with a square-root volatility term. In continuous time, this is analogous to a Cox-Ingersoll-Ross (CIR) or "Feller square-root" process, which ensures positivity of the interest rate. In discrete time, our PE price process is specified as

$$r_{t+1} = (1-\rho_r)\bar{r} + \rho_r r_t + \nu_r \sqrt{\max\{r_t, 0\}} \cdot \varepsilon_{r,t} \qquad \text{where} \qquad \varepsilon_{r,t} \sim \mathcal{N}(0,1)$$

where $\bar{r}$ is the long-run mean level of the interest rate, ρ_r its autocorrelation, and ν_r the innovation volatility. The parameter values we use are reported in Table 2 and are chosen so that the unconditional distribution of interest rates as well as the implied aggregate bond holdings in PE are broadly consistent with general equilibrium in the Huggett calibration.

The idiosyncratic income process y in PE is the same as in the GE model, a three-point discretization of a log AR(1) with persistence ρ_y and volatility ν_y, as reported in Table 2. Thus, the only difference between PE and GE is that in PE the household takes r_t as an exogenous Markov process, while in GE it is determined endogenously from bond market clearing.

For the numerical implementation, we discretize the PE interest rate process on a grid described in Table 3. This grid is constructed using the CIR discretization method in Farmer and Toda (2017), which is designed for square-root processes and preserves positivity. Together with the income grid for y, this yields a fully specified PE environment in which we can solve

Parameter	Description	Value
β	Discount factor	0.96
σ	Coefficient of relative risk aversion	2
ρ_y	Autocorrelation of labor income	0.6
ν_y	Variance parameter of labor income	0.2
ρ_z	Persistence of AR(1) for z_t (log TFP)	0.9
ν_z	Volatility of AR(1) for z_t (log TFP)	0.02
B	Total bond supply	0
$\underline{b}$	Borrowing constraint	-1
$\bar{r}$	Mean interest rate (PE)	0.038
ρ_r	Autocorrelation of interest rate (PE)	0.8
ν_r	Volatility of interest rate (PE)	0.02

Table 2: Huggett model calibration

the household problem using both our SPG algorithm and a conventional VFI method, as described in the main text.

Discretization. The individual state (b, y) consists of bond holdings b and idiosyncratic income y. We discretize bonds on a one-dimensional grid with $n_b = 200$ points and an upper bound $b^{\texttt{max}} = 50$. The income process y takes $n_y = 3$ possible values. On the aggregate side, the two key state variables are the bond price q_t and aggregate income z_t. We approximate q_t on a grid with $n_q = 20$ points covering the interval $[q_L, q_H] = [0.95, 0.99]$. This range is chosen to comfortably contain all equilibrium interest rate realizations observed in our simulations while avoiding an unnecessarily large grid. Aggregate productivity z_t is discretized on a grid with $n_z = 50$ points using a standard Tauchen procedure. These choices strike a balance between accuracy and computational cost. They are fine enough to capture the relevant curvature in individual policies and the dependence of prices on the aggregate state.

Simulation horizon and truncation. We approximate lifetime utility by truncating the infinite sum in (15) at a finite horizon $T_{\texttt{trunc}}$. We choose $T_{\texttt{trunc}}$ to ensure that the tail of the discounted utility is negligible relative to a user-specified tolerance level $\epsilon_{\texttt{trunc}}$,

$$T_{\texttt{trunc}} = \min\left\{T \,:\, \beta^T < \epsilon_{\texttt{trunc}}\right\}.$$

In the baseline Huggett experiment, we use $\epsilon_{\texttt{trunc}} = 10^{-3}$ and obtain $T_{\texttt{trunc}} = 170$, i.e. the contribution of periods beyond $T_{\texttt{trunc}}$ is bounded by 10^{-3} in present-value terms. To avoid numerical issues when wealth is very low, we also impose a minimal consumption floor $c_{\texttt{min}} = 10^{-3}$. This has no discernible effect on the economic results but prevents the utility function from being evaluated at (or extremely close to) zero.

Training schedule and learning rate. We train the SPG algorithm with an exponentially decaying learning rate. Let $lr_{\texttt{ini}}$ denote the initial learning rate and $lr_{\texttt{decay}} \in (0, 1)$ the decay

Parameter	Description	Value
n_b	Number of b grid points	200
$b^{\max}$	Upper bound of b grid	50
n_y	Number of y grid points	3
n_q	Number of q grid points	20
r_L	Lower bound of r grid	0.01
r_H	Upper bound of r grid	0.06
n_z	Number of z grid points	50
$c_{\min}$	Minimum consumption	10^{-3}
T_{trunc}	Truncation horizon for simulations	170
ϵ_{trunc}	Truncation threshold	10^{-3}
N_{epoch}	Maximum number of parameter updates	1000
$N_{\text{warm-up}}$	Number of warm-up epochs	50
lr_{ini}	Initial learning rate	10^{-3}
lr_{decay}	Learning rate decay rate	0.5
lr_{sche}	Learning-rate scheduler	exponential
N_{sample}	Batch size (trajectories per update)	128
$\epsilon_{\text{converge}}$	Convergence threshold	1×10^{-4}

Table 3: Hyperparameters for solving the Huggett model

factor. The learning rate at iteration t is given by

$$lr_t = lr_{\text{ini}} \cdot lr_{\text{decay}}^{t'}, \qquad \text{where} \qquad t' = \frac{\max\{t - N_{\text{warm-up}}, 0\}}{N_{\text{epoch}} - N_{\text{warm-up}}},$$

so that lr_t is held constant during an initial "warm-up" phase of length $N_{\text{warm-up}}$ and then decays smoothly to a lower value by the final epoch N_{epoch}. In the Huggett application, we set $N_{\text{epoch}} = 1000$, $N_{\text{warm-up}} = 50$, $lr_{\text{ini}} = 10^{-3}$, and $lr_{\text{decay}} = 0.5$, and we use an exponential scheduler (denoted by lr_{sche} in Table 3). We declare convergence when the change in the policy parameters across epochs falls below the threshold $\epsilon_{\text{converge}} = 1 \times 10^{-4}$.

Sampling, batching, and memory constraints. Due to GPU memory constraints, we do not use all simulated data to update the policy in each iteration. Instead, we sample data in mini-batches. In each update, the effective data size is $N_{\text{sample}} \times N_{\text{update}}$, where N_{sample} denotes the number of simulated trajectories per batch (we set $N_{\text{sample}} = 128$ in the baseline Huggett experiment) and N_{update} the number of time steps used from each trajectory for the gradient update. This mini-batching keeps memory requirements manageable while preserving enough variation in the data to obtain stable gradient estimates.

Initialization and warm-up. We initialize the policy as described in Footnote **??** to guarantee that the initial aggregate savings schedule is at least weakly responsive to the interest rate. The training process is then split into two phases. During the warm-up phase of length $N_{\text{warm-up}}$, we fix the cross-sectional distribution of agents at some simple initial guess $\boldsymbol{g}_0$ and do not update it. In this phase, the sole objective is to move the policy away from its crude initial guess and toward a reasonable neighborhood of the eventual solution. Keeping $\boldsymbol{g}$ fixed prevents the

Parameter	Description	Value
β	Discount factor	0.95
σ	Utility parameter	3
$\underline{b}$	Borrowing constraint	0
ρ_y	Autocorrelation of idiosyncratic shock	0.6
ν_y	Volatility of idiosyncratic shock	0.2
a	Capital share	0.36
δ	Capital depreciation rate	0.08
ρ_z	Persistence of AR(1) for z_t (log TFP)	0.9
ν_z	Volatility of AR(1) for z_t (log TFP)	0.03

Table 4: Krusell–Smith model calibration

badly informed initial policy from "polluting" the distribution.

After warm-up, we switch to an adaptive phase in which the distribution is updated endogenously, and which may last up to $N_{\texttt{epoch}} - N_{\texttt{warm-up}}$ epochs (though convergence typically occurs earlier). We use the simulated distribution implied by the most recent policy as the initial distribution for each trajectory. In other words, after warm-up, each new batch of trajectories starts from a cross-section that is itself an equilibrium object. This iterative updating of the initial distribution ensures that the policy is trained on data drawn from its own induced stationary distribution, which is important for the accuracy of the final solution.

A.2 Appendix for Krusell-Smith Application

Calibration. Table 4 summarizes the calibration for the Krusell-Smith model used in Section 4.2. We use a discount factor of $\beta = 0.95$. The utility function is CRRA with a coefficient of relative risk aversion $\sigma = 3$, and the borrowing constraint is set at $\underline{b} = 0$.

The idiosyncratic income process is modeled as a log AR(1) with autocorrelation $\rho_y = 0.6$ and innovation volatility $\nu_y = 0.2$, the same specification as in the Huggett model. On the production side, we follow the standard Krusell-Smith calibration: the capital share is $\alpha = 0.36$, the depreciation rate is $\delta = 0.08$, and aggregate productivity z_t follows a log AR(1) with persistence $\rho_z = 0.9$ and volatility $\nu_z = 0.03$.

Discretization and grids. Table 5 reports the hyperparameters used for the SPG solution of the Krusell-Smith model. The individual capital state b is discretized on a grid with $n_b = 200$ points and an upper bound $b^{\texttt{max}} = 100$, which is higher than in the Huggett experiment. The larger upper bound reflects the fact that, in Krusell-Smith, agents accumulate capital rather than unproductive bonds, and the equilibrium wealth distribution is more dispersed. The idiosyncratic income state y again takes $n_y = 3$ values.

The aggregate price vector consists of the interest rate r_t and the real wage w_t. We approximate the price space on two separate grids: the interest rate is discretized with $n_r = 30$ points on $[r_L, r_H] = [0.02, 0.07]$, and the wage with $n_w = 50$ points on $[w_L, w_H] = [0.9, 1.5]$. These ranges comfortably contain the realizations observed in our simulations and allow the policy to respond flexibly to movements in both prices without requiring a prohibitive number of

Parameter	Description	Value
n_b	Number of b grid points	200
$b^{\texttt{max}}$	Upper bound of b grid	100
n_y	Number of y grid points	3
n_r	Number of r grid points	30
r_L	Lower bound of r grid	0.02
r_H	Upper bound of r grid	0.07
n_w	Number of p_2 grid points	50
w_L	Lower bound of w grid	0.9
w_H	Upper bound of w grid	1.5
$c_{\texttt{min}}$	Minimum consumption	10^{-3}
$c_{\texttt{init}}$	Initial guess for consumption share	0.5
$T_{\texttt{trunc}}$	Truncation horizon for simulations	135
$\epsilon_{\texttt{trunc}}$	Truncation threshold	10^{-3}
$N_{\texttt{epoch}}$	Maximum number of parameter updates	1000
$N_{\texttt{warm-up}}$	Number of warm-up epochs	50
$lr_{\texttt{ini}}$	Initial learning rate	5×10^{-4}
$lr_{\texttt{decay}}$	Learning-rate decay rate	0.5
$lr_{\texttt{sche}}$	Learning-rate scheduler	exponential
$N_{\texttt{sample}}$	Batch size (trajectories per update)	128
$\epsilon_{\text{converge}}$	Convergence threshold	1×10^{-4}

Table 5: Hyperparameters for solving the Krusell–Smith model

grid points.

Simulation horizon and truncation. As in the Huggett case, lifetime utility is computed by truncating the infinite sum at a finite horizon $T_{\texttt{trunc}}$. For Krusell-Smith we set $T_{\texttt{trunc}} = 135$ and use a truncation tolerance $\epsilon_{\texttt{trunc}} = 10^{-3}$, i.e.

$$\beta^{T_{\texttt{trunc}}} < \epsilon_{\texttt{trunc}},$$

so that the tail of the discounted utility stream is negligible at the scale of our numerical accuracy. We also impose a minimal consumption level $c_{\texttt{min}} = 10^{-3}$ to avoid evaluating utility at zero or extremely small consumption levels.

Training schedule and convergence. We use the same general training structure as in the Huggett exercise but with slightly different numerical values. The maximum number of epochs is $N_{\texttt{epoch}} = 1000$, with $N_{\texttt{warm-up}} = 50$ warm-up epochs during which the learning rate is kept constant and the initial distribution is fixed. The learning rate starts at $lr_{\texttt{ini}} = 5 \times 10^{-4}$ and decays exponentially at rate $lr_{\texttt{decay}} = 0.5$ according to the scheduler denoted by $lr_{\texttt{sche}}$ in Table 5. As in the Huggett case, the decay is only activated after the warm-up phase. We declare convergence when the change in parameters between successive epochs falls below $\epsilon_{\text{converge}} = 1 \times 10^{-4}$; in practice, the algorithm typically converges well before hitting the hard cap of $N_{\texttt{epoch}}$.

Mini-batching is again used to handle memory constraints and stabilize gradient estimates.

Each update uses $N_{\texttt{sample}} = 128$ simulated trajectories, and a fixed number of time steps per trajectory, to form the stochastic gradient. This yields a total data size per update of $N_{\texttt{sample}} \times N_{\texttt{update}}$, which we choose to fully utilize the available GPU memory without inducing excessive variance in the gradient.

Initialization and warm-up. The warm-up logic mirrors that used in the Huggett application but is adapted to the two-price environment. During the first $N_{\texttt{warm-up}}$ epochs, all trajectories are initialized from a cross-sectional distribution that is held fixed, while the policy is being updated. After warm-up, we update the cross-sectional distribution based on the most recent policy and allow the learning rate to adjust. The initial conditions for each new set of trajectories are drawn from the simulated distribution generated by the most recent policy.

A.3 Appendix for HANK Application

Firm Policy Gradient Method for the Phillips Curve. As stated in the main text, we treat the firm problem (23) exactly as we treat the household problem and solve it using the same SPG method. To this end, we rewrite (23) in terms of firm j choosing the growth rate of its price

$$\Pi_{j,t} = \frac{P_{j,t} - P_{j,t-1}}{P_{j,t-1}}.$$

Note that, in equilibrium, we will have $\Pi_{j,t} = \Pi_t$ for all j. However, when firms choose future prices, they optimize $\{\Pi_{j,t}\}$ *taking as given* $\{\Pi_t\}$. Next note that only the ratio

$$\phi_{j,t} := \frac{P_{j,t}}{P_t}$$

shows up in firm j's problem (23). This ratio satisfies

$$\phi_{j,t} = \frac{1 + \Pi_{j,t}}{1 + \Pi_t} \phi_{j,t-1}$$

with $\phi_{j,-1} = 1$. We therefore write (23) as

$$J_{j,0} = \max_{\{\Pi_{j,t}\}} \mathbb{E}_0 \sum_{t=0}^{\infty} \Lambda_{0\to t} \left[\phi_{j,t}^{1-\varepsilon} - \frac{w_t}{z_t} \phi_{j,t}^{-\varepsilon} - \frac{\theta}{2} \Pi_{j,t}^2 \right] Y_t \quad \text{s.t.} \quad \phi_{j,t} = \frac{1 + \Pi_{j,t}}{1 + \Pi_t} \phi_{j,t-1}$$

with initial condition $\phi_{j,-1} = 1$ and with $\Pi_{j,t} = \Pi_t$ in equilibrium.

This reformulation in terms of inflation motivates the following SPG algorithm. Define the state space as (z, Y, w) and choose policy $\Pi_j(z, Y, w)$ to maximize:

$$J_j(z, Y, w) = \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t \left[\phi_{j,t}^{1-\varepsilon} - \frac{w_t}{z_t} \phi_{j,t}^{-\varepsilon} - \frac{\theta}{2} (\Pi_j(z_t, Y_t, w_t))^2 \right] Y_t \quad \text{s.t.} \quad \phi_{j,t} = \frac{1 + \Pi_j(z_t, Y_t, w_t)}{1 + \Pi_t} \phi_{j,t-1}$$

with initial condition $\phi_{j,-1} = 1$ and *taking as given* Π_t which equals $\Pi_j(z_t, Y_t, w)$ in equilibrium.

Note that we are here imposing the approximation that $\Lambda_{0\to t} \approx \beta^t$. This allows us to drop the bond price q as a state variable, i.e. without this approximation, the state space would be

Parameter	Description	Value
β	Discount factor	0.975
σ	Coefficient of relative risk aversion	1
η	Inverse of Frisch elasticity	1
ϕ	Coefficient of Taylor rule	1.5
θ	Price adjustment cost	100
ϵ	Elasticity of substitution	10
$\bar{R}$	Target for gross interest rate	1.025
ρ_z	Autocorrelation of aggregate TFP shock	0.9
ν_z	Volatility of aggregate TFP shock	0.07
ρ_ϵ	Autocorrelation of monetary policy shock	0.9
ν_ϵ	Volatility of monetary policy shock	0.002

Table 6: Calibration for the HANK model

(z, Y, w, q). Also that, in equilibrium, the relative price $\phi_{j,t} = 1$ for all t; hence, even though firms take into account the effect of their price-setting decisions on $\phi_{j,t}$ via the recursion in the second line, we do not carry ϕ as a state variable.

Calibration. Table 6 reports the calibration of the HANK model used in Section 4.3. One model period corresponds to a year, and the discount factor is set such that the annual real interest rate is in a plausible range; in the baseline we use a discount factor of 0.975. Preferences over consumption and labor are CRRA and separable, with coefficient of relative risk aversion equal to 1 and inverse Frisch elasticity of labor supply $\eta = 1$.

We follow a standard New Keynesian calibration. Price-setting firms face Rotemberg adjustment costs with parameter $\theta = 100$ and an elasticity of substitution across intermediate goods of $\varepsilon = 10$. Monetary policy follows the Taylor rule specified in the main text, with coefficient $\phi = 1.5$ on inflation and a gross steady-state real interest rate target $\bar{R} = 1.025$.

Aggregate risk is two-dimensional. TFP z_t follows an AR(1) process with persistence $\rho_z = 0.9$ and innovation volatility $\nu_z = 0.07$. The monetary policy shock ϵ_t is also AR(1) with the same persistence $\rho_\epsilon = 0.9$ and volatility $\nu_\epsilon = 0.002$. These values are summarized in Table 6 and are chosen so that both real and nominal variables display non-trivial but stable dynamics in the simulations.

Discretization and grids. Table 7 lists the hyperparameters and grid choices for our solution of the HANK model. The individual asset state b is discretized on a grid with $n_b = 200$ points and an upper bound $b^{\max} = 100$. The idiosyncratic income state y again takes $n_y = 3$ values, using the same discretization as in the Huggett and Krusell-Smith applications for ease of comparison.

The aggregate state combines a two-dimensional price vector (q_t, w_t) and output Y_t with the two exogenous shocks z_t and ϵ_t. We approximate the nominal bond price on a grid with $n_q = 40$ points over the interval $[q_L, q_H] = [0.96, 1.00]$. The real wage is discretized with $n_w = 40$ points on $[w_L, w_H] = [0.6, 1.0]$. The real output is discretized with $n_Y = 10$ points on $[Y_L, Y_H] = [0.9, 1.1]$. The ranges are chosen to cover comfortably the realizations observed in

Parameter	Description	Value
n_b	Number of b grid points	200
$b^{\max}$	Upper bound of b grid	100
y	Number of y grid points	3
n_q	Number of r grid points	40
q_L	Lower bound of r grid	0.96
q_H	Upper bound of r grid	1.00
n_w	Number of w grid points	40
w_L	Lower bound of w grid	0.6
w_H	Upper bound of w grid	1.0
n_Y	Number of Y grid points	10
Y_L	Lower bound of w grid	0.9
Y_H	Upper bound of w grid	1.1
n_z	Number of z grid points	50
n_ϵ	Number of e grid points	50
$c_{\texttt{min}}$	Minimum consumption	10^{-3}
$c_{\texttt{init}}$	Initial guess of consumption share	0.5
$n_{\texttt{init}}$	Initial guess of labor supply	1.5
$\Pi_{\texttt{init}}$	Initial guess of inflation	0
$T_{\texttt{trunc}}$	Truncation horizon for simulations	273
$\epsilon_{\texttt{trunc}}$	Truncation threshold	10^{-3}
$N_{\texttt{epoch}}$	Maximum number of parameter updates	1000
$N_{\texttt{warm-up}}$	Number of warm-up epochs	50
$lr_{\texttt{ini}}$	Initial learning rate	1×10^{-3}
$lr_{\texttt{decay}}$	Learning-rate decay rate	0.5
$N_{\texttt{sample}}$	Baseline sampling size	256
$\epsilon_{\text{converge}}$	Convergence threshold	5×10^{-5}

Table 7: Hyperparameters for solving the HANK model

equilibrium simulations, while keeping the price grids small enough for efficient training.

For the aggregate shocks, we use $n_z = 50$ grid points for log TFP z_t and $n_\epsilon = 50$ points for the monetary shock ϵ_t. Both grids are obtained by discretizing the respective AR(1) processes with a standard Tauchen method. The relatively fine grids for (z_t, ϵ_t) help the algorithm capture the interaction between real and nominal disturbances in the HANK model.

Simulation horizon, truncation, and initial guesses. The HANK model combines persistence in both TFP and monetary shocks with sluggish price adjustment. To accommodate this accurately, we use a longer truncation horizon $T_{\texttt{trunc}} = 273$ periods and a truncation tolerance $\epsilon_{\texttt{trunc}} = 10^{-3}$, which implies

$$\beta^{T_{\texttt{trunc}}} < \epsilon_{\texttt{trunc}},$$

so that the contribution of periods beyond $T_{\texttt{trunc}}$ is negligible at the scale of our numerical accuracy. As in the other applications, we impose a minimal consumption level $c_{\texttt{min}} = 10^{-3}$ to avoid evaluating the utility function at zero consumption.

The HANK environment includes both consumption-saving and labor-supply decisions, as well as firm price-setting. We initialize these policy functions using simple guess rules;

for $c_{\texttt{init}}$, we set an initial constant consumption share of total cash-in-hand, a constant $n_{\texttt{init}}$ sets an initial level of hours worked, and $\Pi_{\texttt{init}} = 0$ initializes inflation. These initial values are deliberately crude and their sole purpose is to place the policy in a reasonable region of the parameter space before learning from simulated data. In practice, the final solution is insensitive to these initial guesses once training has converged.

Training and convergence. The remaining training hyperparameters follow the logic of the Huggett and Krusell-Smith applications. We use a baseline batch size of $N_{\texttt{sample}} = 256$ simulated trajectories per update, chosen to saturate GPU memory without generating excessive variance in the policy gradient. The convergence threshold is set at $\epsilon_{\text{converge}} = 5 \times 10^{-5}$ for the HANK model, which reflects the greater complexity of the joint household-firm problem and the fact that small changes in the policy parameters can translate into larger differences in aggregate dynamics.

Other aspects of the training schedule — notably the total number of epochs, the length of the warm-up phase, and the learning-rate schedule — are chosen in line with the Krusell-Smith specification discussed in Appendix A.2 and are not repeated here. In practice, the HANK model converges somewhat more slowly than Huggett and Krusell-Smith but still within a few minutes on a single GPU, as reported in Table 1 in the main text.