DSPG: Finding Heterogeneous-Agent Equilibra with Distribution-based Differentiable Policy Gradients

Anonymous Author(s)
Affiliation
Address
email

Abstract

We extend the Structural Reinforcement Learning (SRL[19]) framework by employing a U-Net to efficiently process the full distributional and structural information of a macroeconomic environment, solving in minutes a Heterogeneous-Agent Model that features both a non-trivial market-clearing condition and aggregate uncertainty—a class of macroeconomic models widely regarded as extremely challenging to solve. We formally show that our algorithm, DSPG (Distribution-based Structural Policy Gradient), obtains global nonlinear solutions without numerically relaxing the market-clearing condition or approximating aggregate shocks via perturbation. Additional experiments demonstrate that DSPG handles not only general-equilibrium but also partial-equilibrium models, and significantly outperforms conventional Deep Reinforcement Learning (DRL) baselines, including PPO, SAC, and DDPG. To our knowledge, DSPG is the first algorithm to obtain an accurate global solution of the Huggett [8] economy. Our results also confirm the validity of the SPG[19] algorithm under its dimensionality reduction and condition relaxation, whose solutions closely match ours. In the equilibrium, the simulation reproduces a negative relationship between interest rates and TFP shocks, capturing the general-equilibrium tension in a small open economy without domestic production, where aggregate shocks are absorbed entirely through interest-rate adjustments that clear the bond market by reconciling the competing responses of heterogeneous agents. Code is available at GitHub[1].

1 Introduction

Modern macroeconomists use heterogeneous-agent models to study important problems such as inequality, business cycles, and optimal policy[2]. However, as models become more complex, the difficulty of solving them increases substantially. Two features that are computationally challenging are *aggregate uncertainty* and *non-trivial market clearing*. Huggett [8] was the first to incorporate a non-trivial market-clearing condition into a heterogeneous-agent model and solved it using binary search. Krusell and Smith [10] introduced aggregate risk into an Aiyagari [1]-style model and proposed a numerical solution. However, each of these seminal models incorporates only one of the two challenging features. Economists have long found it difficult to solve models that combine aggregate uncertainty with non-trivial market clearing. As Moll [12] observes, this technical bottleneck has constrained the development of macroeconomic research; the natural aspiration is that researchers should be able to specify rich, complex models without being blocked by computational solvability.

[1]The entire codebase is written in JAX, natively supporting GPU-accelerated parallel computation.

[2]Heterogeneous-agent models are also used to study housing, health, the environment, and many other topics. See Sargent et al. [14], Huggett [8], Aiyagari [1], Krusell and Smith [10], Kaplan et al. [9], Dietz and Venmans [4], Sun [17].

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

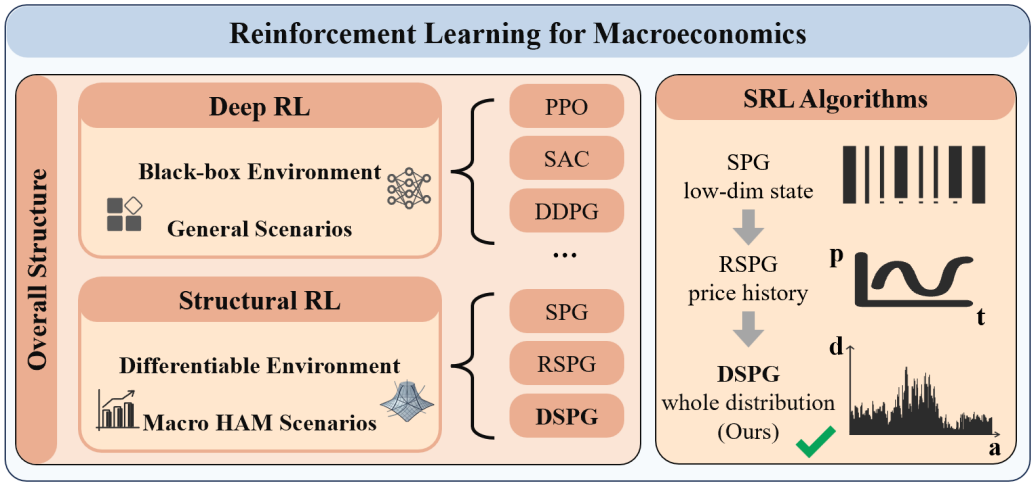


Figure 1: A general framework for reinforcement learning in macroeconomics. Conventional DRL assumes that the structural information of the environment is unknown, which facilitates generalization across diverse scenarios; in contrast, SRL exploits structural economic information in the HAM environment and improves solution efficiency by constructing a differentiable environment. The differences among SRL-style methods are as follows: (1) SPG uses a low-dimensional state space to simplify the problem; (2) RSPG augments the state with price history to strengthen the agent's decision-making capacity; (3) DSPG uses the full distribution as the state representation for transition dynamics, preserving both computational efficiency and mathematical rigor.

Recently, several groups have sought to tackle complex heterogeneous-agent models. Han et al. [7] solved the Krusell and Smith [10] model within an hour using deep neural networks and agent-based simulation. Auclert et al. [2] solved a complex HANK [9] model with aggregate risk using first-order perturbation to handle aggregate uncertainty. Azinovic et al. [3] proposed an equilibrium neural network training method that exploits full distribution information to solve heterogeneous-agent models. Gabriele et al. [5] applied several classical reinforcement learning algorithms to the Krusell and Smith [10] model and compared their performance. Yang et al. [19] relaxed rational expectations and proposed Structural Reinforcement Learning (SRL), solving several complex models within minutes. SRL is efficient and easy to use; however, the structural policy gradient (SPG) algorithm in the original paper relies on simplifying assumptions that shrink the state space and thereby constrain agent rationality. This simplification introduces a discrepancy between the model's theoretical state space and the one actually used in computation, which may compromise solution accuracy in models with richer dynamics. Wibault et al. [18] propose RSPG (RNN+SPG), which encodes the entire price history in the state, to address this issue in the Krusell and Smith [10] setting. Nevertheless, it remains an open question whether a model combining non-trivial market clearing with aggregate risk can be solved with both full state-space consistency and computational efficiency.

Therefore, in this paper we propose DSPG, a novel method to solve the Huggett [8] economy with aggregate risk. Despite its importance for understanding the interaction between macroeconomic uncertainty (i.e. business cycles and TFP shocks) and non-trivial market clearing (supply matching demand), this class of models remains difficult to solve with existing computational techniques. DSPG extends SPG by incorporating the full wealth distribution into the state representation, retaining computational efficiency while substantially improving equilibrium accuracy. Unlike DEQN [3], which also conditions on distribution information but relies on a different training paradigm, DSPG embeds the entire distribution in the state space within a differentiable policy-gradient framework, remaining computationally tractable. To handle the resulting high-dimensional inputs efficiently, we employ a one-dimensional U-Net—first introduced by Ronneberger et al. [13] for image segmentation—to parameterize agents' policy functions. In the Huggett [8] economy where non-trivial market clearing and aggregate risk coexist, DSPG recovers the optimal consumption–saving policy within minutes and achieves market-clearing errors below 10^{-7} on a server with a single NVIDIA RTX 3090 GPU.

Further experiments show that DSPG is consistent with SPG while reducing average market-clearing error significantly. To our knowledge, DSPG is the first method to obtain an accurate global non-

linear solution to the Huggett [8] economy in the simultaneous presence of aggregate uncertainty and non-trivial market clearing. Compared with classical DRL baselines[3] on a simplified benchmark environment, DSPG exhibits strong stability and efficiency[4]. Figure 1 summarizes how DSPG relates to existing DRL and SRL algorithms. Together with this paper, we release two benchmark models[5] for validating reinforcement learning algorithms in this setting. We believe that, owing to its efficiency and accuracy, DSPG can serve as an effective computational paradigm for solving heterogeneous-agent macroeconomic models.

2 Preliminaries

Structural Reinforcement Learning First introduced by Yang et al. [19] and later extended by Wibault et al. [18], SRL solves heterogeneous-agent models by combining RL with structural knowledge of the economic environment. Consider agents with individual state $s_t \in \mathcal{S}$ who choose actions $a_t \in \mathcal{A}$[6] according to a policy $\pi_\theta(a_t|s_t, p_t)$. The agent receives per-period payoff

$$\mathcal{R}_t = \mathcal{R}(s_t, a_t, p_t, z_t),$$

where z_t denotes aggregate shocks and p_t equilibrium prices. Individual states evolve according to known structural dynamics, usually a budget constraint,

$$s_{t+1} = f(s_t, a_t, z_t, p_t),$$

and agents maximize expected discounted returns[7]

$$J(\theta) = \lim_{T\to\infty} \mathbb{E}\left[\sum_{t=0}^{T} \beta^t \mathcal{R}(s_t, a_t, p_t, z_t)\right].$$

In heterogeneous-agent models, prices depend on the cross-sectional distribution of agents, which makes the environment high-dimensional. SRL addresses this by simulating equilibrium trajectories and directly optimizing the policy parameters θ. A key feature of SRL is that the individual transition function $f(\cdot)$ is known and differentiable. This makes the environment partially differentiable, allowing gradients of $J(\theta)$ to be computed by differentiating through the transition dynamics. As a result, policy parameters can be optimized using gradient-based methods rather than relying solely on likelihood-ratio policy gradients.

3 A Huggett Economy with Aggregate Uncertainty

Households There is a continuum of households indexed by $i \in [0, 1]$. Each household earns income $z_t e_{i,t}$ at time t, consisting of an aggregate component z_t and an idiosyncratic component $e_{i,t}$, both following Markov processes

$$z_{t+1} \sim \mathcal{T}_z(\cdot \mid z_t) \quad \textit{and} \quad e_{i,t+1} \sim \mathcal{T}_e(\cdot \mid e_{i,t}). \tag{1}$$

The wealth $a_{i,t}$ of household i at time t evolves according to

$$a_{i,t+1} + c_{i,t} = (1 + r_t)a_{i,t} + z_t e_{i,t}, \tag{2}$$

where $c_{i,t}$ is consumption and r_t is the interest rate. Households also face a borrowing constraint

$$a_{i,t} \geq \underline{a}, \tag{3}$$

where $\underline{a} \leq 0$. Each household chooses consumption $c_{i,t}$ to maximize lifetime expected utility

$$\max_{c_{i,t}} \lim_{T\to\infty} \mathbb{E}\left[\sum_{t=0}^{T} \beta^t u(c_{i,t})\right], \tag{4}$$

subject to (1), (2), and (3), taking as given the interest rate r_t.

[3] Including PPO [15], SAC [6], and DDPG [16]. A more detailed comparative study of classical DRL methods on the Krusell and Smith [10] HAM appears in TaxAI [11].

[4] These gains stem largely from the differentiable environment and the convexity of the optimization problem—features common to SRL methods. Note that although DSPG, RSPG and SPG differ algorithmically, they are all SRL methods.

[5] GE and PE versions of the dynamic Huggett [8] economy.

[6] Here a_t denotes a generic action variable, not to be confused with asset holdings $a_{i,t}$ defined in Section 3.

[7] In practice, we set a relatively large T cutoff so that the program can complete running.

Market We consider a closed economy with exogenous foreign demand for asset in which the domestic bond market interacts with an exogenous foreign sector. Each period, the foreign sector demands a fixed volume $B > 0$ of bonds from domestic households[8]—this can be interpreted as external sovereign or institutional borrowing, reflecting a constant foreign appetite for domestic safe assets—and repays principal plus interest at the equilibrium rate r_t. Here r_t is determined by a non-trivial market-clearing condition following Huggett [8]: it adjusts until aggregate domestic saving absorbs the foreign borrowing requirement,

$$\int_0^1 a_{i,t+1}\, di = B, \quad \forall t \geq 0. \tag{5}$$

Domestic households are price-takers with respect to both foreign demand and the interest rate. Since B is fixed and exogenous, the equilibrium r_t is fully pinned down by the cross-sectional distribution of domestic wealth—the hallmark of non-trivial market clearing in the Huggett [8] tradition.

Equilibrium Since the aggregate shock z_t renders the cross-sectional distribution non-degenerate as a payoff-relevant state, we adopt a recursive formulation. Let μ denote the joint distribution of individual states (a, e) across households, so that (z, μ) constitutes the aggregate state. A *recursive competitive equilibrium* consists of household policies $c(a, e; z, \mu),\ a'(a, e; z, \mu)$, a pricing function $r(z, \mu)$, and a law of motion $\mu' = H(\mu, z, z')$ such that: (i) given $r(\cdot)$ and $H(\cdot)$, the policies solve the household problem (4) subject to (2), (3), and (1)—equivalently, the value function satisfies the Bellman optimality condition $V(a, e; z, \mu) = \max_{a'}\{u(c) + \beta\, \mathbb{E}[V(a', e'; z', \mu') \mid e, z]\}$; (ii) price $r(z, \mu)$ clears the bond market (5); and (iii) H is induced by $a'(\cdot)$ and (1); for measurable subsets $\mathcal{B}, \mathcal{E}'$ of the asset and efficiency state spaces, $\mu'(\mathcal{B} \times \mathcal{E}') = \int \mathcal{T}_e(\mathcal{E}' \mid e)\, \mathbf{1}\{a'(a, e; z, \mu) \in \mathcal{B}\}\, d\mu(a, e)$, under which the joint process (z_t, μ_t) admits a unique ergodic distribution.[9]

What distinguishes this model from the original Huggett [8] economy is the presence of aggregate risk z_t, which renders the equilibrium interest rate r_t time-varying and only *approximately Markov*[10]. The economy settings ensure that no domestic production side absorbs the aggregate shock; instead, r_t must adjust endogenously to clear the bond market given the entire cross-sectional distribution, which is the central computational challenge.[11]

4 Our Method

4.1 Economic Dynamics Reformulation

Global solution Since aggregate shock z_t affects income level $z_t e_{i,t}$ through (2), the optimal policy of households depends on the cross-sectional distribution and the current aggregate state. In general, the optimal policy may depend on the entire history of distributions; however, when the aggregate shock z_t is Markov and the distribution $\mathbf{g}_t$ is a sufficient statistic for the cross-sectional state, we define the policy as a function of $(\mathbf{g}_t, z_t)$ alone. Since $\mathbf{g}_t$ lives on a finite grid of dimension $n_a \times n_e$, this yields a well-defined (though high-dimensional) state representation

$$c_t^* = \pi(\mathbf{g}_t, z_t) \quad \textit{where} \quad \mathbf{g}_t = \begin{bmatrix} d(a_1, e_1) & \cdots & d(a_1, e_{n_e}) \\ \cdots & & \cdots \\ d(a_{n_a}, e_1) & \cdots & d(a_{n_a}, e_{n_e}) \end{bmatrix} \quad \textit{and} \quad \sum_{i=1}^{n_a} \sum_{j=1}^{n_e} d(a_i, e_j) = 1, \tag{6}$$

where c_t^* denotes the optimal consumption policy evaluated on the full grid (yielding the matrix $\mathbf{c}_t \in \mathbb{R}^{n_a \times n_e}$ defined in (10)), $\mathbf{g}_t$ the distribution, $d(\cdot) \in (0, 1)$ the density of distribution, $\mathbf{a} = \{a_1, \cdots, a_{n_a}\}$ the wealth grid and $\mathbf{e} = \{e_1, \cdots, e_{n_e}\}$ the income grid. See Appendix A.2 for more details of discretization. (6) requests distribution $\mathbf{g}_t$ and aggregate shock z_t both in the households' state space to get the optimal global solution. Therefore, the decision variable of household is high-dimensional which makes the problem hard to solve.

[8] Note that this differs from the original Huggett setup where $B = 0$. This is to make the model definition cleaner; however, these two setups are entirely equivalent. When $\underline{a} < 0$, one can redefine $\tilde{a} = a - \underline{a}$ as the distance from the borrowing constraint, and then the model would revert to our definition.

[9] Our method targets the policy and pricing functions evaluated along this ergodic measure. In the notation of Section 2, the individual state is $s = (a, e)$, the action is c, the equilibrium price is $p \equiv r$, and the per-period payoff is $\mathcal{R} = u(c)$.

[10] In the Krusell and Smith [10] economy, prices are approximately AR(1), since tomorrow's prices (r_{t+1}, w_{t+1}) can be predicted accurately from current aggregate capital K_t and aggregate state z_t alone; see Appendix B.1 for the model and Appendix B.2 for the solution method.

[11] See Appendix A.1 for the model parameters.

Non-trivial market clearing From (2) and (5), we know that the aggregate saving is

$$S_t(\mathbf{g}_t, z_t, r_t) = \int_0^1 a_{i,t+1}\, di = \int_0^1 [(1+r_t)a_{i,t} + z_t e_{i,t} - c_{i,t}]\, di = B, \tag{7}$$

where S_t is the saving supply curve. To express (7) in vector form, we define the following notation on the discretized economy: let $\mathbf{a} \in \mathbb{R}^{n_a}$ denote the wealth grid vector, $\mathbf{g}_t \in \mathbb{R}^{n_a \times n_e}$ the distribution matrix, and $\mathbf{c}_t \in \mathbb{R}^{n_a \times n_e}$ the consumption policy matrix. We further write $\mathbf{a} \odot \mathbf{g}_t \equiv \sum_{i=1}^{n_a} \sum_{j=1}^{n_e} a_i \cdot d(a_i, e_j)$ for the cross-sectional mean of wealth (element-wise multiplication followed by summation, broadcasting $\mathbf{a}$ across income states); $\mathbf{c}_t \odot \mathbf{g}_t$ is defined analogously for consumption. We adopt this notation throughout the paper. Then we have

$$\mathbf{a} \odot \mathbf{g}_{t+1} = S_t(\mathbf{g}_t, z_t, r_t) = (1+r_t) \cdot \mathbf{a} \odot \mathbf{g}_t + z_t \cdot \overline{e} - \mathbf{c}_t \odot \mathbf{g}_t = B, \tag{8}$$

where $\overline{e}$ is the ergodic mean of idiosyncratic income $e_{i,t}$ and $\mathbf{c}_t$ is the consumption policy vector over distribution. Then, we can directly get the interest rate that clears the market

$$1 + r_t = \frac{B - z_t \cdot \overline{e} + \mathbf{c}_t \odot \mathbf{g}_t}{\mathbf{a} \odot \mathbf{g}_t}. \tag{9}$$

Equation (9) shows that the market-clearing interest rate r_t can be obtained in closed form from a single formula[12]. This is possible because the full cross-sectional distribution $\mathbf{g}_t$ is included in the state space: given $\mathbf{g}_t$ and the policy $\mathbf{c}_t$, all aggregate quantities required for market clearing are directly computable, so r_t is determined without any iterative root-finding or fixed-point loop.

4.2 Distribution-based Structural Policy Gradient (DSPG)

U-net policy function From (6), the state space (input) is a distribution of dimension $n_a \times n_e$ and the policy (output) is the consumption value at each grid point. We parameterize this mapping with a 1D U-Net for three reasons: (i) the encoder–decoder structure captures both local interactions between neighboring wealth grid points and global distributional features through progressively coarser representations; (ii) skip connections preserve fine-grained grid-level information that would otherwise be lost in deep networks, which is critical for satisfying the borrowing constraint at specific grid points; (iii) the convolutional architecture exploits the ordered structure of the wealth grid, sharing parameters across grid points—a fully connected (MLP) network would require $\mathcal{O}((n_a n_e)^2)$ parameters in its first layer alone, making it computationally prohibitive for high-dimensional grids, whereas the U-Net scales linearly in $n_a n_e$. The policy network is thus

$$\mathbf{c}_t = \textit{U-net}(\mathbf{g}_t, z_t; \Theta) = \begin{bmatrix} c(a_1, e_1) & \cdots & c(a_1, e_{n_e}) \\ \cdots & & \cdots \\ c(a_{n_a}, e_1) & \cdots & c(a_{n_a}, e_{n_e}) \end{bmatrix}, \tag{10}$$

where Θ is the parameters of the 1D U-net and the detailed structure is in Appendix E.

Differentiable learning Unlike classical reinforcement learning algorithms, we create a differentiable environment where agents know the analytical policy gradient of the cumulative utility (4), and directly optimize the objective function by using the loss function

$$\mathbb{J}(\Theta) = -\int_0^1 \mathbb{E}\Big[\sum_{t=0}^{\infty} \beta^t u(c_{i,t})\Big] di \;\approx\; -\frac{1}{n} \sum_{m=1}^{n} \sum_{t=0}^{T} \beta^t \int_0^1 u(c_{i,t})\, di, \tag{11}$$

where the approximation truncates the infinite horizon to T periods and replaces the expectation with an empirical average over n sampled trajectories. (11) can also be written in vector form

$$\mathbb{J}(\Theta) \approx -\frac{1}{n} \sum_{m=1}^{n} \sum_{t=0}^{T} \beta^t\, u(\mathbf{c}_t) \odot \mathbf{g}_t, \tag{12}$$

which is efficient when using JAX and GPU acceleration in practice. Empirically, we observe that the loss landscape of (12) behaves as a convex optimization problem: training consistently converges

[12]Note that this formula is a price update formula, not a price definition formula. It represents the clearing price for the current period. Of course, under equilibrium, it is the equilibrium price.

to the same solution from different random initializations, which is consistent with the concavity of $u(\cdot)$ and the linearity of the budget constraint.

Every step of the DSPG computation graph—U-Net evaluation (10), closed-form market clearing (9), the budget-constraint wealth update, and the distributional transition—is a differentiable function of its inputs. Conditional on an exogenous shock realization $\{z_t\}$, reverse-mode automatic differentiation therefore yields $\nabla_\Theta \mathbb{J}(\Theta)$ *exactly*, with no score-function (REINFORCE) term (Appendix F, Proposition F.3). The only stochasticity in our estimator comes from averaging over n independent shock trajectories, so its variance decays at the standard $O(1/n)$ rate (Proposition F.5); truncating the horizon at T introduces a bias bounded by $M\beta^{T+1}/(1-\beta)$ (equation (36)), which is negligible for the horizons used in our experiments.

Monotonicity-preserving parameterization In consumption-saving models, the optimal consumption policy is monotonically increasing in wealth. To enforce this structural property and prevent the network architecture from producing non-monotone policies, we parameterize the first-order difference rather than the level of consumption:

$$\textit{U-net}(\mathbf{g}_t, z_t; \Theta) = \begin{bmatrix} c(a_1,e_1) & \cdots & c(a_1,e_{n_e}) \\ \delta(a_2,e_1) & \cdots & \delta(a_2,e_{n_e}) \\ \vdots & \ddots & \vdots \\ \delta(a_{n_a},e_1) & \cdots & \delta(a_{n_a},e_{n_e}) \end{bmatrix} \quad \textit{and} \quad c(a_i,e_j) = c(a_1,e_j) + \sum_{k=2}^{i} \delta(a_k,e_j), \tag{13}$$

where $\delta(a_i,e_j) = c(a_i,e_j) - c(a_{i-1},e_j) \geq 0$ with $i \geq 2$ and $j \geq 1$ is the first-step difference of the policy vector along wealth grids. Note that (13) is equivalent to (10). In order to ensure stability at the beginning of the training, we firstly solve a Huggett [8] model without aggregate uncertainty (usually called steady state and easy to solve) and pre-train the U-net to fit the optimal policy function of the steady state. Figure 2 shows the steady state results, where the policy is used to initialize the network and the distribution is used to initialize the simulation.

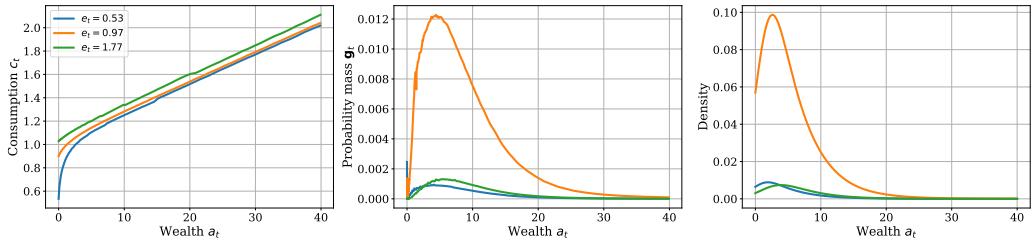


Figure 2: Steady-state solution of the Huggett [8] economy. **Left**: optimal consumption policy as a function of wealth for each idiosyncratic income level. **Middle**: probability mass function of the ergodic wealth distribution. **Right**: kernel density estimate (KDE) of the same distribution.

Relationships to SPG and RSPG All three algorithms—SPG, RSPG, and DSPG—belong to the SRL family and share a differentiable environment for variance-free gradient computation. They differ in the state representation fed to the policy function:

- **SPG** [19] uses a low-dimensional state space (a_t, e_t, r_t, z_t), which is a summary of the distribution μ_t, being computational friendly.
- **RSPG** [18] augments the state space with a history of prices $\{r_{t-\ell}, \ldots, r_t\}$ encoded by an RNN, partially recovering distributional information through price signals.
- **DSPG** directly conditions on the full distribution μ_t, preserving all cross-sectional information at the cost of a higher-dimensional input, which we handle via the U-Net architecture.

As shown in Section 5, DSPG retains SRL-level efficiency—solving the model within minutes—while targeting the exact global equilibrium; the close agreement with SPG solutions in turn validates SPG's approximation. Beyond SRL, DSPG produces global nonlinear solutions (unlike perturbation methods [2]), optimizes through a differentiable simulation rather than an equilibrium network (unlike DEQN [3]), and eliminates score-function variance relative to classical DRL baselines (Appendix F). A limitation is the fixed-grid cost $\mathcal{O}(n_a n_e)$: for very high-dimensional distributions,

backpropagation through T steps may exhaust GPU memory regardless of the network architecture, motivating future work on adaptive distributional representations.

4.3 Algorithm Summary

Algorithm 1 DSPG: Distribution-based Structural Policy Gradient

Phase 1: Steady-state pre-training
Solve the steady-state Huggett [8] economy (no aggregate risk) via Value Iteration
Pre-train U-Net parameters Θ_0 to fit the steady-state consumption policy
Initialize distribution $\mathbf{g}_0$ from the steady-state distribution
Phase 2: Policy optimization
for iteration $k = 1, 2, \ldots, K$ **do**
  for trajectory $m = 1, \ldots, n$ **do**
    Sample aggregate shock sequence $\{z_t\}_{t=0}^{T}$ from $\mathcal{T}_z$
    for $t = 0, 1, \ldots, T$ **do**
      Compute consumption policy: $\mathbf{c}_t = \text{U-Net}(\mathbf{g}_t, z_t; \Theta_k)$ ▷ Eq. (10)
      Compute equilibrium interest rate r_t ▷ Eq. (9)
      Update distribution $\mathbf{g}_{t+1}$ via budget constraint and idiosyncratic transition $\mathcal{T}_e$
    end for
  end for
  Compute loss $\mathbb{J}(\Theta_k)$ by averaging over n trajectories ▷ Eq. (12)
  Update $\Theta_{k+1} \leftarrow \Theta_k - \alpha \nabla_\Theta \mathbb{J}(\Theta_k)$ via backpropagation through differentiable dynamics
end for
return Trained policy parameters Θ_K

Implementation details The entire pipeline is implemented in JAX for GPU-accelerated differentiable computation. All training hyperparameters—including learning rate, number of trajectories n, and truncation length T—are reported in Appendix D.1. The 1D U-Net architecture details (number of levels, channel widths) are provided in Appendix E; model calibration parameters are in Appendix A.1. Each forward pass evaluates the U-Net and computes r_t in $\mathcal{O}(n_a n_e)$; the backward pass through T time steps has complexity $\mathcal{O}(T \cdot n_a n_e \cdot |\Theta|)$. Memory scales linearly with T due to storing intermediate activations for backpropagation.

5 Experiments

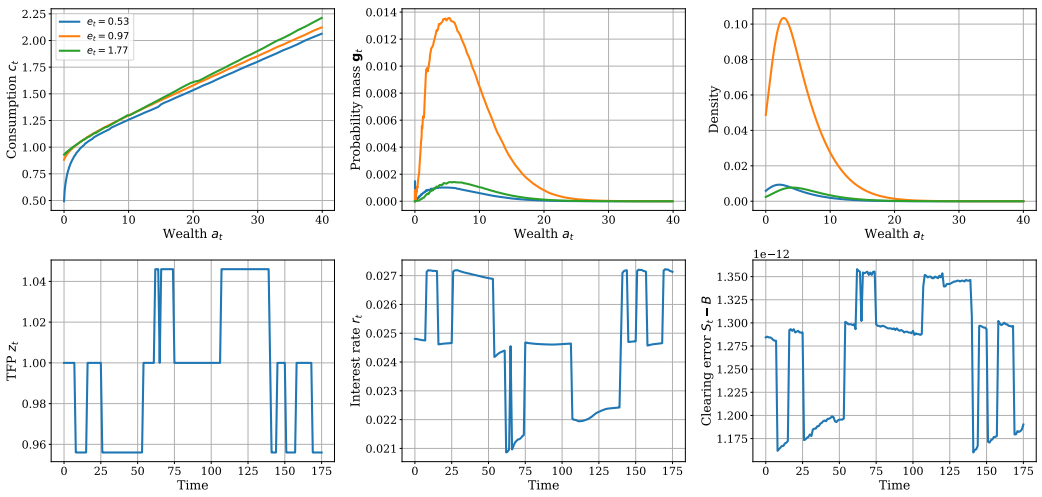


Figure 3: DSPG solution of the Huggett [8] economy with aggregate risk. **Top row**: optimal consumption policy, probability mass, and KDE density of the ergodic distribution. **Bottom row**: a simulated path of aggregate shock z_t, equilibrium interest rate r_t, and market clearing error $S_t - B$.

5.1 Optimal Policy

Figure 2 shows the steady-state solution used for initialization. Figure 3 presents the DSPG solution of the full model with aggregate risk: the policy and ergodic distribution closely match those of the steady-state economy under the same calibration, while producing even smoother curves than value iteration. The simulation path reveals a negative relationship between the equilibrium interest rate r_t and TFP z_t, and aggregate saving residuals remain below 10^{-10} throughout—both consistent with economic theory.

We also ablate the number of trajectories n per update (Figure 4). Increasing n from 32 to 512 progressively reduces training variance at the cost of longer wall-clock time; $n = 128$ offers a practical balance. All three settings converge to comparable final loss values. Other hyperparameters are in Appendix D.1.

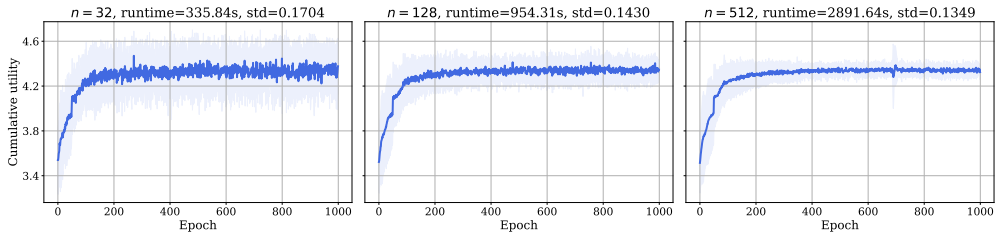


Figure 4: Ablation on the number of trajectories $n \in \{32, 128, 512\}$ (10 independent runs each). **Left**: training loss curves (solid line: mean; shaded: ± 1 std). **Middle**: wall-clock time per run. **Right**: final training variance. Larger n reduces variance but increases computation.

Together, these results demonstrate that DSPG solves the Huggett [8] economy with aggregate risk and positive bond supply in a stable, accurate, and robust manner.

5.2 Economic Findings

Table 1 reports the equilibrium interest rate r_t conditional on each TFP level z_t, revealing a clear negative relationship. The mechanism is intuitive: classify households with $a_{i,t} > B$ as net savers and those with $a_{i,t} < B$ as net borrowers. When TFP is high, all households' income rises, increasing the aggregate supply of savings relative to borrowing demand; the interest rate falls to restore market clearing. Conversely, when TFP is low, saving supply contracts and the interest rate must rise to attract sufficient lending. In short, r_t adjusts to reconcile the shifting balance between aggregate saving and borrowing, generating the observed negative co-movement with z_t.

Table 1: Equilibrium interest rate r_t vs TFP shock z_t

TFP shock z_t	0.91	0.96	1.00	1.05	1.09
AVG r_t	0.0283	0.0267	0.0250	0.0222	0.0188
STD r_t	0.0007	0.0006	0.0008	0.0011	0.0016

5.3 Comparision to SPG

Figure 5 compares the mean ergodic distributions produced by DSPG and SPG[13]. The DSPG distribution is very close to that of SPG yet noticeably smoother, reflecting the continuity of the U-Net policy in contrast to SPG's grid-based interpolation. Figure 6 shows that the aggregate consumption paths C_t also agree closely. Quantitatively, DSPG achieves average market-clearing residuals below 10^{-11}, compared with roughly 10^{-5} for SPG—an improvement of six orders of magnitude. To our knowledge, this is the first high-accuracy global solution of this economy.

[13] SPG [19] was previously the only effective algorithm for this model and yields solutions close to the rational-expectations equilibrium under its bounded-rationality assumptions, making it a natural reference.

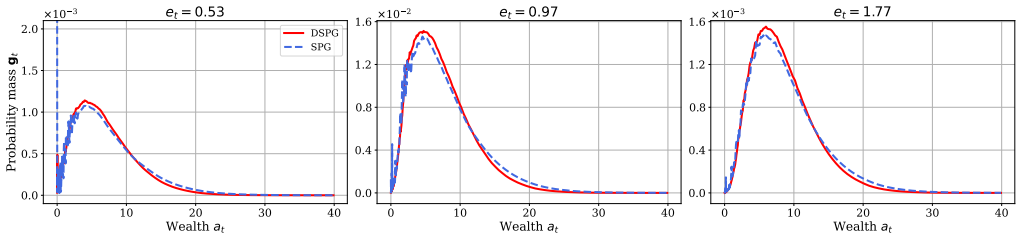


Figure 5: Mean ergodic distribution $\mathbf{g}_t$: DSPG (ours) vs. SPG. The two distributions are nearly indistinguishable, with DSPG producing smoother curves.

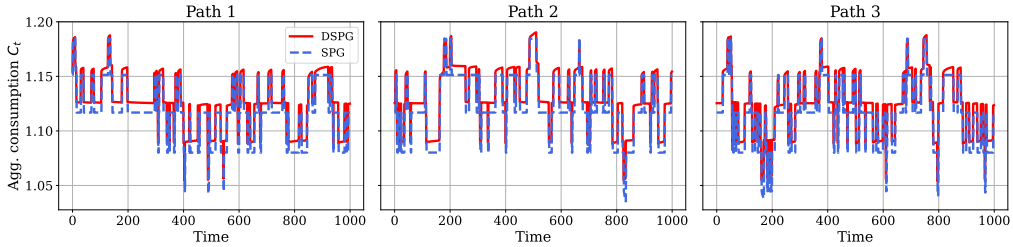


Figure 6: Aggregate consumption C_t: DSPG (ours) vs. SPG. Under three fixed representative TFP shocks, the total consumption paths from both methods closely agree.

5.4 Validation

We compare DSPG with conventional DRL algorithms—PPO, SAC, and DDPG—on the partial-equilibrium (PE) version of our economy, where a value-function iteration (VFI) solution serves as the ground-truth baseline (see Appendix C.1 for the PE setup, Appendix C.2 for the VFI solution, and Appendix C.3 for the rationale of using the PE benchmark). Table 2 and Figure 7 show that DSPG converges faster, with lower variance and closer final performance to VFI than all three DRL baselines. Hyperparameters for each method are reported in Appendices D.2–D.5.

Table 2: Summary of performance

Algorithm	Last eval		Best eval	
	AVG	STD	AVG	STD
DSPG	**5.661**	**0.003**	5.732	**0.001**
PPO	5.128	0.132	**5.910**	0.067
SAC	4.266	0.060	4.716	0.019
DDPG	4.502	0.501	5.225	0.221
VFI	5.653	—	5.653	—

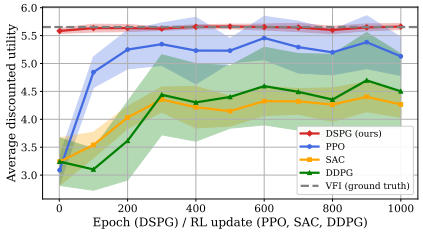


Figure 7: Training curves on the PE model (10 runs; shaded: ± 1.96 std).

6 Conclusion

We introduced DSPG, an efficient distribution-based structural policy gradient method that solves heterogeneous-agent models with both aggregate uncertainty and non-trivial market clearing. By conditioning a U-Net policy on the full cross-sectional distribution and computing exact path-wise gradients through a differentiable simulation, DSPG obtains global nonlinear equilibria within minutes—without linearizing around a steady state or relaxing the market-clearing condition. On the dynamic Huggett [8] economy with aggregate risk, DSPG achieves market-clearing residuals below $1e-7$, improving upon prior SRL methods by two orders of magnitude. To our knowledge, this is the first high-accuracy global solution of this economy. Future work includes extending DSPG to richer environments such as HANK models, improving scalability via adaptive or mesh-free distributional representations, and establishing formal convergence guarantees under NN parameterization.

References

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

[2] Adrien Auclert, Bence Bardóczy, Matthew Rognlie, and Ludwig Straub. Using the sequence-space Jacobian to solve and estimate heterogeneous-agent models. *Econometrica*, 89(5):2375–2408, 2021.

[3] Marlon Azinovic, Luca Gaegauf, and Simon Scheidegger. Deep equilibrium nets. *International Economic Review*, 2022.

[4] Simon Dietz and Frank Venmans. Cumulative carbon emissions and economic policy: in search of general principles. *Journal of Environmental Economics and Management*, 96:108–129, 2019.

[5] Federico Gabriele, Aldo Glielmo, and Marco Taboga. Heterogeneous rbcs via deep multi-agent reinforcement learning. *arXiv preprint arXiv:2510.12272*, 2025.

[6] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

[7] Jiequn Han, Yucheng Yang, and Weinan E. DeepHAM: A global solution method for heterogeneous agent models with aggregate shocks. *arXiv preprint arXiv:2112.14377*, 2021.

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

[9] Greg Kaplan, Benjamin Moll, and Giovanni L Violante. Monetary policy according to HANK. *American Economic Review*, 108(3):697–743, 2018.

[10] Per Krusell and Anthony A Smith, Jr. Income and wealth heterogeneity in the macroeconomy. *Journal of Political Economy*, 106(5):867–896, 1998.

[11] Qirui Mi, Siyu Xia, Yan Song, Haifeng Zhang, Shenghao Zhu, and Jun Wang. Taxai: A dynamic economic simulator and benchmark for multi-agent reinforcement learning. *arXiv preprint arXiv:2309.16307*, 2023.

[12] Benjamin Moll. The trouble with rational expectations in heterogeneous agent models: A challenge for macroeconomics. *Economic Journal Lecture, Royal Economic Society*, 2024.

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[14] Thomas J Sargent, Christopher A Sims, et al. Business cycle modeling without pretending to have too much a priori economic theory. *New methods in business cycle research*, 1:145–168, 1977.

[15] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[16] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. Pmlr, 2014.

[17] Jeffrey Sun. The distributional consequences of climate change: The role of housing wealth, expectations, and uncertainty. Technical report, Princeton University, 2023.

[18] Clarisse Wibault, Johannes Forkel, Sebastian Towers, Tiphaine Wibault, Juan Duque, George Whittle, Andreas Schaab, Yucheng Yang, Chiyuan Wang, Michael Osborne, et al. Recurrent structural policy gradient for partially observable mean field games. *arXiv preprint arXiv:2602.20141*, 2026.

[19] Yucheng Yang, Chiyuan Wang, Andreas Schaab, and Benjamin Moll. Structural reinforcement learning for heterogeneous agent macroeconomics, 2025. URL `https://arxiv.org/abs/2512.18892`.

Appendix

A Details of the Huggett Economy

A.1 Calibration

Following the literature, we set the household discount factor $\beta = 0.975$ and utility function $u(c) = \frac{c^{1-\sigma}}{1-\sigma}$ with $\sigma = 1$. Then, we set borrowing constraint $\underline{a} = 0$ and total bond supply $B = 5$, a setting that is different from the original [8] model. The idiosyncratic shock follows a three-state Markov chain to approximate the log AR(1) process

$$log(e_{i,t+1}) = \rho_e \cdot log(e_{i,t}) + \nu_e \cdot \sqrt{1-\rho_e^2} \cdot \epsilon_{i,t}, \quad \epsilon_{i,t} \sim N(0,1), \tag{14}$$

where the persistence $\rho_e = 0.6$ and the variance $\nu_e = 0.2$, same as [1] model. The grids $\mathbf{e}$ and transition matrix $\mathbf{P_e}$ of idiosyncratic shock are

$$\mathbf{e} = \begin{bmatrix} 0.5343 \\ 0.9735 \\ 1.7739 \end{bmatrix} \quad \textit{and} \quad \mathbf{P_e} = \begin{bmatrix} 0.6460 & 0.3539 & 0.0001 \\ 0.0304 & 0.9392 & 0.0304 \\ 0.0001 & 0.3539 & 0.6460 \end{bmatrix}, \tag{15}$$

with $\mathbf{e}$ and $\mathbf{P_e}$ rounded to 4 decimals and the ergodic mean average $\bar{e}$ is 1.

The aggregate shock follows a five-state Markov chain to approximate the log AR(1) process similar to (14)

$$log(z_{t+1}) = \rho_z \cdot log(z_t) + \nu_z \cdot \sqrt{1-\rho_z^2} \cdot \epsilon_t, \quad \epsilon_t \sim N(0,1), \tag{16}$$

where the persistence $\rho_z = 0.9$ and $\nu_z = 0.03$ to make the aggregate shock more steady and controllable, thereby reducing the difficulty of solving. The grids $\mathbf{z}$ and transition matrix $\mathbf{P_z}$ of aggregate shock are

$$\mathbf{z} = \begin{bmatrix} 0.9139 \\ 0.9560 \\ 1.0000 \\ 1.0460 \\ 1.0942 \end{bmatrix} \quad \textit{and} \quad \mathbf{P_z} = \begin{bmatrix} 0.8478 & 0.1522 & 0.0000 & 0.0000 & 0.0000 \\ 0.0195 & 0.8962 & 0.0843 & 0.0000 & 0.0000 \\ 0.0000 & 0.0427 & 0.9147 & 0.0427 & 0.0000 \\ 0.0000 & 0.0000 & 0.0843 & 0.8962 & 0.0195 \\ 0.0000 & 0.0000 & 0.0000 & 0.1522 & 0.8478 \end{bmatrix}, \tag{17}$$

with $\mathbf{z}$ and $\mathbf{P_z}$ rounded to 4 decimals.

A.2 Discretization

There are two heterogeneous attributes, wealth $a_{i,t}$ and income $e_{i,t}$. The latter has already been discretized to $n_e = 3$ states. We discretize wealth to $n_a = 200$ grid points with $a_{\texttt{min}} = \underline{a} = 0$ and $a_{\texttt{max}} = 150$. So the distribution $\mathbf{g_t}$ is a vector with $n_a \times n_e = 600$ dimensions. There are $n_z = 5$ different aggregate states.

B Krusell and Smith [10] model

B.1 Model Setup

Households There's a continuum of households indexed by $i \in [0,1]$ and each household has productivity level $e_{i,t}$ and wealth $a_{i,t}$. Productivity level (idiosyncratic shock) $e_{i,t}$ follows a three-state Markov process which is the same as (15).

The aggregate shock z_t follows a two-state Markov process

$$\mathbf{z} = \begin{bmatrix} z_b & z_g \end{bmatrix} = \begin{bmatrix} 0.99 & 1.01 \end{bmatrix} \quad \textit{and} \quad \mathbf{P_z} = \begin{bmatrix} 0.875 & 0.125 \\ 0.125 & 0.875 \end{bmatrix}, \tag{18}$$

where $\mathbf{z}$ is the grid of aggregate shock and $\mathbf{P_z}$ is the transition matrix.

Wealth $a_{i,t}$ evolves according to the budget constraint

$$a_{i,t+1} + c_{i,t} = (1 + r_t)a_{i,t} + w_t e_{i,t}, \tag{19}$$

where r_t is interest rate and w_t is wage. Households take prices as given and live permanently to optimize cumulative utility

$$\max_{c_{i,t}} \mathbb{E}_0[\sum_{t=0}^{\infty} \beta^t u(c_{i,t})], \tag{20}$$

where $\beta = 0.975$ is the discount factor and $u(c) = \frac{c^{1-\sigma}}{1-\sigma}$ with $\sigma = 1$ is the utility function.

Firm There is a representative firm who borrows capital from households and produces goods according to Cobb-Douglas function

$$Y_t = z_t K_t^{\alpha} L_t^{1-\alpha}, \tag{21}$$

where K_t is aggregate capital, L_t is aggregate labor supply and $\alpha = 0.36$ is capital share. Firm use FOC condition to determine interest rate and wage

$$r_t = \frac{\partial Y_t}{\partial K_t} - \delta = \alpha z_t k_t^{\alpha-1} - \delta, \quad w_t = \frac{\partial Y_t}{\partial L_t} = (1-\alpha) z_t k_t^{\alpha}, \tag{22}$$

where $\delta = 0.025$ is the capital depreciation rate and $k_t = \frac{K_t}{L_t}$ is capital per capita.

Equilibrium Households maximize cumulative utility (20) subject to the budget constraint (19) and taking as given prices according to firm's optimality condition (22).

B.2 KS Method

From (22), we know that as long as we get the knowledge of the law of motion of aggregate capital K_t, we get the knowledge of interest rate r_t and wage w_t. So in [10] method, we assume the aggregate capital K_t evolves according to

$$log(K_{t+1}) = \begin{cases} x_g \cdot log(K_t) + y_g, & z_t = z_g \\ x_b \cdot log(K_t) + y_b, & z_t = z_b \end{cases}, \tag{23}$$

where $\theta = \begin{bmatrix} x_g & y_g \\ x_b & y_b \end{bmatrix}$ is unknown parameter. Then, there is a fixed point problem between unknown parameter θ and optimal consumption policy π^*.

Given parameter $\theta \in \mathbb{R}^{2\times 2}$, we can find the corresponding optimal policy $\pi^*(\theta)$ using Value Iteration, for all transition dynamics are known. And at the same time, given a policy function π, we can simulate the economy to get massive samples of (K_t, z_t, K_{t+1}) and then run regression of (23) to get optimal parameter $\theta(\pi)$ that satisfy (23). So as long as we can find

$$\theta = \theta(\pi^*), \quad \pi^* = \pi^*(\theta), \tag{24}$$

we get the solution of Krusell and Smith [10] model.

KS method can be extended to solve nearly all heterogeneous agent models. However, not all models exist a fixed point including what we solved in this paper. Economists have tried many function forms of (23) to solve the Huggett [8] model with aggregate risk, yet all failed. A potential reason why there does not exist a fixed point is the sensitiveness of the market clearing price.

C A PE Model of the Huggett [8] Economy

C.1 Model Setup

Households There's a representative household with wealth a_t and productivity level e_t at period t. There are two prices, interest rate r_t and wage w_t, that evolves exogenously. Productivity level e_t, interest rate r_t and wage w_t all follow a Markov process,

$$e_{t+1} \sim \mathcal{T}_e(\cdot \mid e_t), \quad r_{t+1} \sim \mathcal{T}_r(\cdot \mid r_t) \quad \textit{and} \quad w_{t+1} \sim \mathcal{T}_w(\cdot \mid w_t), \tag{25}$$

where we discretize $\mathcal{T}_e$ to three states, $\mathcal{T}_r$ to five states and $\mathcal{T}_w$ to seven states.

The discrete values and transition matrix of e_t are

$$\mathbf{e} = \begin{bmatrix} 0.5343 \\ 0.9735 \\ 1.7739 \end{bmatrix} \quad \textit{and} \quad \mathbf{P_e} = \begin{bmatrix} 0.6460 & 0.3539 & 0.0001 \\ 0.0304 & 0.9392 & 0.0304 \\ 0.0001 & 0.3539 & 0.6460 \end{bmatrix}, \tag{26}$$

where we use the same process of our original model.

The discrete values and transition matrix of r_t are

$$\mathbf{r} = \begin{bmatrix} 0.0193 \\ 0.0218 \\ 0.0247 \\ 0.0279 \\ 0.0315 \end{bmatrix} \quad \textit{and} \quad \mathbf{P_r} = \begin{bmatrix} 0.5655 & 0.4172 & 0.0173 & 0.0000 & 0.0000 \\ 0.0920 & 0.6079 & 0.2966 & 0.0035 & 0.0000 \\ 0.0034 & 0.1423 & 0.7032 & 0.1507 & 0.0004 \\ 0.0000 & 0.0056 & 0.2287 & 0.7131 & 0.0526 \\ 0.0000 & 0.0000 & 0.0081 & 0.3438 & 0.6480 \end{bmatrix}, \tag{27}$$

with the ergodic interest rate $\bar{r} = 2.5\%$.

The discrete values and transition matrix of w_t are

$$\mathbf{w} = \begin{bmatrix} 0.9418 \\ 0.9608 \\ 0.9802 \\ 1.0000 \\ 1.0202 \\ 1.0408 \\ 1.0618 \end{bmatrix} \quad \textit{and} \quad \mathbf{P_w} = \begin{bmatrix} 0.6657 & 0.3312 & 0.0031 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0541 & 0.7002 & 0.2442 & 0.0014 & 0.0000 & 0.0000 & 0.0000 \\ 0.0001 & 0.0842 & 0.7363 & 0.1787 & 0.0007 & 0.0000 & 0.0000 \\ 0.0000 & 0.0003 & 0.1254 & 0.7487 & 0.1254 & 0.0003 & 0.0000 \\ 0.0000 & 0.0000 & 0.0007 & 0.1787 & 0.7363 & 0.0842 & 0.0001 \\ 0.0000 & 0.0000 & 0.0000 & 0.0014 & 0.2442 & 0.7002 & 0.0541 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0031 & 0.3312 & 0.6657 \end{bmatrix}, \tag{28}$$

with the ergodic wage $\bar{w} = 1$.

Wealth a_t evolves according to the budget constraint

$$a_{t+1} + c_t = (1 + r_t)a_t + w_t e_t, \tag{29}$$

where we use the natural borrowing constraint $a_t \geq 0$ for all t. And the representative household optimizes the cumulative utility

$$\max_{c_t} \mathbb{E}_0[\sum_{t=0}^{\infty} \beta^t u(c_t)], \tag{30}$$

where we use the same parameters of $\beta = 0.975$ and $u(c) = \frac{c^{1-\sigma}}{1-\sigma}$ with $\sigma = 1$.

C.2 Numerical Solution (Value Iteration)

Since all transition dynamics are known, we use Value Iteration to solve it. Given the current state (a_t, w_t, r_t, e_t), we know the transition probability of (e_t, r_t, w_t) follows (25) and the transition dynamic of a_t follows (29). Then, we can write the Bellman euqation

$$\begin{aligned} \max_{c_t} \quad & u(c_t) + \beta \mathbb{E}[V(a_{t+1}, e_{t+1}, r_{t+1}, w_{t+1}) \mid (a_t, e_t, r_t, w_t)] \\ \text{s.t.} \quad & a_{t+1} = (1 + r_t)a_t + w_t e_t - c_t \\ & e_{t+1} \sim \mathcal{T}_e(\cdot \mid e_t) \\ & r_{t+1} \sim \mathcal{T}_r(\cdot \mid r_t) \\ & w_{t+1} \sim \mathcal{T}_w(\cdot \mid w_t) \end{aligned}, \tag{31}$$

therefor we can solve the optimal control by using Value Iteration.

C.3 Why PE the baseline?

The reason why we use PE as the baseline is we want to compare our DSPG algorithm with conventional deep reinforcement learning algorithms, including PPO, SAC and DDPG. PE is a real low-dimensional single-agent problem which can be easily extended as a reinforcement learning environment and solved by existing open-source RL repos. Though we discussed in previous section that our complex environment is also equivalent to a single-agent problem, it's high-dimensional. The dimension of inputs and outputs are several hundreds, which is difficult for current open-source DRL repos to handle. So instead we use PE as the baseline.

D Hyper-parameters

D.1 USPG (Huggett Economy)

Symbol	Description	Value
n_a	Number of asset grid points	200
$a_{\min}$, $a_{\max}$	Borrowing limit and asset upper bound	0, 150
n_e, n_z	Labor productivity states, TFP states	3, 5
β	Discount factor	0.975
σ	Coefficient of relative risk aversion	1 (log utility)
B	Total bond supply (market clearing)	5
$\varepsilon_{\text{trunc}}$	Truncation error for infinite horizon	10^{-2}
T	Training rollout length	$\lfloor \ln(\varepsilon_{\text{trunc}}) / \ln \beta \rfloor$ (as in script)
B_{train}	Batch size	64
N_{epoch}	Training epochs	1000
η_0	Adam initial learning rate	2×10^{-3}
LR schedule	`optax.exponential_decay` (multiplicative per epoch)	Factor 0.5
Optimizer		Adam
Warm-up	Initial distribution g_0 for early epochs	Epochs 1–50: `steady_dist`; thereafter previous `final_g`
U-Net width	Output channels per `DoubleConv` block	4
Convolution	`Conv1D` / `Conv1DTranspose`	Kernel 3; `SAME` padding
Activation	Hidden layers	`LeakyReLU`
Downsampling	Pooling	`AvgPool`, window 2, stride 2
Output head	Consumption parameterization	`sigmoid` then cumulative sum for $c(a, e, z)$
Repeats	Outer loop in `ablation_study.py`	10

D.2 USPG (PE Economy)

Symbol	Description	Value
N_a (`NA`)	Number of asset grid points	200
$a_{\min}$, $a_{\max}$	Asset bounds (`PEEnv`)	0, 100
n_e, n_r, n_w	States for productivity, interest rate, wage factor	3, 5, 7
β	Discount factor	0.975
σ	Coefficient of relative risk aversion	1
$c_{\min}$	Minimum consumption (feasibility)	10^{-3}
T	Training and evaluation horizon	Default `PEEnv.T` ($\lceil \ln(10^{-2}) / \ln \beta \rceil$, ≈ 182); override with `-horizon`
B_{train}	Batch size	64
N_{epoch}	Training epochs	1000
N_{rep}	Independent runs	10
η_0	Adam initial learning rate	2×10^{-3}
LR schedule	`optax.exponential_decay`	Factor 0.5 per epoch
Optimizer		Adam
N_{warm}	Epochs with ergodic g_0 warm-start	50 (`WARMUP_EPOCHS`)
N_{eval}	Interval for fixed-g_0 curve evaluation	100 (`eval_every`; first and last epoch always evaluated)
U-Net width	Channels per `DoubleConv`	4
Conv. / pooling		Same pattern as Huggett USPG (kernel 3, `LeakyReLU`, `AvgPool`)
Policy output	Relative to VFI upper bound	Consumption over n_e; (r, w) enter through wealth, no separate head

D.3 PPO (PE Economy)

Symbol	Description	Value
N_{update}	Total PPO updates	500
N_{env}	Parallel environments	32
H	Rollout length per env per update	64
d_h	Shared trunk hidden width (MLP)	128 (two `tanh` layers)
η	Learning rate	3×10^{-4}
γ	Discount factor	Default β (`PEEnv.beta`, 0.975)
λ_{GAE}	GAE parameter	0.95
$\varepsilon_{\text{clip}}$	PPO clip range	0.2
E_{ppo}	PPO epochs per update	10
$\lvert\mathcal{M}\rvert$	Minibatch size	256
c_v	Value loss coefficient	0.5
c_H	Entropy bonus coefficient	0.01
–	Obs. normalization scale (`normalize_obs`)	$[a_{\max}-a_{\min},\ 2.5,\ 0.035,\ 0.15]$
–	Obs. normalization offset (`normalize_obs`)	$[a_{\min},\ 0,\ r_{\min}-0.01,\ w_{\min}-0.02]$
Eval paths	Post-training Monte Carlo evaluation	4096 (`-eval_paths`)
Ergodic eval paths	Ergodic eval when logging	256 (`-log_ergodic_eval_paths`)

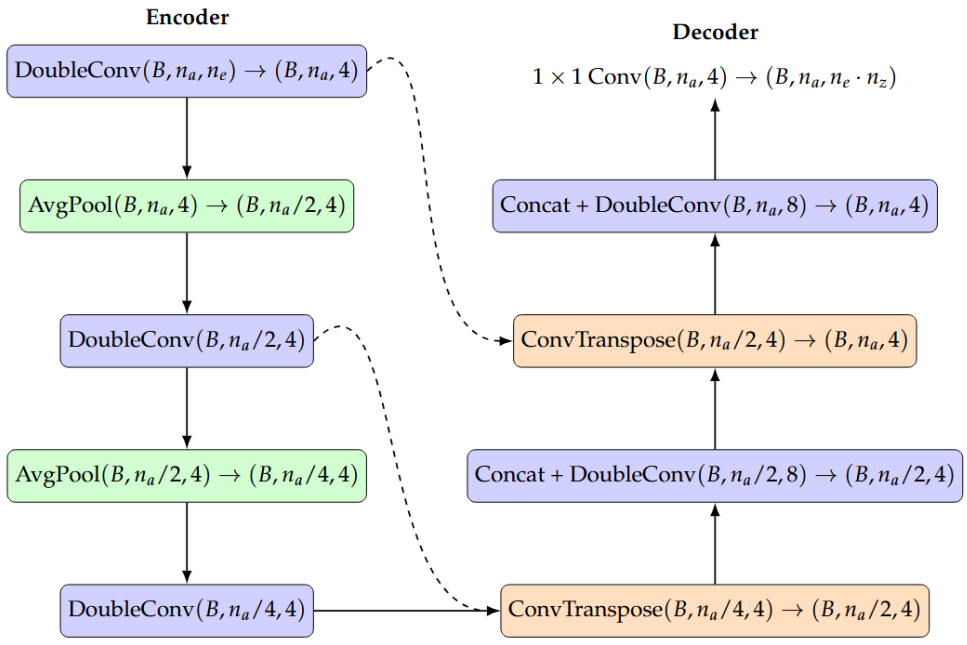
D.4 SAC (PE Economy)

Symbol	Description	Value
N_{update}	Total gradient-update rounds	500
N_{env}	Parallel environments	32
H	Rollout-related horizon	64
d_h	Actor / critic MLP hidden width	128 (actor: two `ReLU` layers)
η	Learning rate (Q and actor)	3×10^{-4}
γ	Discount factor	Default $\beta = 0.975$
τ	Target soft-update coefficient	0.005
α_{SAC}	Entropy temperature (fixed)	0.2
$\lvert\mathcal{B}\rvert$	Replay buffer capacity	500 000
B_{train}	Batch size per gradient step	256
N_{grad}	Gradient steps per environment step (after warm-up)	80
N_{start}	Minimum transitions before learning	5000
$\log\sigma$ clip	Stability	$[-20,\ 2]$ (`LOG_STD_MIN/MAX`)
Eval paths	Post-training MC evaluation	4096 (`-eval_paths`)
Ergodic eval paths	Ergodic eval when logging	256 (`-log_ergodic_eval_paths`)

D.5 DDPG (PE Economy)

Symbol	Description	Value
N_{update}	Total update rounds	500
N_{env}	Parallel environments	32
H	Horizon	64
d_h	Actor / Q hidden width	128 (two `ReLU` layers)
η	Learning rate	3×10^{-4}
γ	Discount factor	Default $\beta = 0.975$
τ	Target soft-update coefficient	0.005
$\sigma_{\text{noise}}^{(0)}$	Initial exploration noise std (on share)	0.1
$\sigma_{\text{noise}}^{(\text{end})}$	Final exploration noise std	0.02
$\lvert\mathcal{B}\rvert$	Replay buffer capacity	500 000
B_{train}	Batch size	256
N_{grad}	Gradient steps per update	80
N_{start}	Minimum samples before learning	5000
Eval paths	Post-training MC evaluation	4096 (`-eval_paths`)
Ergodic eval paths	Ergodic eval when logging	256 (`log_ergodic_eval_paths`)

E Overall Structure of the 1D U-Net



We employ a 1D U-Net architecture operating on the asset grid. The input tensor has shape (B, n_a, n_e), where n_a denotes the asset grid and n_e the productivity states. Convolutions are applied along the asset dimension while treating the productivity states as channels. The network consists of two down-sampling layers followed by two symmetric up-sampling layers with skip connections.

F Theoretical Analysis of DSPG

This appendix analyzes the gradient computation and variance properties of DSPG using the notation of Sections 3–4: Θ denotes the U-Net parameters, $\mathbf{g}_t \in \mathbb{R}_+^{n_a \times n_e}$ the discretized wealth–productivity distribution with entries $g_{k\ell,t} = d(a_k, e_\ell)$, $\mathbf{c}_t \in \mathbb{R}^{n_a \times n_e}$ the consumption matrix, z_t the aggregate shock, and r_t the equilibrium interest rate. We write $u(\mathbf{c}_t) \odot \mathbf{g}_t := \sum_{k=1}^{n_a} \sum_{\ell=1}^{n_e} g_{k\ell,t}\, u(c_{k\ell,t})$ for the aggregate welfare at date t, following the convention of Section 4.

F.1 The DSPG Computation Graph

At each period t, DSPG executes four operations:

(i) **Policy.** $\mathbf{c}_t = \text{U-Net}(\mathbf{g}_t, z_t; \Theta)$ via (10).

(ii) **Market clearing.** r_t is obtained in closed form from the aggregated budget constraint via (9): $1 + r_t = (B - z_t \bar{e} + \mathbf{c}_t \odot \mathbf{g}_t)/(\mathbf{a} \odot \mathbf{g}_t)$, which involves only weighted sums and one scalar division.

(iii) **Wealth update.** At each grid point (k, ℓ), next-period wealth is $a'_{k\ell,t} = (1 + r_t)a_k + z_t e_\ell - c_{k\ell,t}$, projected onto $[\underline{a}, \infty)$ to enforce the borrowing constraint (3).

(iv) **Distributional update.** $\mathbf{g}_{t+1} = \Phi(\mathbf{g}_t, \mathbf{c}_t, r_t, z_t, z_{t+1})$, where Φ redistributes mass across wealth grid points (via linear-interpolation weights that are piecewise linear in $a'_{k\ell,t}$) and composes with the exogenous transition matrix for e.

Composing steps (i)–(iv) over T periods yields the graph

$$\Theta \xrightarrow{\text{U-Net}} \mathbf{c}_0 \xrightarrow{(9)} r_0 \longrightarrow \mathbf{g}_1 \xrightarrow{\text{U-Net}} \mathbf{c}_1 \longrightarrow \cdots \longrightarrow \mathbf{g}_T, \tag{32}$$

every arrow of which is a (piecewise) differentiable map. Aggregate shocks $\{z_t\}$ and idiosyncratic realizations $\{e_{i,t}\}$ are exogenous and do not depend on Θ, so they act as fixed random inputs for a given sample path.

F.2 Assumptions

[Differentiability] The utility $u : \mathbb{R}_{++} \to \mathbb{R}$ is twice continuously differentiable, strictly increasing, and strictly concave. The U-Net map $\Theta \mapsto \text{U-Net}(\mathbf{g}, z; \Theta)$ is continuously differentiable. The redistribution map Φ is piecewise continuously differentiable in $(\mathbf{g}_t, \mathbf{c}_t, r_t)$. Non-smoothness is confined to (a) the borrowing-constraint projection in step (iii) and (b) the boundaries between neighboring grid cells in the interpolation of Φ.

[Bounded rewards] Consumption is bounded along any trajectory: $c_{k\ell,t} \in [\underline{c}, \bar{c}]$ with $0 < \underline{c} \le \bar{c} < \infty$, so that $|u(c_{k\ell,t})| \le M$ for some $M < \infty$.

[Positive aggregate wealth] The wealth grid contains at least one strictly positive point and $\mathbf{g}_t$ assigns positive mass to it, so that $\mathbf{a} \odot \mathbf{g}_t > 0$ for all t; with $B > 0$, the denominator in (9) is bounded away from zero.

F.3 DSPG Objective and Gradient

Recall from (12) that the DSPG sample objective is

$$\mathbb{J}(\Theta) \;\approx\; -\frac{1}{n} \sum_{m=1}^{n} \sum_{t=0}^{T} \beta^t \underbrace{u\big(\mathbf{c}_t^{(m)}\big) \odot \mathbf{g}_t^{(m)}}_{=: U_t^{(m)}}, \tag{33}$$

where m indexes independent shock trajectories. For a fixed m, $U_t^{(m)}$ depends on Θ through the graph (32).

[Exact pathwise gradient] Under Assumptions F.2–F.2, for each trajectory m and almost every realization of exogenous shocks, the map $\Theta \mapsto \sum_{t=0}^{T} \beta^t U_t^{(m)}$ is piecewise differentiable, and

$$\nabla_\Theta \, \mathbb{J}(\Theta) \;=\; -\frac{1}{n} \sum_{m=1}^{n} \sum_{t=0}^{T} \beta^t \, \frac{d\, U_t^{(m)}}{d\, \Theta}, \tag{34}$$

where $dU_t^{(m)}/d\Theta$ is computed by reverse-mode automatic differentiation through the graph (32). No score-function (REINFORCE) estimator is used.

Fix a trajectory m and its exogenous shock sequence. Given these, the sequence $(\mathbf{g}_0, \mathbf{c}_0, r_0, \mathbf{g}_1, \mathbf{c}_1, r_1, \ldots, \mathbf{g}_T)$ is a deterministic, piecewise differentiable function of Θ:

- $\mathbf{g}_0$ is an exogenous initial distribution (independent of Θ).
- $\mathbf{c}_t = \text{U-Net}(\mathbf{g}_t, z_t; \Theta)$ is C^1 in Θ by Assumption F.2.
- r_t is a ratio of affine functions of $(\mathbf{g}_t, \mathbf{c}_t)$ via (9); by Assumption F.2 the denominator is bounded away from zero, so r_t is C^1 in $(\mathbf{g}_t, \mathbf{c}_t)$.
- $\mathbf{g}_{t+1} = \Phi(\mathbf{g}_t, \mathbf{c}_t, r_t, z_t, z_{t+1})$ is piecewise C^1 by Assumption F.2.

By the chain rule, $U_t^{(m)}$ is piecewise differentiable in Θ for every t, and so is the finite sum $\sum_{t=0}^{T} \beta^t U_t^{(m)}$. The gradient (34) follows from applying the chain rule through the entire graph, which is exactly what reverse-mode automatic differentiation computes.

Because $\{z_t\}, \{e_{i,t}\}$ enter as fixed exogenous inputs (not functions of Θ), no likelihood-ratio term appears. Expectation and differentiation commute by dominated convergence: the integrand is bounded by $M \sum_{t=0}^{T} \beta^t < M/(1-\beta)$ under Assumption F.2.

Gradient decomposition. The total derivative of U_t with respect to Θ admits the natural decomposition

$$\frac{d\,U_t}{d\,\Theta} \;=\; \underbrace{\frac{\partial U_t}{\partial \mathbf{c}_t}\,\frac{\partial \mathbf{c}_t}{\partial \Theta}}_{\text{direct effect}} \;+\; \underbrace{\frac{\partial U_t}{\partial \mathbf{g}_t}\,\frac{d\,\mathbf{g}_t}{d\,\Theta}}_{\text{distributional effect}}\;, \tag{35}$$

where $d\mathbf{g}_t/d\Theta$ depends recursively on all prior periods through Φ, r_s, and $\mathbf{c}_s$ for $s < t$. The first term captures the immediate impact of Θ on current consumption; the second captures the cumulative effect of Θ on the wealth distribution, which feeds back through market clearing (9) into r_t. Both are computed automatically by backpropagation.

F.4 Truncation Bias

Let $\mathbb{J}^{\infty}(\Theta) = -\mathbb{E}[\sum_{t=0}^{\infty} \beta^t U_t]$ be the infinite-horizon objective. DSPG uses a finite truncation $T < \infty$. Under Assumption F.2,

$$\left|\mathbb{J}^{\infty}(\Theta) - \mathbb{J}(\Theta)\right| \;\le\; \frac{M\,\beta^{T+1}}{1-\beta}, \tag{36}$$

i.e., the bias decays geometrically in T. The same bound applies to gradients, with M replaced by the product of M and the Lipschitz constant of the graph (32). For $\beta \approx 0.96$, truncation at T on the order of hundreds makes the bias numerically negligible.

F.5 Variance Reduction over REINFORCE

[Zero conditional variance] For a fixed exogenous shock sequence, the DSPG gradient (34) is a deterministic function of Θ; it therefore has zero variance conditional on the shocks. In contrast, a REINFORCE estimator applied to the same trajectory carries strictly positive variance from the score-function weighting $\nabla_\Theta \log \pi_\Theta \cdot R$.

The residual variance of (34) arises only from averaging over n independent shock trajectories and decreases at the standard $O(1/n)$ rate. DSPG thus eliminates one of the two variance sources carried by REINFORCE-style estimators; empirically, this permits training with far fewer trajectories than classical DRL baselines.

F.6 Convexity of the Inner Problem

The per-period objective $U_t = \sum_{k,\ell} g_{k\ell,t}\, u(c_{k\ell,t})$ is concave in $\mathbf{c}_t$ (non-negative weighted sum of concave functions), and the budget constraint (2) is linear in $c_{k\ell,t}$. The inner problem of choosing $\mathbf{c}_t$ given a fixed distribution and price is therefore convex in the minimization sense of $\mathbb{J}$. This structure

does *not* imply that $\mathbb{J}(\Theta)$ is globally convex in the U-Net parameters Θ, but it yields a benign loss landscape: empirically (Section 5), gradient descent converges to the same solution from diverse random initializations, consistent with the observation in Section 4.

F.7 Remarks on Non-smoothness

Proposition F.3 asserts *piecewise* differentiability rather than global smoothness. Two sources of non-smoothness arise in the DSPG graph:

Borrowing constraint. Step (iii) projects next-period wealth onto $[\underline{a}, \infty)$ via $a'_{k\ell,t} \mapsto \max(a'_{k\ell,t}, \underline{a})$. The $\max$ operator is non-differentiable at $a'_{k\ell,t} = \underline{a}$. However, for a given Θ the set of parameters at which some grid point exactly hits the kink is a measure-zero subset of the parameter space. Away from this set, the projection is locally linear ($a'_{k\ell,t}$ or $\underline{a}$, whichever is active) and poses no obstacle to automatic differentiation. In practice, JAX computes a valid subgradient at the kink, which is sufficient for SGD convergence [**?**].

Grid interpolation. Step (iv) redistributes mass using linear-interpolation weights that are piecewise linear in the continuous wealth value $a'_{k\ell,t}$. The interpolation weights change their functional form when $a'_{k\ell,t}$ crosses a grid boundary. As with the borrowing constraint, the set of Θ values where $a'_{k\ell,t}$ lies exactly on a grid boundary has measure zero, and the interpolation is smooth almost everywhere. Furthermore, because linear interpolation is a convex combination, the resulting gradients are bounded, so the occasional subgradient at a grid boundary does not destabilize training.

In summary, the non-smooth points form a measure-zero set in Θ-space at each time step. The composition over T steps remains piecewise C^1 almost everywhere, and gradient descent with subgradients at non-smooth points is well studied and known to converge under mild conditions [**?**].

F.8 Lipschitz Regularity of the Gradient

The truncation-bias bound on gradients in Section F.4 (equation (36)) involves the Lipschitz constant of the computational graph. We argue here that this constant is finite under our assumptions.

[Finite Lipschitz constant] Under Assumptions F.2–F.2, let L_{net} denote the Lipschitz constant of the U-Net map $(\mathbf{g}, z) \mapsto \text{U-Net}(\mathbf{g}, z; \Theta)$ with respect to its inputs (for a fixed Θ), and let L_Φ denote the Lipschitz constant of the redistribution map Φ with respect to $(\mathbf{g}_t, \mathbf{c}_t, r_t)$. Then the one-step Jacobian of the graph (32) satisfies

$$\left\| \frac{d\,\mathbf{g}_{t+1}}{d\,\mathbf{g}_t} \right\| \;\le\; L_\Phi\,(1 + L_{\text{net}})(1 + L_r), \tag{37}$$

where L_r is the Lipschitz constant of r_t with respect to $(\mathbf{g}_t, \mathbf{c}_t)$, which is finite by Assumption F.2 (the denominator in (9) is bounded away from zero, and the numerator is Lipschitz in its arguments).

From the graph, $\mathbf{g}_{t+1}$ depends on $\mathbf{g}_t$ through three channels: (a) directly via Φ, (b) via $\mathbf{c}_t = \text{U-Net}(\mathbf{g}_t, z_t; \Theta)$, and (c) via r_t which depends on $(\mathbf{g}_t, \mathbf{c}_t)$. By the chain rule and the triangle inequality:

$$\left\| \frac{d\,\mathbf{g}_{t+1}}{d\,\mathbf{g}_t} \right\| \le \left\| \frac{\partial \Phi}{\partial \mathbf{g}_t} \right\| + \left\| \frac{\partial \Phi}{\partial \mathbf{c}_t} \right\| \left\| \frac{\partial \mathbf{c}_t}{\partial \mathbf{g}_t} \right\| + \left\| \frac{\partial \Phi}{\partial r_t} \right\| \left\| \frac{\partial r_t}{\partial \mathbf{g}_t} \right\| + \left\| \frac{\partial \Phi}{\partial r_t} \right\| \left\| \frac{\partial r_t}{\partial \mathbf{c}_t} \right\| \left\| \frac{\partial \mathbf{c}_t}{\partial \mathbf{g}_t} \right\|.$$

Each factor is finite: $\|\partial \Phi / \partial \cdot\| \le L_\Phi$ by assumption; $\|\partial \mathbf{c}_t / \partial \mathbf{g}_t\| \le L_{\text{net}}$; $\|\partial r_t / \partial \cdot\| \le L_r$ since r_t is a ratio with bounded-away-from-zero denominator. Collecting terms yields (37).

By induction, the T-step Jacobian is bounded by $(L_\Phi(1 + L_{\text{net}})(1 + L_r))^T$. When multiplied by the discount β^t, the contribution of period-t gradients to the total gradient is bounded by $M \cdot (\beta \cdot L_\Phi(1 + L_{\text{net}})(1 + L_r))^t$. In the regime $\beta \cdot L_\Phi(1 + L_{\text{net}})(1 + L_r) < 1$ (which holds when β is sufficiently small relative to the per-step expansion), the gradient series converges absolutely and the truncation bias on the gradient also decays geometrically. In practice, the effective per-step expansion is moderate because the market-clearing mechanism acts as a stabilizing feedback: if households over-consume, r_t rises to restore equilibrium, dampening the distributional response.

F.9 Pre-training and Initialization

The steady-state pre-training described in Section 4 produces an initial parameter vector Θ_0 by fitting the value-iteration policy of a Huggett [8] economy without aggregate risk. This step is a separate optimization and does not affect the theoretical properties of DSPG's main training phase: Θ_0 is treated as a fixed starting point, and all gradient computations in (34) are with respect to the current Θ only. The initial distribution $\mathbf{g}_0$ is likewise set to the steady-state distribution and is independent of Θ. Pre-training serves only to place Θ_0 in a region where the policy produces economically reasonable consumption values from the outset, ensuring numerical stability (in particular, avoiding negative or exploding consumption that would violate Assumption F.2).

Summary. DSPG assembles a fully differentiable simulation—U-Net policy, closed-form market clearing (9), wealth update, and distributional transition—and obtains exact pathwise gradients via backpropagation. Compared with REINFORCE, it eliminates score-function variance entirely; the only remaining stochasticity arises from sampling over exogenous shock trajectories and decays at the $O(1/n)$ rate. Non-smooth points (borrowing constraint, grid boundaries) form a measure-zero set and do not impede SGD convergence. The per-step Lipschitz constant of the computational graph is finite, ensuring that both truncation bias and gradient norms remain controlled. Horizon truncation contributes a geometrically decaying bias that is negligible in practice.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state that DSPG solves the Huggett [8] economy with aggregate risk and non-trivial market clearing, achieving market-clearing errors below 10^{-11}. These claims are supported by the theoretical analysis (Appendix F) and experimental results (Section 5). Limitations regarding grid scalability are also discussed in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 4 explicitly discusses that DSPG relies on a fixed grid whose cost scales with $n_a \times n_e$, and that backpropagation through T time steps may exhaust GPU memory for very high-dimensional distributions regardless of network architecture. Appendix F discusses non-smoothness at borrowing constraints and grid boundaries, and the truncation bias introduced by finite horizons.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Appendix F states three explicit assumptions (Assumptions F.2–F.2) and provides complete proofs for Proposition F.3 (exact pathwise gradient), Proposition F.5 (zero conditional variance), and Proposition F.8 (finite Lipschitz constant). The truncation bias bound is also derived with an explicit formula.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The model specification is fully described in Section 3. The algorithm is presented as pseudocode in Algorithm 1. All hyperparameters (learning rate, number of trajectories, truncation length, etc.) are reported in the appendices. The U-Net architecture is detailed in Appendix E and model calibration in Appendix A.1. Hardware specifications are provided in the introduction.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The complete codebase (written entirely in JAX) is publicly available at `https://github.com/leafDancer/DSPG`. The repository includes code for both GE and PE benchmark models, training scripts, and steady-state value-iteration solvers.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: The optimizer (gradient descent via JAX autodiff), learning rate, number of trajectories n, truncation length T, and all model calibration parameters are reported in the appendices (Appendix D.1 for DSPG on the GE model, Appendices D.2–D.5 for all methods on the PE benchmark). The U-Net architecture is specified in Appendix E. Hardware details (RTX 3090 GPUs) are stated.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All main experiments are repeated over 10 independent runs. Table 2 reports both mean and standard deviation for each method. Training curves in Figure 7 and Figure 4 show ± 1.96 standard deviation shading across runs. The variability source (different random seeds for aggregate shock trajectories) is stated.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The introduction specifies the hardware used (a server with NVIDIA RTX 3090 GPUs and 32 AMD CPU cores). Section 4 reports per-step computational complexity. The ablation study (Figure 4) reports wall-clock training time under different trajectory counts. The introduction states that the model is solved within minutes.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

Justification: This work develops a computational method for solving macroeconomic models. It does not involve human subjects, personal data, or any activity that could conflict with the NeurIPS Code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: DSPG is a computational method for solving academic macroeconomic models. It does not directly enable deployment decisions, surveillance, or other applications with foreseeable negative societal impact. The positive impact is enabling economists to study richer models of inequality and policy, which is discussed in the introduction.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The released code solves stylized economic models and does not pose risks of misuse comparable to generative models or scraped datasets.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: The paper does not use pre-existing code packages, datasets, or pre-trained models from external sources. All code is written from scratch by the authors. The economic models referenced are theoretical constructs from the academic literature and are properly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release two benchmark models (GE and PE versions of the dynamic Huggett [8] economy) along with the DSPG codebase at `https://github.com/leafDancer/DSPG`. The code is written entirely in JAX and includes documentation for reproducing all experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

Justification: This work does not involve crowdsourcing or human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This work does not involve human subjects research.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not used as a component of the DSPG algorithm or any part of the core methodology. Any use of LLMs was limited to writing assistance and does not affect the scientific content.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.