DSPG: LEARNING DISTRIBUTION-CONDITIONED POLICIES FOR HETEROGENEOUS-AGENT ECONOMIES

Anonymous authors
Paper under double-blind review

ABSTRACT

With richer micro data and cheaper computation, macroeconomic policy evaluation has shifted from aggregates to the distribution behind them. Heterogeneous-agent models are the standard tool for that question, yet solving them in general equilibrium means confronting an infinite-dimensional fixed point: household policies govern how the cross-sectional distribution of assets and income evolves, and that distribution feeds back into policies through equilibrium prices. Existing solvers face a trade-off between distributional information and computational cost: low-dimensional methods (Krusell–Smith, DeepHAM, StructuralRL) reduce the distribution to a few moments, while high-dimensional methods such as DEQN take the distribution itself as input but solve too slowly to be practical. We propose DSPG (Distribution-based Structural Policy Gradient), which introduces the complete cross-sectional distribution directly into structural reinforcement learning. In a Huggett economy with aggregate risk and an explicit market-clearing condition, DSPG attains a low Euler residual of 2.5×10^{-3} in 956 s, against 6.5×10^{-3} for the strongest convergent baseline solver. A U-Net is the best DSPG network, which needs only 783 parameters to beat a 480k-parameter MLP (614$\times$ smaller) at only 3.5% higher runtime cost, and whose parameter count does not grow with the asset grid, the dominant dimension of the distribution. DSPG is the state of the art on this economy once accuracy, runtime, and model size are considered together, and it is robust across all 75 economies of a sweep that varies aggregate risk, idiosyncratic risk, and risk aversion, without retuning.

1 INTRODUCTION

Aggregate growth, per-capita income, and consumption remain central to macroeconomic analysis, but modern tax, transfer, monetary, and social-security policy must evaluate heterogeneous households (Kaplan & Violante, 2018; Ahn et al., 2018): who gains, who loses, who is liquidity constrained (Kaplan et al., 2014), and who bears risk. The same change in the interest rate moves savers and borrowers in opposite directions, a redistributive effect that is invisible in aggregate data (Auclert, 2019). Macroeconomic policy analysis has therefore shifted from a representative-agent view of aggregates toward tracking the economy's entire distribution (Krueger et al., 2016; Heathcote et al., 2009); and heterogeneous-agent models (HAMs), the standard framework for representing that distribution (Achdou et al., 2022), face a general-equilibrium problem whose feasibility and efficiency are themselves a bottleneck for policy analysis (Moll, 2026). The difficulty lies in the model specification and its equilibrium concept: household decisions, the cross-sectional distribution, and equilibrium prices are mutually determined, forming a fixed point whose difficulty has grown steadily as the models have grown richer.

Heterogeneous-agent models began with Bewley, Huggett, and Aiyagari (Bewley, 1986; Huggett, 1993; Aiyagari, 1994), gained aggregate risk with Krusell and Smith (Krusell & Smith, 1998), then acquired New Keynesian elements such as monetary policy and sticky prices with HANK (Kaplan et al., 2018), each step closer to real households. Two features are hardest to handle. (1) *Aggregate shocks* move the economy away from a steady state, so that optimal decisions and prices are no longer constants but depend on the current cross-sectional distribution of assets and income, a measure rather than a finite list of aggregate variables, which those decisions in turn reshape. That is the outer fixed point. (2) *The market-clearing condition* adds an inner one: within each period a fixed point must be solved to obtain the equilibrium price. Reliable solution therefore matters: many

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: Solution methods for Huggett-type HAMs with aggregate risk and market clearing. *Solution* separates global methods from local perturbations; *Params* is the policy parametrization: a grid, an MLP, or a U-Net; *Speed* rates each method by its fastest configuration; *State-space* is what a method must carry, where low-dim means aggregates or a few moments of the distribution and high-dim means the cross-sectional distribution itself; *Conv.* records convergence on the economy studied in this paper; *Acc.* is solution accuracy; *Cmp.* marks the methods benchmarked here.

Method	Solution	Params	Speed	State-space	Conv.	Acc.	Cmp.
SSJ (Auclert et al., 2021)	perturbation	grid	xfast	low-dim	✓	moderate	✗
KS (Krusell & Smith, 1998)	—	grid	slow	low-dim	✗	xhigh	✗
StructuralRL (Yang et al., 2025)	global	grid	fast	low-dim	✓	moderate	✓
DEQN (Azinovic et al., 2022)	global	MLP	slow	high-dim	✓	moderate	✓
DSPG (this work)	global	U-Net	moderate	high-dim	✓	high	✓

policies operate through liquidity, risk exposure, and differences in the tails of the distribution, and these are precisely what a solver must represent (Azinovic et al., 2022; Han et al., 2026).

Progress in AI algorithms has already succeeded on high-dimensional, dynamic, complex problems across many fields. These breakthroughs have spurred AI for computational economics (Maliar et al., 2021; Han et al., 2026; Azinovic et al., 2022; Yang et al., 2025), yet none balances accuracy against efficiency: low-dimensional methods must compress the distribution, so information loss is unavoidable, while high-dimensional methods take the distribution itself as input but converge too slowly to be practical. Central banks and consultancies fall back on perturbation solutions converging in seconds (Auclert et al., 2021). Such methods assume linear impulse responses, sidestepping the policy's nonlinear part, and are regarded as an over-simplification (Bianchi & Kaplan, 2026).

To address this accuracy-efficiency trade-off, we propose DSPG, a distribution-based policy gradient method. DSPG follows the structural reinforcement learning recipe: it builds a differentiable environment from the model's structure and represents the distribution as a vector, as in non-stochastic simulation (Young, 2010). Within a mean-field game formulation (Lasry & Lions, 2007), it solves a Huggett economy with *both aggregate shocks and market clearing* in minutes. Because the policy is conditioned on the complete distribution, the inner market-clearing fixed point is resolved by exact clearing inside the environment rather than internalized by the policy; only the outer fixed point remains, and an analytic policy gradient optimizes it directly. Table 1 places DSPG against the alternatives.[1] Among the convergent solvers DSPG attains the lowest Euler residual and uses the fewest parameters, and is the fastest of the high-dimensional methods.

Ablations show that a U-Net is the best architecture for the framework, which we trace to its natural fit for distribution-to-policy mappings: it attains the highest accuracy with the fewest parameters, outperforming an MLP on both counts, and DSPG is otherwise robust to its hyperparameters. It also solves accurately across a wide range of economy parameterizations, and the sweep yields two findings that match the literature closely. The standard deviation of aggregate risk drives the volatility of the equilibrium interest rate while barely moving its level; the standard deviation of idiosyncratic risk moves the level, welfare, and inequality through precautionary saving (Huggett, 1993; Aiyagari, 1994). Together these results establish DSPG as robust and broadly applicable.

2 Related work

Macroeconomic perspective. Bewley, Huggett, and Aiyagari established the framework: households solve a dynamic program against uninsurable income risk, and the cross-sectional distribution aggregates their decisions (Bellman, 1957; Bewley, 1986; Huggett, 1993; Aiyagari, 1994). Krusell & Smith (1998) added aggregate shocks and one-moment forecasting, yielding a self-confirming solution. We solve their union: both Huggett's market-clearing condition and the Krusell–Smith shock are present, so both fixed points are in play and the problem is harder than either. Young's

[1] We do not compare SSJ: it is a local linearization, so an accuracy comparison against global solvers would measure the wrong object. We do not compare KS either: on this economy the Krusell–Smith algorithm does not converge to a fixed point, so the numbers it produces are not a solution in any rigorous sense.

non-stochastic simulation writes the distribution's evolution as a mean-field transport on a grid; that makes the update linear and underpins our solver's efficiency (Young, 2010), letting DSPG differentiate through it directly.

Machine learning perspective. Maliar, Maliar, and Winant first applied deep networks to dynamic models, with three loss designs (lifetime utility, Bellman, Euler); we take the first (Maliar et al., 2021). DeepHAM characterizes the distribution through generalised moments (Han et al., 2026) and DEQN feeds it the full histogram (Azinovic et al., 2022), the two ends of deep learning for HAMs. StructuralRL inherits DeepHAM's update but puts the price signal in the state with a tabular policy, making it extremely fast (Yang et al., 2025)[2]. DSPG combines the three: it keeps StructuralRL's efficiency and, like DEQN, carries the full distribution rather than compressing it to moments, but handles it more cheaply. DSPG also draws on the reinforcement learning literature, combining model-free Monte Carlo sampling (Mnih et al., 2013; Schulman et al., 2017; Haarnoja et al., 2018; Lillicrap et al., 2016) with the information efficiency of model-based methods (Deisenroth & Rasmussen, 2011; Janner et al., 2019).

3 Economic model

A unit mass of households has state $s = (a, e)$, where a is assets and e is idiosyncratic income. A common state z scales income. Households have preferences $\mathbb{E}\sum_{t\geq 0}\beta^t u(c_t)$ and budget constraint

$$a_{t+1} = (1 + r_t)a_t + z_t e_t - c_t, \qquad c_t > 0, \quad a_{t+1} \geq \underline{a}. \tag{1}$$

The utility is logarithmic at the original calibration and CRRA with curvature σ in the extensions. We construct the Markov matrices P_e and P_z by applying the Rouwenhorst discretization (Rouwenhorst, 1995) to log AR(1) processes, with shock standard deviations written σ_e and σ_z. The baseline uses three idiosyncratic-income states with persistence $\rho_e = 0.6$ and five aggregate states with persistence $\rho_z = 0.8$; both grids are normalized to have mean one. The asset supply is fixed at $B = 5$ and $\underline{a} = 0$. The complete grid construction, shock discretization, and boundary conventions are given in Appendix B.

Let g be the probability distribution on the asset–income grid. For a consumption rule $c_\theta(s; g, z)$, aggregate assets, income, and consumption are

$$A(g) = \sum_s a g(s), \qquad Y(g, z) = \sum_s ze\, g(s), \qquad C_\theta(g, z) = \sum_s c_\theta(s; g, z) g(s).$$

Integrating the household budget and imposing next-period asset clearing gives

$$1 + r_\theta(g, z) = \frac{B + C_\theta(g, z) - Y(g, z)}{A(g)}, \qquad A(g) > 0. \tag{2}$$

This is the price update associated with a consumption plan. Feasibility and numerical boundary handling are described in Appendix B.

The household policy and the income transition induce a row-stochastic matrix $K_\theta(g, z, r)$. Linear asset interpolation implements the histogram transition (Young, 2010):

$$g_{t+1} = K_\theta(g_t, z_t, r_t)^\top g_t, \qquad z_{t+1} \sim P_z(z_t, \cdot). \tag{3}$$

A competitive equilibrium combines household optimality given the price and aggregate-state process with this population law and asset-market clearing. In household improvement, the population environment is taken as given; after a policy update, simulating the population refreshes that environment. Appendix A provides the conditional gradient formulation.

4 Distribution-based structural policy gradients

Algorithm and policy representation. Figure 1 places DSPG among existing solution methods and outlines the procedure below. DSPG denotes the structural policy-learning procedure together

[2]RSPG refines the neural version by pairing price histories with a recurrent network (Wibault et al., 2026).

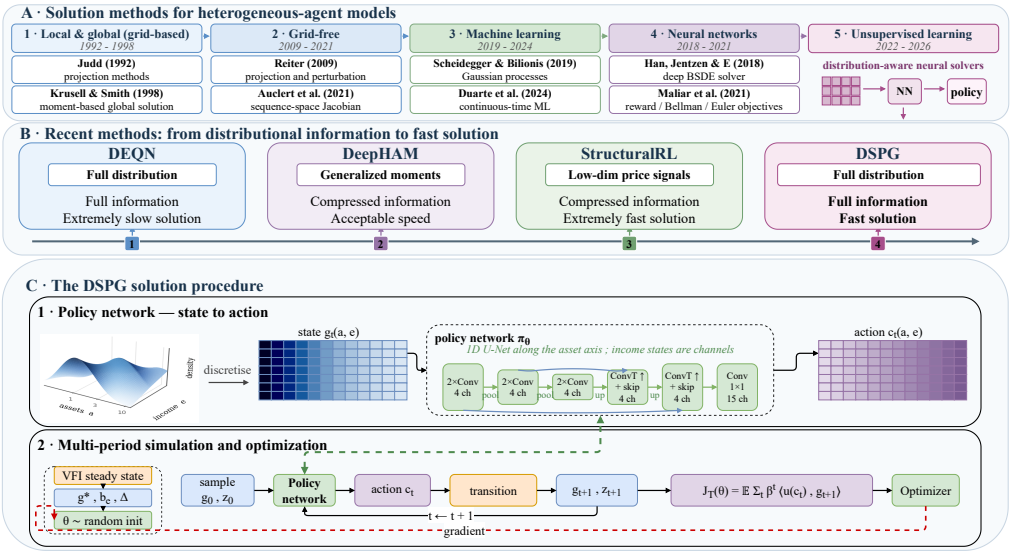


Figure 1: Overview of DSPG. (A) Solution methods for heterogeneous-agent models, from grid-based local and global methods through grid-free and machine-learning approaches to distribution-aware neural solvers. (B) Recent methods, placed by how much distributional information they carry against how fast they solve. (C) The procedure in the compact implementation: a one-dimensional U-Net along the asset axis maps the discretized distribution to a consumption schedule, and the simulation loop clears the market each period and updates the policy through the household transitions.

with distribution conditioning, not a specific neural network. The algorithm simulates the population state, computes prices and transitions from the economic model, and improves a policy that conditions on the resulting histogram. A policy representation is then chosen within this framework. Two representations instantiate it: a dense MLP and the compact ordered-grid U-Net developed below. The equilibrium-method comparison runs both, separating the algorithmic gain of the framework from the representation gain of the U-Net; the economic applications hold MLP-32 fixed, and the architecture study varies the representation.

Structural computation. For each simulated common-shock path, the method computes consumption, the clearing price, feasible asset choices, and the next histogram. The household step treats prices and the population environment as given while retaining the effect of saving on the household's own future states. Known individual transition probabilities are integrated on the grid; randomness enters through sampled common-shock paths. Conditional automatic differentiation then supplies the household policy direction, with the derivation in Appendix A.

The transition makes the role of economic structure explicit. For a feasible asset choice $a'_{i,e}$ between adjacent grid points a_ℓ and $a_{\ell+1}$, define

$$\omega_\ell(a') = \frac{a_{\ell+1} - a'}{a_{\ell+1} - a_\ell}, \qquad \omega_{\ell+1}(a') = 1 - \omega_\ell(a'). \tag{4}$$

Other asset weights are zero. The household transition and population update are then

$$\begin{aligned} K_\theta[(i,e),(j,e')] &= \omega_j(a'_{i,e}) P_e(e,e'), \\ g_{t+1}(j,e') &= \sum_{i,e} g_t(i,e)\,\omega_j(a'_{i,e}) P_e(e,e'). \end{aligned} \tag{5}$$

Each household asset choice therefore redistributes mass to two adjacent asset points per next income state, integrating idiosyncratic transitions without drawing an individual shock per household. Nonnegative weights preserve population mass; mean-preserving interpolation also links the next asset stock to the clearing equation (Appendix B). Differentiating through the household budget and

these transition weights captures how current consumption changes future household resources. Averaging conditional household gradients over common-shock paths supplies the policy-improvement direction; the finite-horizon identity and its assumptions are given in Appendix A.

This structure separates what is learned from what the model specifies: the policy network approximates household decisions, while budget accounting, income transitions, and market clearing come from the economic model. Of the two nested fixed points of the general equilibrium, only the outer one is learned. The inner market-clearing fixed point is resolved inside the environment: the price follows from the clearing identity of Equation 2 and is detached from the policy gradient, so the interest rate enters training as an output of clearing rather than as a target. The representation can therefore be evaluated using household accuracy, aggregate consistency, and the resulting distributions, rather than only a training score.[3]

Compact U-Net representation. Within that framework the representation is a free choice; the compact one we develop here keeps the distribution on the asset grid. In the compact implementation the asset axis is a spatial index and income states are channels. The U-Net encoder contains three double-convolution blocks and two average-pooling stages, with four channels per block. The decoder mirrors this: two transposed convolutions, each followed by a double-convolution block after the skip connection. A final pointwise convolution outputs a consumption schedule for each income and aggregate state. With 200 asset points, three income states, and five aggregate states, the network maps a 200×3 histogram to $200 \times 3 \times 5$ consumption values.

We normalize the input histogram and scale it by its number of entries. The policy head parameterizes nonnegative consumption increments:

$$\widehat{c}_{i,e,z} = b_e + \sum_{j \le i} \mathrm{sigmoid}(\ell_{j,e,z})\, \Delta_{j,e}, \qquad \Delta_{j,e} \ge 0. \tag{6}$$

The steady-state solution supplies the initial distribution, the per-income intercepts b_e, and the increments $\Delta_{j,e}$. This construction imposes increasing consumption along the asset grid and limits the range of the network outputs. The original policy and distribution are in Appendix D.

Parameter sharing yields a small network without first compressing the distribution to a prescribed moment vector, and simulation alone scales with the grid. The U-Net has 783 parameters, compared with 119,288 for MLP-32 and 480,440 for MLP-128. Fewer learned coefficients also mean fewer gradient entries and Adam states, sharply reducing network-state storage as the histogram grows. The nonuniform asset grid means that convolutions share weights in index space; exact translation invariance in asset units is not assumed.

Why an ordered distribution map? The input and output are both functions on the same asset grid, and the U-Net reuses filters along it rather than adding coefficients as the representation expands. Fine-resolution features capture changes near the borrowing constraint; pooling combines information across broader asset ranges; and skip connections restore detail when constructing the consumption schedule. This local-to-global structure is well matched to policies whose curvature varies with wealth while aggregate prices depend on the entire cross section.

For comparison, a prescribed moment representation has the form

$$m_k(g) = \sum_s \phi_k(s) g(s), \qquad c_\theta(s; g, z) = f_\theta(s, m_1(g), \ldots, m_M(g), z). \tag{7}$$

It imposes the same policy for distributions sharing those moments, which is accurate when approximate aggregation holds but a model-dependent restriction when tails or joint-distribution features affect prices and household responses. DSPG instead preserves the discretized cross section at the policy input, and the U-Net limits the learned coefficients, resolving the central representation tradeoff: a rich distribution without the parameter growth of a dense full-histogram map.

[3]Training uses JAX (Bradbury et al., 2018) and Adam (Kingma & Ba, 2015) for 1,000 epochs, with a 50-epoch initial-distribution warmup and a rollout horizon of 181. Reward and error conventions are in Appendix B; each comparison below retains its original protocol.

5 EXPERIMENTS

Setup and metrics. All experiments use the Section 3 economy under one protocol: 512 simulated economies, a 2,000-period burn-in, a 4,000-period panel, exact clearing. Metrics are the distribution-weighted Euler diagnostic $|\mathrm{EE}|$, the mean rate's deviation from its $z \equiv 1$ steady state, the wealth Gini, the clearing residual, runtime, and parameter count; definitions are in Appendix B and baseline protocols in Appendix E.

5.1 EQUILIBRIUM SOLUTION WITH AGGREGATE RISK

The baseline experiment reports the equilibrium solution under log utility, $\beta = 0.975$, and a five-state aggregate shock. Figure 2 displays the policy, stationary distribution, equilibrium price path, and clearing residual for one simulated trajectory. Consumption increases with assets, steeply out of the borrowing constraint then nearly linearly, and rises with income. Probability mass concentrates at low wealth in every income group, the middle group dominant by an order of magnitude. The common shock moves the rate in the opposite direction; over this trajectory it ranges from 2.13% to 2.78% around the $z \equiv 1$ value of 2.47%. The asset-market residual stays below 4×10^{-13} throughout, at float64 precision.

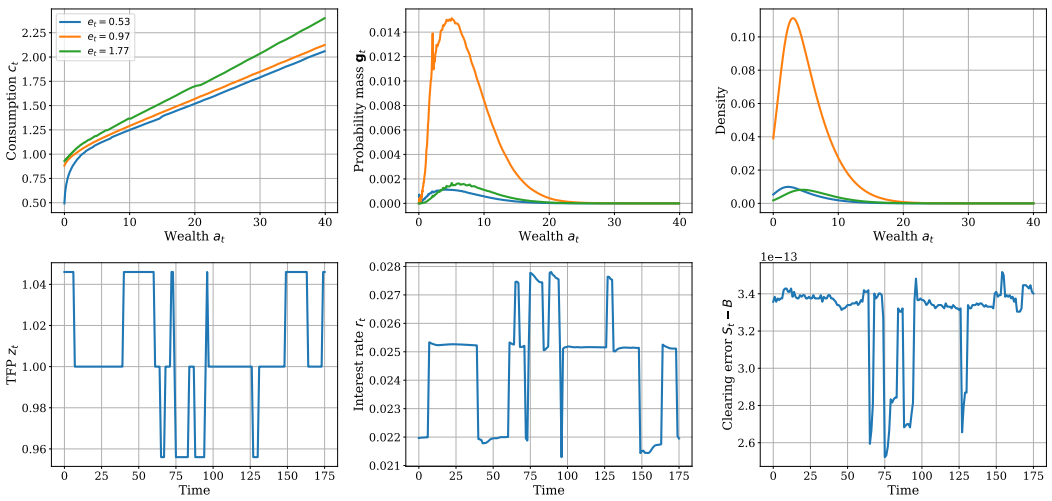


Figure 2: DSPG solution with aggregate risk. Top: consumption policy, probability mass, and density. Bottom: aggregate shock, interest rate, and asset-market residual.

Figure 3 compares stationary wealth distributions across income states with StructuralRL. The baseline comparisons find closely aligned distributions and consumption paths. Appendix D includes the original aggregate-consumption plot and the groupwise error decomposition.

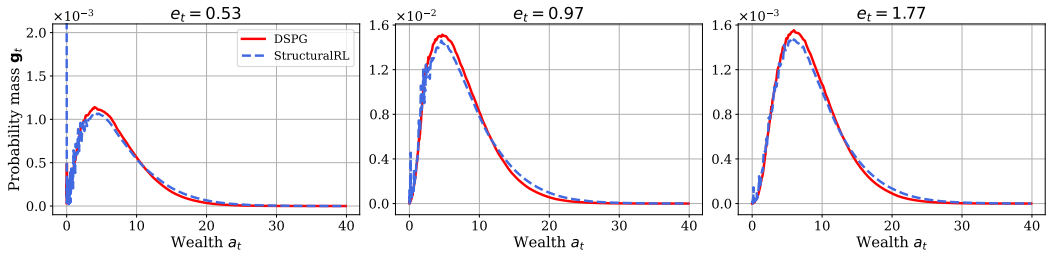


Figure 3: DSPG–StructuralRL stationary wealth distributions by income state, baseline calibration.

5.2 Comparison with baseline equilibrium algorithms

The extended benchmark uses CRRA curvature $\sigma = 2$, five seeds, and a common continuation-error definition, and it compares three baseline solvers: StructuralRL-NN, StructuralRL-GRID, and DEQN (Yang et al., 2025; Azinovic et al., 2022). To separate the two contributions, this table compares both DSPG architectures, MLP-32 and U-Net. DSPG delivers the lowest weighted Euler diagnostic, 2.52×10^{-3}, against 6.50×10^{-3} for StructuralRL-NN, 7.25×10^{-3} for StructuralRL-GRID, and 6.66×10^{-3} for DEQN, and it is faster than the recorded StructuralRL-NN and DEQN runs. StructuralRL-GRID has the shortest runtime but more than twice DSPG's Euler error.

Table 2: Five-seed equilibrium-method comparison in the extension calibration. LD/HD denote low- and high-dimensional solution representations. Bold marks the best. Runtime is the full procedure.

Method	Space	Dist-w. $\lvert\mathrm{EE}\rvert$	$\log_{10}\lvert\mathrm{EE}\rvert$	$\lvert\bar{r} - r_{\mathrm{ss}}\rvert$	Runtime (s)
DSPG-MLP-32	HD	3.08×10^{-3}	-2.51	0.026%	837
DSPG-U-Net	HD	$\mathbf{2.52 \times 10^{-3}}$	$\mathbf{-2.60}$	$\mathbf{0.024}\%$	956
StructuralRL-NN	LD	6.50×10^{-3}	-2.19	0.132%	1776
StructuralRL-GRID	LD	7.25×10^{-3}	-2.14	0.143%	**114**
DEQN	HD	6.66×10^{-3}	-2.18	0.303%	4501

The classical reference is reported separately. Its retained household policy has the lowest diagnostic of any method here, but the outer perceived-law-of-motion iteration enters a two-cycle on this economy, so it is not a converged equilibrium and is excluded from the comparison. Appendix E documents initialization, the DEQN port, and the reference-pipeline checks.

5.3 Ablation over algorithm parameters

This study turns to the second contribution. Holding the DSPG learning rule and economy fixed at $(\sigma_e, \sigma_z, \sigma) = (0.2, 0.05, 2)$, it compares five policy representations, four batch sizes, three learning rates, and ten seeds: 600 runs. Table 3 reports the reference-rate comparison for four of the five, where the U-Net has the lowest mean diagnostic at every batch: at $N = 128$ its mean is 2.5×10^{-3} against 3.0–3.1 $\times 10^{-3}$ for the dense alternatives, with near-identical utility.

Table 3: Architecture comparison at learning rate 2×10^{-3}, ten seeds per configuration. Each batch group reports error extrema and means, utility means and standard deviations, and runtime.

Batch N	Architecture	Params	Dist-w. $\lvert EE\rvert$ ($\times 10^{-3}$)			Final utility		Runtime (s)
			min	mean	max	mean	std	
$N = 32$	MLP-32	119,288	2.91	3.11	3.32	**-35.574**	0.042	**298**
	MLP-64	237,624	2.73	3.02	3.20	-35.576	0.042	311
	MLP-128	480,440	2.83	2.95	3.15	-35.576	0.044	342
	U-Net	**783**	**2.27**	**2.59**	**3.04**	-35.576	**0.038**	349
$N = 64$	MLP-32	119,288	2.84	3.03	3.24	**-35.579**	**0.030**	**478**
	MLP-64	237,624	2.81	2.97	3.08	-35.579	0.030	512
	MLP-128	480,440	2.91	2.97	3.05	-35.581	0.038	545
	U-Net	**783**	**2.26**	**2.49**	**2.69**	-35.582	0.031	555
$N = 128$	MLP-32	119,288	2.95	3.05	3.22	-35.583	0.044	**851**
	MLP-64	237,624	2.85	3.00	3.11	-35.582	0.044	884
	MLP-128	480,440	2.89	3.00	3.17	-35.584	0.038	937
	U-Net	**783**	**2.24**	**2.46**	**2.86**	**-35.578**	**0.031**	969
$N = 256$	MLP-32	119,288	2.95	3.03	3.15	-35.599	0.017	**1220**
	MLP-64	237,624	2.78	2.99	3.09	-35.596	0.017	1299
	MLP-128	480,440	2.88	2.98	3.08	-35.597	0.019	1438
	U-Net	**783**	**2.24**	**2.47**	**2.96**	**-35.583**	**0.014**	1495

The representation gain is substantial: the same full-distribution task uses 783 parameters instead of 480,440 for MLP-128 while improving the Euler accuracy, a 614-fold reduction that carries through to gradient and optimizer-state storage and leaves room for finer grids or larger batches at fixed memory. Wider dense networks do not materially improve the solution; the U-Net's wall time is higher at a fixed batch, but its scaling favors richer state spaces. Larger batches help because each update averages gradients over more simulated paths: raising the batch from 32 to 256 at the reference rate lowers the U-Net mean diagnostic from 2.6×10^{-3} to 2.5×10^{-3} and its final-utility standard deviation from 0.038 to 0.014. Its final utility at $N = 256$ is -35.583, higher than every dense alternative at that batch, and its diagnostic remains the lowest; 0.002 is the best balance of accuracy and seed dispersion, and Appendix F reports all learning rates and curves. As a stability check, the MLP-32 specification is held fixed across the 75-calibration risk study, whose median within-configuration seed standard deviations are 5.54×10^{-5} for the mean interest rate and 6.97×10^{-4} for the wealth Gini; accuracy varies with risk, as documented in Appendix G.

5.4 Macroeconomic studies: findings and validation

The risk sweep serves two purposes: it isolates how the two sources of risk shape the equilibrium, and it validates the solver against established macroeconomic evidence. The findings come first, with the accuracy evidence that supports them; the validation follows.

Findings. Both rows share a layout: each panel fixes the curvature σ, its rows are σ_e, and its columns are σ_z. The upper row of Figure 4 shows where the solver is accurate. Quality degrades with idiosyncratic risk, the dominant factor, and improves with curvature; aggregate risk matters mainly where idiosyncratic risk is low. All 75 cells converge, so these patterns are not read off a corner where the solver struggles. The lower row shows what the solution implies for wealth, and the two are aligned rather than in tension: the cells solved most accurately are the least unequal, across both σ_e and curvature. Aggregate risk is the exception: it moves inequality far less than accuracy.

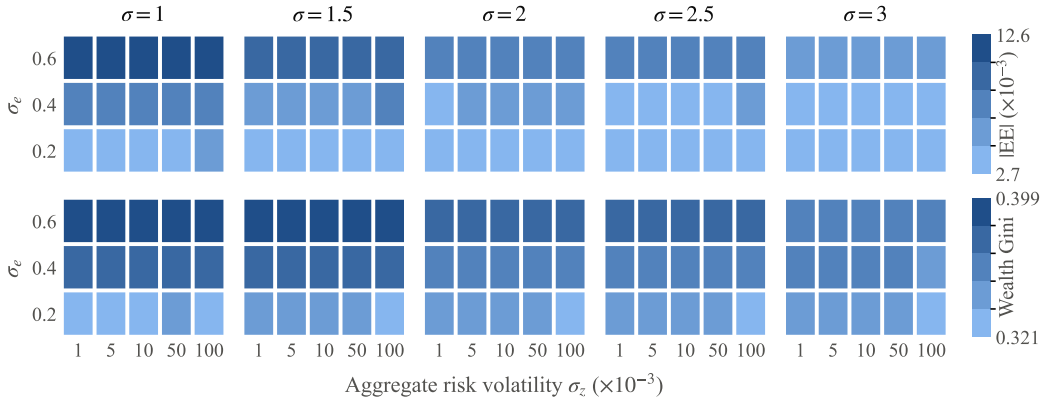


Figure 4: Risk-parameter sweep, five seeds per cell: solution accuracy (top), wealth inequality (bottom). σ is risk aversion (the curvature of utility), and σ_e and σ_z are the volatilities of idiosyncratic and aggregate risk. Each panel fixes σ and holds the $\sigma_e \times \sigma_z$ grid; colour is the cell mean, on the scale beside each row, of the distribution-weighted Euler diagnostic and of the asset Gini.

Because market clearing pins aggregate assets at $B = 5$, the equilibrium distribution answers what aggregate statistics cannot: what happens to household buffers when income risk rises? At $\sigma_z = 0.001$ under log utility, raising σ_e from 0.2 to 0.6 pulls the asset distribution apart, while mean assets stay at the fixed supply. Similar aggregate holdings coexist with thinner buffers at the bottom and more concentrated wealth at the top. Appendix G gives the full risk, price, and welfare results.

Validation. Figure 5 reports the interest-rate response to a common shock alongside the local-projection slope. The response is negative and its magnitude is economically meaningful: at the largest aggregate-risk setting the impact responses are approximately $+1.87$ and -2.50 percentage points in low and high common states, and the response decays with the persistence of the common shock. Both the shape and the scale agree with the impulse responses that incomplete-markets

models with aggregate risk report for the equilibrium return (Kaplan et al., 2018): when productivity rises, income rises more than consumption, desired saving increases at a fixed return, and the return must fall to reconcile demand with a fixed asset supply (Krusell & Smith, 1998). That mechanism makes the return a countercyclical price here: the simulated correlation between the return and the shock reaches -0.997 at the largest setting. The solver reproduces this unprompted. The price is an output of market clearing, not a target of the training objective, so the response is evidence about the economics, not about the fit.

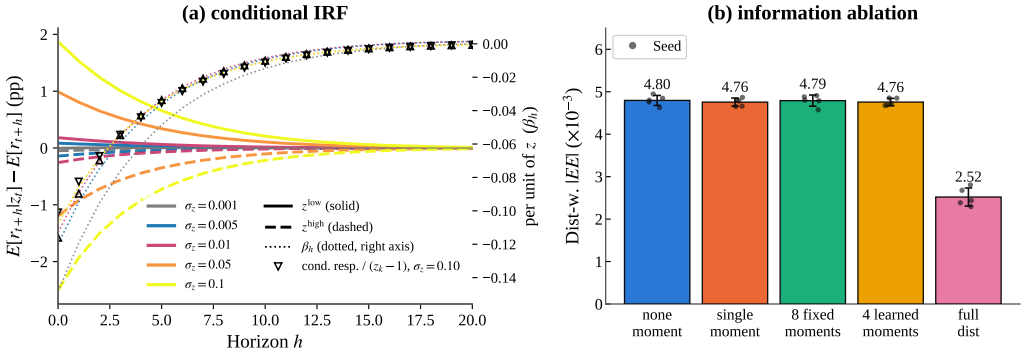


Figure 5: **Left:** conditional interest-rate responses and local-projection slopes, descriptive equilibrium dynamics, not separately identified policy interventions. **Right:** test Euler error against the policy's distributional input, from the policy's own state and the asset mean through 8 fixed and 4 learned moments to the full histogram, with ± 1 seed standard deviation over five seeds; the compressed inputs land within noise of one another, only the full histogram is lowest.

This ablation isolates what the distribution itself contributes, by holding the U-Net, the objective, the clearing step, the training budget and the seeds fixed and varying only what the network is fed as the cross-sectional state. The five configurations receive progressively more: nothing beyond the household's own state, then the asset mean, eight fixed moments of the cross section, four learned moments, and finally the full histogram. They differ in one dimension only, the total information the policy can draw on, from none to everything the cross section carries. Euler accuracy separates them: the four compressed inputs land within noise of one another, from 4.76 to 4.80×10^{-3}, while the full histogram alone reaches 2.52×10^{-3}, 47% below the best of them. The amount of distributional information the policy receives therefore sets the quality of the solution: with the algorithm held fixed, moments, fixed or learned, do not recover the accuracy of the full histogram.

6 Conclusion

Macroeconomic policy analysis needs the distribution, not only its aggregates, and putting it into the recursive state turns general equilibrium into two nested fixed points. DSPG removes the trade-off rather than balancing it: the policy is conditioned on the complete discretized histogram, household dynamics and market clearing come from the model, and the inner price fixed point is resolved by exact clearing inside the environment rather than by the policy. Only the outer fixed point is learned, by an analytic gradient, so the interest rate is an output of clearing rather than a training target. The evidence is consistent: the lowest diagnostic, 2.52×10^{-3} against 6.50×10^{-3} for the strongest baseline, from a 783-parameter U-Net 614 times smaller than the MLP-128 it beats, with one configuration solving all 75 economies of a risk sweep without retuning. The economics come with it: aggregate risk moves the volatility of the return, idiosyncratic risk its level, and raising σ_e from 0.2 to 0.6 lifts the wealth Gini from 0.3212 to 0.3988 while mean assets stay pinned, so nearly unchanged aggregates conceal a redistribution only the distribution reveals. Richer balance sheets and a uniqueness result for the learned outer fixed point are the natural next steps.

AI USE STATEMENT

In this work, we used generative AI tools for the following tasks: drafting English prose from the authors' Chinese outlines, translating the authors' Chinese drafts into English, locating and formatting bibliographic entries, cross-checking numerical claims in the manuscript against the experiment outputs saved in the code repository, and LaTeX formatting fixes. Generative AI was also used to optimise parts of the solver code; the original implementation was written by the authors, and every AI-assisted edit was reviewed and tested by the authors. Generative AI contributed to approximately 20% of the statistical analysis underlying the reported results, which the authors reviewed and checked. The research ideas and all architectural choices in this work are entirely the authors' own. We have reviewed all AI-assisted work. Every AI-drafted passage was rewritten or approved by the authors. Each bibliographic entry was verified against publisher or repository records before inclusion, and each numerical claim was checked against the repository's saved results. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work is a computational method for dynamic economic models. It involves no human subjects, no personal, sensitive or identifiable data, and no experiments on living systems; the economies studied are calibrated to published macroeconomic parameters, and the computations are simulation-based. The method is intended to make quantitative policy analysis more reliable, and its outputs are model-based counterfactuals rather than forecasts or policy recommendations. We are not aware of a specific harm vector for this work, and we note that any such tool inherits the limitations of the model it solves and of the data used to calibrate it.

REPRODUCIBILITY STATEMENT

The conditional-gradient derivation is in Appendix A. The complete model specification, including the grid construction and both shock processes, is in Appendix B, as are the evaluation conventions, the reported training objective, and the Euler diagnostics. The partial-equilibrium comparison is in Appendix C, and the protocol and caveats of each baseline are in Appendix E. Code, configurations, and scripts that produce every number are available at `https://anonymous.4open.science/r/DSPG-EB78`.

REFERENCES

Yves Achdou, Jiequn Han, Jean-Michel Lasry, Pierre-Louis Lions, and Benjamin Moll. Income and wealth distribution in macroeconomics: A continuous-time approach. *The Review of Economic Studies*, 89(1):45–86, 2022.

SeHyoun Ahn, Greg Kaplan, Benjamin Moll, Thomas Winberry, and Christian Wolf. When inequality matters for macro and macro matters for inequality. *NBER Macroeconomics Annual*, 32(1): 1–75, 2018.

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

Adrien Auclert. Monetary policy and the redistribution channel. *American Economic Review*, 109 (6):2333–67, 2019.

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.

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

Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

Truman Bewley. Stationary monetary equilibrium with a continuum of independently fluctuating consumers. *Contributions to mathematical economics in honor of Gérard Debreu*, 79, 1986.

Javier Bianchi and Greg Kaplan. How small is small? non-linearities in heterogeneous agent models. Working Paper 35311, National Bureau of Economic Research, 2026.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. Jax: composable transformations of python+numpy programs, 2018. URL `http://github.com/jax-ml/jax`.

René Carmona and François Delarue. Probabilistic theory of mean field games with applications. i, volume 83 of probability theory and stochastic modelling, 2018.

Marc Deisenroth and Carl E Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 465–472, 2011.

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*, pp. 1861–1870. Pmlr, 2018.

Jiequn Han, Yucheng Yang, and Weinan E. DeepHAM: A global solution method for heterogeneous agent models with aggregate shocks. *Quantitative Economics*, 17(2):297–341, 2026. doi: 10.3982/QE2190.

Jonathan Heathcote, Kjetil Storesletten, and Giovanni L Violante. Quantitative macroeconomics with heterogeneous households. *Annual Review of Economics*, 2009.

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

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, volume 32, pp. 12498–12509, 2019.

Greg Kaplan and Giovanni L Violante. Microeconomic heterogeneity and macroeconomic shocks. *Journal of Economic Perspectives*, 32(3):167–94, 2018.

Greg Kaplan, Giovanni L Violante, and Justin Weidner. The wealthy hand-to-mouth. *Brookings Papers on Economic Activity*, 2014(1):77–138, 2014.

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

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

Dirk Krueger, Kurt Mitman, and Fabrizio Perri. Macroeconomics and household heterogeneity. In *Handbook of macroeconomics*, volume 2, pp. 843–921. Elsevier, 2016.

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

Jean-Michel Lasry and Pierre-Louis Lions. Mean field games. *Japanese journal of mathematics*, 2 (1):229–260, 2007.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations*, 2016.

Lilia Maliar, Serguei Maliar, and Pablo Winant. Deep learning for solving dynamic economic models. *Journal of Monetary Economics*, 122:76–101, 2021.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Benjamin Moll. The trouble with rational expectations in heterogeneous agent models: A challenge for macroeconomics. *The Economic Journal*, 136(676):1173–1216, 2026. doi: 10.1093/ej/ueaf104.

K Geert Rouwenhorst. Asset pricing implications of equilibrium business cycle models. In Thomas F Cooley (ed.), *Frontiers of Business Cycle Research*, pp. 294–330. Princeton University Press, Princeton, NJ, 1995.

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

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.

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

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

A CONDITIONAL HOUSEHOLD GRADIENT

The household interpretation can be stated by fixing the population policy $\bar{\theta}$ and a common-shock path. Let $\mathcal{E}^{\bar{\theta}} = (g_t^{\bar{\theta}}, z_t, r_t^{\bar{\theta}})_{t=0}^{T-1}$ be the resulting environment. A household choosing θ has state law d_t^θ, with a fixed initial law d_0, and objective

$$d_{t+1}^\theta = (K_t^\theta)^\top d_t^\theta, \qquad J_T(\theta; \mathcal{E}^{\bar{\theta}}) = \sum_{t=0}^{T-1} \beta^t (d_t^\theta)^\top u_t^\theta. \tag{8}$$

Here $u_t^\theta(s) = u(c_\theta(s; g_t^{\bar{\theta}}, z_t))$, and the kernel uses the fixed environment price. Differentiation holds the population environment fixed and retains the household's own state transition. Combining this household problem with population consistency gives the usual mean field formulation (Lasry & Lions, 2007; Carmona & Delarue, 2018; Yang et al., 2025; Wibault et al., 2026).

Proposition A.1 (Conditional structural gradient)**.** *For a fixed finite environment path, suppose rewards and transition matrices are differentiable at θ. Define $V_T = 0$ and $V_t = u_t^\theta + \beta K_t^\theta V_{t+1}$. Then*

$$\partial_\theta J_T = \sum_{t=0}^{T-1} \beta^t (d_t^\theta)^\top \left[\partial_\theta u_t^\theta + \beta(\partial_\theta K_t^\theta) V_{t+1}\right]. \tag{9}$$

Proof. Differentiate the value recursion:

$$\partial_\theta V_t = \partial_\theta u_t^\theta + \beta(\partial_\theta K_t^\theta) V_{t+1} + \beta K_t^\theta \partial_\theta V_{t+1}.$$

Multiply by $\beta^t (d_t^\theta)^\top$ and sum over time. Using $(d_t^\theta)^\top K_t^\theta = (d_{t+1}^\theta)^\top$ cancels successive continuation-derivative terms. Since $V_T = 0$ and d_0 is fixed, the remaining left side is $\partial_\theta J_T$. □

Under integrable derivative bounds, expectation and differentiation commute, and independent common-path averaging estimates the finite-horizon household gradient. Individual transitions are integrated directly. The identity applies away from interpolation and clipping breakpoints. If rewards are uniformly bounded by U, the value truncation error is at most $\beta^T U/(1-\beta)$; a gradient-tail bound requires additional derivative assumptions.

B IMPLEMENTATION AND EVALUATION CONVENTIONS

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 $\sigma = 1$, i.e. logarithmic utility $u(c) = \log c$; the extension economies use CRRA curvature $\sigma = 2$. We set the borrowing constraint $\underline{a} = 0$ and the total bond supply $B = 5$, a setting that differs from the original Huggett (1993) model, with three idiosyncratic income states and five aggregate states.

The idiosyncratic shock follows a three-state Markov chain approximating the log AR(1) process

$$\log(e_{i,t+1}) = \rho_e \cdot \log(e_{i,t}) + \sigma_e \cdot \sqrt{1-\rho_e^2} \cdot \epsilon_{i,t}, \quad \epsilon_{i,t} \sim N(0,1), \tag{10}$$

with persistence $\rho_e = 0.6$ and standard deviation $\sigma_e = 0.2$, as in the Aiyagari (1994) model. Its grid and transition matrix are

$$\mathbf{e} = \begin{bmatrix} 0.7388 \\ 0.9803 \\ 1.3007 \end{bmatrix} \quad \text{and} \quad \mathbf{P_e} = \begin{bmatrix} 0.6400 & 0.3200 & 0.0400 \\ 0.1600 & 0.6800 & 0.1600 \\ 0.0400 & 0.3200 & 0.6400 \end{bmatrix}, \tag{11}$$

rounded to four decimals; the grid is scaled so that its stationary mean is one, $\bar{e} = 1$.

The aggregate shock follows a five-state Markov chain approximating the same log AR(1) form,

$$\log(z_{t+1}) = \rho_z \cdot \log(z_t) + \sigma_z \cdot \sqrt{1-\rho_z^2} \cdot \epsilon_t, \quad \epsilon_t \sim N(0,1), \tag{12}$$

with persistence $\rho_z = 0.8$ and standard deviation $\sigma_z = 0.05$, chosen to keep the aggregate shock steady and controllable and thereby reduce the difficulty of solving. Its grid and transition matrix

are

$$\mathbf{z} = \begin{bmatrix} 0.9037 \\ 0.9500 \\ 0.9988 \\ 1.0500 \\ 1.1038 \end{bmatrix} \quad \text{and} \quad \mathbf{P}_{\mathbf{z}} = \begin{bmatrix} 0.6561 & 0.2916 & 0.0486 & 0.0036 & 0.0001 \\ 0.0729 & 0.6804 & 0.2214 & 0.0244 & 0.0009 \\ 0.0081 & 0.1476 & 0.6886 & 0.1476 & 0.0081 \\ 0.0009 & 0.0244 & 0.2214 & 0.6804 & 0.0729 \\ 0.0001 & 0.0036 & 0.0486 & 0.2916 & 0.6561 \end{bmatrix}, \tag{13}$$

rounded to four decimals.

Discretization. Wealth a and income e are the two heterogeneous attributes, and income is discretized to $n_e = 3$ states above. Wealth uses $n_a = 200$ points with $a_{\min} = \underline{a} = 0$ and $a_{\max} = 149.75$, spaced as $\text{geomspace}(0.25, 150, 200) - 0.25$, so the distribution $\mathbf{g}_t$ is a vector of dimension $n_a n_e = 600$; there are $n_z = 5$ aggregate states. The steady-state calculation supplies the consumption bounds and the starting distribution, and the first 50 epochs start from it.

Architecture. The five U-Net double-convolution blocks have 604 parameters, the two transposed convolutions have 104, and the final convolution has 75, for a total of 783. All blocks use four channels. A two-hidden-layer full-input MLP of width h has $h^2 + 3602h + 3000$ parameters for this grid. The extended architecture calibration uses $\sigma_e = 0.2$, $\sigma_z = 0.05$, and curvature $\sigma = 2$, with batch sizes 32, 64, 128, 256 and learning rates 0.0002, 0.002, 0.02. Each configuration has ten seeds. These results are kept distinct from the log-utility baseline.

Forward update and reported objective. The implementation generates a plan $\widehat{c}_t$, computes the price in equation 2, and applies

$$c_t(s) = \min\{\widehat{c}_t(s), (1 + r_t)a + z_t e - \underline{a} - 10^{-3}\}.$$

It then transfers probability mass according to the chosen next assets and P_e. The original training score uses the updated-histogram convention $\sum_t \beta^t \sum_s u(c_t(s)) g_{t+1}(s)$. We retain this convention when reporting existing training objectives, separately from the conditional household formulation and the economic report's stationary welfare statistic. The price expression is treated as a stop-gradient quantity during training.

Proposition B.1 (Histogram conservation and clearing)**.** *If asset interpolation weights are nonnegative, sum to one, and exactly reproduce each chosen next asset, a row-stochastic income transition preserves nonnegative unit population mass. If the consumption plan is feasible without modification and the price satisfies equation 2, next-period mean assets equal B.*

Proof. For each current state the asset weights and the income probabilities each sum to one, so the joint transition does too. Mean preservation of the asset interpolation gives

$$A(g_{t+1}) = (1 + r_t)A(g_t) + Y(g_t, z_t) - C_\theta(g_t, z_t) = B.$$

□

If $\widehat{C}$ is planned consumption, C is executed consumption, and $\varepsilon_{\text{grid}}$ is the mean projection error, the residual is $A(g_{t+1}) - B = \widehat{C} - C + \varepsilon_{\text{grid}}$. The baseline figures report measured residuals for their own simulations.

Euler diagnostics. The evaluation uses distribution-weighted absolute Euler diagnostics. In the saved DSPG evaluator, the continuation policy tensor is evaluated at the current input distribution and the current return:

$$c^E(a, e; g, z) = \left[\beta(1 + r) \sum_{e', z'} P_e(e, e') P_z(z, z') \widehat{c}(a', e', z'; g)^{-\sigma} \right]^{-1/\sigma},$$

$$\text{EE}(g, z) = \sum_{a, e} g(a, e) \left| 1 - \widehat{c}(a, e, z; g) / c^E(a, e; g, z) \right|.$$

The common-return convention is also stated in the equilibrium-baseline report. Earlier mean-log errors, weighted absolute errors, and groupwise averages have different aggregation rules and are not pooled. These quantities are numerical diagnostics under the reported evaluation convention.

C HOUSEHOLD ACCURACY AGAINST A HIGH-PRECISION NUMERICAL SOLUTION

The partial-equilibrium experiment isolates the household problem and uses value-function iteration (VFI) as a high-precision numerical reference. The evaluation statistic is the household's lifetime discounted utility, the same object maximized in the economic model, so higher values are better. One convention differs: in the reference's evaluation window the aggregate states are drawn independently and uniformly rather than from the transition matrices the learning methods and their environment follow, so its value is a reference point rather than a like-for-like reading of the same statistic. Table 4 summarizes the ten-seed results, and Figure 6 shows the training curves. DSPG reaches final evaluation utility 5.6612, close to the VFI grid value 5.6534 and above the tested PPO, SAC, and DDPG final means. Its across-seed final variance is also the smallest among the learning methods.

Method	Last eval		Best eval	
	avg	var	avg	var
DSPG	**5.6612**	**0.003222**	5.7315	**0.000642**
PPO	5.1279	0.131985	**5.9101**	0.066618
SAC	4.2658	0.059560	4.7163	0.019283
DDPG	4.5016	0.500776	5.2248	0.220691
VFI	5.6534	—	5.6534	—

Table 4: Partial-equilibrium methods: ergodic discounted utility (avg and sample variance across random seeds). Bold: best avg per column; best (lowest) var per column.

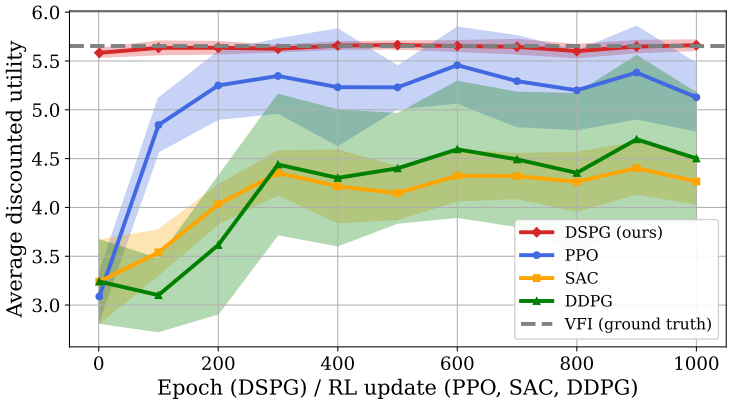


Figure 6: Partial-equilibrium training curves for DSPG, PPO, SAC, DDPG, and the VFI reference.

The result supports the use of known household dynamics in learning the consumption rule. PPO attains a higher best-checkpoint mean, while DSPG has the stronger final mean and lower reported variance. The economic comparison is therefore most informative about final accuracy and stability under the recorded training protocol.

D ADDITIONAL ORIGINAL EXPERIMENTS

The following figures and table reproduce the original experiment assets. They supplement the dynamic solution, StructuralRL distribution comparison, and partial-equilibrium results in the main text.

The original population-weighted mean is 0.002933 (Table 5). Errors are larger near the borrowing constraint and in some income groups. This motivates evaluating the distribution of accuracy together with the distribution of wealth. The earlier U-Net/CNN/large-MLP comparison reports

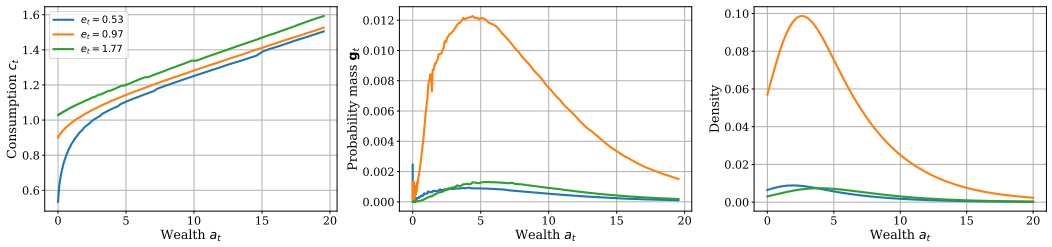


Figure 7: Baseline steady-state consumption policies, probability masses, and densities used as the baseline numerical reference.

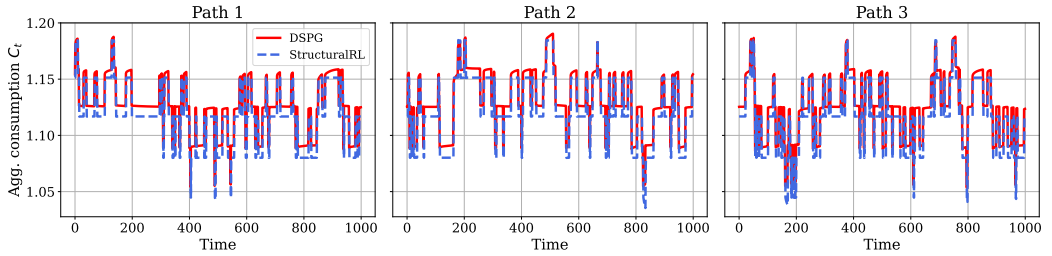


Figure 8: Baseline DSPG and StructuralRL aggregate-consumption paths. The agreement supports the economic consistency of the solutions.

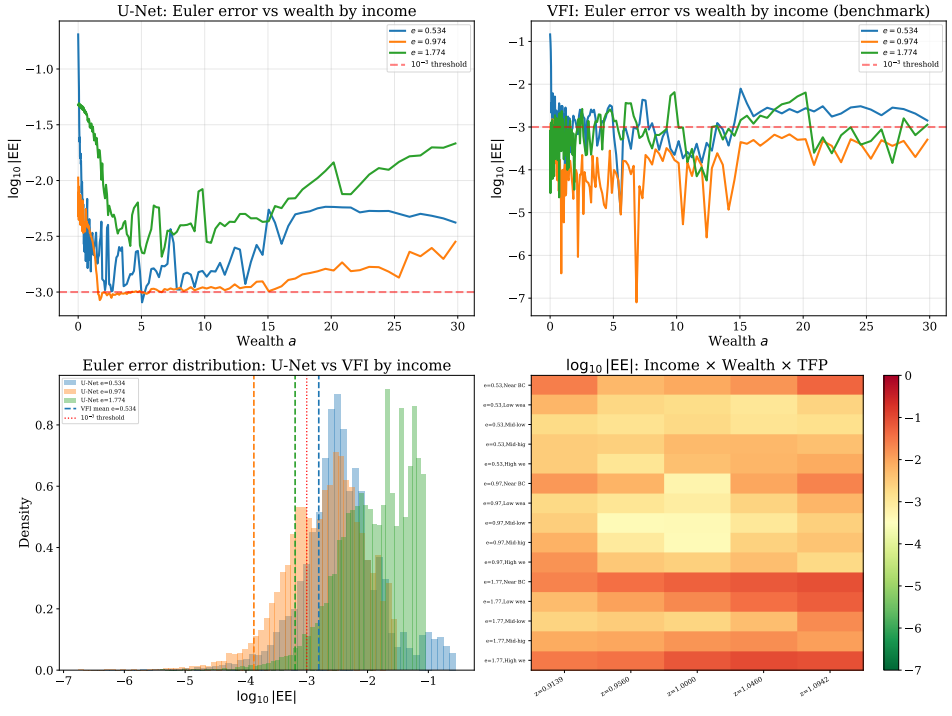


Figure 9: Baseline wealth-profile decomposition of Euler diagnostics. This reveals where errors occur within the population, beyond the aggregate average.

Table 5: Euler error decomposition by income, wealth, and aggregate TFP shock. VFI benchmark: $\log_{10}|\text{EE}| = -3.19$.

Group	wmean $\lvert\text{EE}\rvert$	mean $\lvert\text{EE}\rvert$	max $\lvert\text{EE}\rvert$	$\log_{10}\lvert\text{EE}\rvert$
Overall				
Overall	2.933E-03	1.620E-02	2.788E-01	-2.27
By Income Group				
Low income (e=0.534)	7.329E-03	1.153E-02	2.788E-01	-2.43
Mid income (e=0.974)	1.991E-03	4.773E-03	3.017E-02	-2.63
High income (e=1.774)	9.500E-03	3.228E-02	1.834E-01	-1.75
By Wealth Group				
Near BC	1.908E-02	3.424E-02	2.788E-01	-1.81
Low wealth	2.990E-03	1.145E-02	7.693E-02	-2.39
Mid-low	2.110E-03	3.585E-03	3.379E-02	-2.74
Mid-high	2.790E-03	7.932E-03	6.084E-02	-2.37
High wealth	9.800E-03	2.377E-02	1.834E-01	-2.03
By TFP Shock				
$z = 0.9139$	3.265E-03	1.119E-02	1.582E-01	-2.20
$z = 0.9560$	1.484E-03	9.824E-03	1.749E-01	-2.46
$z = 1.0000$	1.508E-03	1.504E-02	2.116E-01	-2.44
$z = 1.0460$	3.089E-03	2.100E-02	2.463E-01	-2.17
$z = 1.0942$	5.917E-03	2.678E-02	2.788E-01	-2.00

mean-log errors -2.25, -2.17, and -2.29, using 783, 919, and 990,648 parameters, respectively. That experiment is distinct from the extended MLP-width scan.

E EQUILIBRIUM BASELINE PROTOCOLS

The existing comparison uses the extension economy with $B = 5$, $\beta = 0.975$, 200 asset points, three income states, five common states, and CRRA curvature 2. Evaluation uses 512 simulated economies, a 2,000-period burn-in and a 4,000-period panel, as specified in the report. For price-grid methods, the return is parameterized in log gross-return units on a 20-point grid spanning approximately $[-3.5\%, 13.2\%]$. The headline diagnostic uses the current-return continuation convention.

DSPG and StructuralRL. The DSPG row is the MLP-32 reference with batch 128, learning rate 0.002 and 1,000 epochs. StructuralRL-GRID uses a price-conditioned consumption-share tensor; StructuralRL-NN replaces it with a two-layer width-32 network. The StructuralRL variants retain learning rate 10^{-3}, exponential decay, sample size 128 and gradient clipping. StructuralRL uses a constant-share market-clearing initialization rather than a VFI policy warm start. Its range and initialization checks are part of the existing experiment. The reference-pipeline validation on the StructuralRL authors' calibration reports EE 1.53×10^{-3} against the cited 1.93×10^{-3}, supporting the port's behavior on that reference task.

DEQN. DEQN is already included in the completed report. The implementation follows the continuum-of-agents architecture in Azinovic et al. (2022): a consumption/KKT network, a value network and a price network. The loss includes Euler, Bellman, complementarity and clearing residuals, with an additional auctioneer term to train the price network at the moderate training budget. It uses 1,000 episodes, each with 512 simulated periods on 256 tracks and 64 minibatches of 128. Adam has learning rate 10^{-5}, with the price network at one fifth of that rate. This is a budgeted adaptation, not the original paper's full training schedule.

The five-seed DEQN mean is 6.66×10^{-3} and the table reports 4,501 seconds. The recorded seed range is approximately 5.00×10^{-3} to 8.78×10^{-3}, and losses still fall at the final episode. These details qualify the comparison without changing its result: DSPG has the lower reported diagnostic and shorter recorded time at these settings.

VFI+PLM and timing. The household grid solution is wrapped in an outer perceived-law-of-motion iteration. The iteration 2-cycles, and we retain a cycle-point law after nine rounds. Its low diagnostic is useful as a numerical reference, but the outer iteration has not converged. The VFI+PLM timing is CPU-based; the neural methods and their updates differ in hardware use and work per epoch. Table 2 preserves the existing totals, rather than equating 1,000 episodes with an identical number of operations for every method.

F ARCHITECTURE AND OPTIMIZATION EVIDENCE

This section documents the two sweeps behind subsection 5.3: an earlier baseline trajectory-batch experiment, and the extended scan over architecture, batch size, and learning rate. They are reported separately because they were run on different batch grids and should not be pooled.

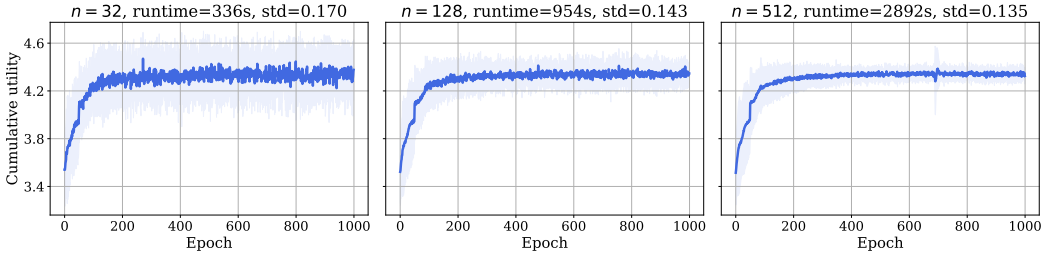


Figure 10: Baseline trajectory-batch experiment, ten repetitions per setting. Larger batches reduce reported training dispersion at increased time cost.

The original 1,000-epoch timings for batches 32, 128 and 512 are 335.84, 954.31 and 2,891.64 seconds (Figure 10), with reported dispersions 0.1704, 0.1430 and 0.1349. The complete baseline training is therefore on the scale of minutes under those conditions, and the dispersion falls monotonically as the batch grows.

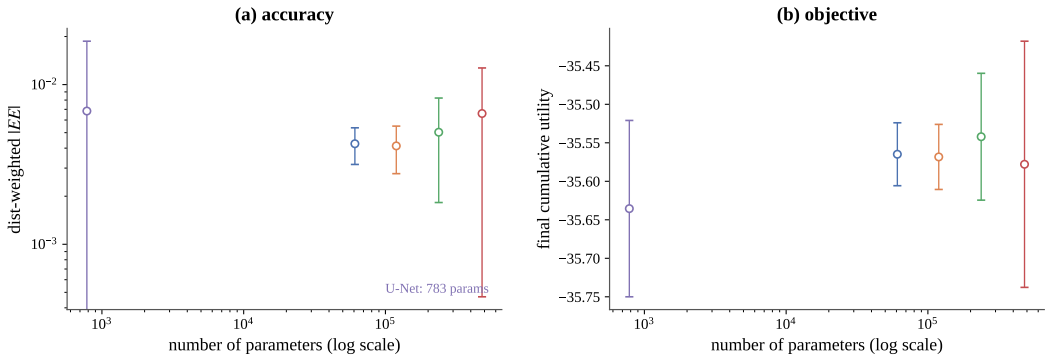


Figure 11: Architecture frontier aggregated over batch sizes and learning rates, 120 runs per architecture. The error bars mix settings as well as seeds. The reference-rate comparison in Table 3 isolates the architecture result more directly.

Figure 11 pools each architecture over the other two factors, so its error bars mix settings as well as seeds; Table 3 holds the learning rate at the reference value and is the cleaner statement of the architecture result. Figure 12 and Table 6 do the same for batch size. Their pooled means sit at 5.1 to 5.8 $\times$ 10^{-3}, well above the 2.5 to 3.1 $\times$ 10^{-3} of Table 3, because the pool includes the aggressive learning rate; pooled and single-rate numbers must not be compared directly. Wall time grows sublinearly in the batch, from 324 s at $N = 32$ to 1,335 s at $N = 256$, so any accuracy gain from a larger batch has to be weighed against that cost.

The learning rate is the most consequential of the three factors. At the reference rate the scan reaches a weighted Euler diagnostic of 3.0×10^{-3} with a dispersion of only 3.0×10^{-4} over the 200 runs;

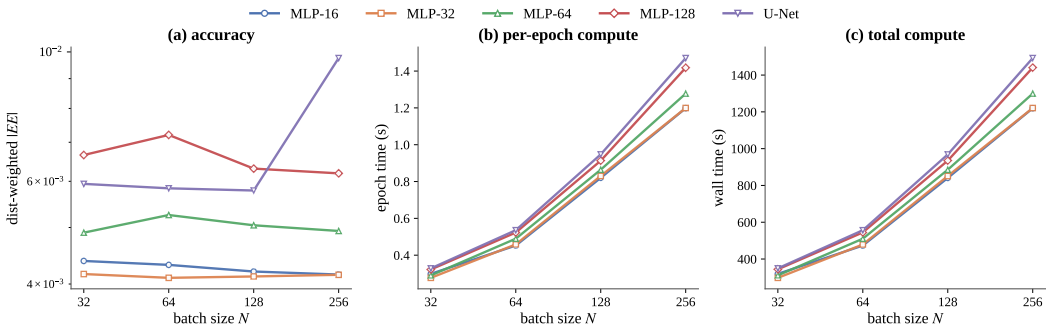


Figure 12: Batch-size comparison: weighted error, per-epoch time, and total wall time. These are aggregated results from the extended scan.

Table 6: Batch summary, 150 runs per batch across the other settings.

Batch size N	dist-w. $\lvert EE\rvert$	epoch time (s)	wall time (s)
32	5.2E-3±3.9E-3	0.30	324
64	5.3E-3±4.8E-3	0.49	513
128	5.1E-3±3.7E-3	0.88	896
256	5.8E-3±1.0E-2	1.31	1335

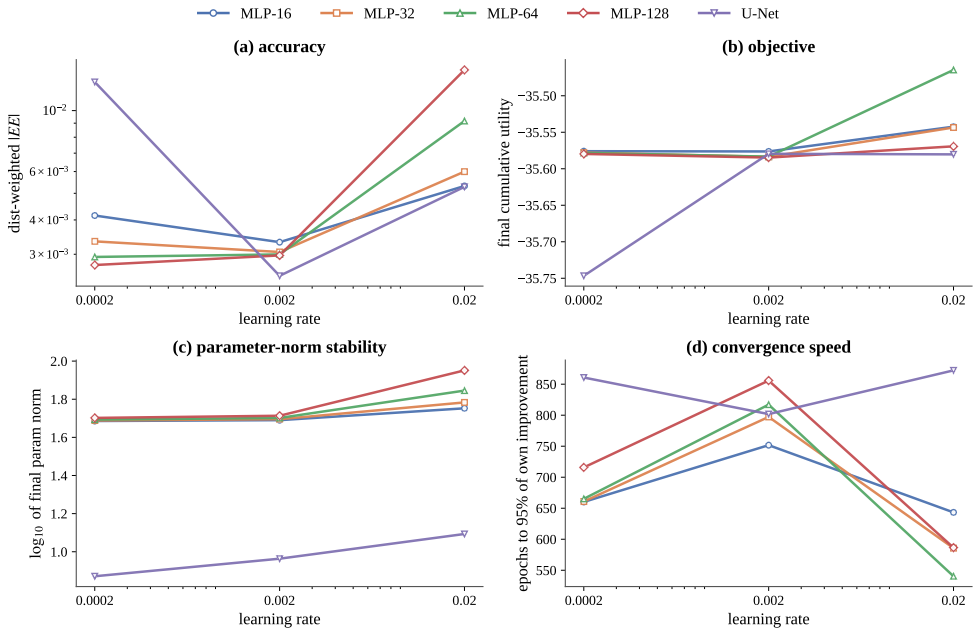


Figure 13: Learning-rate comparison. The objective, Euler diagnostic, parameter norm, and progress measure distinguish numerical quality from a high training score.

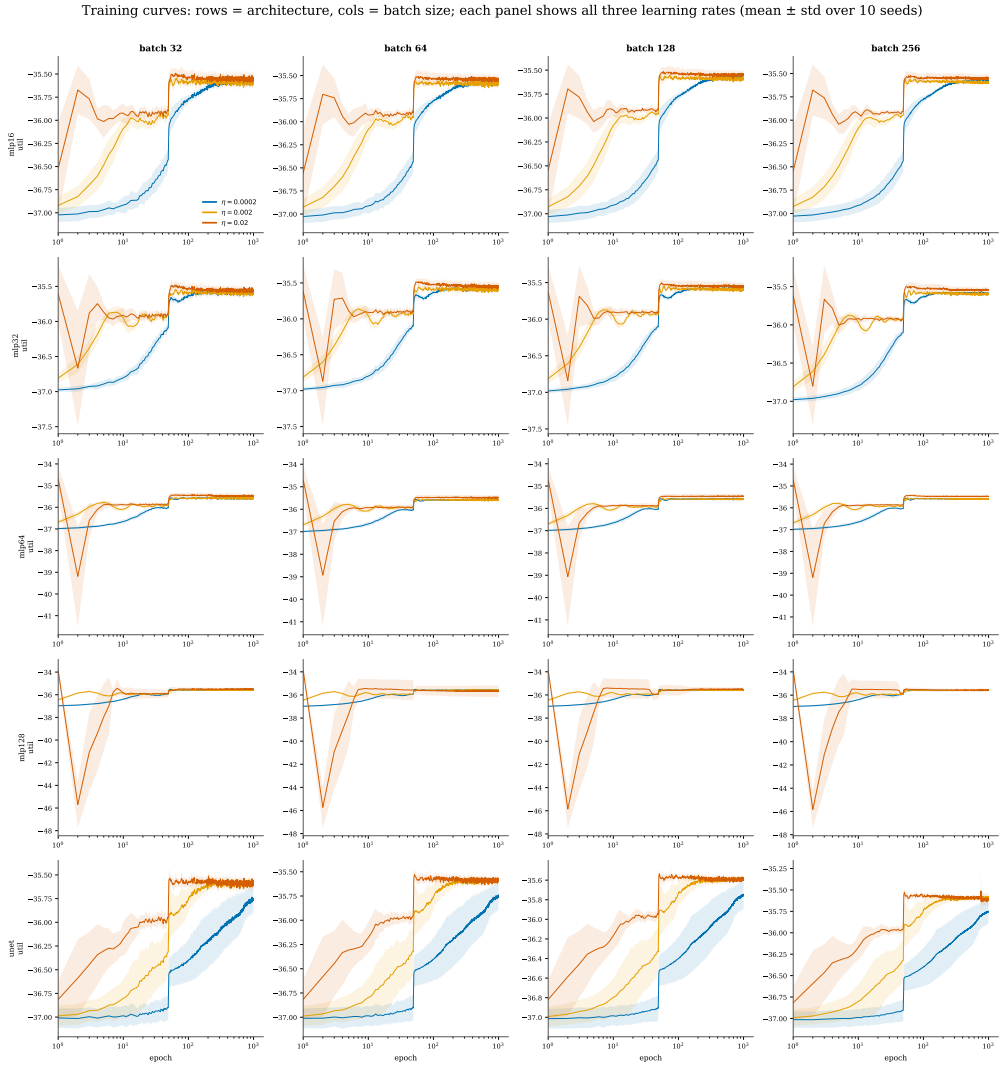


Figure 14: Training curves with seed means and standard deviations under different batch sizes and network architectures.

Table 7: Learning-rate summary, 200 runs per rate.

LR	dist-w. $\lvert EE\rvert$	final utility	$\log_{10}\lVert\theta\rVert$	clearing err (mean / max)	epochs to 95%
0.0002	5.2E-3±3.9E-3	-35.612±0.093	1.53	1.8e-04 / 4.1e-04	713
0.002	3.0E-3±3.0E-4	-35.582±0.034	1.55	1.7e-04 / 4.0e-04	805
0.02	8.0E-3±9.5E-3	-35.540±0.140	1.69	9.3e-04 / 1.4e-01	646

the aggressive rate is both less accurate on average (8.0×10^{-3}) and far more variable (9.5×10^{-3}), and its worst configuration fails to clear at all (maximum clearing residual 1.4×10^{-1}, against 4.0×10^{-4} at the reference rate). The slower rate is closer but still more dispersed than the reference ($5.2 \times 10^{-3} \pm 3.9 \times 10^{-3}$). Figure 13 sets these against the objective, the parameter norm and the progress measure, so the reference rate can be judged on more than one diagnostic, and Figure 14 shows the same settings epoch by epoch with seed means and standard deviations.

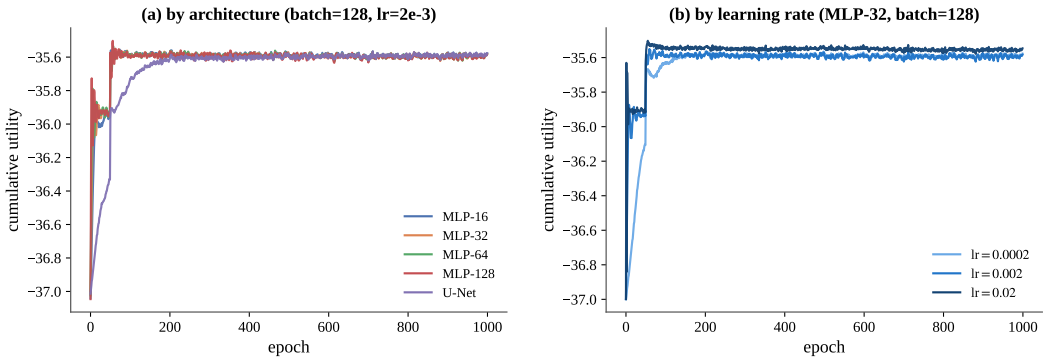


Figure 15: Seed-mean training curves: architectures at batch 128 and learning rate 0.002, and learning rates for MLP-32 at batch 128.

Figure 15 contrasts the architectures at the reference batch and rate with the three learning rates at MLP-32. Both panels are logged at the same 1,000-epoch horizon, so the curves can be read against the endpoint values in Tables 3 and 7. Across all of these exhibits, collected per seed in Figure 16, the figures retain every tested learning rate: finite training over 600 runs does not mean every setting is equally accurate, and the aggressive rate in particular is better read as a stability boundary than as a usable configuration.

G Risk, distribution, and welfare evidence

The risk design uses $\sigma_z \in \{0.001, 0.005, 0.01, 0.05, 0.1\}$, $\sigma_e \in \{0.2, 0.4, 0.6\}$, and utility curvature $\sigma \in \{1, 1.5, 2, 2.5, 3\}$, five seeds each. The persistence parameters are $\rho_z = 0.8$ and $\rho_e = 0.6$. Each policy is MLP-32. The panel has 512 trajectories, a 2,000-period burn-in, and a 4,000-period panel, the section-1 convention (the 500-period / 100-period figures are the training library's internal rollout, not the reported evaluation).

At the largest aggregate-risk setting under log utility and $\sigma_e = 0.2$, the sweep gives conditional impact responses of approximately $+1.87$ and -2.50 percentage points in low and high common states, which the impulse-response exhibit in subsection 5.4 plots. The response decays with the persistence of the common shock. Aggregate consumption is less dispersed than income, consistent with consumption smoothing.

Under log utility and $\sigma_e = 0.2$, raising σ_z from 0.001 to 0.1 leaves the mean rate near 2.5% but lifts its standard deviation from 0.0222 to 1.0806 percentage points. Raising σ_e from 0.2 to 0.6 at low aggregate risk instead lowers the mean rate from 2.52% to 1.85%, alongside the wealth-distribution changes.

Consumption-equivalent welfare. For a fixed utility specification, define $\mathrm{CE} = u^{-1}(\bar{u})$, where $\bar{u}$ is average per-period utility. Compensation relative to a reference cell is $100(\mathrm{CE}_{\mathrm{ref}}/\mathrm{CE} - 1)$. The following tables reproduce the log-utility comparisons and their seed variation.

Within log utility, the reported increase in aggregate risk costs 0.254% of consumption (Figure 19), while the increase in idiosyncratic risk from 0.2 to 0.6 costs 5.565%. The risk increments differ, so these are costs of the specified comparisons, not a unit-normalized ranking.

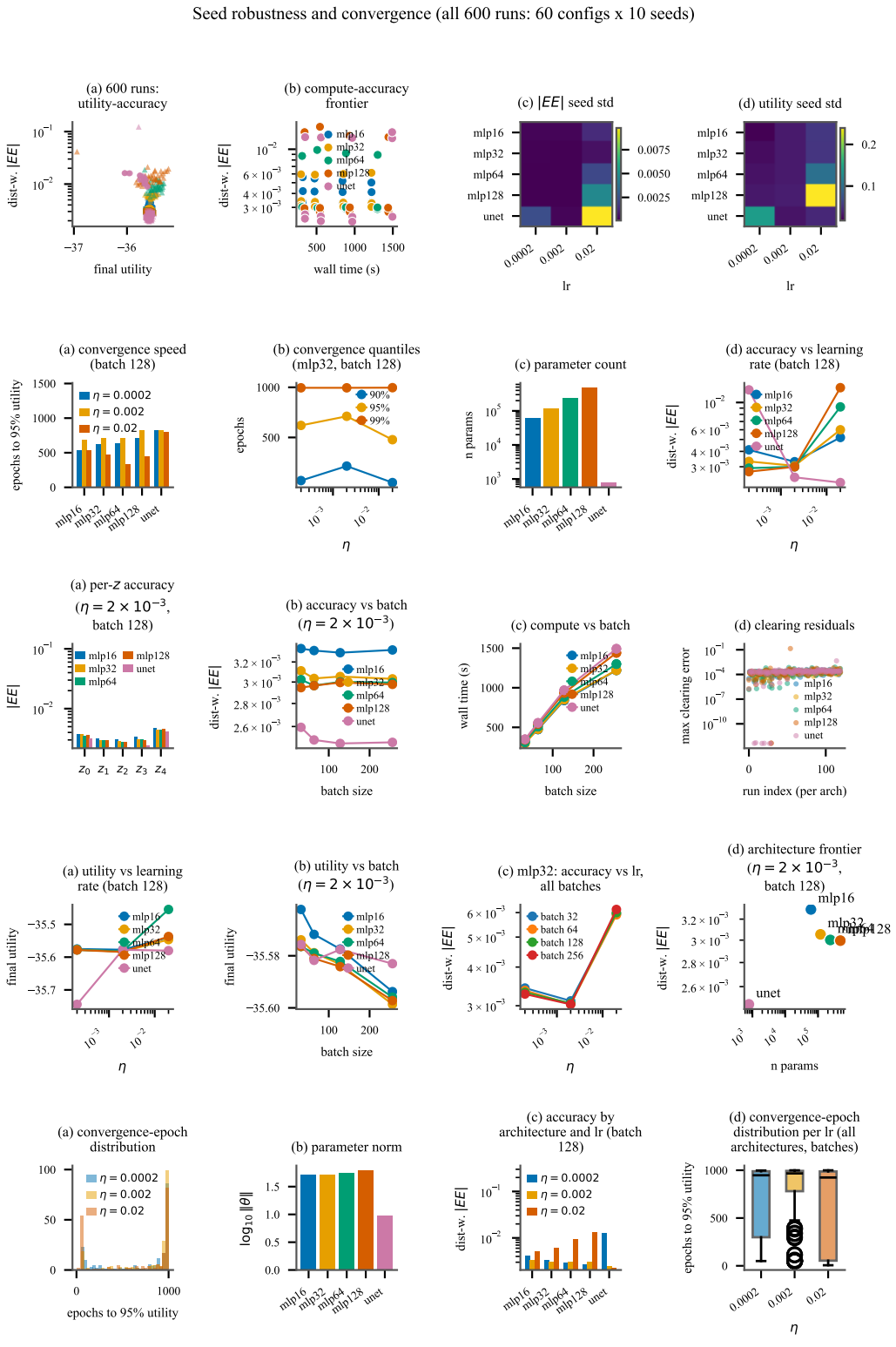


Figure 16: Seed robustness and convergence across all 600 runs (60 configurations, ten seeds each), in twenty panels: the utility–accuracy relation, the compute–accuracy frontier, seed dispersion by architecture and learning rate, convergence speed and its quantiles, parameter counts, per-z accuracy, the batch and learning-rate sweeps, the architecture frontier, and clearing residuals.

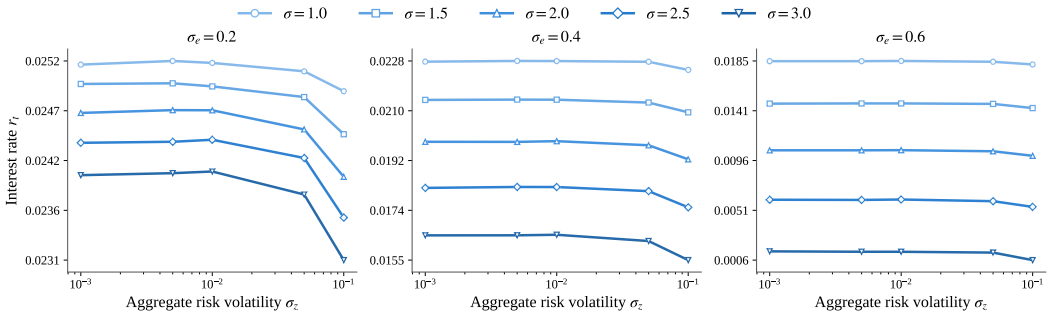


Figure 17: Mean interest-rate results. Panels fix idiosyncratic risk; curves distinguish utility curvature.

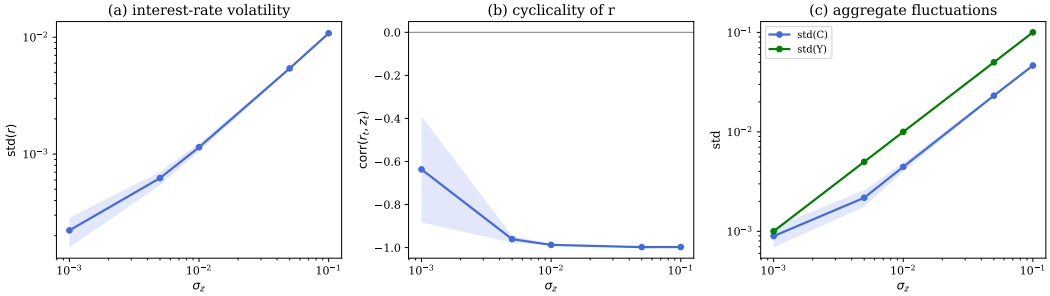


Figure 18: Volatility, cyclicality and aggregate-fluctuation results across aggregate risk.

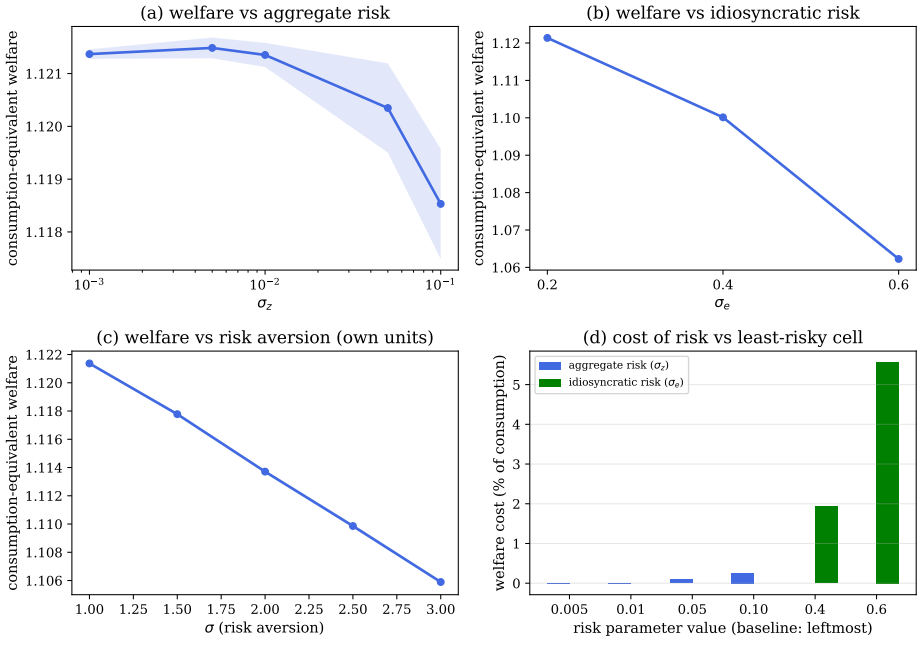


Figure 19: Consumption-equivalent welfare results. Panel (d) reports compensation relative to the lowest risk values. Comparisons are interpreted within a given utility specification.

Table 8: Welfare (consumption-equivalent) as a function of aggregate risk σ_z, at $\sigma_e = 0.2$ and log utility. Cost is the compensating variation relative to the least-risky cell ($\sigma_z = 0.001$), in percent of consumption.

σ_z	CE	cost (%)
0.001	1.1214 (0.0001)	0.000 (0.008)
0.005	1.1215 (0.0002)	-0.010 (0.017)
0.010	1.1214 (0.0002)	0.001 (0.020)
0.050	1.1203 (0.0008)	0.091 (0.076)
0.100	1.1185 (0.0010)	0.254 (0.094)

Table 9: Welfare (consumption-equivalent) as a function of idiosyncratic risk σ_e, at $\sigma_z = 0.001$ and log utility. Cost is relative to the least-risky cell ($\sigma_e = 0.2$), in percent of consumption.

σ_e	CE	cost (%)
0.2	1.1214 (0.0001)	0.000 (0.008)
0.4	1.1001 (0.0001)	1.931 (0.010)
0.6	1.0623 (0.0000)	5.565 (0.001)

The combined consumption cost reaches 5.840% at the extreme two-risk cell, close to the sum of its marginal comparisons. This is a feature of the evaluated grid, not a general separability property. We interpret welfare comparisons within a utility specification.

The accuracy and training exhibits document numerical behavior over the risk grid; Figures 25 and 26 give the complete impulse-response and seed-level libraries.

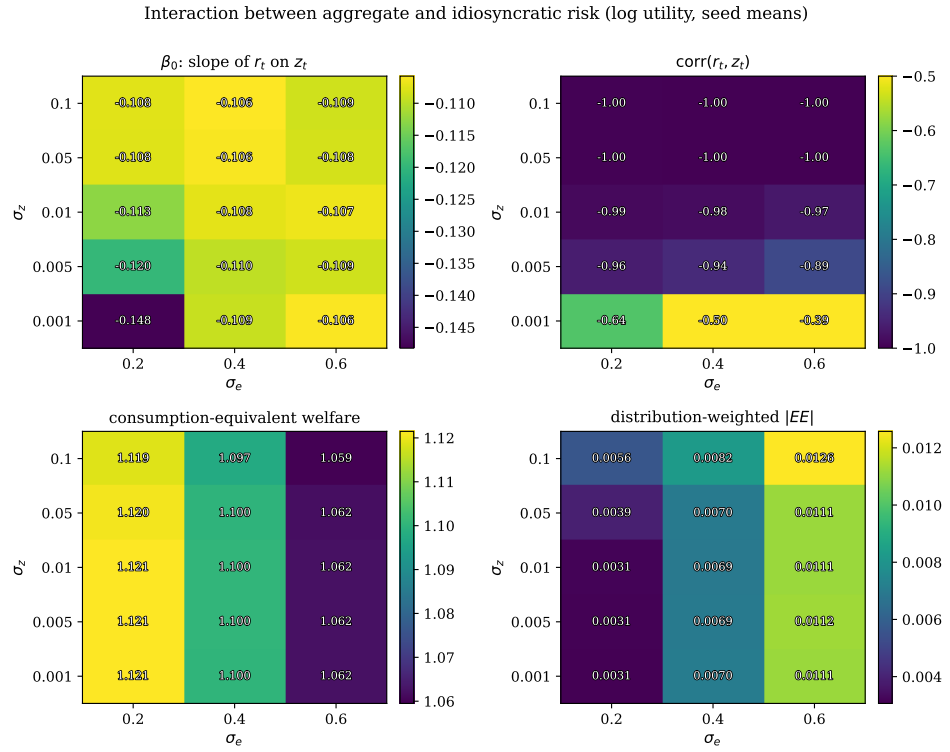


Figure 20: Interaction plots at log utility: interest-rate sensitivity, cyclicality, welfare, and error.

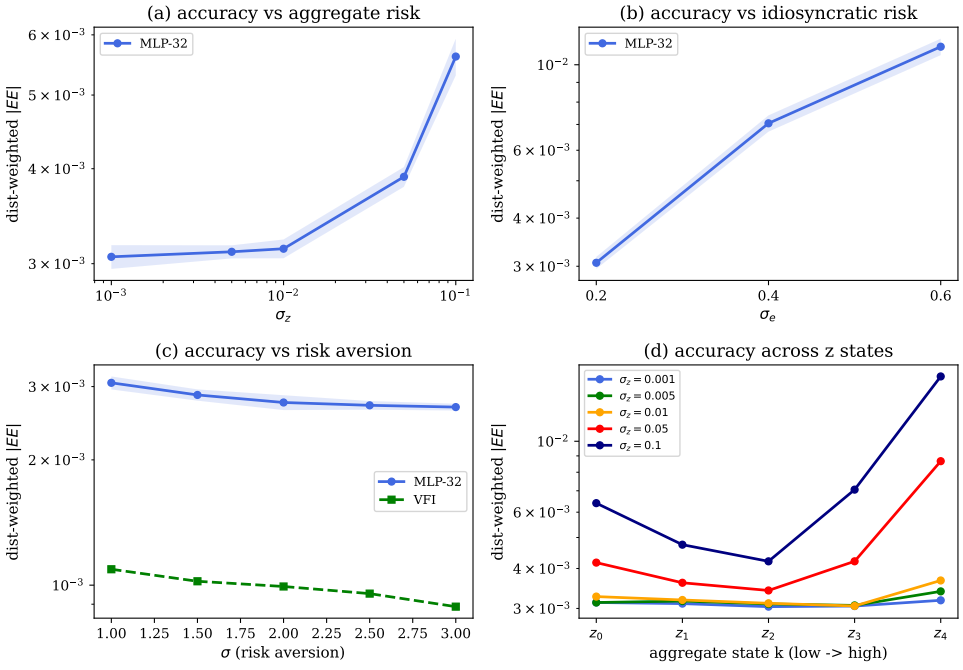


Figure 21: Risk-dependent Euler diagnostics, including state profiles and the corresponding VFI references.

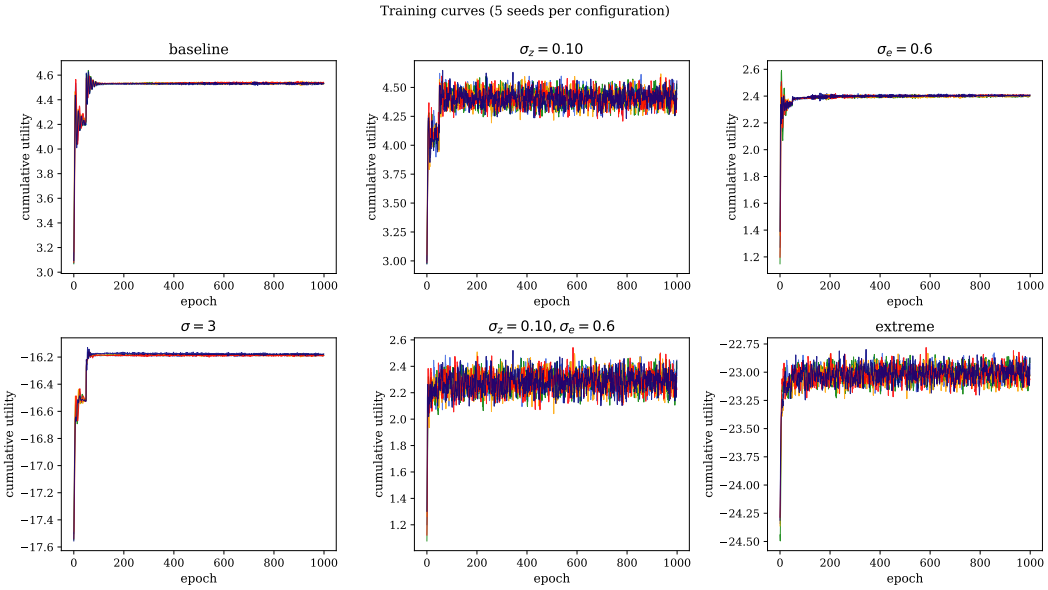


Figure 22: Training curves for representative risk configurations, with five seeds each, including high-risk cases.

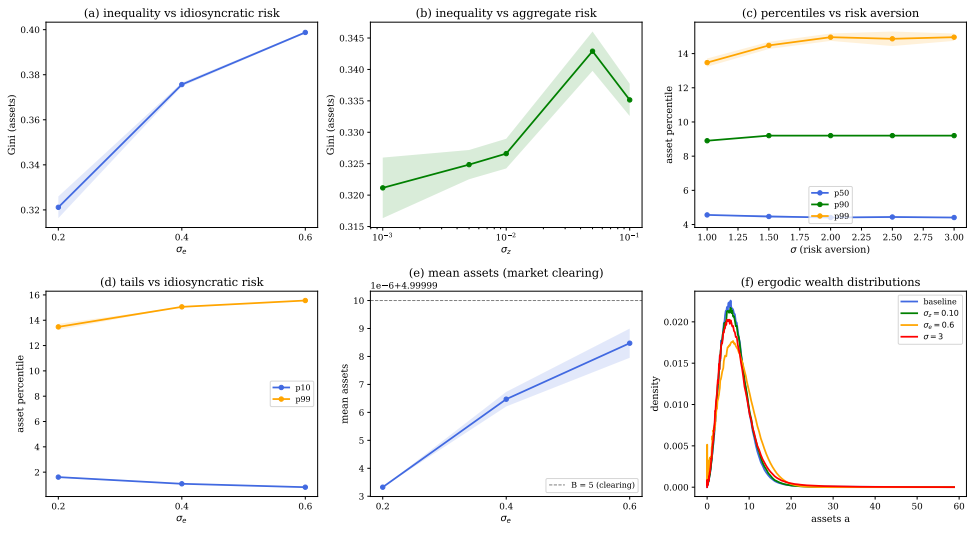


Figure 23: Wealth-distribution results across the risk grid and utility curvature. The discussion in subsection 5.4 focuses on the idiosyncratic-risk comparison under log utility and low aggregate risk, where fixed mean assets coexist with changing inequality and asset quantiles.

Table 10: Cross-seed variation of headline statistics across the 75 configurations: median and maximum of the within-configuration seed standard deviation.

statistic	median seed-std	max seed-std
r_mean	5.54×10^{-5}	4.64×10^{-4}
r_std	4.71×10^{-5}	2.06×10^{-4}
ce	2.28×10^{-4}	1.79×10^{-3}
gini	6.97×10^{-4}	5.33×10^{-3}
ee_wmean_abs	1.64×10^{-4}	6.97×10^{-4}
last50_avg_util	3.70×10^{-3}	1.73×10^{-2}

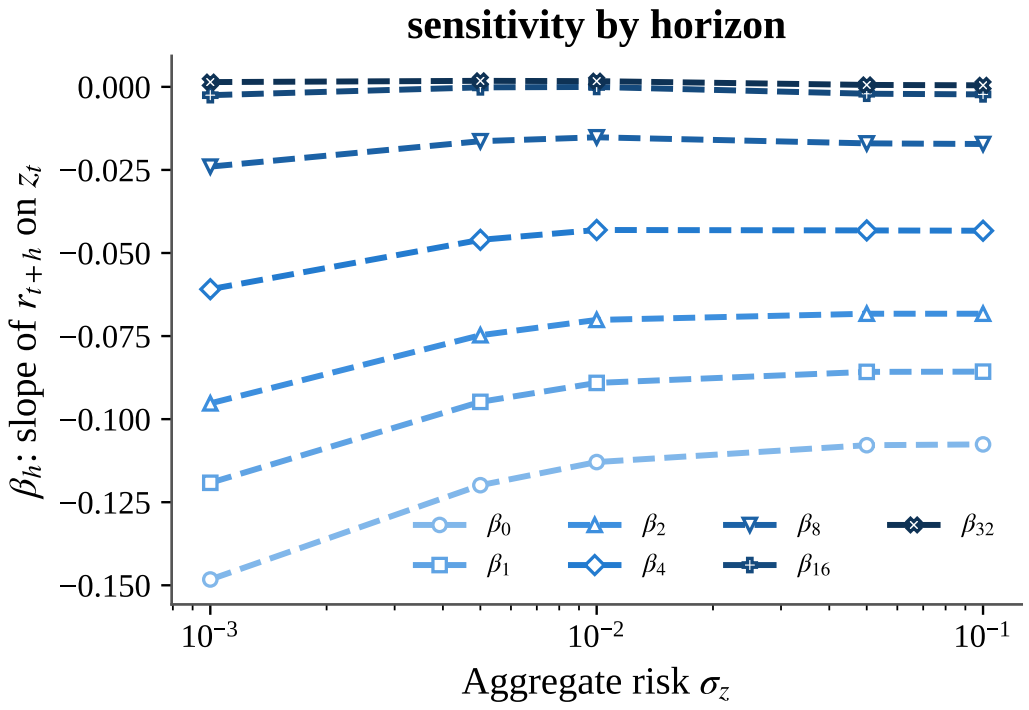


Figure 24: Horizon sensitivity of the return's response: each curve is the local-projection slope β_h of r_{t+h} on z_t at one horizon, against the aggregate-risk volatility σ_z.

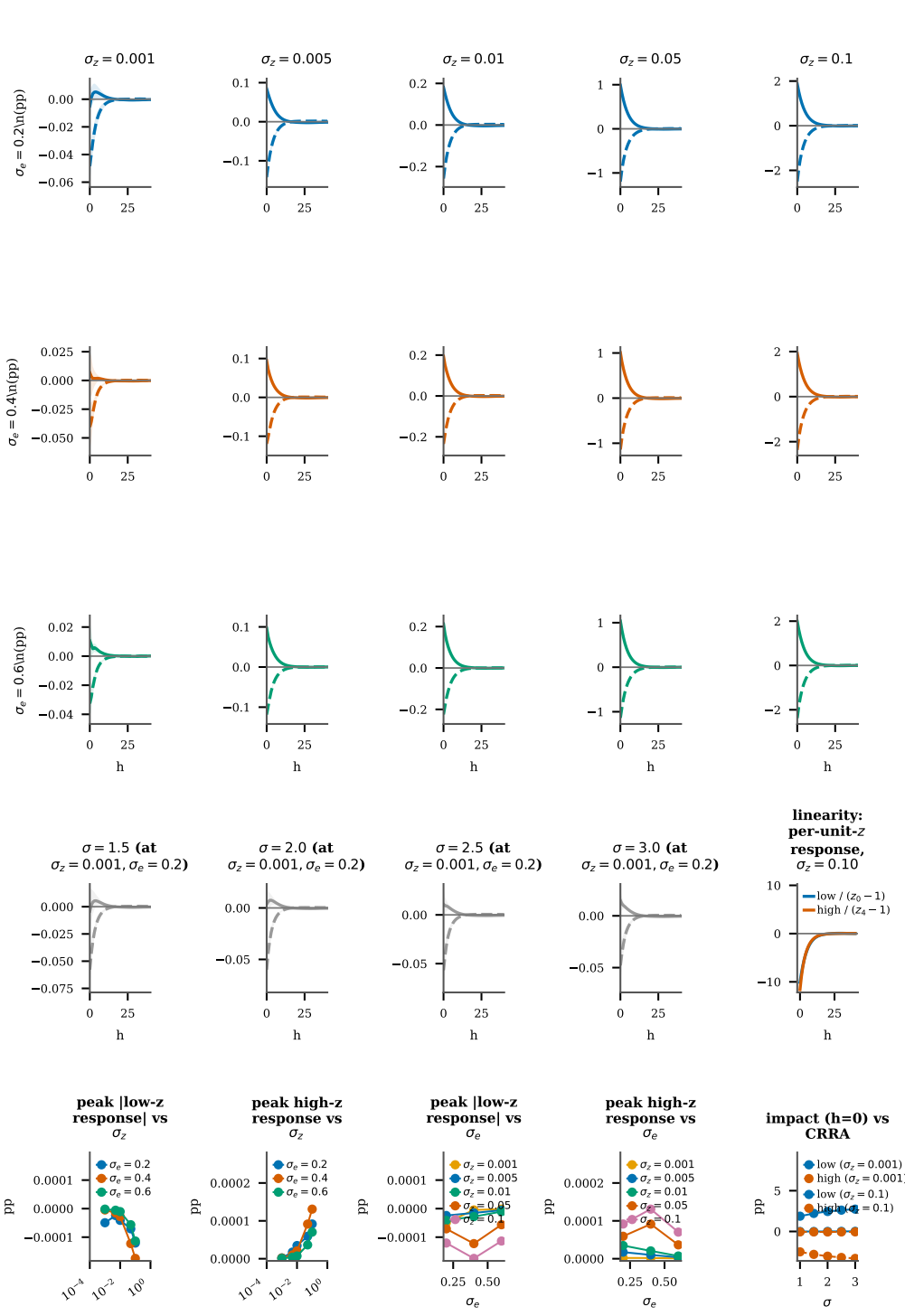


Figure 25: The full impulse-response library, twenty-five panels: fifteen over the (σ_z, σ_e) grid, each showing both conditional responses of the return (low state solid, high state dashed, seed mean with cross-seed standard deviation), four for the curvature variations, and six structural panels covering peak scaling along each risk dimension and a linearity check against the local projection.

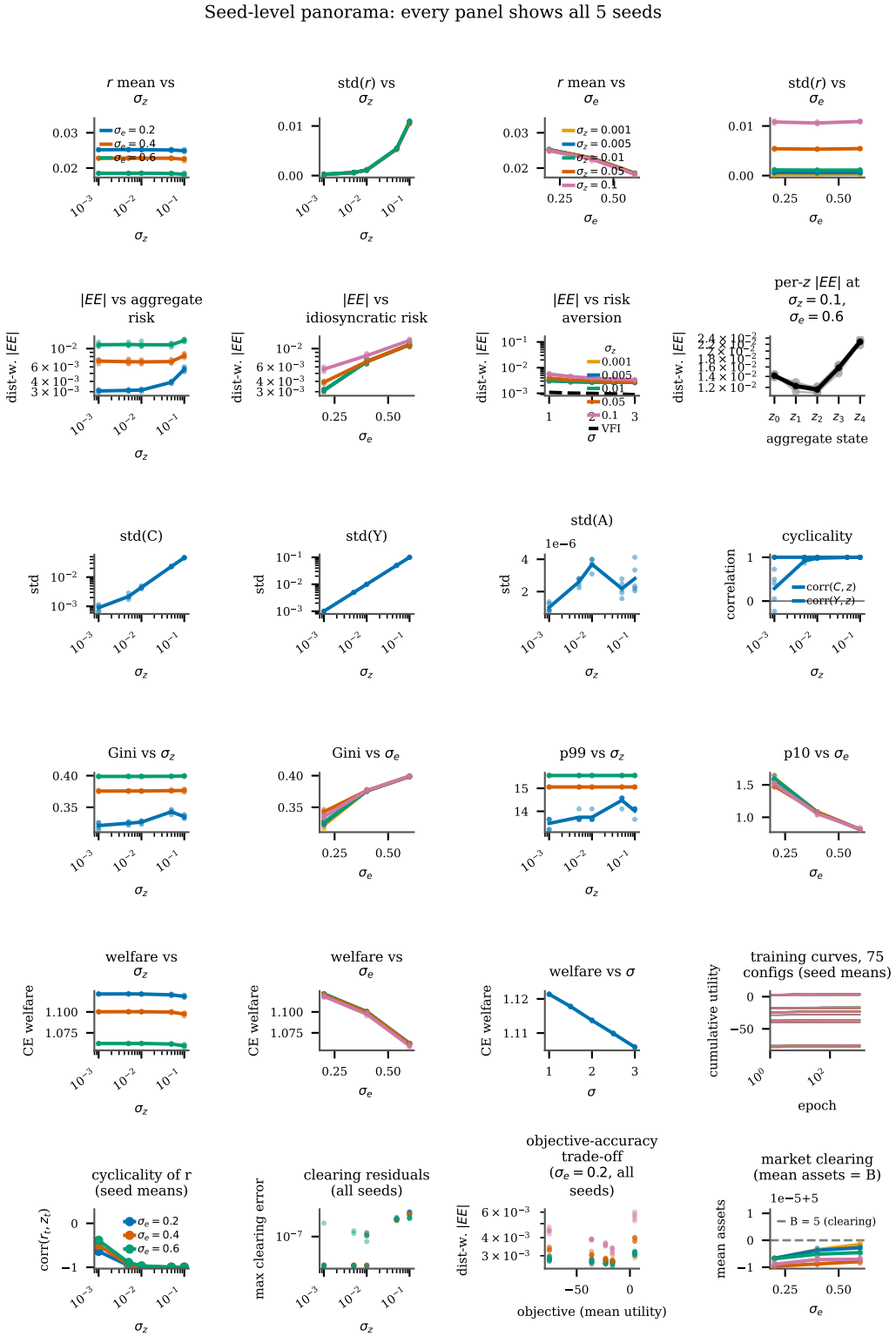


Figure 26: Seed-level panorama: twenty-four panels, each showing all five seeds.