Bounded Rationality as Limited Optimization: Stochastic Gradient Descent Agents in Macroeconomic Models*

Pablo A. Guerrón Quintana
Boston College†

Version: March 2026.
The latest version of this paper is available here

Abstract

This paper proposes a novel equilibrium concept in which agents are fully rational in preferences and constraints but computationally bounded, with decision rules parameterized and improved via stochastic gradient methods applied to simulated expected utility. In a standard RBC model with GHH preferences, we define a *Stochastic Gradient Descent equilibrium* as a fixed point (in expectation) of the stochastic-gradient update rule together with market clearing and firm optimality. The fixed-point condition requires that the expected gradient of the household's truncated utility function vanishes at the equilibrium decision rules, so that the agent has no incentive, on average, to revise her policy further. We establish conditions under which the SGD equilibrium converges to the rational expectations solution in the sequential limit as the planning horizon and training intensity increase without bound. Away from this limit, tighter computational budgets generate systematic, state-dependent deviations summarized by an intratemporal labor wedge and an intertemporal capital wedge, whereas large horizon planning and extensive training deliver policies close to the rational expectations benchmark. These wedges are endogenous and time-varying, providing a structural bridge to the business-cycle accounting framework of Chari et al. (2007) without introducing non-productivity shocks or real frictions.

1 Introduction

Modern dynamic stochastic general equilibrium (DSGE) models typically assume that households and firms solve intertemporal optimization problems exactly. In canonical real business cycle (RBC) environments, this assumption is operationalized by characterizing equilibrium allocations through Lagrangian maximization problems that result in Euler equations and intratemporal first-order conditions, and by computing policy functions using perturbation, projection, or dynamic programming methods. While analytically powerful, the rational-expectations benchmark embeds a strong and

*I thank Ryan Chahrour and Raf Wouters for detailed comments and suggestions, and Chris Naubert, Jaromir Nosal, Robert Ulbricht, and Rosen Valchev for insightful discussions. All errors are mine.
†email: pguerron@gmail.com

implicit premise: agents possess the computational capacity to solve high-dimensional, nonlinear dynamic programs at negligible cost.

This paper proposes a novel equilibrium concept that preserves the primitives of the DSGE problem — preferences, production function, feasibility, and market clearing — while relaxing the assumption of unlimited computation. Households are modeled as *stochastic-gradient-descent (SGD) agents*. The household does not solve Euler equations nor a Bellman equation. Instead, she selects *parameterized decision rules* and improves them through repeated gradient-based updates of expected utility computed along simulated equilibrium paths. In this sense, bounded rationality takes the form of *limited optimization* rather than incorrect beliefs or incomplete information, as the agent knows the economic environment, but approximates the optimal policy subject to a finite planning horizon and a finite computational budget.

A critical step in this setup is the change in object being chosen. Rather than choosing sequences $\{c_t, n_t, k_{t+1}\}_{t\geq 0}$ that satisfy optimality conditions, the household chooses parameters $\theta \in \mathbb{R}^d$ indexing a family of measurable policy rules. In the RBC–GHH model, the policy maps the state (k_t, z_t) into labor and a savings decision,

$$n_t = \pi^n_\theta(k_t, z_t), \qquad s_t = \pi^s_\theta(k_t, z_t),$$

with feasibility imposed by construction:

$$R_t = (1-\delta)k_t + z_t k_t^\alpha n_t^{1-\alpha}, \qquad c_t = (1-s_t)R_t, \qquad k_{t+1} = s_t R_t.$$

Here, R_t denotes available resources at time t. Labor is constrained to $(0, \bar{n})$ and the savings share to $(s_{\min}, s_{\max})$. Consequently, every θ induces an implementable allocation, and the optimization problem is "moved" from enforcing equilibrium conditions to improving a policy within a feasible policy class.

The agent evaluates a truncated discounted objective over a rollout horizon T,

$$J_T(\theta) = \mathbb{E}\left[\sum_{t=0}^{T-1} \beta^t u(c_t(\theta), n_t(\theta))\right],$$

and updates the policy parameters using Monte Carlo estimates of $\nabla_\theta J_T(\theta)$. Gradients are computed *pathwise* by differentiating through the simulated transition equations ("backpropagation through time"), so that the update internalizes how today's decisions affect future states through capital accumulation. Bounded rationality arises from three sources that are explicit in the computational problem: (i) truncation of the objective at T; (ii) Monte Carlo noise from a finite number of simulated paths in each update, number of batches B; and (iii) limited optimization depth due to a finite number of parameter updates (epochs), denoted E.

The SGD framework delivers a sharp and economically interpretable link between computational constraints and equilibrium outcomes. In the rational-expectations benchmark, policy functions solve the Euler equation for capital accumulation and the intratemporal labor condition at every

state:

$$u_c(c_t, n_t) = \beta\, \mathbb{E}_t\Big[u_c(c_{t+1}, n_{t+1}) r^k_{t+1}\Big], \qquad \text{and} \qquad \psi n_t^{\varphi} = w_t,$$

with r^k_{t+1} the return on capital and w_t the marginal product of labor. In the SGD equilibrium, these conditions need not hold exactly. Instead, deviations arise endogenously because the policy parameters solve a *finite* optimization problem and are updated using *noisy* gradients. We therefore interpret Euler residuals and intratemporal labor residuals as *computation wedges*: equilibrium distortions generated by limited optimization rather than by preference shocks, information frictions, or ad hoc exogenous wedges.

This interpretation connects the SGD-agent approach to the business-cycle accounting framework of Chari et al. (2007). In their accounting decomposition, measured wedges—including a labor wedge that distorts the static labor-leisure condition—summarize deviations between the data and a frictionless RBC benchmark. In our model, an analogous labor wedge emerges as

$$\omega^n_t \equiv \frac{w_t}{\psi n_t^{\varphi}} - 1,$$

and it can be computed directly from the learned policy. The wedge is state-dependent and time-varying because optimization errors depend on where the economy is in the state space and on the recent sequence of shocks. Importantly, the wedge shrinks systematically as computational constraints are relaxed: increasing T and E pushes the policy toward the rational-expectations benchmark and drives ω^n_t toward zero. This provides a structural channel through which the accounting wedges of Chari et al. (2007) can be interpreted as reduced-form manifestations of bounded optimization.

We establish conditions under which the SGD agent converges to the rational-expectations solution as computational resources increase. The assumptions fall into four groups. The first group imposes regularity on the policy class and the simulated economy: the parameter space is compact, policy outputs are bounded away from the feasibility constraints by smooth transformations that ensure the GHH composite utility is well-defined, and the state process remains in a compact invariant set under shock truncation. The second group governs the optimization: the objectives are continuous in parameters, the SGD agent achieves near-optimal objective value for any prescribed tolerance as the number of epochs and simulation paths grow, and the finite-horizon objective is well-separated so that near-optimality in objective value implies near-optimality in policy space. The third group characterizes the infinite-horizon target: the policy class contains a unique best policy under the infinite-horizon objective, which coincides with the RE solution if the class is sufficiently rich. The final condition requires that backpropagation through the simulated RBC transitions yields asymptotically unbiased gradient estimates. Under these conditions, Proposition 1 establishes that, in the sequential limit in which epochs and simulation paths grow for each fixed horizon and the horizon then grows to infinity, the policy induced by the SGD agent converges to the RE policy uniformly on the invariant state space, with an explicit truncation error of order $O(\beta^T)$ that provides direct quantitative guidance for the choice of planning horizon.

Away from the theoretical limit, the quantitative analysis studies how learned policy functions vary with computational budgets and links these changes to economically interpretable wedges. When computation is ample, the learned labor and savings rules closely track the rational-expectations benchmark near the steady state and generate standard RBC dynamics. When computation is constrained, distortions emerge in systematic ways: shorter rollout horizons T make the investment payoff less salient because a larger share of the return to capital lies beyond the planning window, while fewer training epochs E leave the policy imperfectly optimized and amplify state-dependent deviations from optimality. We summarize these deviations using wedges: intertemporal Euler residuals capture under- or over-saving incentives, and an intratemporal labor wedge captures misoptimization of the contemporaneous tradeoff between wages and the marginal disutility of labor. This decomposition clarifies the distinct roles of T and E — T governs *what* objective the agent effectively optimizes, whereas E governs *how well* it is optimized given approximation and sampling error — even though the two interact through the gradient attenuation inherent in backpropagation through time.

The SGD framework differs from black-box reinforcement learning (RL) approaches used in macroeconomic applications. In standard RL, the agent interacts with an environment by taking actions and receiving scalar rewards, learning a policy without explicit knowledge of preferences, constraints, or equilibrium conditions. Rather than treating the household as a reward-maximizing agent in an unknown environment, we exploit the differentiable structure of the DSGE model: feasibility is enforced by construction and policy updates follow from pathwise gradients of household utility along equilibrium paths. This distinction matters economically. Because gradients are computed by differentiating through the model's transition equations, the learning algorithm explicitly internalizes how current savings decisions affect future capital, output, and consumption — the same intertemporal channel that generates the Euler equation in the rational-expectations benchmark. As a result, deviations from optimality can be decomposed into economically interpretable wedges rather than attributed to approximation error. This delivers transparent diagnostics — policy-function comparisons, Euler residuals, and intratemporal wedges — that have no natural counterpart in black-box RL.

The remainder of the paper is organized as follows. The next section discusses the related literature. Section 3 presents the RBC–GHH environment and defines the SGD-agent problem. Section 4 discusses the role of the rollout horizon and how truncated planning affects incentives. Section 5 states conditions under which the SGD solution converges to the RE solution. Section 6 presents quantitative results, including policy-function comparisons and the computation wedges that emerge under limited horizons and training. Some empirical implications are discussed in Section 7. The last section concludes.

2 Relation to the Literature

The SGD agent framework is related to, but distinct from, several strands of the macroeconomic literature.

Rational Expectations and Exact Optimization. In standard DSGE models, households are assumed to solve dynamic optimization problems exactly. The SGD framework relaxes this assumption by explicitly modeling the optimization process, allowing deviations from exact optimality to arise endogenously from limited computation.

Adaptive Learning. A large literature on adaptive learning departs from rational expectations by assuming that agents form expectations using recursive forecasting rules—typically least-squares updating of perceived laws of motion—while still solving the underlying optimization problem conditional on those beliefs (Evans and Honkapohja, 2001; Marcet and Sargent, 1989). In these environments, departures from the rational-expectations benchmark reflect *expectational errors*: agents misperceive the mapping from states to future variables and gradually update their forecasting rules as new data arrive. Our SGD-agent framework is complementary but distinct. We hold the economic environment fixed and known, and we impose feasibility by construction; bounded rationality instead arises from *imperfect optimization* of a known objective due to finite planning depth and finite, noisy gradient-based updates. Consequently, the wedges we measure are optimality-condition residuals generated by limited computation, rather than residuals induced by incorrect beliefs about equilibrium laws of motion.

Rational Inattention and Costly Information. Rational inattention (RI) models formalize cognitive limits as constraints on *information acquisition and processing*: agents optimally choose what to pay attention to subject to an information-capacity (or entropy) constraint, so that decision rules are distorted because relevant state information is only imperfectly absorbed (Sims, 2003). This perspective has been widely applied in macroeconomics to generate sluggish adjustment and state-dependent responses even when preferences and technology are standard; for instance, Woodford (2009) embeds a Sims-style RI constraint into a New Keynesian pricing environment to study how limited attention alters state-dependent pricing decisions. Recent surveys emphasize that RI delivers a disciplined notion of bounded rationality grounded in optimal information choice, with clear implications for nominal and real dynamics (Maćkowiak et al., 2023). Our SGD-agent framework is complementary: rather than constraining *information*, we hold the environment and constraints fixed and constrain the agent's *optimization/computation* by restricting planning depth and the number/quality of gradient-based updates. Both approaches yield interpretable wedges, but the primitive differs—information frictions in RI versus limited optimization capacity in the SGD equilibrium.

A key motivation for our truncated-horizon SGD-agent formulation is the finite-horizon planning approach advocated by Woodford (2019). Woodford departs from the standard rational-

expectations paradigm in which agents compute complete state-contingent plans over an infinite horizon—typically characterized by Euler equations and solved using projection methods or dynamic programming—and instead models decision makers as carrying out *forward planning only up to a finite depth*. Formally, agents optimize over a finite look-ahead horizon and then close the problem with a simple continuation object (a terminal continuation value that is held fixed, rather than solved for as part of a global fixed point). This perspective delivers a disciplined notion of bounded rationality that preserves forward-looking behavior while explicitly limiting intertemporal reasoning. Our SGD-agent setup implements the same discipline in a computationally transparent way: we fix a planning cap T and define the objective as discounted utility up to $T-1$, with a normalized terminal continuation value (set to zero) beyond that point. Rather than solving Euler equations (or minimizing Euler residuals on a grid), the agent chooses parameters θ indexing decision rules and we update θ directly to maximize truncated lifetime utility under the induced law of motion. Thus, the cap T plays the role of Woodford's finite planning horizon, while the terminal normalization provides an explicit and tractable closure that avoids imposing equilibrium optimality conditions period-by-period.

Reinforcement Learning and Deep RL. Recent work applies RL and DRL methods to DSGE models, treating agents as black boxes that learn from realized rewards. While powerful, these approaches abstract from the economic structure of the household problem. The SGD agent, by contrast, optimizes the household's utility function directly, enforces feasibility by construction, and admits a transparent connection to Euler equation residuals and wedges.

An exception is Yang et al. (2025) who proposes a new global solution method for heterogeneous-agent models with aggregate risk that sidesteps the Master equation by replacing the high-dimensional cross-sectional distribution state with low-dimensional equilibrium prices as the objects agents condition on. Agents form expectations about prices directly from simulated equilibrium paths and update their policy functions using a structural policy gradient (SPG) method that differentiates through agents' known micro-level dynamics while treating the equilibrium price mapping as part of the environment. The approach solves for a restricted perceptions equilibrium (RPE) in which policies depend only on current prices (or a short price history), avoiding an inner–outer fixed point loop over perceived laws of motion as in Krusell–Smith. In computational experiments, the authors solve the Krusell–Smith model, a Huggett model with aggregate shocks, and a HANK model with a forward-looking Phillips curve quickly on modern hardware, reporting global solutions "within minutes."

3 Economic Environment

We consider a standard real business cycle (RBC) model with endogenous labor supply and Greenwood–Hercowitz–Huffman (GHH) preferences. Time is discrete and one period in the model corresponds

to one quarter in the data. Output is produced using a Cobb–Douglas technology,

$$y_t = z_t k_t^{\alpha} n_t^{1-\alpha}. \tag{1}$$

Here, k_t denotes capital, n_t labor, and z_t total factor productivity, which is assumed to follow the autoregressive process

$$\log z_t = \rho \log z_{t-1} + \sigma \epsilon_t, \quad \epsilon_t \sim N(0,1). \tag{2}$$

Thus, productivity is a stochastic state variable, and household decisions at date t are functions of both endogenous capital k_t and realized productivity z_t.

The representative household derives utility from consumption and labor according to

$$u(c_t, n_t) = \frac{\left(c_t - \psi \frac{n_t^{1+\varphi}}{1+\varphi}\right)^{1-\gamma}}{1-\gamma}, \tag{3}$$

and faces the resource constraint

$$c_t + k_{t+1} = (1-\delta)k_t + y_t. \tag{4}$$

Firms are competitive and rent capital and labor at their marginal products.

3.1 The Household Problem and the SGD Agent

Now, we describe the structure behind the SGD agents using the household's framework described above. An important object in the model is the computational budget, which is defined by the agent's planning horizon T and the number of optimization steps (epochs, E) that the agent can use to improve the current policies.[1] Our characterization proceeds in multiple steps.

Policy Parameterization

A critical step in our setup is that, rather than characterizing optimal allocations by solving the household problem through Euler equations (as in perturbation/projection methods) or through dynamic programming (value function iteration), we parameterize the household's decision rules for labor and the savings rate and let the household *choose* the parameters of these rules. In particular, we posit measurable mappings from the state (k_t, z_t) to choices,

$$s_t = \pi_\theta^s(k_t, z_t), \qquad n_t = \pi_\theta^n(k_t, z_t), \tag{5}$$

where $\theta \in \mathbb{R}^d$ is a finite-dimensional parameter vector. The explicit dependence on z_t is central: the policy functions are state-contingent not only on accumulated capital, but also on the current

[1] Another critical step in the solution process is the number of stochastic simulations (batches) that the agent has access to compute expectations. For now, we abstract from this dimension. In the implementation below, these expectations are approximated by averaging over simulated paths of productivity innovations.

productivity realization that shifts wages, the rental rate, and feasible output. This change in perspective is substantive. Under Euler-equation based methods, one specifies a functional class for the policy rules and selects coefficients to make the equilibrium conditions hold (approximately) over a set of collocation nodes, typically by minimizing the Euler residual

$$u_c(c_t, n_t) \;-\; \beta\, \mathbb{E}_t[u_c(c_{t+1}, n_{t+1})\, R_{t+1}]$$

subject to feasibility. In contrast, in our approach θ is chosen to maximize the household's expected discounted utility under the law of motion induced by the pair of policy functions $(\pi^s_\theta, \pi^n_\theta)$.

To fix ideas, consider the resource constraint

$$c_t \;=\; z_t k_t^\alpha n_t^{1-\alpha} + (1-\delta)k_t - k_{t+1}.$$

A perturbation of θ changes current choices (c_t, n_t), which alters capital accumulation k_{t+1} and therefore the entire future state path $(k_{t+2}, k_{t+3}, \cdots)$; in turn, this affects all subsequent consumption–labor allocations and lifetime utility. Because the state also includes productivity, a perturbation to θ changes how the agent responds to each realized and future simulated sequence $\{z_t\}_{t\geq 0}$, so the continuation value changes through both endogenous capital dynamics and state-contingent responses to productivity shocks. Our algorithm exploits this intertemporal channel directly: it simulates the economy forward given $(\pi^s_\theta, \pi^n_\theta)$ and computes gradients of the objective with respect to θ by differentiating through the sequence of transition equations ("backpropagation in time"), thereby updating θ toward a policy that improves utility. In practice, the simulation step draws innovations $\{\epsilon_t\}$, constructs the implied productivity path from the AR(1) law of motion for z_t, and then evaluates decisions using the policy rules at each realized state (k_t, z_t). Importantly, feasibility and non-negativity are enforced by construction through smooth transformations, so optimization is carried out over policies that are implementable throughout the relevant region of the state space.

Neural Network Architecture

The policy functions π^s_θ and π^n_θ are each represented by a fully connected feedforward neural network with two hidden layers of 64 units and hyperbolic tangent (tanh) activations (Goodfellow et al., 2016). The input layer receives the two-dimensional state vector (k_t, z_t), which is standardized to have zero mean and unit variance using the ergodic moments of the state process prior to training. The output layer maps to the savings rate s_t and labor n_t through smooth, bounded transformations: a sigmoid function scaled to the interval $[\underline{s}, \overline{s}] \subset (0, 1)$ for the savings rate, and a softplus function bounded to $[\underline{n}, \overline{n}] \subset (0, \infty)$ for labor, ensuring that the positivity and feasibility requirements of the equilibrium concept are satisfied by construction for all inputs and all values of θ. The two networks share the same input layer but have independent weights, so the full parameter vector $\theta \in \mathbb{R}^d$ concatenates the weights and biases of both networks, with d determined by the width and depth of the architecture. Figure 1 shows a schematic representation of the neural network.

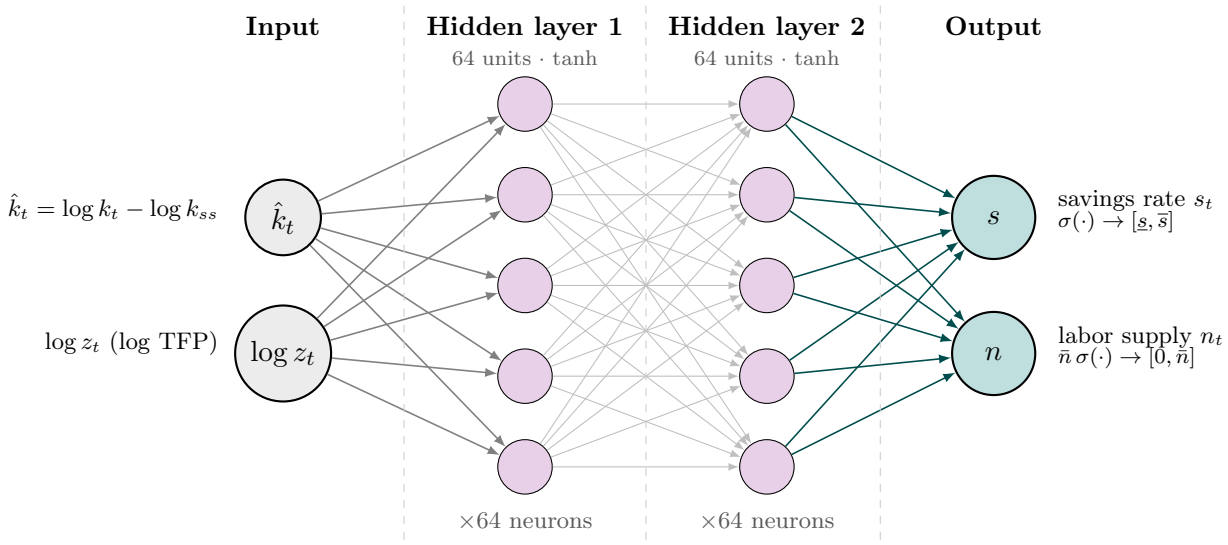


Figure 1: **Neural network architecture of the SGD agent's policy functions.** Each policy π^s_θ (savings rate) and π^n_θ (labor supply) is a fully connected feedforward network with two hidden layers of 64 units and tanh activations. The input layer receives the standardized state vector $(\hat{k}_t, \log z_t)$, where $\hat{k}_t = \log k_t - \log k_{ss}$ is log-deviation of capital from its steady-state value. The output layer maps to the savings rate s_t through a sigmoid transformation scaled to $[\underline{s}, \bar{s}]$, and to labor n_t through a sigmoid scaled to $[0, \bar{n}]$, ensuring feasibility by construction for all $\theta \in \Theta$. The two networks share the same architecture but have independent weight vectors; the full parameter vector $\theta \in \mathbb{R}^d$ concatenates the weights and biases of both networks. Five representative hidden units are shown per layer; each layer contains 64 units.

Objective and Learning

Given a candidate policy $\pi_\theta = (\pi^s_\theta, \pi^n_\theta)$, the household evaluates expected lifetime utility over simulated future paths,

$$J_T(\theta) = \mathbb{E}\left[\sum_{t=0}^{T-1} \beta^t u\big(c_t(\theta), n_t(\theta)\big)\right]. \tag{6}$$

The expectation is taken with respect to the sequence of productivity shocks $\{\epsilon_t\}_{t=0}^{T-1}$ (equivalently, the induced productivity path $\{z_t\}_{t=0}^{T-1}$), conditional on the initial state. Since this expectation is not evaluated analytically, we replace it with a Monte Carlo average over simulated productivity draws.

Specifically, for a batch of B simulated shock paths indexed by $b = 1, \ldots, B$, we compute

$$\widehat{J}_{T,B}(\theta) = \frac{1}{B}\sum_{b=1}^{B}\sum_{t=0}^{T-1} \beta^t \, u\Big(c_t^{(b)}(\theta), n_t^{(b)}(\theta)\Big), \tag{7}$$

where each path $\{c_t^{(b)}(\theta), n_t^{(b)}(\theta), k_t^{(b)}, z_t^{(b)}\}_{t=0}^{T-1}$ is generated by iterating the policy rules and transition equations forward using an independent draw of productivity innovations.
The household updates its policy using stochastic gradient descent,

$$\theta_{m+1} = \theta_m + \eta_m \nabla_\theta J_T(\theta_m),$$

and in implementation $\nabla_\theta J_T(\theta_m)$ is replaced by the simulation-based gradient $\nabla_\theta \widehat{J}_{T,B}(\theta_m)$ computed by differentiating through the simulated transition system. Bounded rationality arises from finite horizons (T), noisy gradient estimates (number of stochastic draws to evaluate expectations, B), and a finite number of gradient updates (number of epochs, E).

3.2 SGD Equilibrium

We now formalize the solution concept implied by this learning process. The definition replaces the standard requirement that agents solve an infinite-horizon optimization problem exactly with the weaker requirement that their parameterized decision rules are a fixed point, in expectation, of the gradient-based updating procedure.

Definition (SGD Equilibrium). An *SGD equilibrium* consists of:

1. a computational budget (T, E, B), comprising a planning horizon T, a number of training epochs E, and a number of Monte Carlo simulation paths B;

2. a policy parameter vector $\theta^\star \in \Theta$ and induced decision rules $(\pi^s_{\theta^\star}, \pi^n_{\theta^\star})$ mapping states (k_t, z_t) to a savings rate and labor supply; and

3. a stationary distribution $\mu^\star$ over states (k, z) induced by iterating the policy rules and the exogenous productivity process forward,

such that the following two conditions hold.

Optimality in expectation. The parameter vector $\theta^\star$ is a fixed point of the stochastic gradient ascent update in the sense that the expected gradient of the truncated objective vanishes at $\theta^\star$,

$$\mathbb{E}[\nabla_\theta J_T(\theta^\star)] = 0, \tag{8}$$

so that the agent has no incentive, on average, to revise her decision rules further. Equation (8) is the stochastic analogue of a first-order condition: it does not require convergence to a specific parameter vector in any given simulation draw, only that the gradient is zero in expectation, reflecting the fact that gradient estimates remain noisy due to finite Monte Carlo sampling.

Market clearing. At every state (k, z) in the support of $\mu^\star$, the allocations induced by $(\pi^s_{\theta^\star}, \pi^n_{\theta^\star})$ satisfy the resource constraint and the capital and labor markets clear at the competitive prices.

4 Impact of the rollout horizon T and the number of training epochs E on the quality of the learned policy

The rollout horizon T (see equation 6) and the epoch count E govern completely different aspects of the learning problem. T determines *what* the agent optimizes — the shape of the truncated objective she sees at every gradient step. E determines *how well* it optimizes that objective — how far the network weights move from their initialization. Nevertheless the two parameters interact through the gradient structure of backpropagation through time (BPTT), and understanding that interaction is essential for interpreting how each imension of the computational budget shapes the resulting policy functions.[2] The remainder of this section analyses each parameter in isolation and then examines the interaction.

4.1 Effect of the horizon T: defining the objective

The standard macroeconomics problem is infinite-horizon: maximize $\sum_{t=0}^{\infty} \beta^t \, u(c_t, n_t)$. In the SGD framework, the quantity the agent actually optimizes is

$$J(\theta;\, k_0, \varepsilon_{1:T}) \;=\; \sum_{t=0}^{T-1} \beta^t \, u(c_t, n_t). \tag{9}$$

This objective ignores the continuation value present in the standard model $\beta^T \; \hat{V}(k_T, z_T)$, where $\hat{V}(\cdot, \cdot)$ is the continuation value given states k_T, z_T. Two distinct channels connect T to the properties of the solution.

[2]The gradient of lifetime utility depends on how policy choices affect future states and utilities. Computing this gradient requires propagating marginal effects backward along the simulated time path. See Appendinx B for details.

Channel 1: the weight on the continuation approximation.

The continuation term enters with weight β^T. Because $\beta = 0.99$ the discount decays slowly, and this weight is substantial for any practically affordable T:

T	β^T	% of infinite-horizon value in the tail	Implication
50	0.605	60%	Most of the value is in the approximation
100	0.366	37%	Approximation still carries large weight
200	0.134	13%	Workable; small residual bias
300	0.049	5%	Tail nearly irrelevant
500	0.007	1%	Negligible

Early in training the omission of $\hat{V}$ introduces a large discrepancy between the SGD solution and the standard solution. As training progresses and the policy improves, the approximation improves with it—the error is *self-correcting*. Beyond $T \approx 300$ the weight β^T is small enough that residual approximation error distorts negligibly the objective, and further increases in T yield diminishing returns on objective fidelity.[3]

Channel 2: visibility of the investment payoff.

The savings decision is intertemporal. Raising the savings rate s_t sacrifices consumption today but increases k_{t+1}, which raises future output $y_{t+1}, y_{t+2}, \ldots$ through the production function. The agent can only *observe* this payoff to the extent that it materializes within the simulated window. The continuation value makes future payoffs *implicit* — it assigns a value to k_T — but it cannot fully substitute for actually simulating the compounding. With very short T the direct gradient signal for "invest more" is weak because most of the payoff from investment falls outside the window; raising T makes more of the compounding visible and strengthens that signal. However, as Section 4.3 shows, the gradient's own attenuation structure limits how far into the future useful signal can reach, so this channel also saturates.

4.2 Effect of the epoch count E: optimization depth

The bias-initialization trick guarantees that at exactly one point in state space—$(k, z) = (k_{ss}, 1)$—the policy outputs the correct steady-state controls (n_{ss}, s_{ss}) before any training has occurred. Everywhere else the policy is governed by the network *weights* W_1, W_2, W_3, which start at their random Xavier values.[4]

Each training epoch moves those weights by roughly $\eta\|\nabla_\theta J_T\|$ in the gradient direction, where $\eta = 3 \times 10^{-4}$ is the Adam step size. After one epoch, the displacement is negligible; the policy is still

[3] This may not be the case if the economic problem contains extreme event shocks.

[4] Weights are drawn i.i.d. from $\mathcal{U}[-a, a]$ with $a = \sqrt{6/(n_{\text{in}} + n_{\text{out}})}$ (Glorot and Bengio, 2010). For the three layers of π_θ this gives $a_1 = a_3 = \sqrt{6/66} \approx 0.30$ and $a_2 = \sqrt{6/128} \approx 0.22$. The choice sets $\text{Var}(W_{ij}) = 2/(n_{\text{in}} + n_{\text{out}})$, which preserves activation and gradient variances through symmetric nonlinearities such as tanh. More details are in appendix A

essentially random away from the steady state. Training therefore proceeds through a self-correcting loop. Early on, any TFP shock pushes the economy into a region where the policy is poor, capital drifts, and the states visited during the rollout may lie far from the true ergodic distribution. As the policy improves, the distribution of visited states converges toward the ergodic distribution, and gradient updates refine the policy over an increasingly representative sample. Closing this loop requires enough epochs for the weights to reach a neighbourhood of the optimum.

4.3 The interaction: gradient depth is finite

Two distinct mechanisms limit the extent to which increasing the rollout horizon T improves the learned policy. The first is *objective truncation*: when the agent optimizes a truncated return, the weight β^T placed on the omitted tail (or on an approximate continuation value) governs how closely the training objective approximates the true infinite-horizon problem (Section 4.1). The second is *gradient attenuation*: even holding the objective fixed, the marginal effect of actions taken far in the future on the gradient with respect to early-period policy parameters tends to decay rapidly under backpropagation through time (BPTT). These mechanisms are conceptually separate but interact in practice: T affects both the fidelity of the objective and the informativeness of the gradients used to optimize it.

Attenuation in the indirect state channel

BPTT computes $\nabla_\theta J$ by differentiating through the unrolled T-period simulation. A change in θ affects the loss both *directly* (through its impact on current actions and current utility) and *indirectly* (through how current actions change future states and therefore future utilities). In our RBC setting, capital is the only endogenous state variable, so the longest-range indirect channel runs through the sensitivity of future capital to earlier policy parameters.

To build intuition, consider the capital transition under the savings-share formulation,

$$k_{t+1} \;=\; s_t\, x_t, \qquad x_t \;=\; (1-\delta)k_t + y_t, \qquad y_t \;=\; z_t k_t^{\alpha} n_t^{1-\alpha},$$

where (s_t, n_t) are produced by the policy $\pi_\theta(k_t, z_t)$. A key distinction is between the *open-loop* Jacobian of the transition map with respect to the state, holding actions fixed,

$$\left.\frac{\partial k_{t+1}}{\partial k_t}\right|_{(s_t,n_t)\,\text{fixed}} \;=\; s_t\big(1-\delta+r_t\big), \qquad r_t = \alpha z_t k_t^{\alpha-1} n_t^{1-\alpha},$$

and the *closed-loop* (policy-induced) sensitivity that governs state propagation along the learned policy,

$$\frac{dk_{t+1}}{dk_t} \;=\; \left.\frac{\partial k_{t+1}}{\partial k_t}\right|_{(s_t,n_t)\,\text{fixed}} \;+\; \frac{\partial k_{t+1}}{\partial s_t}\frac{\partial s_t}{\partial k_t} \;+\; \frac{\partial k_{t+1}}{\partial n_t}\frac{\partial n_t}{\partial k_t}.$$

BPTT propagates gradients through the full computation graph, and therefore the relevant object for long-range attenuation is the *closed-loop* derivative. The additional terms capture policy feedback

($\partial s_t/\partial k_t$ and $\partial n_t/\partial k_t$) and can, in principle, either dampen or amplify state sensitivity relative to the open-loop benchmark.

Nevertheless, the open-loop steady-state Jacobian provides a useful *timescale* benchmark. Evaluating the open-loop expression at the deterministic steady state and using the Euler condition $1 - \delta + r_{ss} = 1/\beta$ yields

$$\left.\frac{\partial k_{t+1}}{\partial k_t}\right|_{SS,\,(s,n)\,\text{fixed}} \;=\; s_{ss}\,(1 - \delta + r_{ss}) \;=\; \frac{s_{ss}}{\beta}. \tag{10}$$

If one further uses this local approximation to describe the propagation of a perturbation over τ steps *through the open-loop state channel*, then an indirect contribution to the gradient that travels from date $t + \tau$ back to date t carries the mechanical factor

$$\beta^\tau \left(\frac{s_{ss}}{\beta}\right)^\tau \;=\; s_{ss}^\tau. \tag{11}$$

This cancellation should be interpreted narrowly: it holds for the open-loop steady-state linearization and abstracts from state dependence and policy feedback. In the full model, the net attenuation rate is governed by the closed-loop Jacobian dk_{t+1}/dk_t along the simulated path, which generally varies with (k_t, z_t) and with the current policy.

For the baseline calibration, $s_{ss} \approx 0.932$, so the benchmark decay rate s_{ss}^τ implies a half-life of approximately

$$\tau_{1/2} = \frac{\log(1/2)}{\log s_{ss}} \approx 10 \text{ steps}.$$

That is, the indirect gradient signal traveling through the capital accumulation channel loses half its magnitude every 10 periods. After $\tau = 50$ steps — five half-lives — the attenuation factor is $s_{ss}^{50} \approx 0.03$, meaning that periods more than 50 steps ahead contribute less than 3% as much gradient information about early-period actions as periods in the near future. In practice, these distant gradient contributions are dominated by Monte Carlo noise and optimization variance well before $\tau = 50$, so the effective informative horizon of BPTT is shorter than the nominal planning horizon T.

Implications for the choice of T and E

The two mechanisms above imply different roles for T.

Objective fidelity. Increasing T reduces the weight β^T on the omitted tail (or on any continuation approximation). With $\beta = 0.99$, $\beta^{300} \approx 0.05$, so for T on the order of a few hundred the contribution of the tail to the objective becomes quantitatively small in our baseline calibration, and further increases yield diminishing returns on this margin.

Gradient informativeness. By contrast, the effective depth of useful gradients is limited by attenuation in the indirect state channel and by sampling noise. Beyond a moderate horizon, additional periods tend to contribute little to $\nabla_\theta J$ relative to the dominant early-period terms.

This is an empirical regularity of smooth DSGE rollouts trained with Monte Carlo gradients rather than a knife-edge theoretical statement: the precise depth depends on the calibration, the current policy, and the magnitude of policy feedback.

Epochs E, in turn, govern *optimization depth* rather than the definition of the objective. Each epoch draws new shock sequences and (after initial transients) visits states closer to the ergodic distribution induced by the evolving policy. Increasing E therefore refines the policy across the relevant region of the state space. In sum, T primarily governs the fidelity of the truncated objective and exhibits diminishing returns once β^T is small, whereas E primarily governs how well the agent optimizes that objective given gradient noise and function approximation.

5 Asymptotic Properties of the SGD Agent Solution

This section discusses the asymptotic properties of the SGD solution as the planning horizon and the number of epoch increase arbitrarily. Recall the household policy is parameterized by a finite-dimensional vector $\theta \in \Theta \subset \mathbb{R}^d$ through a pair of decision rules

$$s_t = \pi_\theta^s(k_t, z_t), \qquad n_t = \pi_\theta^n(k_t, z_t), \tag{12}$$

where s_t is the savings rate out of available resources. Define available resources

$$R_t := (1-\delta)k_t + z_t k_t^\alpha n_t^{1-\alpha}, \tag{13}$$

so that

$$k_{t+1} = s_t R_t, \qquad c_t = (1-s_t)R_t. \tag{14}$$

Given an initial state $x_0 = (k_0, z_0)$, define the infinite-horizon and truncated expected utility objectives:

$$J_\infty(\theta) := \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t \, u(c_t(\theta), n_t(\theta)) \,\middle|\, x_0\right], \tag{15}$$

$$J_T(\theta) := \mathbb{E}\left[\sum_{t=0}^{T-1} \beta^t \, u(c_t(\theta), n_t(\theta)) \,\middle|\, x_0\right]. \tag{16}$$

The expectation is taken with respect to the productivity innovations $\{\epsilon_t\}_{t\geq 1}$.

As discussed above, $J_T(\theta)$ is approximated by Monte Carlo simulation over B independent shock paths:

$$\widehat{J}_{T,B}(\theta) = \frac{1}{B}\sum_{b=1}^{B}\sum_{t=0}^{T-1} \beta^t u\Big(c_t^{(b)}(\theta), n_t^{(b)}(\theta)\Big), \tag{17}$$

where each path is generated by simulating the productivity process and iterating (12)–(14). The SGD agent returns $\hat{\theta}_{T,E,B}$ after E gradient-ascent updates using simulation-based gradients obtained by backpropagation through time.

Let $\Pi_\Theta := \{\pi_\theta : \theta \in \Theta\}$ denote the policy class induced by Θ, where $\pi_\theta = (\pi^s_\theta, \pi^n_\theta)$. We endow Π_Θ with the sup-norm

$$\|\pi - \pi'\|_\Pi := \sup_{(k,z)\in\widetilde{\mathcal{X}}} \left(|\pi^s(k,z) - \pi'^s(k,z)| + |\pi^n(k,z) - \pi'^n(k,z)|\right).$$

(R1) **Compact parameter space and policy continuity.** $\Theta \subset \mathbb{R}^d$ is compact. For each $\theta \in \Theta$, the policy maps $\pi^s_\theta, \pi^n_\theta$ are continuous in (k,z) on the admissible state domain. For each state (k,z), the map $\theta \mapsto \pi_\theta(k,z)$ is continuous.

(R2) **Compact invariance of the simulated state process.** The productivity shock sequence is drawn from a truncated distribution with support $[-\kappa\sigma, \kappa\sigma]$ for some $\kappa > 0$. There exists a compact set

$$\widetilde{\mathcal{X}} = [k_{\min}, k_{\max}] \times [z_{\min}, z_{\max}] \subset (0,\infty) \times (0,\infty)$$

such that, for every $\theta \in \Theta$ and every initial state $x_0 = (k_0, z_0) \in \widetilde{\mathcal{X}}$, the simulated state path induced by (3) and (12) remains in $\widetilde{\mathcal{X}}$ almost surely.[5]

Remark: Assumptions (R1) and (R2) imply that Π_Θ is a metric space.

(R3) **Admissible policy outputs and positivity margins.** The policy parameterization is implemented through smooth output transformations such that, for all admissible states and all $\theta \in \Theta$,

$$s_t \in [\underline{s}, \overline{s}] \subset (0,1), \qquad n_t \in [\underline{n}, \overline{n}] \subset (0,\infty), \tag{18}$$

where the bounds $\underline{s}, \overline{n}$ are chosen consistently with the model parameters ψ and φ such that the GHH composite is bounded away from zero over the invariant set $\widetilde{\mathcal{X}}$. Specifically, let

$$\underline{R} := \inf_{(k,z)\in\widetilde{\mathcal{X}}} R(k,z;\underline{n}) > 0$$

denote the minimum available-resources level on the invariant set. We require

$$\underline{s} \cdot \underline{R} - \psi \frac{\overline{n}^{1+\varphi}}{1+\varphi} \;\geq\; \underline{m} \;>\; 0. \tag{19}$$

Under (18) and (19), the implied allocations satisfy

$$c_t > 0, \qquad c_t - \psi \frac{n_t^{1+\varphi}}{1+\varphi} \geq \underline{m} > 0 \tag{20}$$

for all admissible states and all $\theta \in \Theta$. In particular, utility is well-defined and finite along all simulated paths.

[5] Because the Gaussian innovations in (3) have unbounded support, true compact invariance requires shock truncation. In the implementation we truncate the shocks so we retaining 99.7% of the distribution, and verify numerically that $\widetilde{\mathcal{X}}$ is absorbing under the policy class Π_Θ and the admissible output bounds of (R2).

Remark: The continuity of $\theta \mapsto J_T(\theta)$ and $\theta \mapsto J_\infty(\theta)$ on Θ follows from assumptions (R1)–(R3). Specifically, continuity of the integrand in θ (R1), compact invariance (R2), and uniform boundedness of period utility (R3) together imply, by dominated convergence, that both objectives are continuous on Θ.

(R4) **Finite-horizon approximate optimization.** For each fixed T and each $\varepsilon > 0$, there exist $E_0 = E_0(T, \varepsilon)$ and $B_0 = B_0(T, \varepsilon)$ such that for all $E \geq E_0$ and $B \geq B_0$,

$$J_T\left(\hat{\theta}_{T,E,B}\right) \;\geq\; \sup_{\theta \in \Theta} J_T(\theta) - \varepsilon \qquad \text{with probability } 1 - \varepsilon. \tag{21}$$

That is, the SGD agent is an approximate maximizer of J_T in objective value.[6]

(R4′) **Well-separation (finite horizon).** For each T, the finite-horizon objective J_T has a unique maximizing policy $\pi_T^\star \in \Pi_\Theta$ under $\|\cdot\|_\Pi$.

(R5) **Unique class-optimal infinite-horizon policy.** The infinite-horizon objective admits a unique maximizing policy

$$\pi^\star \in \Pi_\Theta. \tag{22}$$

This holds if the RE solution π^{RE} belongs to Π_Θ and is the unique solution to the Bellman equation, since in that case $J_\infty(\pi^{RE}) > J_\infty(\pi)$ for all $\pi \neq \pi^{RE}$ in Π_Θ by the contraction mapping argument, so $\pi^\star = \pi^{RE}$ is the unique maximizer. If the policy class does not contain π^{RE}, uniqueness of the best approximation $\pi^\star$ is a maintained assumption.

(R6) **RE representability or projection.** Let π^{RE} denote the rational-expectations solution of the infinite-horizon RBC–GHH problem under (3). Assume either:

(i) $\pi^\star = \pi^{RE}$, or

(ii) $\pi^\star$ is the unique best approximation to π^{RE} in Π_Θ under the sup-norm $\|\cdot\|_\Pi$ defined above.

(R7) **Differentiability for backpropagation-through-time.** The policy parameterization and simulated transition system are differentiable almost everywhere in θ, and the backpropagation-through-time gradient estimator provides an asymptotically unbiased estimator of $\nabla_\theta J_T(\theta)$.

Proposition 1 (SGD-to-RE convergence in the RBC–GHH model)**.** *Consider the RBC–GHH economy with policy parameterization* (12)–(14)*. Suppose assumptions (R1)–(R7) hold.*

Then, in the sequential limit *in which one first lets $E, B \to \infty$ for each fixed horizon T, and then lets $T \to \infty$, the induced policies satisfy*

$$\pi_{\hat{\theta}_{T,E,B}} \to \pi^\star \tag{23}$$

[6]This condition replaces the stronger requirements of global concavity or a Polyak–Łojasiewicz condition on the landscape. It does not restrict the geometry of $J_T(\theta)$ and does not require convergence to a specific parameter vector $\theta_T^\star$ — only that the *objective value* achieved is near-optimal. For the GHH–RBC model, this is directly verifiable by comparing $J_T(\hat{\theta}_{T,E,B})$ against the RE benchmark value function across a grid of initial states; see Section 6.

(pointwise in (k, z), and uniformly on $\widetilde{\mathcal{X}}$ in the sup-norm $\|\cdot\|_\Pi$ under uniform continuity of the parameterization).

If, in addition, (R6)(i) holds (the policy class contains the RE policy), then

$$\pi_{\hat{\theta}_{T,E,B}} \to \pi^{RE}. \tag{24}$$

Proof. The proof consists of 3 steps.

Step 1: Uniform boundedness of period utility and truncation error. By (R2), the GHH composite

$$m_t(\theta) := c_t(\theta) - \psi \frac{n_t(\theta)^{1+\varphi}}{1+\varphi}$$

satisfies $m_t(\theta) \geq \underline{m} > 0$ uniformly over admissible states and $\theta \in \Theta$. Since n_t is also uniformly bounded by (18), utility (3) is continuous on a compact admissible range and therefore uniformly bounded:

$$|u(c_t(\theta), n_t(\theta))| \leq \bar{u} < \infty.$$

Hence, for every $\theta \in \Theta$,

$$|J_\infty(\theta) - J_T(\theta)| \leq \sum_{t=T}^{\infty} \beta^t \bar{u} = \frac{\beta^T}{1-\beta}\bar{u}, \tag{25}$$

which implies

$$\sup_{\theta\in\Theta} |J_\infty(\theta) - J_T(\theta)| \leq \frac{\beta^T}{1-\beta}\bar{u} \to 0. \tag{26}$$

Thus, the truncated objective converges uniformly to the infinite-horizon objective, with objective truncation error of order $O(\beta^T)$.

Step 2: Convergence of finite-horizon class-optimal policies. By (R1)–(R4), J_T and J_∞ are continuous on compact Θ, so maximizers exist. Uniform convergence (26) and uniqueness of the class-optimal infinite-horizon policy (R6) imply, by a standard argmax theorem in policy space (Newey and McFadden, 1994, Lemma 2.1), that finite-horizon maximizing policies converge to $\pi^\star$ as $T \to \infty$.

Step 3: Finite-horizon optimization consistency of the SGD agent. For fixed T, assumption (R5) implies that the SGD iterate with Monte Carlo simulation chieves near-optimal objective value (in the sense of (21)) as $E, B \to \infty$. By (R7), the simulation-based gradients used in the implementation provide asymptotically unbiased estimates of $\nabla_\theta J_T(\theta)$, ensuring that the near-optimality in (R4) is achievable by the implemented SGD procedure.

By (R4′) and the argmax theorem applied in the compact metric space $(\Pi_\Theta, \|\cdot\|_\Pi)$, near-optimality in objective value implies near-optimality in policy space:

$$\|\pi_{\hat{\theta}_{T,E,B}} - \pi_T^\star\|_\Pi \to 0 \quad \text{in probability as } E, B \to \infty.$$

Combining Steps 2 and 3 in the sequential limit: for each $\varepsilon > 0$, choose T_0 large enough (from Step 2) that $\|\pi^\star_{T_0} - \pi^\star\|_\Pi < \varepsilon/2$; then for that T_0, choose E_0, B_0 large enough (from Step 3) that $\|\pi_{\hat{\theta}_{T_0,E,B}} - \pi^\star_{T_0}\|_\Pi < \varepsilon/2$ with probability at least $1 - \varepsilon$. The triangle inequality then gives $\|\pi_{\hat{\theta}_{T_0,E,B}} - \pi^\star\|_\Pi < \varepsilon$, establishing (23).

In addition, if (R6)(i) holds, then $\pi^\star = \pi^{RE}$, giving (24). □

5.1 Monte Carlo implementation of expectations in the RBC–GHH theorem

For clarity, the theorem pertains to the implemented learning problem in which expectations over future productivity are replaced by simulation averages. Specifically, for each parameter θ and horizon T, the expectation in (16) is approximated by (17), where each simulation path is generated as follows:

1. draw innovations $\{\epsilon^{(b)}_{t+1}\}_{t=0}^{T-1}$,
2. construct $\{z^{(b)}_t\}$ recursively using (3),
3. evaluate $s^{(b)}_t = \pi^s_\theta(k^{(b)}_t, z^{(b)}_t)$ and $n^{(b)}_t = \pi^n_\theta(k^{(b)}_t, z^{(b)}_t)$,
4. compute $(k^{(b)}_{t+1}, c^{(b)}_t)$ from (13)–(14),
5. accumulate discounted utility and average over $b = 1, \ldots, B$.

Thus, the expectation operator in the household's objective is implemented as a simulation average over productivity draws, and the SGD update uses gradients of this simulation-based objective.

5.2 Policy-function error decomposition in the RBC–GHH model

Let $\pi^\star_T$ denote a maximizing policy for the truncated horizon-T objective within the policy class, and let $\pi^\star$ denote the unique class-optimal infinite-horizon policy. For any norm $\|\cdot\|_\Pi$ on policies (defined on $\widetilde{\mathcal{X}}$), the triangle inequality yields

$$\|\pi_{\hat{\theta}_{T,E,B}} - \pi^{RE}\|_\Pi \le \underbrace{\|\pi_{\hat{\theta}_{T,E,B}} - \pi^\star_T\|_\Pi}_{\text{(I) optimization \& simulation error}} + \underbrace{\|\pi^\star_T - \pi^\star\|_\Pi}_{\text{(II) truncation (finite-horizon) error}} + \underbrace{\|\pi^\star - \pi^{RE}\|_\Pi}_{\text{(III) approximation-class error}}. \tag{27}$$

Interpretation of the three components.

- **(I) Optimization & simulation error.** This component captures finite optimization time ($E < \infty$) and finite Monte Carlo sampling ($B < \infty$). It shrinks as the SGD routine converges and simulation noise is reduced.
- **(II) Truncation (finite-horizon) error.** This component captures the discrepancy between the horizon-T class-optimal policy and the infinite-horizon class-optimal policy. The objective-level truncation error decays at rate $O(\beta^T)$ by (25). Translating this into a policy-distance rate

requires additional curvature/identification conditions linking objective differences to policy distances.

- **(III) Approximation-class error.** This component captures the distance between the best policy in the chosen parameterized class and the exact RE policy. It vanishes if the class contains the RE policy and decreases as the policy class is enriched.

5.3 Remark on computational bounded rationality in RBC–GHH

For a fixed policy class Π_Θ, the RBC–GHH SGD agent defines a sequence of computationally bounded policies indexed by the computational budget (T, E, B). Proposition 1 implies that, as computational resources increase, the learned policy converges to the class-optimal infinite-horizon policy, and to the RE policy whenever the latter is contained in the policy class. In this sense, bounded rationality in the model arises from finite planning horizon, finite simulation, and finite optimization time rather than from misspecified beliefs about the economy .

6 Quantitative Analysis

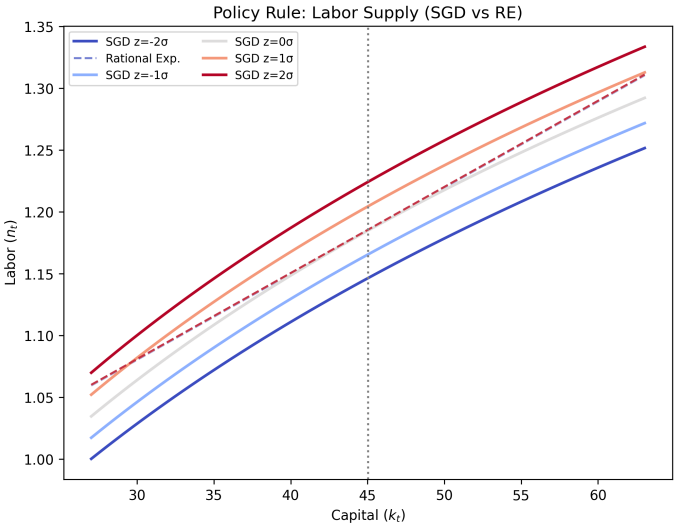
6.1 Large T/Large E: The Unconstrained SGD Agent

In the first case, let's consider an agent with large time horizon $T = 500$ and large optimization capacity $E = 3000$. This setup captures the case of the standard RBC setup model/solution (see Section 6). Figure 2 display the policy functions for labor, consumption, and the savings share. The solid lines correspond to the policies from the SGD-agent model, with different shades representing different productivity levels with the highest in red and the lowest in blue. The dashed line corresponds to the policy functions implied by the linear rational expectation model when productivity it is at its steady state. There are several points worth discussing. First, the policies in the two models are practically indistinguishable from each other when capital is in the interval $[0.89k_{ss}, 1.1k_{ss}]$. The SGD-agent model displays features such as higher productivity leads to lower saving rates (compare blue versus red lines in panel c), higher consumption, and higher labor.

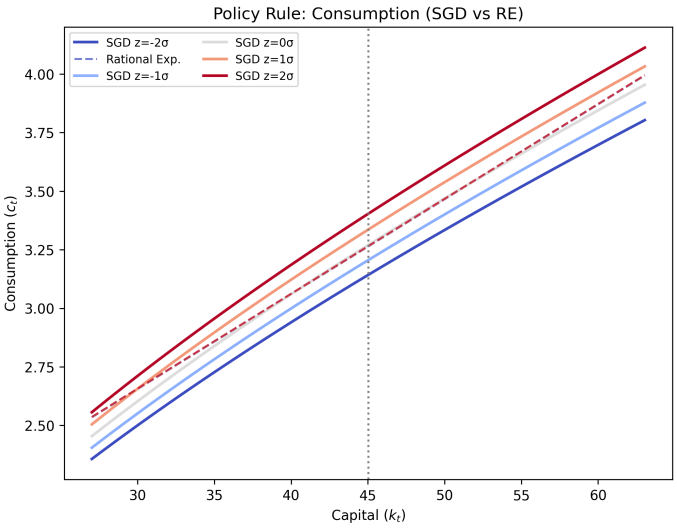
6.2 Limited T and E: The Constrained SGD Agent

This section documents the behavior of an SGD agent that faces computation limitation reflected by short planning horizon (T), limited computation capacity (E), or both. To this end,five variation of the model were analyzed: $(T = 10, E = 100), (T = 10, E = 1000), (T = 100, E = 100), (T = 100, E = 1000), (T = 200, E = 2000)$. The results are reported in Figure 3.

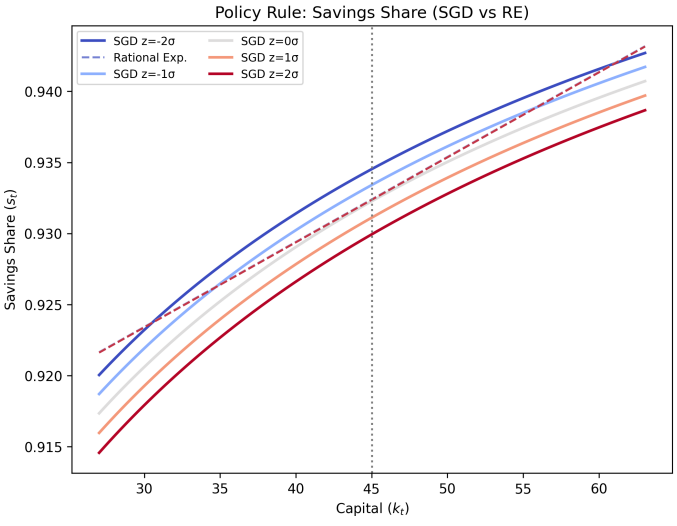
The results for small – $T = 10$ in orange and red lines – look counterintuitive with downward sloping policy functions. One possibility is that these shapes are, at least in part, an artifact of Monte Carlo noise from using a single batch to approximate conditional expectations in the SGD update. With short rollouts, the effective BPTT depth is limited, so the gradient estimate $\nabla_\theta J_T(\theta)$ can be dominated by the particular shock realizations in that batch; the resulting policy may



(a) Labor Supply (n_t)



(b) Consumption (c_t)



(c) Savings Share (s_t)

Figure 2: **Policy Functions: SGD Agent vs. Rational Expectations.** The solid lines represent the neural network policy learned via SGD. The dashed lines represent the linear Rational Expectations solution derived from Dynare. Shades indicate different productivity shock levels (z).

therefore partly overfit those draws and display spurious non-monotonicities. A useful diagnostic is to compute the implied policy functions across multiple independent batches (or training seeds) holding T fixed, and then examine the cross-batch average policy (and its dispersion).

Figure 4 reports the corresponding *average* policy functions computed by evaluating the short-horizon (T) case across 100 independent Monte Carlo batches and averaging the implied policies pointwise. The goal is to filter out batch-specific expectation error in the gradient updates and isolate the systematic component of the short-T solution. Comparing Figure 4 to Figure 3 helps to see that the distorted policies partially result from the lack of enough draws in the Monte Carlo average (in Figure 3 there is only 1 draw).

A short horizon makes the agent effectively *myopic*: because the objective places little weight on outcomes beyond T (and because BPTT attenuates the influence of distant periods on the gradient), the policy internalizes the return to investment only weakly. Relative to the large-T case, the agent therefore tends to save less (accumulate less capital) and to consume more on impact. Lower capital accumulation reduces future productive capacity, which translates into persistently lower output in the short-T scenarios.

6.3 Cost of computation

The setup proposed in this paper allows for a direct measure of the cost of attention, which is defined as the energy consumption for an SGD-agent with a given time horizon and number of epochs (for simplicity we abstract from the number of batches). Table 1 shows the energy results based on a MacBook Pro Laptop 14-inch with a M4 processor.

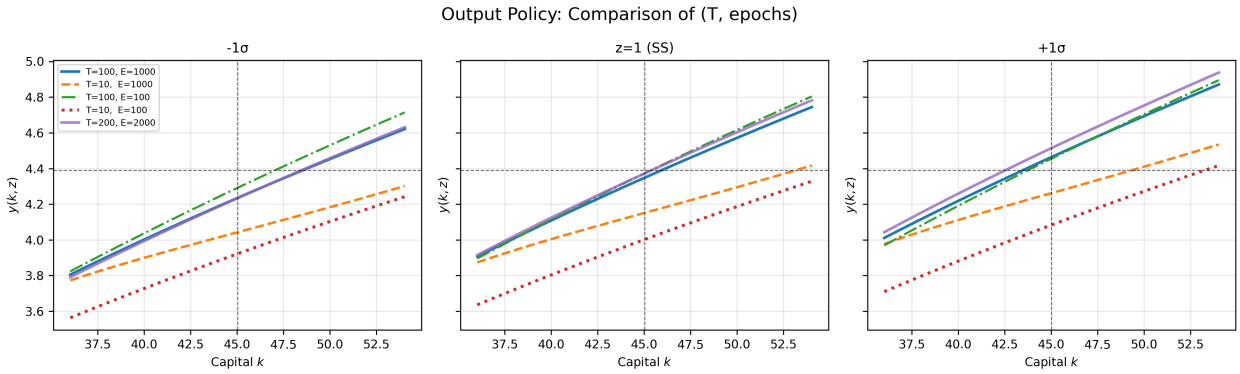
Table 1: Energy consumption per training epoch on Apple M4 (batch size 512)

Horizon	Time per epoch	Energy per epoch (Wh)		
T	(seconds)	Conservative (20W)	Typical (30W)	Intensive (40W)
10	0.01	0.0000	0.0001	0.0001
50	0.03	0.0002	0.0002	0.0003
100	0.06	0.0003	0.0005	0.0006
200	0.12	0.0006	0.0010	0.0013
500	0.29	0.0016	0.0024	0.0032

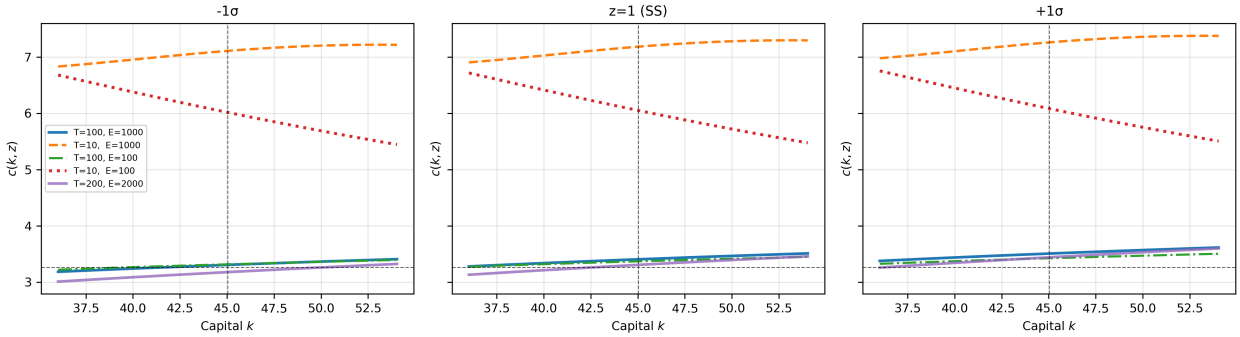
Note: Power estimates based on typical M4 behavior during sustained neural network training. Conservative scenario assumes light CPU load (20W); typical scenario assumes moderate sustained training (30W); intensive scenario assumes heavy compute burst (40W). Actual power varies with thermals, background processes, and battery vs. AC power.

7 Empirical Implications

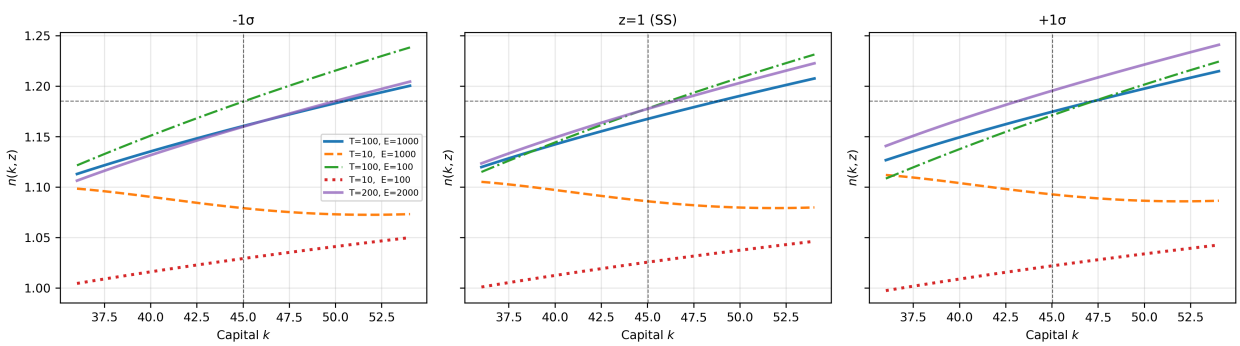
The stochastic gradient descent (SGD) agent framework delivers a set of empirical implications that are distinct from both rational expectations models and existing learning-based approaches. Because



(a) Output

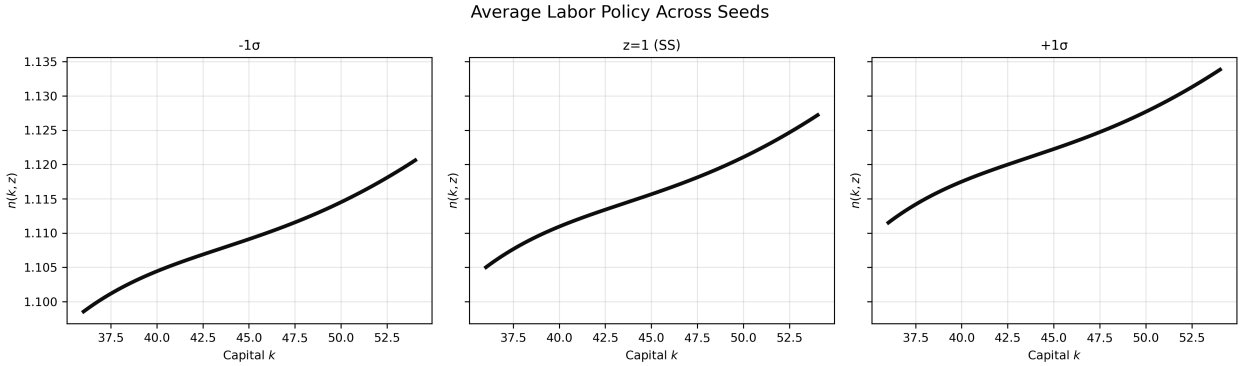


(b) Consumption

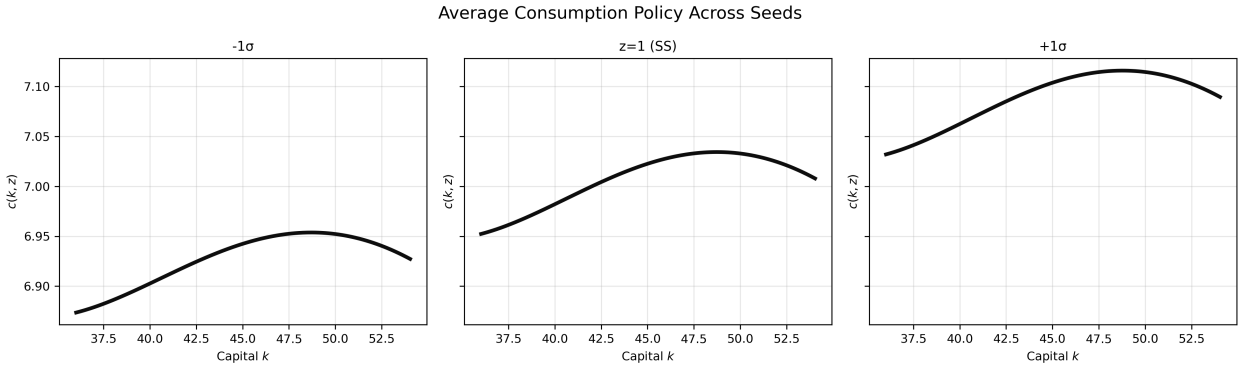


(c) Labor

Figure 3: **Limited-computation SGD agents:** Plots correspond to the alternative scenarios.



(a) Average labor policy



(b) Average consumption policy

Figure 4: **Average policies across batches.** Policy functions averaged across 100 independent Monte Carlo batches (short-T case).

bounded rationality arises from limited optimization rather than incorrect beliefs or incomplete information, the framework generates testable predictions for wedges, dynamics, and heterogeneity that can be confronted with macroeconomic data.

7.1 Endogenous and Time-Varying Wedges

A central implication of the SGD equilibrium is the emergence of endogenous, time-varying wedges in optimality conditions. Because agents do not solve the household problem exactly, Euler equations and intratemporal first-order conditions may hold only approximately.

In particular, the intertemporal Euler equation implies an *Euler wedge*

$$\omega_t^k \equiv u_c(c_t, n_t) - \beta \mathbb{E}_t[u_c(c_{t+1}, n_{t+1})(1 + r_{t+1} - \delta)], \tag{28}$$

while the intratemporal condition under GHH preferences implies a *labor wedge*

$$\omega_t^n \equiv w_t / \psi n_t^{\varphi} - 1. \tag{29}$$

In the SGD framework, these wedges are not exogenous shocks or measurement error. Instead, they arise endogenously from finite planning horizons, noisy gradients, and limited optimization depth. As computational capacity increases, both wedges converge to zero, nesting the rational expectations benchmark. This interpretation provides a structural foundation for the time-varying wedges commonly measured in business cycle accounting exercises (Chari et al., 2007), without invoking preference shocks, technology wedges, or real frictions not present in the baseline RBC model.

Figure 5 plots the labor wedge $\log_{10} |w_t / \psi n_t^{\varphi} - 1|$ as a function of capital for different levels of productivity. This figure illustrates how limited optimization generates a sizable and time-varying intratemporal distortion. The rational-expectations benchmark corresponds to $\omega_t^n = 0$ at all dates and states of the economy. In the constrained case ($T = 100$, $E = 1000$), the wedge departs systematically from zero, indicating that the learned labor policy does not fully internalize the contemporaneous tradeoff between the marginal disutility of work and the real wage. The fact that the wedge varies over time is consistent with state-dependent mis-optimization: following shocks and along transition dynamics, the truncated-horizon objective and noisy gradient updates can deliver labor choices that are locally suboptimal, and these errors show up as persistent deviations of ψn_t^{φ} from w_t.

Figure 6 shows that as the planning horizon increases, the labor wedge closes. That is, the SGD-agent model gets closer to the fully rational equilibrium solution. For completeness, Figure 7 show the Euler wedge in log 10 basis.

These wedge patterns provide a structural interpretation of the time-varying distortions emphasized by business-cycle accounting. In the accounting framework of Chari et al. (2007), wedges are measured as residuals in the equilibrium conditions of a prototype RBC model and then interpreted as reduced-form "shocks"—most prominently, an intratemporal (labor) wedge and an intertemporal

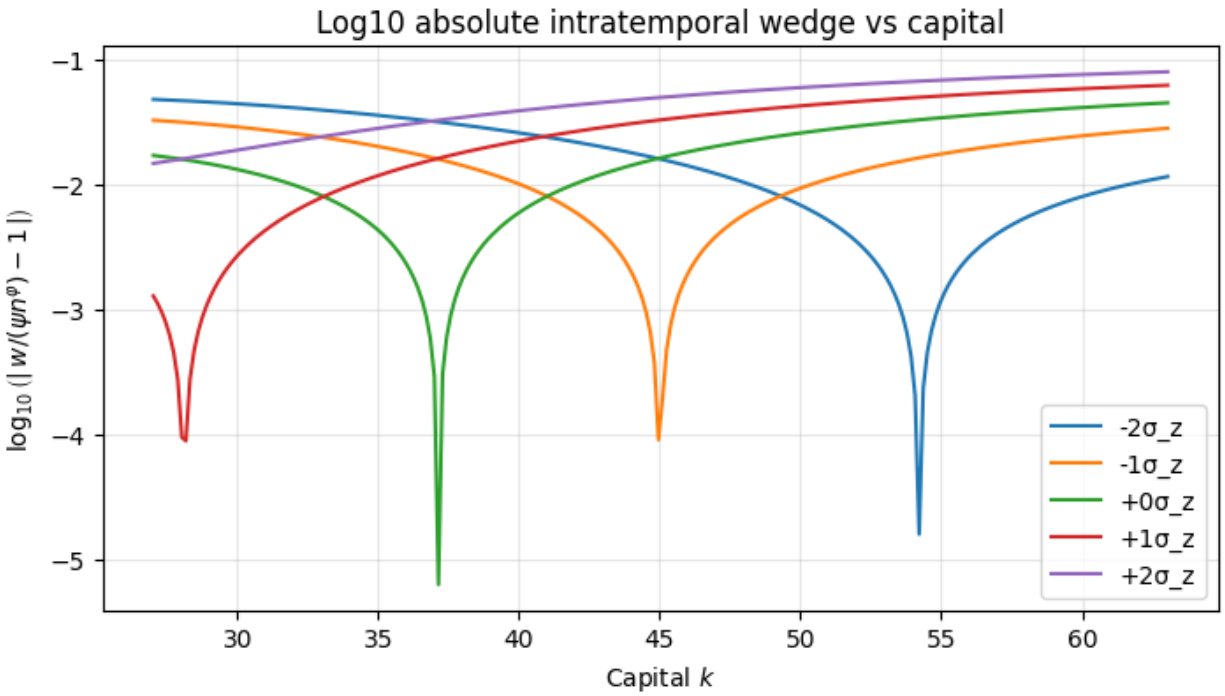


Figure 5: **Labor wedge under limited optimization.** The figure reports the implied labor wedge $\omega_t^n \equiv w_t/\psi n_t^{\varphi} - 1$ in $\log 10$ basis for a representative short-horizon/limited-training case (here, $T = 100$ and $E = 1000$).

(investment) wedge—that summarize the forces needed to reconcile the prototype model with observed allocations. Our results suggest a complementary, micro-founded reading of these objects. In the SGD equilibrium, the analogues of the Chari et al. (2007) wedges arise endogenously as optimality-condition residuals generated by limited computation: finite planning horizons, Monte Carlo noise in gradient estimates, and finite training intensity imply that the learned policy does not satisfy the Euler equation and the intratemporal labor condition exactly. The resulting residuals therefore have the same accounting form as the wedges in Chari et al. (2007), but their economic content differs: they are not exogenous disturbances or omitted frictions, but state-contingent misoptimization.

The figures underscore two implications that are difficult to obtain from a purely reduced-form wedge process. First, the wedges are intrinsically state dependent. Even at steady-state productivity, the intertemporal wedge varies sharply with capital: deviations from the rational-expectations benchmark are small at low k but become substantially larger at high k, indicating that bounded optimization is more severe in regions where the marginal value of additional capital accumulation is flatter. Second, the wedges depend systematically on productivity. Holding k fixed, both the intertemporal Euler wedge and the intratemporal labor wedge are amplified in high-productivity states (e.g., when z is two standard deviations above its mean), implying that optimization errors co-move with the endogenous incentives to adjust saving and labor. In this sense, the SGD equilibrium generates endogenous wedge dynamics that resemble the time variation

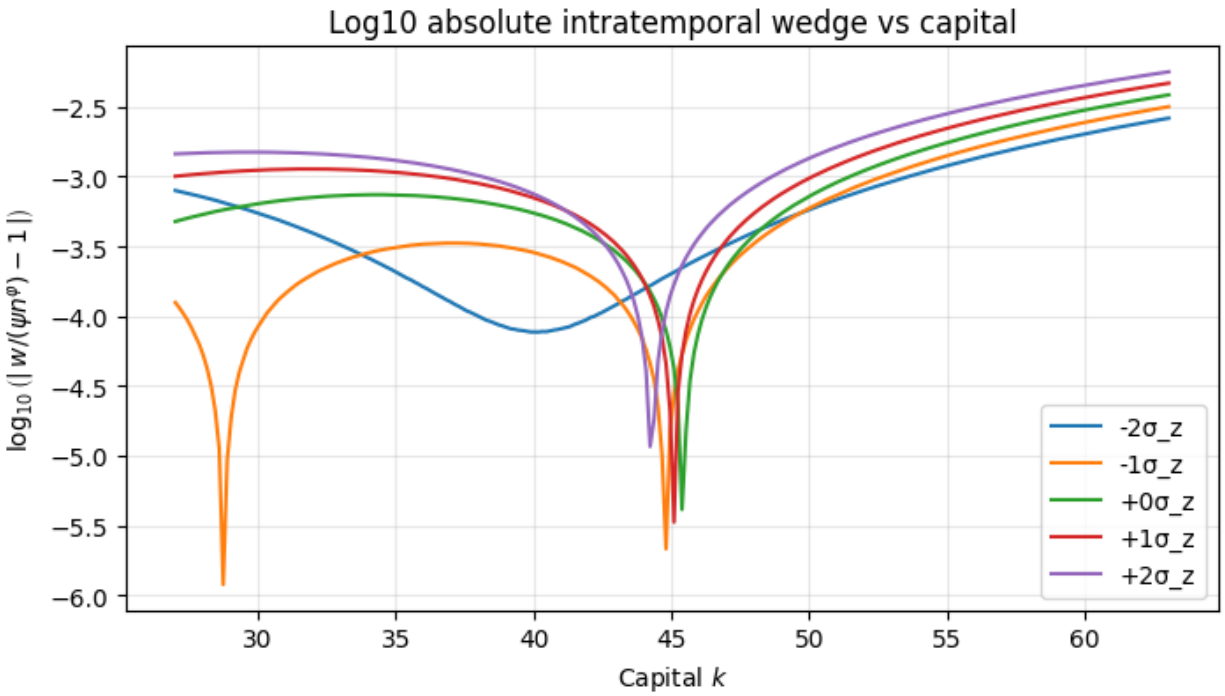


Figure 6: **Labor wedge under limited optimization.** The figure reports the implied labor wedge $\omega_t^n \equiv w_t/\psi n_t^{\varphi} - 1$ in $\log 10$ basis for a representative longer-horizon/limited-training case (here, $T = 500$ and $E = 1000$).

documented by Chari et al. (2007) but do so without introducing preference shocks, markup shocks, or other ad hoc distortions. Instead, wedge variation reflects the interaction between the economy's state and the agent's computational budget.

This perspective also delivers a discipline for comparative statics that is absent in standard accounting exercises. Tightening the computational constraint by shortening the rollout horizon T primarily shifts the intertemporal wedge by weakening the internalization of long-run returns to investment, whereas limiting the number of training epochs E primarily increases dispersion and steepens state dependence by reducing the accuracy of state-contingent responses. Consequently, changes in computational capacity map into predictable changes in the level and curvature of the measured wedges, providing a direct bridge between the Chari et al. (2007) objects and an explicit primitive—bounded optimization—that can be varied, measured, and potentially estimated.

7.2 State-Dependent Deviations from Rational Expectations

Unlike models with constant wedges, the deviations from full optimality generated by SGD agents are inherently state-dependent. When the economy is close to its steady state and shocks are small, gradient updates are weak and the policy function remains close to optimal. Following large shocks or during periods of heightened volatility, optimization errors increase, leading to larger wedges and slower adjustment.

Table 2: Business Cycle Accounting: Cyclical Properties of SGD Wedges (HP-filtered, $\lambda = 1600$, E = 100)

	σ (%)	$\sigma/\sigma(Y)$	AC_1	$\rho(Y)$	$\rho(C)$	$\rho(I)$	$\rho(N)$
Panel A: SGD agent (E = 5000)							
Output Y	0.908	1.000	0.773	1.000	0.993	0.785	-0.868
Consumption C	0.810	0.892	0.812	0.993	1.000	0.705	-0.802
Investment I	0.071	0.078	0.627	0.785	0.705	1.000	-0.989
Labor N	0.119	0.131	0.640	-0.868	-0.802	-0.989	1.000
Labor wedge τ_l	1.940	2.137	0.739	0.994	0.974	0.845	-0.915
Investment wedge τ_x	0.226	0.248	0.287	-0.376	-0.426	-0.020	0.105
$\rho(\tau_l, \tau_x)$	—	—	—	-0.327	—	—	—
Panel B: CKM (2007) postwar U.S. benchmark							
Output Y	—	1.000	0.840	1.000	—	—	—
Consumption C	—	0.740	0.870	0.880	—	—	—
Investment I	—	2.910	0.800	0.930	—	—	—
Labor N	—	0.990	0.880	0.850	—	—	—
Labor wedge τ_l	—	0.500	0.900	-0.800	—	—	—
Investment wedge τ_x	—	0.300	0.850	-0.450	—	—	—

Notes: All series are HP-filtered with $\lambda = 1600$. Macro aggregates (Y, C, N) are logged before filtering; investment I and wedges are filtered in levels. The labor wedge is $\tau_l = 1 - \psi n^{\varphi}/w$ and the investment wedge is $\tau_x = 1 - \beta\,\mathbb{E}_t[u_c(c_{t+1}, n_{t+1})\,R_{t+1}]\,/\,u_c(c_t, n_t)$. CKM benchmark columns not available are marked —. AC_1 denotes the first-order autocorrelation.

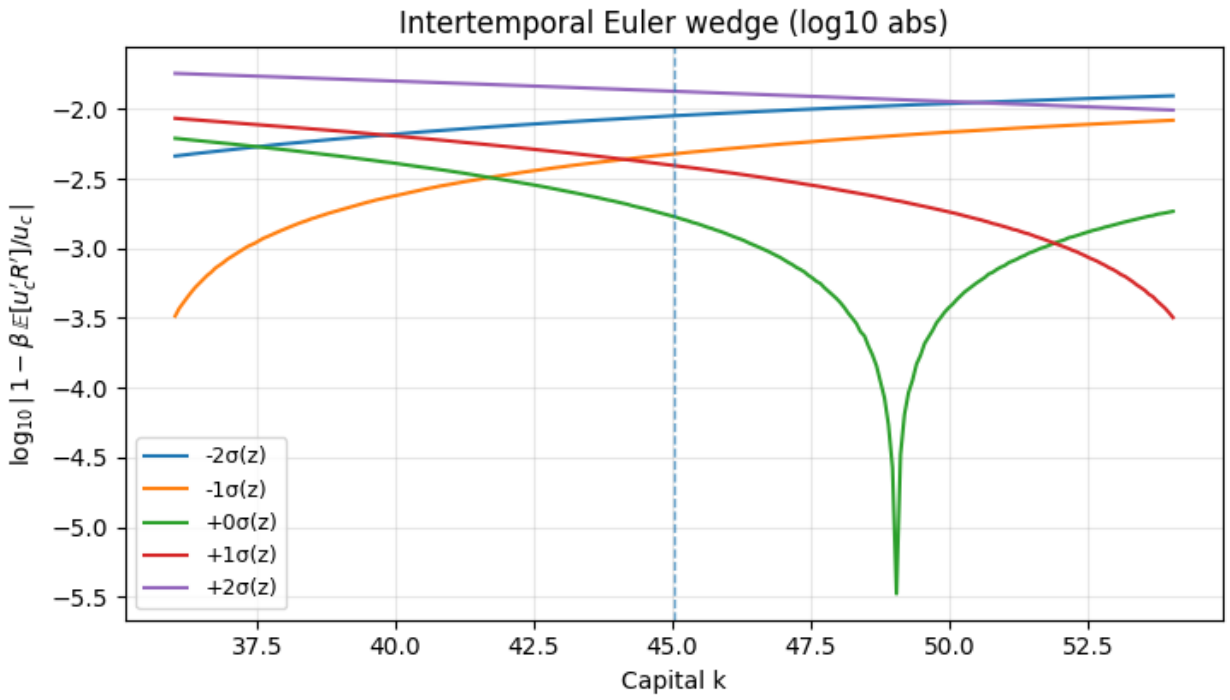


Figure 7: **Euler wedge under limited optimization.** The figure reports the implied Euler wedge $\omega_t^k \equiv 1 - \beta E\left[u_c(c)R'/u_c(c)\right]$ in $\log 10$ basis for a representative short-horizon/limited-training case (here, $T = 100$ and $E = 1000$).

Specifically, Figure 5 showcases pronounced state dependence in the intertemporal wedge. Along the capital dimension, the schedule is asymmetric even at steady-state productivity (green line): deviations from the rational-expectations benchmark are small at low levels of capital but become substantially larger as capital rises, indicating that mis-optimization is more severe in high-k states. Along the productivity dimension, the wedge also varies systematically with z. Holding capital fixed, the wedge is markedly larger when productivity is two standard deviations above its steady state than when z is at its mean, implying that optimization errors are amplified in high-productivity states rather than being a constant distortion.

7.3 Compute as a Structural Parameter

A distinctive feature of the SGD agent framework is that computational capacity enters as an explicit structural object. The number of gradient steps, learning rates, and planning horizons jointly determine how closely agents approximate optimal policies.

From an empirical perspective, this implies that:

- economies or agents with greater computational capacity should exhibit smaller wedges,
- changes in effective computation (e.g., due to technological progress, institutional learning, or experience) should alter macroeconomic dynamics even if economic fundamentals remain unchanged,

- heterogeneity in computational capacity across agents can generate cross-sectional dispersion in behavior and aggregate dynamics.

This perspective opens the door to empirical work that treats computation as an observable or estimable dimension of economic behavior.

8 Conclusion

Modeling households as stochastic gradient descent agents provides a structural and interpretable approach to bounded rationality in macroeconomics. By replacing exact optimization with an explicit learning process, the framework preserves the discipline of DSGE models while relaxing the assumption of unlimited computational capacity. Unlike black-box reinforcement learning, the SGD agent approach remains grounded in economic theory and offers a tractable avenue for studying the macroeconomic implications of limited optimization.

References

Chari, V. V., Kehoe, P. J., and McGrattan, E. R. (2007). Business cycle accounting. *Econometrica*, 75(3):781–836.

Evans, G. W. and Honkapohja, S. (2001). *Learning and Expectations in Macroeconomics*. Princeton University Press.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. and Titterington, M., editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256. PMLR.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

Maćkowiak, B., Matějka, F., and Wiederholt, M. (2023). Rational inattention: A review. *Journal of Economic Literature*, 61(1):226–273.

Marcet, A. and Sargent, T. J. (1989). Convergence of least squares learning mechanisms in self-referential linear stochastic models. *Journal of Economic Theory*, 48(2):337–368.

Newey, W. K. and McFadden, D. (1994). Large sample estimation and hypothesis testing. *Handbook of Econometrics*, 4:2111–2245.

Sims, C. A. (2003). Implications of rational inattention. *Journal of Monetary Economics*, 50(3):665–690.

Woodford, M. (2009). Information-constrained state-dependent pricing. *Journal of Monetary Economics*, 56(Supplement):S100–S124.

Woodford, M. (2019). Monetary policy analysis when planning horizons are finite. In Eichenbaum, M. S. and Parker, J. A., editors, *NBER Macroeconomics Annual 2018, Volume 33*, pages 1–50. University of Chicago Press.

Yang, Y., Wang, C., Schaab, A., and Moll, B. (2025). Structural reinforcement learning for heterogeneous agent macroeconomics. Technical report, University of Zurich and SFI; Peking University; UC Berkeley; London School of Economics. Preliminary. First version: December 2025; this version: December 2025.

A Xavier uniform initialisation

Xavier uniform initialization (Glorot and Bengio, 2010) draws each weight independently from a symmetric uniform distribution whose half-width is calibrated to the input and output dimensions of the layer. For a linear layer with n_{in} inputs and n_{out} outputs, each entry of the weight matrix is drawn from

$$W_{ij} \;\sim\; \mathcal{U}[-a,\, a]\,, \qquad a \;=\; \text{gain} \;\times\; \sqrt{\frac{6}{n_{\text{in}} + n_{\text{out}}}}\,. \tag{30}$$

The code sets gain $= 1$. Applying the formula to each layer of the policy network π_θ (two hidden layers of width $H = 64$, two inputs, two outputs) gives

Layer	Shape	n_{in}	n_{out}	$a = \sqrt{6/(n_{\text{in}} + n_{\text{out}})}$
W_1	$H \times 2$	2	64	$\sqrt{6/66} \approx 0.3015$
W_2	$H \times H$	64	64	$\sqrt{6/128} \approx 0.2165$
W_3	$2 \times H$	64	2	$\sqrt{6/66} \approx 0.3015$

Every entry of every weight matrix is drawn independently from the corresponding uniform. All biases are initialised to zero, and then the two entries of b_3 are immediately overwritten with the steady-state logit targets described in equation (44).

The rationale is *variance preservation*. A uniform random variable on $[-a, a]$ has variance $a^2/3$. Choosing $a = \sqrt{6/(n_{\text{in}} + n_{\text{out}})}$ sets this variance to

$$\text{Var}(W_{ij}) \;=\; \frac{a^2}{3} \;=\; \frac{2}{n_{\text{in}} + n_{\text{out}}}\,, \tag{31}$$

which is the value that keeps the variance of activations approximately constant in the forward pass *and* the variance of gradients approximately constant in the backward pass, provided the nonlinearity is symmetric about zero. The policy network uses tanh activations, which satisfy this symmetry exactly; Xavier initialization is therefore the natural default for this architecture.

B Solution by Stochastic-Gradient-Descent Agents

This section describes how to solve a representative-agent real business-cycle model with Greenwood–Hercowitz–Huffman (GHH) preferences by training a neural-network policy with stochastic gradient descent (SGD) through a differentiable simulation of the economy. The exposition proceeds in four stages: the economic environment and the Bellman problem it induces, the neural-network parameterisation of the policy and its initialisation, the differentiable rollout objective and its terminal-value correction, and the gradient structure that determines how quickly the policy can be learned. A consolidated pseudocode listing and the Euler-residual diagnostic used to assess solution quality are given in Sections B.5 and B.6, respectively.

B.1 Economic environment and the agent's problem

Production and shocks.

A representative firm rents capital k_t and labour n_t under constant returns to scale:

$$y_t \;=\; z_t\, k_t^{\alpha}\, n_t^{1-\alpha}, \qquad \alpha \in (0,1). \tag{32}$$

Total-factor productivity (TFP) evolves as a first-order autoregressive process in logs:

$$\log z_{t+1} \;=\; \rho \log z_t \;+\; \sigma_z\, \varepsilon_{t+1}, \qquad \varepsilon_{t+1} \sim \mathcal{N}(0,1), \qquad \rho \in [0,1). \tag{33}$$

Preferences.

The household values consumption net of the disutility of labour according to the GHH specification:

$$u(c_t, n_t) \;=\; \frac{\left(c_t \;-\; \psi\, \dfrac{n_t^{1+\varphi}}{1+\varphi}\right)^{1-\gamma}}{1-\gamma}, \qquad \gamma > 0,\;\; \psi > 0,\;\; \varphi > 0. \tag{34}$$

The argument of the power function must be strictly positive for utility to be well defined; this defines the *GHH domain*:

$$c_t \;>\; \psi\, \frac{n_t^{1+\varphi}}{1+\varphi}. \tag{35}$$

In the numerical implementation the power function, $(\cdot)^{1-\gamma}$, is replaced by a softplus-smoothed version, $\mathrm{softplus}_{\mathcal{B}}(x) = \mathcal{B}^{-1}\log(1+e^{\mathcal{B}x})$, to keep the computation graph differentiable even when transient trajectories approach the domain boundary.

Resource constraint and savings-share parameterisation.

Define wealth as undepreciated capital plus output:

$$x_t \;=\; (1-\delta)\, k_t \;+\; y_t. \tag{36}$$

The household chooses a *savings share* $s_t \in (0,1)$ that splits wealth into next-period capital and current consumption:

$$k_{t+1} = s_t\, x_t, \qquad c_t = (1-s_t)\, x_t. \tag{37}$$

This is equivalent to the standard formulation $c_t + k_{t+1} = (1-\delta)k_t + y_t$ but maps the choice set to the bounded interval $(0,1)$, which is convenient for the sigmoid parameterisation described below.

Steady state.

The deterministic steady state ($z = 1$, all variables constant) is obtained in closed form. The Euler equation pins down the net rental rate,

$$r_{ss} \;=\; \frac{1}{\beta} - 1 + \delta, \tag{38}$$

from which the capital-labour ratio, wage, and optimal labour follow sequentially:

$$\frac{k_{ss}}{n_{ss}} = \left(\frac{\alpha}{r_{ss}}\right)^{\frac{1}{1-\alpha}}, \tag{i}$$

$$w_{ss} = (1-\alpha)\left(\frac{k_{ss}}{n_{ss}}\right)^{\alpha}, \tag{ii}$$

$$n_{ss} = \left(\frac{w_{ss}}{\psi}\right)^{\frac{1}{\varphi}}. \tag{iii}$$

Equation (iii) is the GHH intratemporal first-order condition $\psi n^{\varphi} = w$. Capital, output, consumption, and the steady-state savings share then follow from the definitions:

$$k_{ss} = \tfrac{k_{ss}}{n_{ss}} \cdot n_{ss}, \qquad y_{ss} = k_{ss}^{\alpha} n_{ss}^{1-\alpha}, \qquad c_{ss} = y_{ss} - \delta k_{ss}, \qquad s_{ss} = \frac{k_{ss}}{x_{ss}}. \tag{39}$$

A necessary condition for the solution to exist is that the steady state satisfies the GHH domain constraint (35). A second requirement, specific to the neural-network solution, is that $n_{ss} < \bar{n}$, where $\bar{n}$ is the upper bound of the labor grid; the calibration must be chosen so that both conditions hold simultaneously. With the baseline parameters $(\alpha, \beta, \delta, \rho, \sigma_z, \gamma, \psi, \varphi) = (0.36,\ 0.99,\ 0.025,\ 0.95,\ 0.007,\ 2,\ 2,\ 1)$ the intratemporal FOC yields $n_{ss} \approx 1.185$.

B.2 Neural-network policy

The optimal policy maps each state (k_t, z_t) to the pair of controls (n_t, s_t). We approximate this mapping with a feedforward neural network $\pi_\theta : \mathbb{R}^2 \to \mathbb{R}^2$.

Architecture.

The network has two hidden layers each of width H with tanh activations. Its input is centred at the steady state:

$$\text{input} \;=\; \big(\hat{k}_t,\ \log z_t\big), \qquad \hat{k}_t = \log k_t - \log k_{ss}. \tag{40}$$

At the steady state both inputs are exactly zero, which is important for the initialisation described below. The network output is a pair of unconstrained real scalars $(m_t^n,\ m_t^s)$:

$$\begin{pmatrix} m_t^n \\ m_t^s \end{pmatrix} = W_3\ \tanh\Big(W_2\ \tanh\big(W_1 \begin{pmatrix} \hat{k}_t \\ \log z_t \end{pmatrix} + b_1\big) + b_2\Big) + b_3. \tag{41}$$

Squashing to controls.

The sigmoid $\sigma(x) = (1 + e^{-x})^{-1}$ maps $\mathbb{R}$ onto $(0, 1)$. Labour and the savings share are recovered as

$$n_t = \bar{n}\, \sigma(m_t^n), \tag{42}$$

$$s_t = s_{\min} + (s_{\max} - s_{\min})\, \sigma(m_t^s), \tag{43}$$

where $s_{\min}$ and $s_{\max}$ are small margins inside $(0, 1)$ that prevent the pathologies $c \to 0$ or $k' \to 0$.

Steady-state bias initialization.

All weights are initialised with Xavier uniform draws. The *biases of the final layer* $b_3 = (b_3^n, b_3^s)$ are then overwritten so that the network reproduces the steady-state controls when both inputs are zero (and hence all hidden activations are near zero):

$$b_3^n = \operatorname{logit}\Big(\frac{n_{ss}}{\bar{n}}\Big), \qquad b_3^s = \operatorname{logit}(s_{ss}), \qquad \operatorname{logit}(p) = \log\frac{p}{1-p}. \tag{44}$$

This is exact at $(k_{ss}, 1)$ and provides a consistent starting point for training. Away from the steady state the policy is governed by the weights, which are refined during training.

B.3 Differentiable simulation and the training objective

Reparameterization.

The transition (33) is *additively separable* in the shock ε_{t+1}. Consequently, once a shock sequence $\{\varepsilon_1, \ldots, \varepsilon_T\}$ is drawn, the entire trajectory $(k_0, \ldots, k_T)$ is a deterministic, differentiable function of the network parameters θ and the initial state k_0. No score-function estimator or REINFORCE-style correction is needed: the expectation over shocks can be replaced by a Monte-Carlo average, and every realization produces a computation graph that supports exact backpropagation through time (BPTT).

Truncated-horizon objective with terminal value.

A single rollout of length T produces the truncated return

$$\hat{J}(\theta;\, k_0, \varepsilon_{1:T}) \;=\; \sum_{t=0}^{T-1} \beta^t\, u(c_t, n_t), \tag{45}$$

where every c_t, n_t is produced by the policy π_θ and the environment dynamics described in Section 3.

The training loss averages over a batch of B independent rollouts, each with its own i.i.d. shock sequence and with k_0 drawn from a small neighbourhood of k_{ss}:

$$\mathcal{L}(\theta) \;=\; -\,\frac{1}{B}\sum_{i=1}^{B} \hat{J}\Big(\theta;\; k_0^{(i)},\; \varepsilon_{1:T}^{(i)}\Big)\,. \tag{46}$$

B.4 Gradient structure and effective depth

Automatic differentiation computes $\nabla_\theta \mathcal{L}$ by backpropagating through the T-step simulation graph. The magnitude of the gradient signal from period t back to the policy parameters used at period 0 is governed by two compounding factors at each intermediate step: the discount β and the Jacobian of the capital transition.

Near the steady state the Jacobian is

$$\left.\frac{\partial k_{t+1}}{\partial k_t}\right|_{SS} = s_{ss}\,(1 - \delta + r_{ss}) = \frac{s_{ss}}{\beta}, \tag{47}$$

where the second equality follows from the Euler condition $1 - \delta + r_{ss} = 1/\beta$. The *effective per-step attenuation* of the indirect gradient channel (utility at $t + \tau$ back through capital to the policy at t) is therefore

$$\beta^\tau \left(\frac{s_{ss}}{\beta}\right)^\tau = s_{ss}^\tau. \tag{48}$$

For the baseline calibration $s_{ss} \approx 0.932$, giving a gradient half-life of

$$\tau_{1/2} = \frac{\log(1/2)}{\log s_{ss}} \approx 10 \text{ steps}. \tag{49}$$

Consequently, the gradient is dominated by the first 20–40 steps of the rollout regardless of T. Increasing T beyond roughly 200 improves the *objective* (less weight on the continuation term) but not the *gradient quality*. Epochs, by contrast, are the only lever for refining the policy over the full state space: each epoch provides a fresh gradient over a (slightly) different set of visited states.

B.5 Algorithm

Algorithm 1 consolidates the procedure.

B.6 Convergence diagnostic: Euler-equation residuals

A trained policy π_θ is assessed by its violation of the Euler equation, which must hold exactly at the true optimum. Define the marginal utility of consumption

$$u_c(c, n) \;=\; \left(c - \psi\,\frac{n^{1+\varphi}}{1+\varphi}\right)^{-\gamma} \tag{50}$$

and the gross marginal product of capital

$$R(k, n, z) \;=\; 1 - \delta + \alpha\, z\, k^{\alpha-1}\, n^{1-\alpha}. \tag{51}$$

Algorithm 1 SGD agent for the RBC–GHH model

Require: Structural parameters $(\alpha, \beta, \delta, \rho, \sigma_z, \gamma, \psi, \varphi, \bar{n})$

Require: Hyperparameters: horizon T, batch size B, epochs E, learning rate η, gradient-clip threshold G

Steady state. Compute (k_{ss}, n_{ss}, s_{ss}) in closed form. Verify $n_{ss} < \bar{n}$ and the GHH domain constraint.

Initialize network. Draw W_1, W_2, W_3 from Xavier uniform. Set $b_3^n = \text{logit}(n_{ss}/\bar{n})$, $b_3^s = \text{logit}(s_{ss})$.

Initialize optimiser (Adam with learning rate η).

for epoch $= 1, \ldots, E$ **do**

  Draw $k_0^{(1:B)}$ i.i.d. from $k_{ss} \cdot \mathcal{U}[1-\delta_k,\, 1+\delta_k]$.

  Set $\log z_0^{(1:B)} = 0$.

  Initialise return $J^{(1:B)} \leftarrow 0$, discount $d \leftarrow 1$.

  for $t = 0, \ldots, T-1$ **do**

    $(m^n, m^s) \leftarrow \pi_\theta(\hat{k}_t, \log z_t)$

    $n_t \leftarrow \bar{n}\, \sigma(m^n)$

    $s_t \leftarrow s_{\min} + (s_{\max} - s_{\min})\, \sigma(m^s)$

    $y_t \leftarrow z_t\, k_t^\alpha\, n_t^{1-\alpha}$

    $x_t \leftarrow (1-\delta)k_t + y_t$

    $c_t \leftarrow (1-s_t)\, x_t; \quad k_{t+1} \leftarrow s_t\, x_t$

    $J \leftarrow J + d \cdot u(c_t, n_t); \quad d \leftarrow d \cdot \beta$

    Draw $\varepsilon_{t+1} \sim \mathcal{N}(0,1)$

    $\log z_{t+1} \leftarrow \rho \log z_t + \sigma_z \varepsilon_{t+1}$

  end for

  $$\mathcal{L} \leftarrow -\frac{1}{B} \sum_{i=1}^{B} J^{(i)}$$

  $\nabla_\theta \mathcal{L} \leftarrow \texttt{backprop}(\mathcal{L})$

  Clip: $\nabla_\theta \mathcal{L} \leftarrow G \cdot \dfrac{\nabla_\theta \mathcal{L}}{\max(G, \|\nabla_\theta \mathcal{L}\|)}$

  $\theta \leftarrow \theta - \eta \cdot \text{Adam}(\nabla_\theta \mathcal{L})$

end for

Given a state (k, z) the policy produces (c, n, k'). Drawing N i.i.d. shocks $\varepsilon^{(1)}, \ldots, \varepsilon^{(N)}$ generates N next-period realisations $(c'^{(j)}, n'^{(j)}, R'^{(j)})$. The *Euler residual* at (k, z) is

$$e(k,z) \;=\; 1 \;-\; \frac{\beta \, \dfrac{1}{N} \displaystyle\sum_{j=1}^{N} u_c(c'^{(j)}, n'^{(j)}) \; R'^{(j)}}{u_c(c, n)}. \tag{52}$$

At the true optimum $e(k, z) = 0$ for all (k, z). A well-trained policy satisfies $|e| \lesssim 10^{-3}$ across the ergodic support of (k, z). Residuals that are systematically positive indicate over-saving; negative residuals indicate under-saving. In practice, common random numbers (a single draw of $\varepsilon^{(1:N)}$ shared across all grid points in k) are used so that curves at different z-levels are directly comparable.

C Jacobian Derivation

We derive the formula $\left.\frac{\partial k_{t+1}}{\partial k_t}\right|_{SS} = \frac{s_{ss}}{\beta}$ by computing the mechanical effect of a perturbation in k_t on next-period capital k_{t+1}, holding current-period actions (s_t, n_t) and productivity z_t fixed.

Step 1: State the transition equations

The capital accumulation equation is

$$k_{t+1} = s_t \, x_t, \tag{53}$$

where disposable resources are

$$x_t = (1 - \delta) k_t + y_t, \tag{54}$$

and output is given by the production function

$$y_t = z_t \, k_t^{\alpha} \, n_t^{1-\alpha}. \tag{55}$$

Substituting, the full transition is

$$k_{t+1} = s_t \left[(1 - \delta) k_t + z_t \, k_t^{\alpha} \, n_t^{1-\alpha} \right]. \tag{56}$$

Step 2: Compute the derivative holding actions fixed

We compute the partial derivative $\frac{\partial k_{t+1}}{\partial k_t}$ treating (s_t, n_t, z_t) as constants. Differentiating equation (56):

$$\frac{\partial k_{t+1}}{\partial k_t} = s_t \left[(1 - \delta) + \frac{\partial}{\partial k_t} \left(z_t \, k_t^{\alpha} \, n_t^{1-\alpha} \right) \right] \tag{57}$$

$$= s_t \left[(1 - \delta) + z_t \, \alpha \, k_t^{\alpha - 1} \, n_t^{1-\alpha} \right]. \tag{58}$$

Step 3: Recognize the marginal product of capital

The firm's first-order condition for capital implies that the rental rate equals the marginal product of capital:

$$r_t = \alpha \, z_t \, k_t^{\alpha-1} \, n_t^{1-\alpha}. \tag{59}$$

Substituting equation (59) into equation (58):

$$\frac{\partial k_{t+1}}{\partial k_t} = s_t \, (1 - \delta + r_t). \tag{60}$$

This formula holds at any point along the simulated trajectory. It shows that the direct effect of capital on future capital operates through two channels: the undepreciated fraction $(1 - \delta)$ and the marginal product r_t, both scaled by the savings share s_t.

Step 4: Evaluate at the steady state

At the deterministic steady state where all variables are constant and $z_{ss} = 1$, the household's Euler equation must hold:

$$1 = \beta \, (1 - \delta + r_{ss}). \tag{61}$$

Rearranging equation (61):

$$1 - \delta + r_{ss} = \frac{1}{\beta}. \tag{62}$$

Substituting equation (62) into equation (60) evaluated at steady state:

$$\left.\frac{\partial k_{t+1}}{\partial k_t}\right|_{SS} = s_{ss} \, (1 - \delta + r_{ss}) = s_{ss} \cdot \frac{1}{\beta} = \frac{s_{ss}}{\beta}. \tag{63}$$

Interpretation

The discount factor β and the gross return on capital $(1 - \delta + r_{ss})$ appear with offsetting effects: the household discounts at rate β, but capital appreciates at the gross rate $(1 - \delta + r_{ss}) = 1/\beta$ in equilibrium. These two forces cancel exactly in equation (63), leaving only the savings share s_{ss} as the net attenuation factor when propagating effects backward through time.

For the baseline calibration with $\beta = 0.99$, $\alpha = 0.36$, $\delta = 0.025$, and GHH preferences with $\psi = 2$ and $\varphi = 1$, the steady-state savings share is $s_{ss} \approx 0.932$. The Jacobian is therefore

$$\left.\frac{\partial k_{t+1}}{\partial k_t}\right|_{SS} \approx \frac{0.932}{0.99} \approx 0.941. \tag{64}$$

Extension: Policy feedback

The derivation above holds (s_t, n_t) fixed, isolating the *mechanical* effect of capital on future capital. In the full model, a change in k_t also induces changes in the policy: $\Delta s_t = \frac{\partial s_t}{\partial k_t} \Delta k_t$ and $\Delta n_t = \frac{\partial n_t}{\partial k_t} \Delta k_t$.

These policy-feedback terms enter the *total derivative* of k_{t+1} with respect to k_t:

$$\frac{dk_{t+1}}{dk_t} = \frac{\partial k_{t+1}}{\partial k_t} + \frac{\partial k_{t+1}}{\partial s_t}\frac{\partial s_t}{\partial k_t} + \frac{\partial k_{t+1}}{\partial n_t}\frac{\partial n_t}{\partial k_t}. \tag{65}$$

However, the Jacobian $\frac{\partial k_{t+1}}{\partial k_t}$ defined in equation (60) refers specifically to the first term: the direct sensitivity of the transition function holding the policy fixed. This is the quantity that governs gradient attenuation in backpropagation through time, because the backward pass of automatic differentiation computes partial derivatives with respect to each intermediate variable in the computation graph, not total derivatives that account for implicit dependencies.

D Why the Solution Algorithm Is Backpropagation Through Time

The stochastic gradient descent (SGD) agent maximizes expected lifetime utility,

$$J_T(\theta) = \mathbb{E}\left[\sum_{t=0}^{T-1} \beta^t u(c_t, n_t)\right], \tag{66}$$

where allocations are generated by a parameterized policy

$$(s_t, n_t) = \pi_\theta(k_t, z_t),$$

and the state evolves according to the endogenous law of motion

$$k_{t+1} = g\big(k_t, z_t, \pi_\theta(k_t, z_t)\big).$$

The key feature of this problem is that the policy parameters θ affect utility at all dates both directly, through contemporaneous consumption and labor, and indirectly, through future state variables. As a result, the gradient of lifetime utility with respect to θ requires accounting for how a marginal change in the policy today alters the entire future trajectory of the economy.

Formally, the gradient can be written as

$$\nabla_\theta J_T(\theta) = \sum_{t=0}^{T-1} \beta^t \left[\frac{\partial u_t}{\partial \theta} + \frac{\partial u_t}{\partial k_t}\frac{\partial k_t}{\partial \theta}\right], \tag{67}$$

where $\partial k_t/\partial\theta$ satisfies the recursion

$$\frac{\partial k_t}{\partial \theta} = \frac{\partial g_{t-1}}{\partial \theta} + \frac{\partial g_{t-1}}{\partial k_{t-1}}\frac{\partial k_{t-1}}{\partial \theta}. \tag{68}$$

Computing this gradient therefore requires propagating marginal effects backward along the simulated time path. The algorithm first performs a forward simulation of the economy under a given policy, storing the resulting computational graph, and then applies the chain rule backward

from period $T-1$ to period 0. Because the backward pass follows the temporal structure of the model, the algorithm is referred to as *backpropagation through time*.

Economically, backpropagation through time is the numerical analogue of the adjoint or costate recursion in dynamic optimization. As the planning horizon grows and learning noise vanishes, the gradient conditions implied by this algorithm converge to the Euler equations and intratemporal optimality conditions of the rational expectations equilibrium.

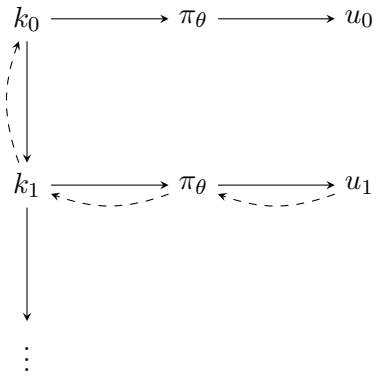


Figure 8: Forward simulation (solid arrows) and backward propagation of gradients through time (dashed arrows). The dashed arrows are bent to avoid overlapping the forward edges.