Deep Learning Methods for Solving and Estimating Structural Corporate Finance Models

Yat Fan Lau

January 19, 2026

Abstract

This report applies deep learning methods to the solution and estimation of dynamic corporate finance models. We employ a neural network-based solver in TensorFlow for a discrete-time investment model with AR(1) shocks, leveraging a Monte Carlo stochastic gradient framework with an All-in-One(AiO) expectation operator. The methodology is then extended to a risky-debt model with endogenous default and bond pricing, where risky debt prices are determined by imposing the zero-profit condition. We also implement and compare different estimation techniques including the Generalized Method of Moments (GMM), the Simulated Method of Moments (SMM), and a Bayesian alternative using Hamiltonian Monte Carlo (HMC) within TensorFlow Probability.

Contents

1 Part 1: Applying Deep Learning to Solve Structural Corporate Finance Model

1.1 Introduction and Literature Review

Dynamic corporate finance models provide a rigorous framework for understanding how firms jointly make investment and financing decisions over time with the goal of maximizing shareholder value. Importantly, dynamic models are particularly well-suited to capturing the inherently forward-looking nature of firm behavior. Their ability to produce both time-series and cross-sectional implications enhances their empirical relevance and allows for more robust evaluation against real-world data than static models. This makes them powerful tools for developing and testing economic theories of corporate decision-making. A foundational approach within this topic is the discrete-time dynamic investment model, in which firms optimally choose next-period capital to maximize the present value of cash flows, accounting for adjustment frictions. This framework, rooted in early contributions by Lucas and Prescott [1, 2], has since been extended and refined in various directions. In this report, we primarily follow the formulation in section 3 of a review by Strebulaev and Whited [3], who outline three core components of such models: **exogenous stochastic state variables**, an **objective function**, and a set of **endogenous state variables** governed by control decisions.

The exogenous state variables typically capture random shocks to firm fundamentals, such as changes in demand or productivity. The objective function often reflects the goal of maximizing shareholder value, quantified as the expected discounted stream of net cash flows. Endogenous state variables include the firm's current capital stock, and may also encompass labor, cash holdings, or debt levels. These evolve dynamically based on the firm's choices regarding future investment, employment, liquidity, and leverage.

Although solving these models analytically is often infeasible due to their complexity, advances in numerical methods have made it possible to approximate solutions and generate rich quantitative predictions. Researchers have developed a rich toolkit of methods for solving this kind of models, including traditional methods and machine-learning methods. In the following text we first review and outline some common traditional approaches to solve this kind of models, followed by the more modern machine learning based methods(such as the deep learning approach) which will be our main focus of this report.

A natural starting point is the **Bellman-operator dynamic programming**. It solves the Bellman equation on discretized grids for the state vector and for the shock process. For example, value function iteration (VFI) approximates the stochastic process with a finite-state Markov chain [4, 5] and iterates on the value function by maximizing over feasible controls at each grid point, delivering policy functions and simulated dynamics. Variants such as Howard's policy improvement and Modified Policy Iteration accelerate convergence by alternating policy evaluation and policy improvement [6, 7, 8, 9] . These methods are robust to non-smooth payoffs and occasionally binding constraints(OBCs). Their main limitation is the curse of dimensionality, which makes dense grids costly as the number of state variables grows.

Next approach is **Euler-Equation Time Iteration (Coleman operator)**, including the **Endogenous Grid Method** (EGM). Time iteration solves for decision rules directly by enforcing the Euler and envelope/Kuhn–Tucker conditions rather than iterating on the value function. EGM constructs "endogenous" grids of next-period states and maps them back to today's states using the Euler equation, eliminating costly root-finding and yielding large speed gains in continuous-state accumulation problems [10, 11, 12]. For many dynamic models, time iteration/EGM is especially effective in the smooth, concave core and has generalized variants that accommodate non-concavities and occasionally binding constraints via piecewise regimes or com-

plementarity conditions [13, 14, 15]. Strengths include speed and accuracy for continuous problems; limitations are that standard EGM assumes invertibility/monotonicity of First-Order Conditions(FOCs) and smoothness—so with kinks or OBCs one must adopt generalized EGM or hybridize with grid or complementarity formulations.

Furthermore, **global approximation via projection or collocation methods**, including sparse-grid approaches like the Smolyak construction. Projection methods approximate unknown value or policy functions with global basis functions (e.g., Chebyshev polynomials) and choose coefficients to minimize the Euler/Bellman errors at collocation nodes [16, 17, 18]. They deliver high global accuracy for smooth problems and scale more gracefully than dense grids when state dimension rises—especially with Smolyak sparse grids that drastically reduce nodes in multi-dimensional models such as cash-and-debt or risky-debt environments [19]. Their strengths are accuracy and better scalability; limitations are sensitivity to non-smoothness (basis approximations can struggle with kinks/OBCs, requiring piecewise bases or local refinement) and the need for careful error diagnostics to ensure reliability near boundaries and constraint switches.

Despite the fact that the above projection- and Euler-equation–based workflows are precise in small dimensions, but they become brittle and slow as states and shocks increase, require hand-crafted derivatives, and struggle with kinks/OBCs. Fortunately, researchers recently have developed deep learning method for solving the models of interest[20]. Maliar *et al.* develop a unified DL framework for solving high-dimensional dynamic economic models. Their approach transforms lifetime reward, Euler, or Bellman equations into nonlinear regression objectives estimated on simulated data via stochastic gradient descent. A key ingredient is an *all-in-one* (AiO) operator that merges nested expectations and thereby makes training tractable, even with many shocks.

We bring these strands together: we shall first implement the basic discrete-time and also risky-debt investment models with an AR(1) shock and solve it using the Euler and Bellman residual minimization respectively. We prefer DL to traditional projection/VFI/Euler-grid methods because it scales to high-dimensional state and shock spaces without tensor grids, needs no hand-coded derivatives thanks to automatic differentiation, collapses nested expectations into a single simulation step, and reuses the same modular training across different model variants, delivering competitive quantitative accuracy with far less brittleness and engineering overhead. Then, we move to the estimations of the structural model using GMM, SMM and HMC methods and compare the three methods.

For the basic investment model of section 3.1 in Ref[3], we employ the Euler residual minimization while in the risky-debt model in section 3.6, we choose the Bellman residual minimization. In the basic model, the firm solves a smooth dynamic optimization problem with interior choices and no OBCs. In this setting, the Euler equations fully characterize the optimum and provide a low-dimensional set of conditions that are convenient to implement. By contrast, the risky-debt model in Section 3.6 features default, limited liability, and regime changes between solvent and default states, so the value function is kinked and the optimal policy can involve corners. In such environments Euler conditions become numerically fragile, because they mix expectations over discontinuous objects, while the Bellman equation remains valid and naturally encodes the max operator over "continue" versus "default" decisions. Using a Bellman-residual objective for the risky-debt case therefore makes it easier to incorporate default, recovery etc. and to let the neural network learn both the value function and policy in the presence of non-smoothness, whereas for the basic model the simpler Euler-equation approach is more efficient.

1.2 Basic Investment Model without Risky-Debt

1.2.1 Mathematical formulation

We consider the basic discrete-time infinite horizon investment problem described in Section 3.1 of Ref[3]. The state comprises beginning-of-period capital $k_t > 0$ and an external shock $z_t > 0$ and we assume the shock follows an AR(1) in logs:

$$\ln z_{t+1} = \rho \ln z_t + \sigma \varepsilon_{t+1}, \quad \varepsilon_{t+1} \sim \mathcal{N}(0, \sigma_\varepsilon^2), \tag{1}$$

with $|\rho| < 1$ and $\sigma > 0$.

The parameter ρ measures the persistence of productivity shocks. If $\rho = 0$, a positive shock today may disappear by tomorrow — the shock is short-lived.

In such cases, firms are unlikely to make large-scale investments just for a temporary spike in profits, because they know the good times won't last.

In contrast, if ρ is high (i.e., the shock is persistent), firms believe that today's high productivity will likely continue into the future. In this case, investing makes sense, because the additional capital can generate high returns over many future periods. As a result, firms respond more strongly to shocks.

This is something unique to dynamic models: static models (which only consider a single period) cannot capture the concept of "persistence" because there is no "future." Only dynamic models can reveal that a firm's response to current shocks fundamentally depends on its expectations about the future.

The firm produces profits $\pi(k_t, z_t)$ and invests I_t with convex adjustment costs $\psi(I_t, k_t)$. The one-period cash flow to shareholders is

$$e(k_t, I_t, z_t) = \pi(k_t, z_t) - \psi(I_t, k_t) - I_t. \tag{2}$$

Before we proceed, we first provide an intuitive understanding of the above equation. Here k represents how much productive capacity a firm currently has (e.g., the size of machines, stores, or servers). This capacity can be increased through investment (e.g., buying new equipment), but it also depreciates over time at a rate δ with the capital stock accounting identity given by

$$k_{t+1} = (1 - \delta)k_t + I_t, \qquad \delta \in (0, 1), \tag{3}$$

On the other hand, z represent today's "business weather"—that is, how favorable or unfavorable demand, efficiency, and the overall economy are. It fluctuates according to some stochastic process (e.g. AR(1) as in our case).

When z is high and expected to remain high, the return on each unit of k is also expected to be higher, which encourages more investment and leads to a higher future capital stock k'. Conversely, when z is low, firms tend to invest less.

However, adjusting k too quickly incurs costs(adjustment costs). Thus, firms weigh today's cost against the expected future value $E[V(k', z')]$ and choose the optimal k' accordingly. This is essentially the central idea of the task we are going to solve in the following.

The value of the firm at time t can be given by the expected discounted sum of the cash flow over time

$$V_t = \max_{\{k_{t+j}\}_{j=1}^{\infty}} \mathbb{E}_t \left[\sum_{j=0}^{\infty} \left(\frac{1}{1+r} \right)^j e\left(k_{t+j}, I_{t+j}, z_{t+j}\right) \right] \tag{4}$$

where $\mathbb{E}_t$ denotes the expectation conditional on information known at time t, and r is the constant risk-free interest rate. Thus, the above problem of finding an optimal set of $\{k_{t+j}\}_{j=1}^{\infty}$ such that

V_t attains its maximum under the constraint of Eqn.(3) is the **lifetime reward** formulation. As is mentioned in Ref[3], this model has no analytical solution. Thus, it would be helpful to recast the model in the Euler or the Bellman formulation in our DL implementation later.

The Euler formulation

To find the extrema of a function under certain constraints, we can introduce Lagrange multipliers χ_{t+j}'s to construct the Lagrangian

$$\mathcal{L} = \mathbb{E}_t \left[\sum_{j=0}^{\infty} \left(\frac{1}{1+r} \right)^j \left(e\left(k_{t+j}, I_{t+j}, z_{t+j}\right) - \chi_{t+j} \left(k_{t+j+1} - k_{t+j}(1-\delta) - I_{t+j} \right) \right) \right] \tag{5}$$

The first-order condition thus can be obtain by taking the derivative of $\mathcal{L}$ w.r.t. I_{t+j}

$$\frac{\partial \mathcal{L}}{\partial I_{t+j}} = 0 \quad \Rightarrow \quad 1 + \frac{\partial \psi}{\partial I_{t+j}} = \chi_{t+j} \tag{6}$$

We note that some of the equations in [3] contains typos in the indexing.
Next

$$\frac{\partial \mathcal{L}}{\partial k_{t+j+1}} = 0 \tag{7}$$

$$\mathbb{E}_t \left[-\chi_{t+j} (\frac{1}{1+r})^j + (\frac{1}{1+r})^{j+1} \left(\frac{\partial \pi}{\partial k_{t+j+1}} - \frac{\partial \psi}{\partial k_{t+j+1}} + (1-\delta)\chi_{t+j+1} \right) \right] = 0 \tag{8}$$

$$\mathbb{E}_t \left[(\frac{1}{1+r}) \left(\frac{\partial \pi}{\partial k_{t+j+1}} - \frac{\partial \psi}{\partial k_{t+j+1}} + (1-\delta)\chi_{t+j+1} \right) \right] = \chi_{t+j} \tag{9}$$

Let $\beta = \frac{1}{1+r}$ and define

$$A_t \coloneqq \frac{\partial \pi}{\partial k_t} - \frac{\partial \psi}{\partial k_t}.$$

The Euler equation is obtained by using Eq.(6)

$$\begin{aligned} \chi_{t+j} &= \beta\, \mathbb{E}_t \left[A_{t+j+1} + (1-\delta)\chi_{t+j+1} \right] \\ 1 + \frac{\partial \psi}{\partial I_{t+j}} &= \beta\, \mathbb{E}_t \left[\frac{\partial \pi}{\partial k_{t+j+1}} - \frac{\partial \psi}{\partial k_{t+j+1}} + (1-\delta) \left(1 + \frac{\partial \psi}{\partial I_{t+j+1}} \right) \right] \end{aligned} \tag{10}$$

Forward-iterating S times gives

$$\chi_{t+j} = \mathbb{E}_t \left[\sum_{s=1}^{S} \beta^s (1-\delta)^{s-1} A_{t+j+s} + \beta^S (1-\delta)^{S-1} \chi_{t+j+S} \right].$$

Under the usual transversality/no-bubbles condition

$$\lim_{S \to \infty} \mathbb{E}_t \left[\beta^S (1-\delta)^{S-1} \chi_{t+j+S} \right] = 0,$$

we obtain the closed-form solution with no χ on the right-hand side:

$$\chi_{t+j} = \mathbb{E}_t \left[\sum_{s=1}^{\infty} \beta^s (1-\delta)^{s-1} \left(\frac{\partial \pi}{\partial k_{t+j+s}} - \frac{\partial \psi}{\partial k_{t+j+s}} \right) \right].$$

Split the sum into the $j = 0$ term and the rest:

$$V_t = \max \left\{ e(k_t, I_t, z_t) + \beta \, \mathbb{E}_t \left[\sum_{j=1}^{\infty} \beta^{j-1} e(k_{t+j}, I_{t+j}, z_{t+j}) \right] \right\}.$$

Recognize the continuation value:

$$\sum_{j=1}^{\infty} \beta^{j-1} e(\cdot) = V_{t+1},$$

so

$$V_t = \max \left\{ e(k_t, I_t, z_t) + \beta \, \mathbb{E}_t \left[V_{t+1} \right] \right\}.$$

Imposing the time stationarity and Markov property, we obtain

$$V(k, z) = \max_{k'} \left\{ \pi(k, z) - \psi \left(k' - (1-\delta)k, k \right) - \left[k' - (1-\delta)k \right] + \frac{1}{1+r} \int V(k', z') \, dg(z' \mid z) \right\} \tag{11}$$

where

$$\mathbb{E}_t \left[V(k_{t+1}, z_{t+1}) \right] = \int V(k', z') \, dg(z' \mid z).$$

This is known as the **Bellman equation**. Maximizing over k' means the firm is choosing next period's capital stock k'(or investment I) as its control and the arg max over k' defines the policy function $k' = \varphi(k, z)$: for each current state (k, z), choose the investment level that maximizes the firm's value (subject to any feasibility constraints like $k' \geq 0$).

The objective is to maximize the expected present value of cash flows,

$$V(k, z) = \max_{\{k' \geq 0\}} \left\{ \pi(k, z) - \psi(k' - (1-\delta)k, k) - \left(k' - (1-\delta)k \right) + \beta \mathbb{E} \left[V(k', z') \,\middle|\, z \right] \right\}, \tag{12}$$

with $\beta = \frac{1}{1+r} \in (0, 1)$ and we use prime to denote next-period quantities. We note the equivalence between Eq. (11) and Eq. (4), namely that Eq. (11) is the Bellman (recursive) formulation of the sequential problem in Eq. (4). In Eq. (4), the firm chooses the entire future path $\{k_{t+1}, k_{t+2}, \ldots\}$ to maximize expected discounted cash flows, whereas in Eq. (11) it chooses only next period's capital $k' \equiv k_{t+1}$ (equivalently I_t via $k_{t+1} = (1-\delta)k_t + I_t$), and the continuation value $V(k', z')$ embeds all future optimal choices.

By Bellman's principle of optimality—and because k_t is predetermined—maximizing over k' at time t delivers the same policy and value as maximizing over the whole sequence in Eq. (4).

To implement the model concretely, we first need to choose the specific expressions for the production and cost functions:

Functional forms In line with Strebulaev *et al.*[3], we use

$$\pi(k, z) = z k^{\theta}, \qquad \theta \in (0, 1), \tag{13}$$

$$\psi(I, k) = \frac{\phi}{2} \frac{(I - k\delta)^2}{k} = \frac{\phi}{2} k (\iota - \delta)^2, \qquad \phi > 0. \tag{14}$$

where we define $\iota = I/k$. These imply

$$\pi_k(k, z) = \theta z k^{\theta - 1}, \qquad \psi_I(I, k) = \phi \frac{I - k\delta}{k} = \phi(\iota - \delta), \qquad \psi_k(I, k) = \frac{\phi}{2}(\delta^2 - \iota^2). \tag{15}$$

Before we proceed, we give intuitive meaning of the above functions: the parameter θ measures the rate at which returns to capital diminish. A smaller θ implies that the production function is more concave, meaning that the marginal return from additional capital declines more rapidly.

As a result, when productivity z changes, its impact on the marginal return to capital is smaller. Firms perceive that "investing a bit more doesn't bring much additional benefit," so their response to shocks is more muted, and investment behavior becomes more stable.

1.2.2 Deep-learning method and algorithm

In this section we describe how we implement the deep-learning (DL) Euler-equation solver of Maliar *et al.* (2021) [20] for the stochastic investment model of Strebulaev *et al.* (2012) [3]. The key ingredients are: (i) the training distribution over states, combining exogenous coverage and on-policy (ergodic) sampling; (ii) an Euler-residual-based All-in-One (AiO) loss with antithetic shocks; and (iii) a neural-network policy for investment that is trained by stochastic gradient descent.

Notation

- State vector:
$$s_t = (k_t, z_t),$$
where k_t is the capital stock and z_t is the productivity shock.

- Shock process:
$$\ln z_{t+1} = \rho \ln z_t + \varepsilon_{t+1}, \qquad \varepsilon_{t+1} \sim \mathcal{N}(0, \sigma_\varepsilon^2).$$
In the stationary distribution, $\ln z_t \sim \mathcal{N}(m_{\ln z}, \sigma_{\ln z}^2)$ with $\sigma_{\ln z}^2 = \sigma_\varepsilon^2/(1-\rho^2)$; in our implementation we set $m_{\ln z}$ so that $\mathbb{E}[z_t] = 1$.

- Policy network (capital accumulation rule): we parameterize the *investment rate* $\iota_t = \iota(k_t, z_t; w)$ and define next-period capital as
$$k_{t+1} \;=\; \varphi_w(k_t, z_t) \;\equiv\; \big(1 - \delta + \iota(k_t, z_t; w)\big)\, k_t.$$
In what follows we write $\varphi_w(\cdot)$ or $\iota(\cdot; w)$ interchangeably for the policy.

Training distribution and state sampling Following Maliar *et al.* (2021), we combine (i) *exogenous coverage sampling* from a fixed domain to guarantee broad coverage of the relevant state space, and (ii) *on-policy (ergodic) sampling* from the stationary distribution generated by simulating the model under the current policy. Although our baseline model has only two state variables, this hybrid scheme keeps accuracy high on the ergodic set and scales naturally to higher-dimensional problems where exogenous grids become impractical.

(i) **Exogenous coverage sampling (Monte-Carlo grid).**

We start from the frictionless steady state with $\mathbb{E}[z] = 1$ and no adjustment costs. Setting $z = z^* = 1$, the Euler equation in Strebulaev *et al.* (2012, Eq. (3.12)) reduces to
$$\theta\, k^{\theta-1} = r + \delta,$$
which implies the steady-state capital level
$$k^* = \left(\tfrac{\theta}{r+\delta}\right)^{1/(1-\theta)}.$$

We choose two constants $m_- < 1 < m_+$ (e.g. $m_- = 0.2$, $m_+ = 5$) and define

$$k_{\min} = m_- k^*, \qquad k_{\max} = m_+ k^*.$$

To generate exogenous draws of k we sample

$$\ln k \sim \text{Uniform}\big(\ln k_{\min}, \ln k_{\max}\big), \qquad k = e^{\ln k}.$$

For the productivity shock we draw from the stationary distribution of $\ln z$. Let

$$\sigma^2_{\ln z} = \frac{\sigma^2_\varepsilon}{1-\rho^2}, \qquad m_{\ln z} = \mathbb{E}[\ln z_t],$$

where in the implementation $m_{\ln z}$ is chosen so that $\mathbb{E}[z_t] = 1$. We then sample

$$\ln z \sim \mathcal{N}\big(m_{\ln z}, \sigma^2_{\ln z}\big), \qquad z = e^{\ln z},$$

and truncate $\ln z$ at $m_{\ln z} \pm 3\sigma_{\ln z}$ to avoid extreme tails. This yields a Monte-Carlo "coverage grid" surrounding the deterministic steady state.

(ii) **On-policy ergodic sampling with replay buffer.**

In addition to exogenous coverage draws, we simulate the model under the *current* policy φ_w and store visited states in a replay buffer. After a short burn-in, this buffer approximates the ergodic distribution induced by φ_w, so that training focuses on the region of the state space that is actually visited in equilibrium.

Algorithm (On-policy simulation with replay buffer).

M1. **Initialisation.**

- Choose initial states, e.g. $k_0 = k^*$, $z_0 = 1$, possibly replicated across many parallel paths.
- Create an empty replay buffer $\mathcal{B} = \varnothing$ with maximum size N_{buf}.

M2. **Roll-out under the current policy.** For n_{roll} consecutive periods and for each parallel path:

store the current state $s_t = (k_t, z_t)$ in $\mathcal{B}$(discard the oldest element if $|\mathcal{B}| > N_{\text{buf}}$);

$\varepsilon_{t+1} \sim \mathcal{N}(0, \sigma^2_\varepsilon)$, $\ln z_{t+1} = \rho \ln z_t + \varepsilon_{t+1}$, $z_{t+1} = e^{\ln z_{t+1}}$;

$k_{t+1} = \varphi_w(k_t, z_t)$ (network output as defined above).

M3. **Sampling for stochastic gradient.** Draw a mini-batch $\{s^{(i)}\}_{i=1}^B$ *uniformly at random* from the buffer $\mathcal{B}$.

M4. **Parameter update.** Compute a suitable loss, for example

$$L(w) = \frac{1}{B}\sum_{i=1}^{B} \xi\big(s^{(i)}; w\big),$$

where ξ denotes the chosen Euler-equation residual evaluated at $s^{(i)}$. Update the network parameters using stochastic gradient descent (SGD), e.g.

$$w \;\leftarrow\; w - \eta_t \nabla_w L(w) \qquad \text{(Adam or another SGD rule)}.$$

M5. **Iterate.** Return to step M2 with the updated parameters w. Data generation (M2) and learning (M3–M4) alternate until the Euler residuals have converged.

Remarks

- After burn-in, the replay buffer contains draws from the stationary distribution induced by the current policy; the network is therefore trained on states that are most relevant for the optimal solution.
- Pure coverage sampling is convenient for debugging and for the early iterations of training. Once the network stabilises, ergodic sampling typically yields faster and more accurate convergence, while retaining a small share of coverage draws helps maintain exploration.

In our implementation, we follow this principle by first *pretraining* the network on pure coverage samples around k^* and from the stationary distribution of z, and then switching to hybrid mini-batches that are drawn primarily from the replay buffer while keeping a small, gradually shrinking fraction of coverage states.

Euler residual, AiO loss and antithetic shocks For a given state (k, z) and a draw ε for the shock to z', the (one-step) Euler residual under policy $\varphi(\cdot; w)$ is defined as

$$\mathcal{R}\big(k, z, \varepsilon; w\big) \;\equiv\; \frac{1}{1+r}\Big(\pi_k(k', z') \;-\; \psi_k(I', k') \;+\; (1-\delta)\big(1+\psi_I(I', k')\big)\Big) \;-\; \big(1+\psi_I(I, k)\big), \quad (16)$$

where $k' = \varphi(k, z; w)$, $z' = \exp\{\rho \ln z + \varepsilon\}$, and $I = k' - (1-\delta)k$, $I' = k'' - (1-\delta)k'$ are investment levels as defined in the model. At the exact solution w^*, the conditional expectation satisfies

$$\mathbb{E}\big[\mathcal{R}(k, z, \varepsilon; w^*) \,\big|\, k, z\big] = 0 \qquad \text{for all relevant } (k, z).$$

A natural training objective is the squared conditional Euler error averaged over the state distribution,

$$L(w) \;\equiv\; \mathbb{E}_{(k,z)}\Big[\big(\mathbb{E}_\varepsilon\big[\mathcal{R}\big(k, z, \varepsilon; w\big)\big]\big)^2\Big]. \quad (17)$$

Computing (17) by direct Monte Carlo requires, for each state (k, z), an inner simulation over many shocks ε to approximate $\mathbb{E}_\varepsilon[\mathcal{R}]$, followed by squaring and averaging over states. If n_s states and n_ε shocks per state are used, each gradient step costs $O(n_s n_\varepsilon)$ model evaluations.

To avoid nested simulation, Maliar *et al.* (2021) propose the *All-in-One (AiO)* formulation. For each fixed state (k, z), take two independent draws $\varepsilon_1, \varepsilon_2$ and observe that

$$\big(\mathbb{E}_\varepsilon[\mathcal{R}]\big)^2 \;=\; \mathbb{E}_{\varepsilon_1}\big[\mathcal{R}(k, z, \varepsilon_1; w)\big]\; \mathbb{E}_{\varepsilon_2}\big[\mathcal{R}(k, z, \varepsilon_2; w)\big] \;=\; \mathbb{E}_{\varepsilon_1, \varepsilon_2}\big[\,\mathcal{R}(k, z, \varepsilon_1; w)\, \mathcal{R}(k, z, \varepsilon_2; w)\,\big].$$

Hence the theoretical loss (17) can be written as a *single* expectation over the composite random vector $\omega = (k, z, \varepsilon_1, \varepsilon_2)$:

$$L(w) \;=\; \mathbb{E}_{(k,z),\,\varepsilon_1,\,\varepsilon_2}\big[\,\mathcal{R}\big(k, z, \varepsilon_1; w\big)\; \mathcal{R}\big(k, z, \varepsilon_2; w\big)\,\big]. \quad (18)$$

Equation (18) is the *AiO Euler loss*: it replaces the square of an inner expectation by the product of two independent one-step residuals, enabling unbiased stochastic gradients with only two shock draws per state.

Given a mini-batch of states $\{(k_i, z_i)\}_{i=1}^N$ and, for each i, two independent shocks $\varepsilon_{i,1}, \varepsilon_{i,2}$, the empirical AiO loss is

$$\widehat{L}_N(w) \;\equiv\; \frac{1}{N}\sum_{i=1}^{N}\Big[\mathcal{R}\big(k_i, z_i, \varepsilon_{i,1}; w\big)\; \mathcal{R}\big(k_i, z_i, \varepsilon_{i,2}; w\big)\Big], \quad (19)$$

where each $\mathcal{R}(\cdot)$ is computed as in (16).

What to do given a batch of states (k, z). For each (k_i, z_i) in the batch we proceed as follows:

1. Compute
$$k_i' = \varphi(k_i, z_i; w) = \big(1 - \delta + \iota(k_i, z_i; w)\big)k_i, \qquad I_i = k_i' - (1-\delta)k_i.$$

2. Draw two independent shocks $\varepsilon_{i,1}, \varepsilon_{i,2} \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ and set
$$z_{i,j}' = \exp\{\rho \ln z_i + \varepsilon_{i,j}\}, \qquad j = 1, 2.$$

3. For each $j \in \{1, 2\}$, propagate one more step under the policy,
$$k_{i,j}'' = \varphi(k_i', z_{i,j}'; w), \qquad I_{i,j}' = k_{i,j}'' - (1-\delta)k_i',$$
and compute the corresponding one-step residual
$$\mathcal{R}_{i,j} = \frac{1}{1+r}\Big(\pi_k(k_i', z_{i,j}') \;-\; \psi_k(I_{i,j}', k_i') \;+\; (1-\delta)\big(1 + \psi_I(I_{i,j}', k_i')\big)\Big) \;-\; \big(1 + \psi_I(I_i, k_i)\big).$$

4. Form the per-state AiO contribution $\mathcal{R}_{i,1}\,\mathcal{R}_{i,2}$ and average across i as in (19).

This yields an unbiased Monte-Carlo estimator of the theoretical loss (18) with only two shock draws per state, avoiding the cost of nested expectations while remaining faithful to the Euler condition.

Antithetic-variance reduction. In practice we further reduce Monte-Carlo variance using antithetic variates. For each state (k_t, z_t) we draw two independent innovations $\varepsilon_{t+1}^{(1)}, \varepsilon_{t+1}^{(2)}$, and for each branch $j \in \{1, 2\}$ we form an antithetic pair $\pm\varepsilon_{t+1}^{(j)}$. For each sign $s \in \{+, -\}$ we define
$$\mathcal{R}^{(j,s)}(k_t, z_t; w) \;\equiv\; \mathcal{R}\big(k_t, z_t, s\,\varepsilon_{t+1}^{(j)}; w\big), \qquad j = 1, 2,$$
and then average over the antithetic pair,
$$\bar{\mathcal{R}}^{(j)}(k_t, z_t; w) \;\equiv\; \tfrac{1}{2}\Big(\mathcal{R}^{(j,+)}(k_t, z_t; w) + \mathcal{R}^{(j,-)}(k_t, z_t; w)\Big), \qquad j = 1, 2.$$
The AiO loss is implemented as
$$L(w) = \mathbb{E}\Big[\bar{\mathcal{R}}^{(1)}(k_t, z_t; w)\,\bar{\mathcal{R}}^{(2)}(k_t, z_t; w)\Big],$$
where the outer expectation is taken over the joint distribution of $(k_t, z_t, \varepsilon_{t+1}^{(1)}, \varepsilon_{t+1}^{(2)})$. Minimizing $L(w)$ drives the conditional expectation $\mathbb{E}_\varepsilon[\mathcal{R}(k_t, z_t, \varepsilon; w)]$ towards zero for all states in the training support, enforcing the Euler condition in an L^2-sense.

Policy network and training algorithm We parameterize the investment-rate policy $\iota(k, z; w)$ by a two-layer feed-forward neural network with 64 hidden units per layer and tanh activations, as in the code:

- **Inputs.** The network takes the transformed state $(\ln k, \ln z)$ as input, which normalises the scale of k and z.
- **Architecture.** Two fully connected hidden layers of size 64, each with tanh activation, followed by a linear output layer producing a raw scalar.

- **Bounded output.** The raw output is mapped into the admissible interval $[\iota_{\min}, \iota_{\max}]$ via a scaled tanh transformation:

$$\iota(k, z; w) = \iota_{\min} + \tfrac{1}{2}\big(\tanh(\text{raw}) + 1\big)\big(\iota_{\max} - \iota_{\min}\big),$$

where we set $\iota_{\max} = 0.5$ and $\iota_{\min} = -0.99(1 - \delta)$. This guarantees

$$k_{t+1} = \big(1 - \delta + \iota_t\big)k_t \;\geq\; 0.01\, k_t,$$

and thus strictly positive capital along simulated paths.

Given the policy $\iota(\cdot; w)$, the capital-accumulation rule $\varphi_w(k_t, z_t)$ used in the Euler residual and in the on-policy simulation is

$$k_{t+1} = \varphi_w(k_t, z_t) = \big(1 - \delta + \iota(k_t, z_t; w)\big)k_t.$$

We train the network using the Adam optimizer with learning rate 10^{-4}. Mini-batches have size 4,096 and are constructed as follows:

- **Pretraining phase.** We begin with 1,000 pretraining steps using *only* coverage samples drawn as in item (i) above. This stabilises the network on a broad state domain around the deterministic steady state.
- **Hybrid training phase.** We then perform a main training phase with 50,000 steps. At training step t,

  1. choose a coverage share $\alpha_t \in [\alpha_{\text{final}}, 1]$, which decreases linearly from 1 to a small final value $\alpha_{\text{final}} \approx 0.1$;
  2. draw $\alpha_t B$ states from the coverage sampler and $(1 - \alpha_t)B$ states uniformly from the replay buffer $\mathcal{B}$;
  3. evaluate the AiO Euler loss with antithetic shocks as described above and compute its stochastic gradient with respect to w;
  4. update w via an Adam step.

  After every few gradient steps (every n_{roll} iterations in the code), we roll out the model one step under the current policy and append the resulting states to $\mathcal{B}$, thereby refreshing the on-policy sample.

This training scheme mixes exogenously sampled *coverage* states with *on-policy* states drawn from long simulations under the current policy $\iota(\cdot; w)$. Combined with the AiO Euler loss and antithetic shocks, it produces an accurate approximation to the optimal investment policy while remaining computationally efficient and scalable.

1.2.3 Results and Validity of the DL Method

In this section, we summarize our results and demonstrate that a single fully-connected neural network, trained by stochastic gradient descent on an all-in-one (AiO) Euler residual, is able to solve the model with quite high accuracy.

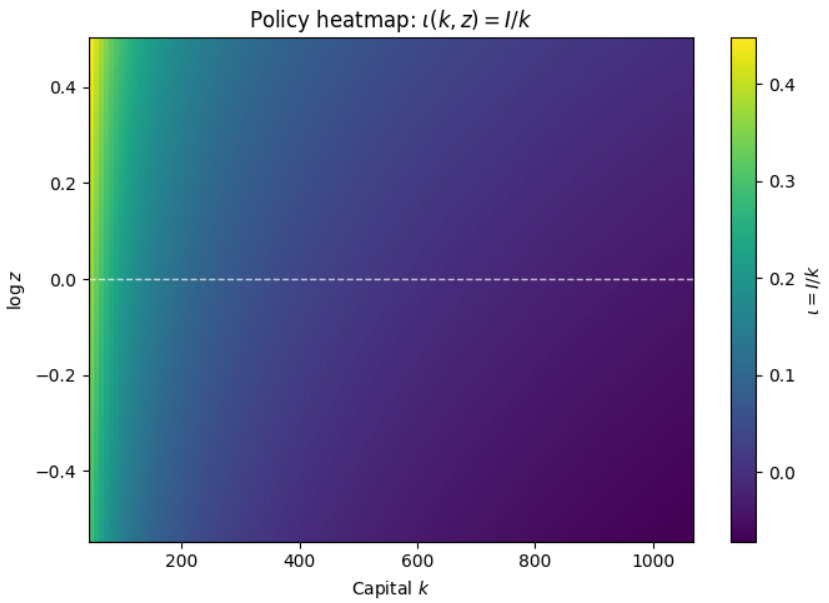

Figure 1:

Policy shape and deterministic dynamics The learned policy has the expected qualitative properties. Figure 1 plots a heatmap of $\iota(k, z)$ on a grid covering $k \in [0.2k^*, 5k^*]$ and the stationary 2.5-standard-deviation band of $\ln z_t$, where k^* is the deterministic steady state solving $\theta k^{\theta-1} = r+\delta$. For any fixed z, the investment rate is strictly decreasing in k, while for any fixed k it is increasing in z. Thus, high productivity and low capital jointly induce a high investment rate, and conversely.

Figure 2 investigates the deterministic transition dynamics when $z_t \equiv 1$. Starting from three initial conditions $k_0 \in \{0.2k^*, k^*, 5k^*\}$, the corresponding capital paths converge monotonically to a neighborhood of k^*, while the investment rate $\iota_t = I_t/k_t$ converges to the depreciation rate $\delta = 0.1$. These dynamics are exactly those implied by the analytical steady-state condition of the model and provide a qualitative check of the policy network.

Figure 3 illustrates the ergodic "cloud" of on-policy states $(\ln k_t, \ln z_t)$ together with the rectangular coverage box used in the off-policy accuracy tests below. The box comfortably contains the ergodic support, spanning approximately $\ln k \in [5.14, 5.62]$ and $\ln z \in [-0.56, 0.52]$.

Euler-equation accuracy and effectiveness measures To assess the quantitative accuracy of the solution we follow the practice in Maliar *et al.* [20] and compute high-precision Euler residuals on independent test sets. For any given state (k, z) we consider the *conditional* Euler residual

$$\bar{\mathcal{R}}(k, z; w) \;\equiv\; \mathbb{E}_\varepsilon\big[\mathcal{R}(k, z, \varepsilon; w) \mid k, z\big],$$

which is the conditional expectation appearing in the Euler equation formulation. We approximate this conditional expectation using n-node Gauss–Hermite (GH-n) quadrature for the Normal innovation $\varepsilon_{t+1} \sim \mathcal{N}(0, \sigma_\varepsilon^2)$. Denote by $\widehat{\mathbb{E}}_n[\cdot]$ the resulting quadrature operator and define

$$\widehat{\bar{\mathcal{R}}}_n(k, z; w) \;\equiv\; \widehat{\mathbb{E}}_n\big[\mathcal{R}(k, z, \varepsilon; w)\big] \approx \bar{\mathcal{R}}(k, z; w).$$

Our primary accuracy measures are:

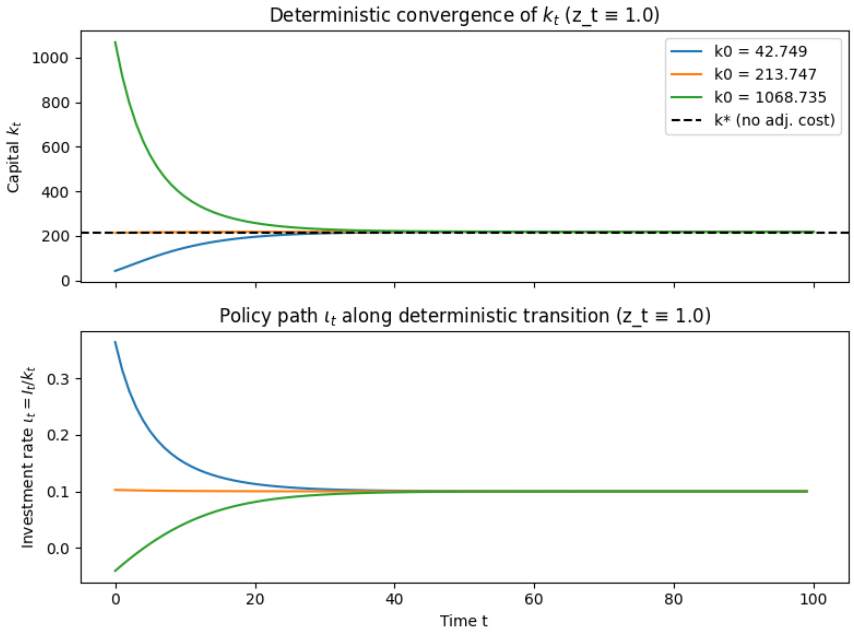

Figure 2:

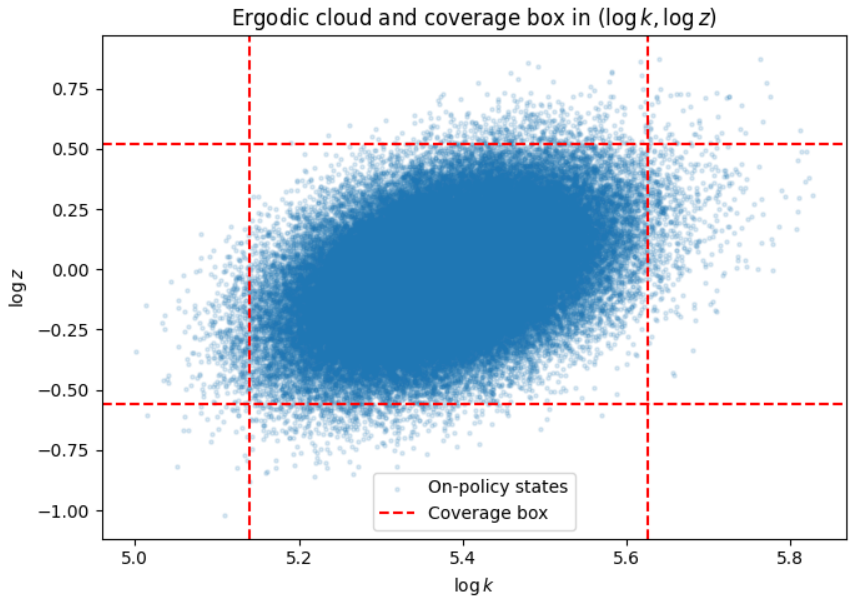

Figure 3:

1. **Absolute Euler residuals.** For a test set $\{(k_i, z_i)\}_{i=1}^N$ we compute

$$\bar{\mathcal{R}}_i \equiv \hat{\bar{\mathcal{R}}}_n(k_i, z_i; w), \qquad \left|\bar{\mathcal{R}}_i\right| = \left|\hat{\bar{\mathcal{R}}}_n(k_i, z_i; w)\right|,$$

and summarize the collection $\{|\bar{\mathcal{R}}_i|\}_{i=1}^N$ by the mean absolute error (MAE), the root mean squared error (RMSE), the median, the 95th percentile, and the maximum. Following the quantitative macroeconomics literature, we consider errors below 10^{-3} in absolute value to be negligible.

2. **Relative (dimensionless) residuals.** Since $\bar{\mathcal{R}}(k, z; w)$ is a linear combination of economically meaningful terms from the Euler equation, we define the relative residual

$$r_i \equiv \frac{|\bar{\mathcal{R}}_i|}{\left|1 + \psi_I(I_i, k_i)\right| + \left|\beta\, \widehat{\mathbb{E}}_n[\text{term}(k_i, z_i; w)]\right|},$$

where I_i is the investment implied by the policy at (k_i, z_i) and "term" denotes the bracketed expectation term on the right-hand side of the Euler equation. This quantity measures the Euler error as a fraction of the typical scale of the left- and right-hand sides.

3. **Share-of-states statistics.** We report the share of states satisfying $|\bar{\mathcal{R}}_i| \leq 10^{-3}$ and $|\bar{\mathcal{R}}_i| \leq 10^{-4}$, both on the ergodic set and on a wide coverage region that contains the ergodic set with high probability.

4. **GH-node robustness.** To verify that Euler errors are not driven by an inaccurate quadrature rule, we compare the median and 95th percentile of $|\bar{\mathcal{R}}_i|$ when increasing the number of nodes from 10 to 15 and 20. Relative changes at or below the level of numerical round-off indicate that GH–10 is already sufficiently accurate.

We perform four types of tests using GH–10 as the baseline integration rule:

Ergodic on-policy test. Under the final policy we simulate 2,048 independent paths, apply a 10,000-period burn-in, and then collect $N = 100{,}000$ states (k_t, z_t) from subsequent periods to approximate the ergodic distribution. On this on-policy set we obtain

$$\text{MAE}(|\bar{\mathcal{R}}|) = 3.139{\times}10^{-4}, \quad \text{RMSE} = 3.190{\times}10^{-4}, \quad \text{P95} = 4.008{\times}10^{-4}, \quad \max_i |\bar{\mathcal{R}}_i| = 5.631{\times}10^{-4}.$$

The corresponding relative residuals have mean 1.581×10^{-4}, median 1.582×10^{-4}, and 95th percentile 2.108×10^{-4}. All states satisfy $|\bar{\mathcal{R}}_i| \leq 10^{-3}$ and 0.6% satisfy $|\bar{\mathcal{R}}_i| \leq 10^{-4}$.

Coverage test. We construct an outer *coverage box* in $(\ln k, \ln z)$ space from the 1st–99th percentiles of the ergodic distribution, expanded by 5% in each dimension, and draw $N = 20{,}000$ states uniformly inside this box (see Figure 3). On this wider region we obtain

$$\text{MAE}(|\bar{\mathcal{R}}|) = 3.136{\times}10^{-4}, \quad \text{RMSE} = 3.302{\times}10^{-4}, \quad \text{P95} = 4.989{\times}10^{-4}, \quad \max_i |\bar{\mathcal{R}}_i| = 5.881{\times}10^{-4},$$

and the relative residuals have mean 1.606×10^{-4}, median 1.515×10^{-4}, and 95th percentile 2.784×10^{-4}. All coverage points satisfy $|\bar{\mathcal{R}}_i| \leq 10^{-3}$, and 0.8% satisfy $|\bar{\mathcal{R}}_i| \leq 10^{-4}$.

Edge and corner stress test. To probe the boundaries of the coverage region we evaluate Euler residuals on $N = 196$ states located along the edges and at the corners of the coverage box. On this stress-test set we obtain

$$\text{MAE}(|\bar{\mathcal{R}}|) = 3.029\times10^{-4}, \quad \text{RMSE} = 3.389\times10^{-4}, \quad \text{P95} = 5.593\times10^{-4}, \quad \max_i |\bar{\mathcal{R}}_i| = 5.904\times10^{-4}.$$

The relative residuals have mean 1.585×10^{-4}, median 1.308×10^{-4}, and 95th percentile 3.228×10^{-4}. Even at these boundary points all states satisfy $|\bar{\mathcal{R}}_i| \le 10^{-3}$, and 12.2% satisfy $|\bar{\mathcal{R}}_i| \le 10^{-4}$.

Quadrature robustness. On a random subsample of coverage points we recompute Euler residuals using GH–15 and GH–20. Relative to GH–10, the median and 95th percentile of $|\bar{\mathcal{R}}_i|$ change by 0.0% at the reported numerical precision for both GH–15 and GH–20, confirming that quadrature error is negligible.

In summary, the proposed deep-learning approach yields a policy function that is economically sensible and numerically accurate. Across the relevant state space, absolute Euler residuals are uniformly below 6×10^{-4}, well within the 10^{-3} tolerance commonly adopted in the quantitative macroeconomics literature, and relative Euler equation errors are on the order of 10^{-4}. These results suggest that the method can serve as a reliable benchmark for discrete-time investment models of the type studied in Strebulaev *et al.* [3].

1.3 Investment model with risky-debt

The basic investment model (Sec. 3.1 of Strebulaev *et al.*) has a representative firm choosing next period capital k' given current capital k and a productivity shock z. There is no corporate debt, no taxes, and therefore no financing margin: the decision is purely real. The Bellman equation is standard and the state is $s \equiv (k, z)$. This benchmark is ideal for isolating investment dynamics, but it cannot address capital structure, default, credit spreads, or tax policy because the cost of funds is exogenous and flat, there is no interest tax shield, and limited liability never binds.

To bring the model closer to corporate finance reality, we consider adding a one-period risky debt choice b', corporate taxes with an interest tax shield, and limited liability so equity can default endogenously when equity value hits zero(Sec. 3.6 of Strebulaev *et al.*). Lenders are risk-neutral and impose a zero-profit condition, so the debt yield $\tilde{r}(z, k', b')$ is priced *inside* the model. The state expands to $s \equiv (k, b, z)$ and the firm jointly chooses (k', b'). The problem becomes a coupled system: the firm's dynamic program interacts with a debt-pricing fixed point. Intuitively, leverage is chosen by trading off tax benefits of interest deductibility against expected deadweight default costs, under an endogenous borrowing rate that rises with default risk.

The main computational difficulty in the risky–debt model is a "loop within a loop." The endogenous debt yield $\tilde{r}(z, k', b')$ must satisfy lenders' zero–profit condition, which depends on default probabilities and recoveries implied by the firm's optimal policy and value; but the firm's optimal (k', b') choices themselves depend on $\tilde{r}$. A traditional solution begins by solving the value function under the assumption that debt is priced at the risk-free rate. Once the value function has been obtained, use it to identify default regions and compute the corresponding interest rate as a function of the model's state variables. This updated interest rate is then used to re-solve the value function. The process is repeated iteratively until the value function converges(see Ref [3]). While transparent, this approach is computationally heavy: the state (k, b, z) suffers the curse of dimensionality(especially when more state variables are included), limited liability induces kinks and nonconvexities that demand fine grids/interpolation, and accurate pricing requires costly nested integration over shocks.

We instead extend the DL framework(Maliar *et al.* 2021) we employed in previous section to tackle the risky debt model. The idea is to cast the whole equilibrium system into a single expectation–based loss and train neural networks to approximate the decision rules/value function. Concretely, we parameterize the policy functions by $\{k', b'\} = \{\varphi_k(k, b, z; w_k), \varphi_b(k, b, z; w_b)\}$ and the value function $V = \varphi_V(k, b, z; w_V)$. We then minimize an empirical loss that stacks (i) the Bellman–equation residuals (computed via the AiO product trick to avoid bias) and (ii) the FOC residuals. The lenders' zero–profit condition is enforced in the forward pass when computing the risky interest rate $\tilde{r}$, and thus its residual is not explicitly included in the loss function. This DL approach removes the explicit nested fixed point, so policies and pricing co–evolve during training until the residuals jointly vanish.

1.3.1 Model Formulation and training with Risky Debt

In this section, we outline the basic formulation of the risky–debt model.

Financing: one–period risky debt and net cash

At date t the firm chooses next–period net debt b' and capital k'. If $b' > 0$, it issues a one–period risky bond with *face value* b' to be repaid at $t{+}1$; if $b' \le 0$, it is a net lender holding a risk-free asset. We denote by $q_t(k', b', z)$ the time–t price per unit of (nominal) face value, so that the proceeds are

$$\text{debt proceeds}_t \;=\; q_t(k', b', z)\, b'.$$

In the implementation we treat very small positions as risk–free:

$$\text{if } b' \le \varepsilon_b, \qquad q_t(k', b', z) \;=\; \frac{1}{1+r},$$

for a small threshold $\varepsilon_b > 0$. This both avoids numerical problems when dividing by b' in the risky–debt pricing formula, and consistently treats net savings (and tiny debt positions) as risk–free claims.

Default, limited liability, and recovery

Equity has limited liability: if the continuation value is negative, equity defaults and receives zero. Let $C(k, b, z)$ denote the (pre–truncation) continuation value. The equity value is

$$V(k, b, z) \;=\; \max\{C(k, b, z), 0\}.$$

In the code this is implemented via a smooth softplus projection:

$$V(k, b, z) \;\approx\; \tau_V \,\log\!\Big(1 + e^{C(k,b,z)/\tau_V}\Big), \tag{20}$$

with a temperature parameter $\tau_V > 0$ that is gradually annealed during training. For small τ_V, (20) approximates $V = \max\{C, 0\}$.

If default occurs at $t{+}1$, debtholders seize the firm and receive a recovery value that is a fraction $1 - \alpha$ of the after–tax operating cash flow plus undepreciated capital:

$$R(k', z') \;:=\; (1-\alpha)\Big((1-\tau)\,\pi(k', z') + (1-\delta)\,k'\Big), \qquad \alpha \in [0, 1). \tag{21}$$

The bankruptcy cost parameter reflects the value lost when a firm defaults—due to legal fees, fire-sale losses, damaged relationships, and more. Higher bankruptcy costs mean creditors and shareholders recover less, making default more painful. As a result, firms are less inclined to take on high debt and tend to choose safer, lower–leverage strategies in the model.

Pricing of risky debt

Lenders are risk–neutral and discount at rate r. At date t, for $b' > \varepsilon_b$ (true risky debt), a bond with face value b' is sold at price

$$P_t \;=\; q_t(k', b', z)\, b', \qquad q_t(k', b', z) \;\equiv\; \frac{1}{1 + \tilde{r}(z, k', b')}.$$

At $t{+}1$, if there is no default the lender receives the promised face value b', while in default it receives $R(k', Z')$ as in Eq. (21). Thus the realized gross return to the lender is

$$\frac{b'}{q_t b'} \;=\; \frac{1}{q_t} \;=\; 1 + \tilde{r}(z, k', b') \quad \text{(no default)}, \qquad \frac{R(k', Z')}{q_t b'} \quad \text{(default)}.$$

Risk–neutral pricing requires that the expected discounted payoff equals the issue price $P_t = q_t b'$:

$$q_t(k', b', z)\, b' \;=\; \frac{1}{1 + r}\, \mathbb{E}\big[D' R(k', Z') + (1 - D')\, b' \,\big|\, Z = z\big], \tag{22}$$

where $D' = \mathbf{1}\{\text{default at } t{+}1\}$ is the default indicator. Dividing by b' and using $q_t = 1/(1 + \tilde{r})$ yields

$$1 + \tilde{r}(z, k', b') \;=\; \frac{1 + r}{\mathbb{E}\left[D' \,\dfrac{R(k', Z')}{b'} + (1 - D') \,\middle|\, Z = z\right]}. \tag{23}$$

In our implementation, the conditional expectation in (23) is approximated with M_{pricing} Monte Carlo draws of Z', using the *target* value network to evaluate $C(k', b', Z')$ and a smooth default gate

$$D' \;\approx\; \sigma\!\left(-\frac{C_{\bar{w}}(k', b', Z')}{\tau_D}\right), \tag{24}$$

where $\sigma(\cdot)$ is the logistic function, $\tau_D > 0$ is a temperature parameter, and $C_{\bar{w}}$ denotes the target continuation–value network (parameters frozen by Polyak averaging[1]). When $b' \le \varepsilon_b$ we bypass (23) and set

$$q_t(k', b', z) \;=\; \frac{1}{1 + r},$$

so that net saving is priced at the risk–free rate. In all cases, the implementation clips extreme payoff ratios $R(k', Z')/b'$ to mitigate numerical outliers, and clips q_t into $[0.01,\, 1/(1 + r)]$.

Cash flow and costly external finance

Let $q = q_t(k', b', z)$. The gross cash flow to equity at date t *before* external financing costs, denoted $e(k, k', b, b', z)$, is

$$\begin{aligned} e(k, k', b, b', z) \equiv (1 - \tau)\, \pi(k, z) \;-\; \Big(k' - (1 - \delta)k\Big) \;-\; \psi\big(k' - (1 - \delta)k, k\big) \\ +\; q\, b' \;+\; \tau\, (1 - q)\, b'\, \beta \;-\; b. \end{aligned} \tag{25}$$

The first line is after–tax operating profit net of investment and quadratic adjustment costs. The second line contains (i) the time–t proceeds from issuing (or repurchasing) debt with face value b',

[1]The purpose is to provide a slowly moving, smoothed approximation of the continuation value that is used inside pricing and the Bellman RHS, so that the bootstrapped targets, bond prices, and default probabilities change gradually and training remains stable and convergent.

and (ii) a discrete–time approximation of the tax effect of interest, implemented as $\tau(1-q)b'\beta$. For $b' > 0$, this term acts as a tax shield on expected interest payments; for $b' < 0$ it captures the tax on interest income from net saving.

As in Ref. [3], we allow for costly external equity finance and specify

$$\eta\big(e\big) \;\equiv\; \big(\eta_0 + \eta_1|e|\big)\,\mathbf{1}_{\{e<0\}}, \tag{26}$$

so that equity injections ($e < 0$) are penalized by a fixed component η_0 and a linear component proportional to $|e|$, while distributions ($e \geq 0$) are costless. The *net* payout to shareholders is then

$$d(k, k', b, b', z) \;\equiv\; e(k, k', b, b', z) - \eta\big(e(k, k', b, b', z)\big).$$

Bellman equation and continuation value

Let $C(k, b, z)$ denote the pre–truncation continuation value at state (k, b, z) and $V(k, b, z)$ the equity value under limited liability. The exact dynamic program can be written as

$$C(k, b, z) \;=\; \max_{k', b'} \big\{ d(k, k', b, b', z) + \beta\, \mathbb{E}\big[V(k', b', Z') \,\big|\, Z = z\big] \big\}, \tag{27}$$

$$V(k, b, z) \;=\; \max\{C(k, b, z), 0\}. \tag{28}$$

In the implementation we parameterize:

- a *policy* network π_w mapping normalized features $s = (\log k,\ b/k,\ \log z)$ to controls (k', b'):

  $$(k', b') \;=\; \pi_w(s),$$

  where $k' > 0$ is enforced by a softplus activation plus a small floor ($k' \geq 0.1$), and b' is unconstrained (can be positive or negative);

- a *continuation–value* network C_w mapping s to $C_w(k, b, z)$;

- the corresponding equity value via the softplus limited–liability transform in Eq. (20).

In the Bellman right–hand side we use a slowly moving *target* copy $C_{\bar{w}}$ (and $V_{\bar{w}}$) for the expectation over Z', updated by Polyak averaging. Given a state (k, b, z) and controls $(k', b') = \pi_w(k, b, z)$, the model–implied continuation value used in the loss is

$$\mathrm{RHS}(k, b, z; w) \;:=\; d(k, k', b, b', z) \;+\; \beta\, \mathbb{E}\big[V_{\bar{w}}(k', b', Z') \,\big|\, Z = z\big], \tag{29}$$

where the expectation is approximated via Monte Carlo. The Bellman equation then requires

$$C_w(k, b, z) \;\approx\; \mathrm{RHS}(k, b, z; w),$$

while Eq. (28) is enforced by the softplus mapping from C_w to V_w.

Soft default indicator

The exact default indicator at $t{+}1$ is $D' = \mathbf{1}\{C(k', b', Z') \leq 0\}$. For differentiability we use the smooth gate

$$D' \;\approx\; \sigma\!\left(-\frac{C_{\bar{w}}(k', b', Z')}{\tau_D}\right),$$

as in (24). This D' enters the risky–debt pricing kernel (23) and the diagnostic default–probability computation.

First–order conditions and FOC residuals

Optimality of (k', b') requires that the Bellman right–hand side be maximized with respect to these controls. For interior solutions,

$$\frac{\partial}{\partial k'}\mathrm{RHS}(k, b, z; w) \;=\; 0, \qquad \frac{\partial}{\partial b'}\mathrm{RHS}(k, b, z; w) \;=\; 0.$$

Analytical expressions are cumbersome, so in the code we compute these derivatives by automatic differentiation. To reduce Monte Carlo noise, we still use the AiO product estimator. For each state (k, b, z) we form two independent Monte Carlo evaluations

$$\mathrm{RHS}_a(k, b, z; w), \qquad \mathrm{RHS}_b(k, b, z; w),$$

and then compute the corresponding action–gradients

$$g_k^a \;:=\; \frac{\partial \mathrm{RHS}_a}{\partial k'}, \quad g_b^a \;:=\; \frac{\partial \mathrm{RHS}_a}{\partial b'}, \qquad g_k^b \;:=\; \frac{\partial \mathrm{RHS}_b}{\partial k'}, \quad g_b^b \;:=\; \frac{\partial \mathrm{RHS}_b}{\partial b'},$$

using a nested `GradientTape`. The FOC loss is then

$$L_{\mathrm{FOC}} \;:=\; \mathbb{E}\Big[\mathrm{stop\_grad}(g_k^a)\, g_k^b \;+\; \mathrm{stop\_grad}(g_b^a)\, g_b^b\Big], \tag{30}$$

where the expectation is over the training batch and Monte Carlo draws. Because g^a and g^b are independent stochastic estimates of the true gradient, (30) is an AiO estimator of the squared gradient up to sampling error, while gradients flow only through g^b. The implementation replaces non–finite entries in g^a, g^b by zeros for numerical robustness.

Our training enforces (i) Bellman consistency between the continuation–value network and the risky–debt environment induced by the current policy (via L_{Bell}), (ii) approximate optimality of the policy (via L_{FOC}), (iii) limited liability (via the softplus mapping), and (iv) internally consistent pricing of risky debt (via the pricing kernel and soft default indicator). In practice, jointly updating policy and value in this way is akin to an actor–critic method and converges to a fixed point approximating the optimal policy and value function for the risky–debt problem.

To illustrate the key differences, we summarize in the following table.

State, controls, constraints, and pricing

	Basic Model	**Risky-Debt Model**
States	$k \in \mathbb{R}_+$ (capital), z (productivity/demand shock, Markov)	(k, b, z) with $b \in \mathbb{R}$ (net debt; $b < 0$ interpreted as cash)
Controls	k' (equivalently $I = k' - (1-\delta)k$)	(k', b') (next-period capital and debt)
Cash flow (sources/uses)	$e = \pi(k, z) - \psi(I, k) - I$	$e = (1-\tau)\pi(k, z) - \psi(I, k) - I + \dfrac{b'}{1+\tilde{r}} + \dfrac{\tau \tilde{r}\, b'}{(1+\tilde{r})(1+r)} - b$
Law of motion	$k' = (1-\delta)k + I$	same for k', plus choice of b'
Debt pricing	not applicable (no debt pricing)	risky rate $\tilde{r} = \tilde{r}(z, k', b')$ via lender zero-profit
Default	none	Equity value V satisfies $V = \max\{0, \text{going-concern value}\}$; default $\iff V = 0$
Recovery	not applicable	$R(k', z') = (1-\alpha)\big[(1-\tau)\pi(k', z') + (1-\delta)k'\big]$

Table 1: Final accuracy diagnostics

	Coverage states	On–policy states
Mean relative Bellman error	6.87×10^{-3}	5.68×10^{-3}
Max. relative Bellman error	3.12×10^{-2}	3.43×10^{-2}
Mean squared FOC residual	1.57×10^{-2}	1.74×10^{-2}
Max. absolute FOC residual	3.80×10^{-1}	4.12×10^{-1}
Mean relative limited–liability error	3.01×10^{-5}	1.69×10^{-4}
Max. relative limited–liability error	1.29×10^{-3}	2.75×10^{-3}

1.3.2 Results and Discussion

This section reports the numerical performance of the deep–learning solution to the risky–debt model, as well as some economic consistency checks.

Training dynamics and accuracy We train the model for 2,500 epochs with a warm–up phase of 400 epochs, during which the state sampling is purely from a coverage distribution and the FOC penalty is kept small. After warm–up, each minibatch is a mixture of coverage states and on–policy states generated by an ergodic simulation, and the FOC weight is gradually increased.

Figure 4 (left panel) shows the evolution of the absolute value of the AiO loss on a logarithmic scale. The total loss declines monotonically from order unity at the start of training to roughly 10^{-3}–10^{-4} by the end and then fluctuates in this range. Because the AiO loss uses products of residuals rather than squared residuals, the scalar objective is not sign–definite and can occasionally become negative; for monitoring we therefore track its absolute value, as in [20]. The right panel of Figure 4 reports the mean relative Bellman error on held–out coverage and on–policy test sets. Both errors fall from values above one at initialization to below 10^{-2} within about 1,500 epochs and remain stable thereafter, with on–policy errors slightly below coverage errors on average, which is expected.

To assess accuracy we compute out–of–sample diagnostics on separate sets of coverage and on–policy states. The main statistic is a relative Bellman error,

$$\varepsilon^B(s) = \frac{\big|C(s) - \mathbb{E}[\mathrm{RHS}(s)]\big|}{\big|\mathbb{E}[\mathrm{RHS}(s)]\big| + 10^{-3}},$$

where $C(s)$ is the network's continuation value and $\mathbb{E}[\mathrm{RHS}(s)]$ is a Monte Carlo approximation of the right–hand side of the Bellman equation. We also report an FOC diagnostic $\varepsilon^F(s) = \|\nabla_{(k',b')}\mathrm{RHS}(s)\|^2$ and a limited–liability diagnostic comparing the softplus equity value V with $\max\{0, C\}$.

Table 1 summarizes the final diagnostics. Mean relative Bellman errors are 6.87×10^{-3} on coverage states and 5.68×10^{-3} on on–policy states, with maxima of about 3×10^{-2}. Limited–liability violations are extremely small: the mean relative error is on the order of 10^{-5}–10^{-4} and the worst–case error remains below 3×10^{-3}. The mean squared FOC residuals are about 1.6×10^{-2}, corresponding to an average first–order condition violation of roughly $\sqrt{1.6\times10^{-2}} \approx 0.13$ in absolute value. These magnitudes are economically modest and comparable across coverage and ergodic states.

Ergodic distributions of leverage and default probability After training, we simulate an ergodic panel of 256 firms for $T = 400$ periods and discard the first 100 periods as burn–in. Let k_t and b_t denote capital and debt in levels and define period–t leverage by $\ell_t = b_t/k_t$.

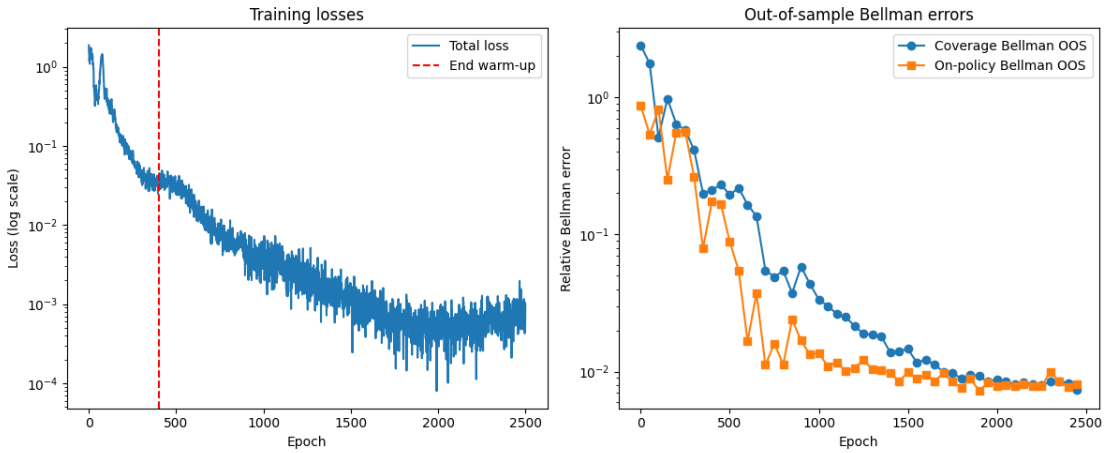

Figure 4: Training loss and out–of–sample Bellman error diagnostics. Notes: The left panel plots the absolute value of the AiO training loss on a logarithmic scale over 2,500 epochs; the red dashed vertical line marks the end of the 400–epoch warm–up phase. The loss falls from order unity to roughly 10^{-3}–10^{-4} and then fluctuates around this level. The right panel reports mean relative Bellman errors on held–out coverage and on–policy state sets. Both measures decrease from values above one at the start of training to below 10^{-2} and remain stable, with on–policy errors slightly below coverage errors.

Table 2: Ergodic leverage and default probability statistics

Statistic	Leverage $\ell_t = b_t/k_t$	Default probability π_t^{def}
Mean	1.5418	0.0017
Median	1.6005	0.0010
Std. dev.	0.2164	0.0017
10th pct.	1.2141	0.0002
50th pct.	1.6005	0.0010
90th pct.	1.7693	0.0044
Maximum	1.7809	0.0093

Table 2 reports descriptive statistics of the ergodic distribution of leverage and the one–step–ahead default probability π_t^{def}. The leverage distribution is tightly concentrated at relatively high levels: the mean is 1.5418 with a median of 1.6005 and a standard deviation of 0.2164. The 10th and 90th percentiles are 1.2141 and 1.7693, respectively, and the maximum observed leverage is 1.7809. The left panel of Figure 5 shows that the density rises gradually from about 0.6 and peaks close to the upper end of this range, indicating that most ergodic mass is at leverage ratios between roughly 1.6 and 1.8.

The one–period default probability is low on average but strongly right–skewed. The mean of π_t^{def} is about 0.0017, the median is 0.0010, and the 90th percentile is 0.0044; the maximum value in the panel is below one percent (0.0093). The right panel of Figure 5 illustrates that the distribution is highly concentrated near zero with a thin upper tail.

Cross–sectional default regions on the (k, b) grid To visualize how default risk varies across states, we compute the one–step–ahead default probability on a (k, b) grid for a relatively low

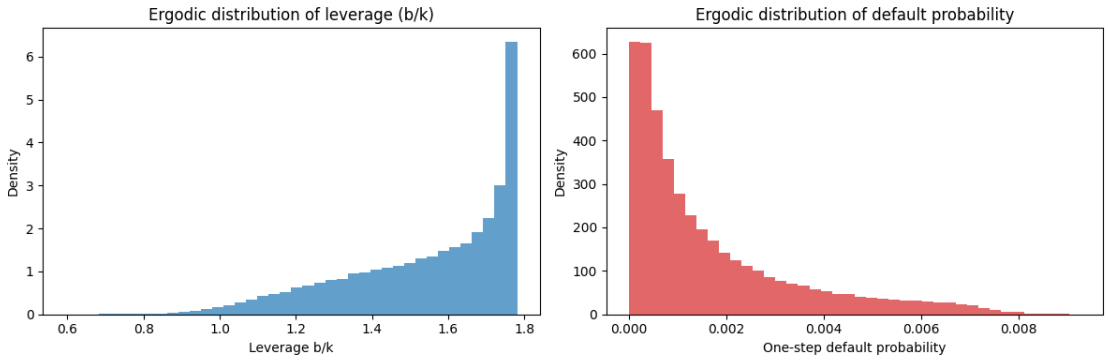

Figure 5: Ergodic distributions of leverage and default probability. Notes: The left panel plots the simulated ergodic distribution of leverage $\ell_t = b_t/k_t$ for the panel of 256 firms after burn–in. Leverage is concentrated between about 1.2 and 1.8, with most mass near the upper end of this range. The right panel shows the ergodic distribution of the one–period default probability π_t^{def}. Default probabilities are highly concentrated near zero, with a thin right tail and all mass below one percent.

productivity level $z = 0.3$; see Figure 6. The color map displays $\pi^{\text{def}}(k, b, z = 0.3)$, and black contour lines mark level sets of this probability. Default risk is highest in the region with low capital and high debt and decreases monotonically as either capital increases or debt declines. For sufficiently high k and low b the default probability becomes negligible, so that these states are effectively risk–free. The contours are smooth and well behaved, with no spurious oscillations or irregular boundaries, providing an additional consistency check on the learned policy and value functions.

Debt policy, value slices, and partial adjustment Further insight into the policy and value functions is provided by one–dimensional slices at $z = 0.3$; see Figure 7. For five representative debt levels $b \in \{-0.25, 0.30, 0.86, 1.43, 2.00\}$, the left panel plots the equity value $V(k, b, z = 0.3)$ as a function of capital k. Equity value is increasing and concave in k and decreasing in b, with smooth, well–ordered curves and no evidence of local irregularities.

The middle panel of Figure 7 shows the corresponding one–step–ahead default probabilities. For each b, $\pi^{\text{def}}(k, b, z = 0.3)$ is highest at low k and declines monotonically as capital increases, while higher debt levels shift the entire curve upward. Quantitatively, default probabilities on this adverse–productivity slice range from about 0.002 for high k and low b up to roughly 0.02 in the region with high debt and low capital, which is consistent with the ergodic statistics reported above.

The right panel depicts next–period leverage $\ell' = b'/k'$ implied by the policy network for the same values of b. For given productivity and current debt, optimal leverage is decreasing in current capital and increasing in current debt, and it remains in a moderate band between roughly 0.4 and 0.7 across the plotted range of k. In particular, leverage does not approach unity or explode as we move toward the edges of the grid. Together with the ergodic distribution, this pattern indicates that firms adjust toward an interior leverage range rather than pursuing ever–increasing leverage, which is consistent with partial adjustment and mean reversion in capital structure emphasized in [3].

Overall assessment From a numerical perspective, the deep–learning implementation achieves relative Bellman errors on the order of 10^{-3}–10^{-2} and mean squared FOC residuals around 10^{-2},

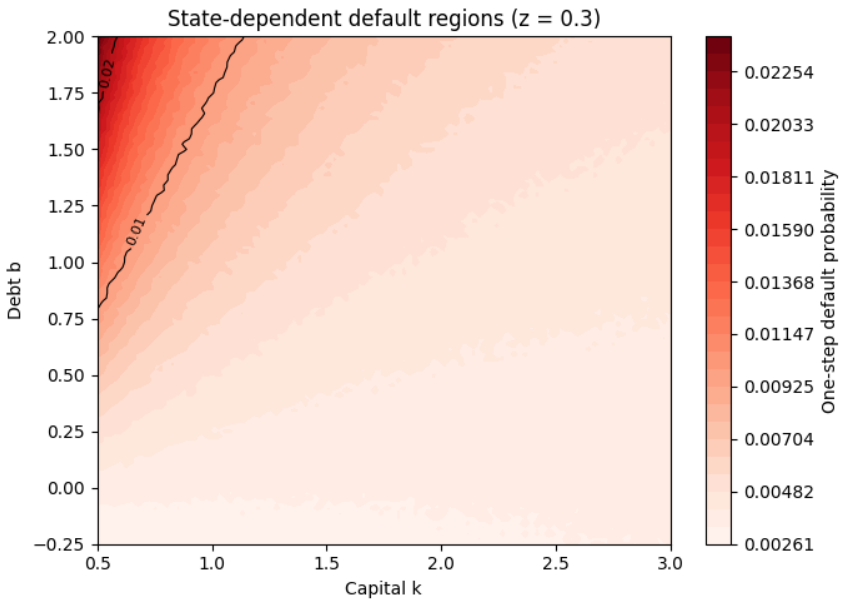

Figure 6: State–dependent default regions on the (k, b) grid at $z = 0.3$. Notes: The figure shows a contour plot of the one–step–ahead default probability as a function of capital k and debt b for a low productivity level $z = 0.3$. Warmer colors correspond to higher default probabilities, and black contour lines indicate level sets of the default probability. Default risk is concentrated in the region with low capital and high debt, while states with high k and low b are effectively default–free.

with very small violations of the limited–liability constraint, both on coverage and along the ergodic distribution. The implied leverage and default probability distributions are internally consistent: leverage is high but tightly concentrated, one–step default probabilities remain well below one percent even in adverse states. Cross–sectional patterns of equity value, default risk, and future leverage on the (k, b) grid exhibit the expected monotonicities and smoothness, and the policy functions keep leverage in an interior band, generating gradual mean reversion in capital structure.

In sum, the results suggest that the AiO deep–learning approach of [20] can solve the risky–debt model of [3] with reasonable accuracy and efficiency, producing policy functions and ergodic implications that are qualitatively and quantitatively in line with dynamic trade–off theory.

1.3.3 Issues and practices in NN and training design

As we have already noticed, the DL implementation here is quite different from conventional DL training in most data science applications using real-world data. Thus, in this section, we summarize the issues and practices that we have considered or need to be aware of in our DL implementation for the dynamic models discussed in this report.

- **State scaling and parameterization**
  - Work with transformed and normalized state variables (e.g. logs, leverage) to stabilize gradients and enforce basic constraints (such as positivity).
  - Parameterize controls in scale-free or normalized form, and recover levels from these normalized outputs.

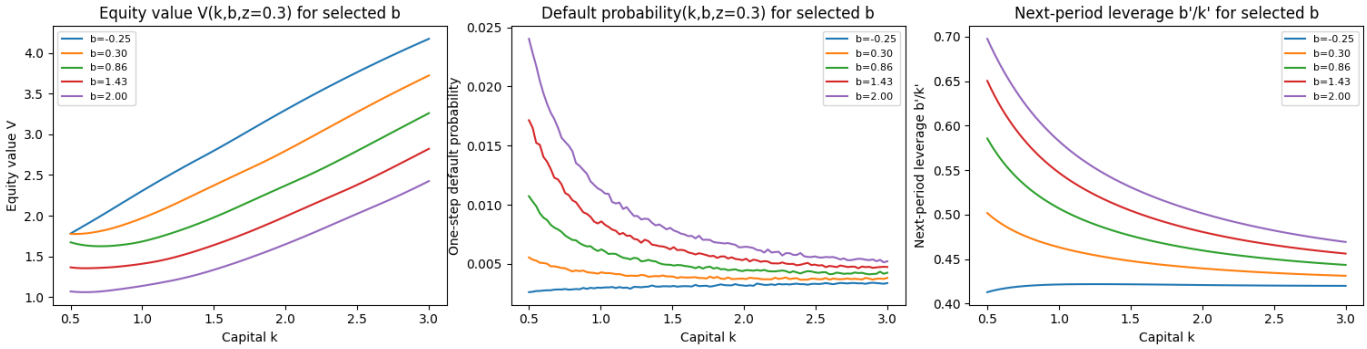

Figure 7: Policy and value slices at $z = 0.3$ for selected debt levels. Notes: The panels show (from left to right) the equity value $V(k, b, z = 0.3)$, the one–period default probability, and next–period leverage $\ell' = b'/k'$ as functions of capital k for five representative debt levels $b \in \{-0.25, 0.30, 0.86, 1.43, 2.00\}$. Equity is increasing in k and decreasing in b; default probabilities decline with k and rise with b; and future leverage is higher for higher current debt and lower current capital.

- **Feasibility-by-construction output maps**
  - Map raw network scores to economically feasible controls (e.g. positive capital, bounded investment rates) using smooth, bounded activation functions.
  - Impose limited liability and default rules via smooth approximations that are easy to differentiate.
  - Use clipping only as a secondary safeguard for robustness, not as the main way to impose constraints.
- **Sampling of training states**
  - Use a *coverage sampler* that draws states from a broad predefined region of the state space, with shocks drawn from their stationary distribution.
  - Use *on-policy* or *ergodic* sampling by simulating the model forward under the current policy and collecting the visited states.
  - Combine both in a *hybrid schedule*: start with coverage-only training and gradually increase the share of on-policy states while keeping some coverage states for global accuracy.
- **Expectation approximation and AiO operator**
  - During training, approximate expectations by Monte Carlo, using an All-in-One (AiO) structure where residuals or gradients under multiple independent shocks are combined within a single loss.
  - For evaluation, use accurate numerical quadrature (e.g. Gauss–Hermite) on fixed test sets to compute expectations and residuals.
- **Loss design**
  - In the basic model, train on an Euler-equation-based AiO loss that matches Euler residuals under independent shock draws.

- In the risky debt model, use a Bellman-type AiO loss for the value function and a first-order-condition (FOC) AiO loss for the policy, then combine them with a tunable weight.
- Stabilize Bellman updates with a slowly updated target value network (e.g. via Polyak averaging).

- **Optimization and numerical stability**
  - Use the Adam optimizer with a small fixed learning rate and a small number of shock draws per state (stochastic-gradient style).
  - Apply gradient norm clipping and zero out non-finite gradients to avoid numerical explosions, especially with nested automatic differentiation.
  - Gradually reduce the smoothness of limited-liability and default approximations during training (anneal "temperature" parameters).
  - Fix random seeds in all libraries for reproducibility and keep evaluation sets generated by independent random number streams.

- **Evaluation and convergence diagnostics**
  - Do not rely on a separate validation set or early stopping; instead, judge convergence using economically meaningful residuals and equilibrium conditions.
  - Maintain two fixed test sets: a coverage test set (uniform over the coverage region) and an on-policy test set (an ergodic sample under the trained policy).

1.3.4 Potential applications of the risky-debt model(Part 2b)

We recall that the risky-debt model describes a *discrete-time* dynamic corporate model with (i) endogenous investment, (ii) costly external equity finance, (iii) *risky* one-period debt priced endogenously via a competitive lender (zero-profit) condition, and (iv) *endogenous default* (equity value is the max of continuation value and zero). This workhorse structure is useful whenever we want real decisions (investment) and financing/default to be jointly determined. In the following, we summarize the some applications, limitations and potential solutions of risky-debt model(see also Ref[3]).

(A1) Interpreting and stress-testing investment–cash flow sensitivity

A first key application of the risky-debt model in is to use it as a laboratory for interpreting investment–cash flow sensitivities as measures of financing constraints. In such models, firms jointly and optimally choose investment, financing, and default policies over time in response to productivity and financing shocks. This structure makes it possible to generate synthetic panel data in which the researcher observes by construction which firms are financially constrained and which are not. One can then run standard investment regressions on these simulated data such as regressing investment on Tobin's q and cash flow, and compare estimated cash flow coefficients across truly constrained and unconstrained firms. Moyen [21] implements this type of exercise in a risky-debt setting and shows that investment–cash flow sensitivity can be a misleading proxy for financial constraints: in their model, truly constrained firms (those that cannot raise external funds) can display *lower* sensitivity than unconstrained firms because constraints limit the firm's ability to scale investment in response to shocks.

(A2) Structural estimation of external financing and bankruptcy costs

A second major application is to structurally estimate economically meaningful primitives—such as equity flotation costs and bankruptcy (or distress) costs—by fitting a dynamic risky-debt model with endogenous investment, leverage, payouts, and default directly to firm-level data. In this approach, one specifies the stochastic environment and the firm's problem, then uses simulation-based estimators (eg:SMM) to choose parameter values so that moments in the simulated data match those observed in the data. Hennessy and Whited [22] pursue precisely this strategy, estimating a model with taxation, costly bankruptcy, and linear–quadratic equity flotation costs. They find sizable marginal equity issuance costs and nontrivial bankruptcy costs as fractions of capital, and show that many standard empirical proxies for financing constraints can actually move in counterintuitive directions when financing frictions are increased. This application is important because it answers questions like "how costly is external finance?" in structural, quantitatively disciplined terms and enables counterfactual policy analysis on how changes in financing frictions would alter investment, leverage, and default.

(A3) Quantifying dynamic leverage with investment and agency frictions

A third key application is to quantify how leverage evolves over time when firms can continually adjust their capital structure and investment in the presence of distress costs and debtholder–equityholder agency conflicts. While many such models are written in continuous time, they are close in spirit to the risky-debt framework in Stebualev §3.6: the firm chooses both its capital structure and its investment policy in response to shocks, taking into account the option value of adjusting leverage and the incentives created by limited liability and default. Titman and Tsyplakov [23] develop such a dynamic model and study how financial distress costs and agency problems interact with investment opportunities to generate both time-series and cross-sectional variation in debt ratios. By running "regressions on simulated data" from their model that mirror standard empirical specifications, they evaluate how well the model can match observed leverage dynamics and the deviations of actual debt ratios from target levels. This application is central to corporate finance because it directly addresses the core question of how investment opportunities, distress costs, and agency frictions jointly shape firms' leverage paths over time.

1.3.5 Limitations and potential solutions(Part 2c)

(L1) One-period debt and the absence of a maturity structure

A first limitation of the risky-debt model is that debt is typically modeled as a one-period claim that is rolled over each period, with default occurring when the continuation value of equity falls to zero. Such a specification is natural for short-term bank lending, but it cannot capture the rich spectrum of observed corporate debt maturities or generate implications for the term structure of credit spreads. It therefore has limited ability to address questions about optimal debt maturity choice, rollover risk, or how credit spreads vary with horizon. A standard fix is to introduce long-term debt and allow the firm to choose both the *amount* and the *maturity* of its debt. Leland and Toft [24] develop exactly this type of model in continuous time, with endogenous bankruptcy and an explicit maturity structure, and show how the joint choice of leverage and maturity affects leverage levels, credit spreads, default rates, and writedowns. Incorporating these ideas into the discrete-time risky-debt framework would enable it to speak to optimal maturity and the term structure of credit spreads in a more realistic way.

(L2) Treating cash as "negative debt"

A second limitation is that cash is treated simply as "negative debt": the firm's single financial state variable is net debt, and holding cash is isomorphic to reducing debt. This restriction is analytically convenient but empirically problematic, because many firms simultaneously hold large cash balances and positive amounts of debt. Such joint borrowing and lending behavior is central for understanding liquidity management, precautionary savings, and the value of financial flexibility. A natural fix is to introduce separate state variables for cash and debt, together with issuance or adjustment costs, so that the firm can optimally choose when to borrow, when to save, and when to adjust either position. Gamba and Triantis [25] adopt this approach in a dynamic model that endogenizes financing, investment, and cash retention/payout policies, and show via simulations that firms facing financing frictions optimally *simultaneously* borrow and lend. Embedding this richer liquidity structure into the risky-debt framework greatly expands its ability to analyze cash holdings, payout, and liquidity management.

(L3) Stylized default without strategic renegotiation or covenants

A third important limitation is that default and bankruptcy are often modeled as a simple stopping time when equity value hits zero, followed by an exogenously specified recovery rule. This treatment abstracts from key institutional features of distress, such as strategic renegotiation between debtholders and equityholders, covenant violations, bargaining over reorganization, and systematic deviations from absolute priority. As a result, the model may misstate default timing, recovery rates, credit spreads, and payout or issuance policies around distress events. A standard remedy is to model debt contracts and distress explicitly as the outcome of a game between debtholders and equityholders, in which renegotiation and reorganization boundaries are determined endogenously. Anderson and Sundaresan [26] construct such a dynamic game-theoretic framework for the design and valuation of debt contracts, in which strategic debt service and endogenous reorganization boundaries generate realistic default premia and deviations from absolute priority. Fan and Sundaresan [27] further extend this line of research by embedding renegotiation and covenant effects into a model of debt valuation, dividend policy, and equity issuance, and by distinguishing liquidity-induced defaults from strategic defaults. Incorporating these strategic and contractual elements into the risky-debt model of §3.6 would yield a richer and more empirically accurate description of distress and reorganization.

2 Part 2 Estimation Methods: GMM/SMM

2.1 Introduction and Literature Review

Part 1 of this project focused on the *solution* side of dynamic structural corporate finance models. Starting from the basic discrete-time investment model in Section 3.1 of Ref[3], we specified the profit shock z_t as an AR(1) process and approximated the policy/value function using a deep learning approach in the spirit of Ref[20]. This first part thus provided a flexible computational framework for solving high-dimensional dynamic programming problems that arise in structural corporate finance, including extensions with risky debt as in Section 3.6 of Ref[3].

Part 2 shifts the focus from *solving* the model to *estimating* it. Instead of treating the structural parameters as known, we now ask how they can be inferred from data in a statistically disciplined way. Following Section 4 of Ref[3], we concentrate on two workhorse estimation techniques for dynamic structural models: the Generalized Method of Moments (GMM) and the Simulated Method of Moments (SMM). Both methods are method–of–moments estimators, but they differ in how

model-implied moments are computed and in the class of models for which they are practically tractable.

In this part we will: (1) conduct a literature review on estimation methods; (2) introduce and implement GMM & SMM in TensorFlow, reusing the solution machinery developed in Part 1 to evaluate or simulate model-implied moments; (3) compare the *effectiveness* of the two methods along both statistical and computational dimensions; and (4) discuss and illustrate potential empirical applications of the risky-debt model in Section 3.6 of Ref[3], including a critical assessment of its limitations and a brief literature survey of extensions that address those limitations.

Note that we do not compare GMM and SMM "in the abstract", because neither is a single estimator: both are families of estimators whose properties depend crucially on implementation choices, including the set of moments or summary statistics, the weighting matrix, and the optimization routine. Any blanket claim that one is better than another would therefore mostly reflect these design choices rather than the methods themselves. Instead, we specify and document two concrete estimators—one GMM and one SMM—built around clearly defined, economically motivated moments, comparable weighting schemes, and similar optimization settings(we also extend the method to HMC+filtering). We then compare their effectiveness in a controlled synthetic-data environment using common metrics: parameter accuracy, fit to key economic features of the data, and computational performance.

Throughout, "effectiveness" will be defined explicitly: on the statistical side in terms of finite-sample bias, variance, robustness to misspecification, and identification strength; and on the computational side in terms of convergence reliability, speed, scalability in state-space dimension, and ease of integration with modern machine-learning toolkits such as TensorFlow.

Intuitive overview of GMM and SMM. Most structural corporate finance models boil down to the same basic problem: the model implies that certain averages or "moments" computed from the data should satisfy a set of restrictions, and we want to choose the model's parameters so that those predicted moments line up with what we actually see in the data.

Generalized method of moments (GMM) is the workhorse when those moment conditions can be written explicitly as functions of observed variables and parameters. For example, an investment Euler equation says that, under the model, a particular nonlinear function of investment, cash flow, and instruments should have expectation zero. In practice, we compute the sample averages of this function and then search for the parameter vector that makes these sample moments as close to zero as possible. Intuitively, GMM picks the parameter values under which the model's theoretical restrictions are least violated in the data.

Simulated method of moments (SMM) (often also called MSM or SME) is the natural extension of this idea to models that are too complex for their moments to be written in closed form. In many dynamic corporate finance settings—featuring default, refinancing, taxes, adjustment costs, and occasionally binding constraints—the model-implied expectations are high-dimensional integrals with no analytic expression. What remains feasible, however, is to simulate artificial data from the model for any candidate parameter vector. SMM replaces analytically intractable expectations with sample averages from these simulations: it chooses parameters so that a vector of moments computed from simulated data matches the corresponding vector of empirical moments. Conceptually, SMM asks: *for which parameter values does the model, when we simulate it, look like the real world along a set of carefully chosen dimensions?*

GMM is therefore attractive when the model delivers explicit moment conditions in terms of observables, as in Euler-equation–based investment or asset-pricing tests. SMM becomes the tool of choice once we move to richer dynamic models whose only practical implication for the data is

"we can simulate it."

Foundations of GMM. Formally, GMM traces back to Hansen[28], who showed how to construct consistent and asymptotically normal estimators from general sets of orthogonality conditions $E[m(Y_t, \theta_0)] = 0$, even when the system is overidentified(see also [29]). His framework nests the classical method-of-moments and instrumental-variables estimators, and it quickly became standard in macroeconomics, asset pricing, and empirical corporate finance.

In finance, Hanse and Singleton[30] demonstrate how nonlinear rational-expectations models can be estimated by GMM: one derives Euler equations that imply that discounted asset payoffs are orthogonal to instruments, and then chooses preference and technology parameters so that these sample orthogonality conditions are as close as possible to zero. This logic carries over directly to structural investment and financing models in corporate finance, where Euler equations or first-order conditions generate tractable moment restrictions in terms of observable firm-level variables.

GMM in structural corporate finance. The earliest structural applications of GMM in corporate finance focus on investment Euler equations under financing frictions. Whited[31] estimates a nonlinear investment Euler equation on U.S. panel data and introduces a debt constraint/shadow cost that alters the Euler equation. Using GMM with lagged instruments, it is found that the standard frictionless Euler equation fits well for financially healthy firms but is rejected for distressed firms, which supports the view that asymmetric information in credit markets distorts investment.

Bond and Meghir[32] develop a dynamic investment model with convex adjustment costs and an explicit hierarchy-of-finance approach. They derive Euler equations that incorporate both financial assets and debt, and estimate them by GMM using U.K. company panel data. Their results suggest that internal funds affect investment primarily for firms that are more likely to be financially constrained, consistent with a pecking-order view nested in a structural model.

Using an international panel, Love[33] estimates an investment Euler equation in which the stochastic discount factor depends on firms' liquidity positions. The author's GMM estimates show that the sensitivity of investment to internal funds, interpreted as a measure of financing constraints, declines with country-level financial development, providing structural evidence that better-developed financial markets relax financing frictions.

A parallel literature uses GMM to estimate structural models that link investment and financing decisions to asset-pricing implications. Liu *et. al.*[34] derive a q-theory-based investment model in which expected stock returns equal levered returns on investment. They then use GMM to match average model-implied investment returns to average observed portfolio returns sorted on earnings surprises, book-to-market, and investment. The model captures several well-known return patterns, illustrating how GMM can be used to discipline joint models of real decisions and asset prices.

More recent GMM applications exploit the Euler-equation framework to study regulatory and institutional changes. For example, Kang *et. al.*[35] estimate an investment Euler equation in which the stochastic discount factor is allowed to shift after the Sarbanes–Oxley Act of 2002 (SOX). Using GMM on U.S. and U.K. panels, they find that the effective discount rate U.S. managers apply to investment projects rises significantly post-SOX, while the implied rate for U.K. firms is unchanged.[35] This provides a structural interpretation—through managers' perceived discount rates—of how governance regulation can feed into real investment.

These Euler-equation GMM studies share several features. First, they work with relatively low-dimensional state spaces (capital, productivity, simple financial variables), so that the expectations entering the Euler equations can be expressed analytically. Second, they make explicit use of rational-expectations arguments: the unobserved conditional expectations in the Euler equa-

tions are replaced by realized variables plus expectational errors that are orthogonal to lagged instruments[29, 30]. Finally, the GMM framework provides a natural overidentifying-restrictions test of the structural model: failures of the Euler equation to fit particular subsamples (e.g. distressed firms in [31]) are directly interpretable as evidence on the presence and nature of financial frictions.

Foundations of SMM and related simulation estimators. SMM generalizes GMM to settings in which the model's theoretical moments cannot be evaluated in closed form. The key idea is to replace analytically intractable expectations with simulated counterparts computed from artificial data generated by the structural model. One then chooses structural parameters to minimize a quadratic distance between simulated and empirical moments, using essentially the same objective as in GMM.

The asymptotic behavior of simulation-based extremum estimators was analyzed by Pakes and Pollard[36], who prove a general central limit theorem for estimators defined as minimizers of random criterion functions, without imposing smoothness on the criterion. Building on this foundation, Duffie and Singleton[37] formalize the *simulated moments estimator* for Markov models of asset prices and provide primitive conditions—most notably geometric ergodicity of the Markov process—under which the estimator is consistent and asymptotically normal.

In parallel, Gourieroux *et. al.*[38] develop the closely related *indirect inference* approach, in which one matches parameters of an auxiliary statistical model estimated on actual and simulated data. GMM itself can be viewed as a special case of indirect inference in which the auxiliary model consists of a vector of moments.

Simulation-based methods were also a breakthrough in dynamic discrete choice. Hotz and Miller[39] show that, under fairly general conditions, there is an inversion between conditional choice probabilities and value functions: one can recover the latter from the former, which can be estimated nonparametrically from data. Hotz *et. al.*[40] then build a simulation estimator for dynamic discrete-choice models that exploits this inversion and avoids solving the full dynamic programming problem at each trial parameter vector. Together with the broader literature on indirect inference, these contributions established SMM and related simulation-based estimators as flexible and powerful tools whenever likelihood-based methods are infeasible or prohibitively expensive.

SMM and simulation-based estimation in structural corporate finance. Dynamic structural corporate finance models almost invariably feature strong nonlinearities, occasionally binding financing and default constraints, and high-dimensional or latent state variables. These features make it difficult or impossible to write down closed-form likelihoods or analytical expressions for model-implied moments. As a result, SMM and related simulation-based estimators have become central in this literature[3].

A first generation of papers used simulated moments in a relatively informal way to discipline dynamic capital-structure models. Hennessy and Whited[41]. develop a dynamic trade-off model in which firms choose leverage, investment, and payouts in the presence of graduated corporate taxes, personal taxes, bankruptcy costs, and equity-issuance costs. They estimate/calibrate parameters so simulated moments match data and the he resulting model reproduces several facts that are hard to reconcile with static trade-off theory, including the absence of a sharp target leverage ratio and the negative relation between leverage and lagged measures of internal liquidity.

Hennessy and Whited take the next step by explicitly applying SMM to a dynamic model with endogenous investment, leverage, default, and costly external equity. They estimate equity

flotation and bankruptcy cost parameters by minimizing the distance between simulated and empirical moments summarizing firms' leverage, default, and payout behavior. The estimated marginal equity-issuance and bankruptcy costs are sizable but economically plausible, and the model implies that many common proxies for financing constraints behave in the opposite way suggested by static intuition, in the sense that increasing financing frictions can *reduce* investment–cash-flow sensitivities, because constrained firms optimally smooth investment rather than tracking cash-flow shocks one-for-one.

DeAngelo *et. al.*[42] estimate a dynamic capital-structure model in which firms hold conservative permanent leverage targets but deliberately and temporarily deviate from them by issuing transitory debt to fund investment. In their framework, leverage targets embed the option to issue such transitory debt, so that the evolution of leverage closely tracks the timing and magnitude of investment outlays. The estimated model replicates industry leverage very well, explains debt issuances and repayments better than standard tradeoff models, and accounts for the leverage changes that accompany investment "spikes." It also produces leverage ratios that exhibit slow average speeds of adjustment to target, driven by intentional temporary departures from target rather than by high debt issuance costs.

Finally, SMM and related methods have been used to quantify more specialized structural mechanisms in corporate finance. For instance, Albuquerque and Schroth[43] estimate a structural block-pricing model to back out private benefits of control from negotiated block trades. Li *et. al.*[44] estimate a dynamic model in which collateral constraints and taxes jointly determine leverage, using simulated moments to quantify the cost of lost financial flexibility from moving close to a collateral constraint; they find that this flexibility cost is of similar magnitude to the tax benefit of debt.

Taken together, these applications illustrate the comparative advantages of SMM in structural corporate finance. When the model can be expressed in terms of relatively simple orthogonality conditions involving observable variables—as in Euler-equation investment models—GMM remains the natural tool. When the mapping from parameters to observable implications is available only through simulation, SMM and related simulation-based estimators allow researchers to bring rich, fully dynamic models—with default, refinancing, taxes, collateral, and agency—directly to the data by matching a broad set of cross-sectional and time-series moments.

2.2 Formulation and Implementation

Since the main goal of this section is to implement and compare the Generalized Method of Moments (GMM) and the Simulated Method of Moments (SMM), we chose to apply both methods to the basic model presented in Section 3.1 of Ref[3] as a concrete example. The same workflow can also be extended to more complex models, although additional considerations may be required depending on the specific features of each model. In what follows, we first present a unified formulation that is common to both GMM and SMM, and then specialize it to each estimator. We then introduce our implementation subsection that emphasizes practical aspects (reparameterization, simulation, optimization etc.) in TensorFlow.

2.2.1 Formulation

Data, information set, and model-implied policy. We consider a panel of firms indexed by $i = 1, \ldots, N$, observed over periods $t = 0, \ldots, T$. Each firm solves the investment problem described in Section 3.1 of Ref[3]. The endogenous state is beginning-of-period capital k_{it} and the exogenous state is productivity z_{it}. Investment I_{it} is chosen at time t and updates the next-period capital

stock via

$$k_{i,t+1} = (1-\delta)k_{it} + I_{it}.$$

Throughout this subsection we work with synthetic panel data generated from the structural model. In particular, we observe the productivity shock z_{it} directly for each firm and period. This is convenient for exposition because it allows us to implement GMM and SMM using the "true" structural state (k_{it}, z_{it}) in the Euler residuals and (for GMM) as part of the instrument set. In empirical applications with real firm-level data, however, productivity is typically latent and must be proxied (e.g., via estimated TFP). Measurement error in such proxies can break orthogonality conditions and complicate instrument choice. Since our focus here is on design and implementation of GMM/SMM under ideal conditions, we abstract from latent productivity and measurement error.

For SMM, we must be able to simulate the model under candidate parameters. This requires the optimal policy function as a function of states and parameters. We therefore assume that for any parameter vector ϑ [2] we can evaluate an (approximate) optimal policy function

$$k' = \varphi(k, z; \vartheta), \qquad I = \varphi(k, z; \vartheta) - (1-\delta)k.$$

Conceptually, this is a nested fixed-point structure: for each trial ϑ in the outer estimation loop, an inner "solution" step provides $\varphi(\cdot;\vartheta)$ (e.g. via value function iteration, projection methods, or our deep learning solver from Part 1). As we shall see, our approach is to solve a parametrized policy as a callable routine. (For Euler-equation GMM, φ is not needed to evaluate the GMM objective given a panel $\{k_{it}, I_{it}, z_{it}\}$, but it is needed to generate synthetic panels in Monte Carlo experiments.)

In principle, evaluating the SMM objective at a given parameter vector ϑ would require re-solving the dynamic program in order to obtain the corresponding optimal policy $\varphi(\cdot;\vartheta)$, which induces a computationally costly nested fixed point. In this report we avoid this by training a single parameter-conditioned (amortized) policy network that approximates the mapping $(k, z, \theta, \phi) \mapsto \iota = I/k$ over a prespecified range of structural parameters (θ, ϕ), holding $(r, \delta, \rho, \sigma_\varepsilon)$ fixed. The network takes as inputs $\ln k$, $\ln z$, θ, and $\log \phi$, and outputs an investment rate ι constrained to an admissible interval $[\iota_{\min}, \iota_{\max}]$. It is trained once, offline, by minimizing an AiO Euler-equation loss with antithetic shocks on a mixture of coverage and on-policy states. During SMM estimation this amortized policy $\hat{\iota}(k, z; \theta, \phi)$ is kept fixed and plays the role of $\varphi(\cdot;\vartheta)$: for any candidate (θ, ϕ) we can simulate panels under the corresponding policy by forward evaluation of the network, eliminating the inner solution step and leaving only a controlled approximation error that we monitor via out-of-sample Euler residuals across the training parameter domain.

Euler equation and Euler residual (common to GMM/SMM). Assume the optimal investment choice is interior so that first-order conditions hold with equality. Then the optimality condition can be written as the Euler equation (cf. Strebulaev & Whited (2012), Eq. (3.8))

$$1 + \psi_I(I_{it}, k_{it}) \;=\; \beta\,\mathbb{E}\Big[\pi_k(k_{i,t+1}, z_{i,t+1}) - \psi_k(I_{i,t+1}, k_{i,t+1}) + (1-\delta)\big(1 + \psi_I(I_{i,t+1}, k_{i,t+1})\big)\;\Big|\;\mathcal{F}_{it}\Big], \tag{31}$$

where $\beta = 1/(1+r)$ and $\mathcal{F}_{it}$ is the firm-i information set at time t.

Define the *realized Euler residual* at generic parameter vector ϑ by replacing the conditional expectation with the realized future value:

$$\begin{aligned} u_{i,t+1}(\vartheta) \equiv\; & 1 + \psi_I(I_{it}, k_{it}; \vartheta) \\ & - \beta\Big[\pi_k(k_{i,t+1}, z_{i,t+1}; \vartheta) - \psi_k(I_{i,t+1}, k_{i,t+1}; \vartheta) + (1-\delta)\big(1 + \psi_I(I_{i,t+1}, k_{i,t+1}; \vartheta)\big)\Big]. \end{aligned} \tag{32}$$

[2] In our implementation, we shall estimate θ and ϕ for both GMM/SMM/HMC. Again this choice is for simplicity and the parameter set can certainly be extended to include more.

At the *true* parameter vector ϑ_0, equation (31) implies

$$\mathbb{E}\big[u_{i,t+1}(\vartheta_0) \,\big|\, \mathcal{F}_{it}\big] = 0 \quad \text{for all } i, t,$$

so that we can write

$$u_{i,t+1}(\vartheta_0) = \varepsilon_{i,t+1}, \qquad \mathbb{E}[\varepsilon_{i,t+1} \mid \mathcal{F}_{it}] = 0, \tag{33}$$

where $\varepsilon_{i,t+1}$ is the forecast error induced by replacing $\mathbb{E}[\cdot \mid \mathcal{F}_{it}]$ with the realized future value. Equation (33) embodies rational expectations and correct model specification.

GMM: population and sample moment conditions. Let $Z_{it} \in \mathbb{R}^q$ be a vector of *instruments* measurable with respect to $\mathcal{F}_{it}$. Then at the true parameter we have the orthogonality condition

$$\mathbb{E}[Z_{it}\, u_{i,t+1}(\vartheta_0)] = 0. \tag{34}$$

Define the moment function

$$m(X_{it}, \vartheta) \equiv Z_{it}\, u_{i,t+1}(\vartheta), \tag{35}$$

where X_{it} collects all variables needed to compute m (e.g. $X_{it} = (k_{it}, I_{it}, z_{it}, k_{i,t+1}, I_{i,t+1}, z_{i,t+1}, Z_{it})$). The *population* moment vector is

$$g(\vartheta) \equiv \mathbb{E}[m(X_{it}, \vartheta)] \in \mathbb{R}^q, \tag{36}$$

and equation (34) is equivalent to $g(\vartheta_0) = 0$.

Identification. If the parameter vector ϑ has dimension p and the number of moments q satisfies $q \geq p$, then the system is:

- *exactly identified* if $q = p$ and $g(\vartheta) = 0$ has a unique solution ϑ_0;
- *overidentified* if $q > p$ and the equation $g(\vartheta) = 0$ has a unique solution ϑ_0 but more restrictions than unknowns;
- *underidentified* if there is either no solution or infinitely many solutions.

GMM requires at least local identification: the Jacobian matrix

$$D(\vartheta) \;\equiv\; \frac{\partial g(\vartheta)}{\partial \vartheta'}$$

must have full column rank p at ϑ_0, meaning that the p columns of $D(\vartheta_0)$ are linearly independent. Intuitively, each column of $D(\vartheta_0)$ tells us how the population moments change if we nudge one parameter, holding the others fixed. Linear independence of those columns means: there is no non-trivial direction in parameter space we can move such that all the moments stay unchanged to first order.

Choosing instruments in the investment Euler equation The vector of instruments Z_{it} must satisfy two conditions:

1. *Relevance:* Z_{it} should be correlated with the Euler residual $u_{i,t+1}(\vartheta)$ through its dependence on ϑ, so that the moments are informative about ϑ. Formally, the population moment function

$$g(\vartheta) \equiv \mathbb{E}\big[Z_{it}\, u_{i,t+1}(\vartheta)\big]$$

must vary with ϑ, so that its Jacobian

$$D(\vartheta_0) \equiv \left.\frac{\partial g(\vartheta)}{\partial \vartheta'}\right|_{\vartheta=\vartheta_0}$$

has full column rank. This guarantees that only the true parameter ϑ_0 solves $\mathbb{E}[Z_{it}u_{i,t+1}(\vartheta)] = 0$, i.e. the instruments have identifying power for ϑ.

2. *Exogeneity:* Z_{it} must be measurable with respect to $\mathcal{F}_{it}$ and uncorrelated with the forecast error at the true parameter, i.e. $\mathbb{E}[Z_{it}\,\varepsilon_{i,t+1}] = 0$ where $\varepsilon_{i,t+1}$ is defined in (33). Measurability $Z_{it} \in \mathcal{F}_{it}$ means the instruments only use information available at time t and cannot depend on future shocks. Combined with the rational expectations condition $\mathbb{E}[\varepsilon_{i,t+1} \mid \mathcal{F}_{it}] = 0$ in (33), this implies, by the law of iterated expectations,

$$\mathbb{E}[Z_{it}\,\varepsilon_{i,t+1}] = \mathbb{E}[Z_{it}\,\mathbb{E}[\varepsilon_{i,t+1} \mid \mathcal{F}_{it}]] = 0,$$

so the instruments are orthogonal to the forecast error at the true parameter.

In our synthetic setup, a natural baseline choice is to use functions of the current state and investment rate, e.g.

$$Z_t = \begin{pmatrix} 1 \\ \log k_t \\ \log z_t \\ \iota_t \\ (\log k_t)^2 \\ (\log z_t)^2 \\ \iota_t^2 \end{pmatrix},$$

where all non-constant instruments (all components except the intercept) are standardized to have zero mean and unit variance in the sample. One can also include lagged state and control variables such as $k_{i,t-1}$, lagged proxies for profitability, and the lagged investment rate $I_{i,t-1}/k_{i,t-1}$ as instruments, following Whited (1992) [45] and Bond & Meghir (1994) [46].

Sample moments and GMM criterion (large-N, fixed-T). To match standard panel GMM practice, we adopt large-N, fixed-T asymptotics with independence across firms i and allowing arbitrary within-firm serial correlation over t. Define the firm-level average moment

$$\bar{m}_i(\vartheta) \equiv \frac{1}{T}\sum_{t=0}^{T-1} m(X_{it}, \vartheta),$$

and the sample moment vector

$$g_N(\vartheta) \equiv \frac{1}{N}\sum_{i=1}^{N} \bar{m}_i(\vartheta) \in \mathbb{R}^q. \tag{37}$$

By the law of large numbers, if the number of observations are sufficiently large (e.g. $N \to \infty$), then $g_N(\vartheta)$ converges in probability to $g(\vartheta)$.

Let W_N be a $q \times q$ positive definite weighting matrix. The GMM criterion is

$$J_N(\vartheta) \equiv g_N(\vartheta)^\top W_N g_N(\vartheta), \tag{38}$$

and the GMM estimator is

$$\hat{\vartheta}_N \equiv \arg\min_{\vartheta \in \Theta} J_N(\vartheta), \tag{39}$$

where Θ is the admissible parameter space. Intuitively, $g_N(\vartheta)$ measures how far the sample moments are from their theoretical value of zero for a given ϑ. The GMM estimator chooses ϑ to make these discrepancies as small as possible in the quadratic norm defined by W_N.

Under the stated asymptotics, the asymptotic covariance of $\sqrt{N}\, g_N(\vartheta_0)$ is

$$S \equiv \mathrm{Var}\big(\bar{m}_i(\vartheta_0)\big) = \mathbb{E}\Big[(\bar{m}_i(\vartheta_0) - g(\vartheta_0))(\bar{m}_i(\vartheta_0) - g(\vartheta_0))^\top\Big]. \tag{40}$$

It is well known in [47] that asymptotic variance is minimized when $W_N \to_p W^* = S^{-1}$ (the optimal two-step GMM weight).

Two-step GMM and cluster-by-firm $\hat{S}_N$. A standard two-step procedure is:

1. First step: set $W_N^{(1)} = I_q$ and compute

$$\hat{\vartheta}_N^{(1)} = \arg\min_{\vartheta \in \Theta} g_N(\vartheta)^\top g_N(\vartheta).$$

2. Estimate S: compute fitted moments $\hat{m}_{it} = m(X_{it}, \hat{\vartheta}_N^{(1)})$ and firm averages $\hat{\bar{m}}_i = T^{-1} \sum_{t=0}^{T-1} \hat{m}_{it}$. Let $\hat{g}_N = g_N(\hat{\vartheta}_N^{(1)})$. A cluster-by-firm estimator consistent under large-N, fixed-T with arbitrary within-firm serial correlation is

$$\hat{S}_N \equiv \frac{1}{N} \sum_{i=1}^{N} (\hat{\bar{m}}_i - \hat{g}_N)(\hat{\bar{m}}_i - \hat{g}_N)^\top. \tag{41}$$

   (In real data, additional cross-firm correlation from aggregate shocks would require further adjustments, e.g. common factors or alternative covariance estimators.)

3. Second step: set $W_N^{(2)} = \hat{S}_N^{-1}$ and compute

$$\hat{\vartheta}_N^{(2)} = \arg\min_{\vartheta \in \Theta} g_N(\vartheta)^\top W_N^{(2)} g_N(\vartheta).$$

Under regularity conditions,

$$\sqrt{N}\big(\hat{\vartheta}_N^{(2)} - \vartheta_0\big) \xrightarrow{d} \mathcal{N}(0, V),$$

with the usual sandwich form. Let $D = D(\vartheta_0)$. For a generic weight W,

$$V = (D^\top W D)^{-1} (D^\top W S W D)(D^\top W D)^{-1}, \tag{42}$$

and for the optimal $W^* = S^{-1}$ this simplifies to $V = (D^\top S^{-1} D)^{-1}$.

When $q > p$ (overidentification), the Hansen J-test is based on the scaled minimized criterion:

$$N\, J_N\big(\hat{\vartheta}_N^{(2)}\big) \xrightarrow{d} \chi^2_{q-p}, \tag{43}$$

so large values of NJ_N relative to χ^2_{q-p} lead to rejection of the joint validity of the moment conditions (which can reflect misspecified instruments, misspecified structure, or both).

SMM: simulated moments and distance matching. SMM uses the same structural model but does not require closed-form population moments. Instead it matches *evaluation moments* computed from data to their simulated counterparts.

Let y denote a (stacked) panel data set (e.g. $\{k_{it}, z_{it}, I_{it}\}_{i,t}$). Choose a vector of K evaluation moments

$$h(y) \in \mathbb{R}^K, \tag{44}$$

In our implementation we take $K = 13$ and let $h(y)$ collect a set of unconditional, dynamic, and cross moments of investment, capital, and productivity. These moments are designed to jointly identify the technology parameter θ and the adjustment-cost parameter ϕ.

Given a panel

$$y = \{k_{it}, I_{it}, z_{it}\}_{i=1,\dots,N;\, t=0,\dots,T},$$

define the investment rate, log capital, and log productivity by

$$\iota_{it} \equiv \frac{I_{it}}{k_{it}}, \qquad \kappa_{it} \equiv \log k_{it}, \qquad \zeta_{it} \equiv \log z_{it}.$$

For any generic variable x_{it}, define the panel mean and variance by

$$\bar{x}(y) \equiv \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} x_{it},$$

$$s_x^2(y) \equiv \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \big(x_{it} - \bar{x}(y)\big)^2.$$

For two variables x_{it} and w_{it} we define the correlation as

$$\operatorname{corr}(x, w; y) \equiv \frac{\dfrac{1}{NT} \displaystyle\sum_{i=1}^{N} \sum_{t=1}^{T} \big(x_{it} - \bar{x}(y)\big)\big(w_{it} - \bar{w}(y)\big)}{\sqrt{s_x^2(y)}\,\sqrt{s_w^2(y)}}.$$

We also use the first-order autocorrelation of x_{it},

$$\rho_x(1; y) \equiv \operatorname{corr}\big(x_{it}, x_{i,t-1}; y\big),$$

and, for any x_{it}, the within-firm first difference $\Delta x_{it} \equiv x_{it} - x_{i,t-1}$ (defined for $t \geq 1$).

Productivity follows a known AR(1) process in logs,

$$\zeta_{it} = \mu_z + \rho_z\, \zeta_{i,t-1} + \varepsilon_{it},$$

with μ_z and ρ_z fixed at their true values in the SMM implementation. The corresponding innovation is

$$\varepsilon_{it} \equiv \zeta_{it} - \mu_z - \rho_z\, \zeta_{i,t-1}.$$

We then set

$$h(y) = \begin{bmatrix} h_1(y) \\ \vdots \\ h_{13}(y) \end{bmatrix} \in \mathbb{R}^{13}, \tag{45}$$

with components

$$
\begin{aligned}
h_1(y) &\equiv \bar{\kappa}(y) && \text{mean log capital,}\\
h_2(y) &\equiv \mathrm{corr}(\iota, \zeta; y) && \text{contemporaneous corr. between investment and log productivity,}\\
h_3(y) &\equiv \rho_\iota(1; y) && \text{AR(1) autocorrelation of the investment rate,}\\
h_4(y) &\equiv \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\big(\iota_{it} - \delta\big)^2 && \text{mean squared deviation of } \iota_{it} \text{ from depreciation } \delta,\\
h_5(y) &\equiv s^2_{\Delta\kappa}(y) && \text{variance of } \Delta\kappa_{it} = \kappa_{it} - \kappa_{i,t-1},\\
h_6(y) &\equiv \mathrm{corr}\big(\iota_{i,t+1}, \zeta_{it}; y\big) && \text{lead–lag corr. between } \iota_{i,t+1} \text{ and } \zeta_{it},\\
h_7(y) &\equiv \bar{\iota}(y) && \text{mean investment rate,}\\
h_8(y) &\equiv \sqrt{s^2_\iota(y)} && \text{standard deviation of the investment rate,}\\
h_9(y) &\equiv s^2_{\Delta\iota}(y) && \text{variance of } \Delta\iota_{it} = \iota_{it} - \iota_{i,t-1},\\
h_{10}(y) &\equiv \mathrm{corr}(\iota, \varepsilon; y) && \text{corr. between } \iota_{it} \text{ and the productivity innovation } \varepsilon_{it},\\
h_{11}(y) &\equiv \mathrm{corr}\big(\iota_{i,t+1}, \varepsilon_{it}; y\big) && \text{lead–lag corr. between } \iota_{i,t+1} \text{ and } \varepsilon_{it},\\
h_{12}(y) &\equiv b_\iota(y) && \text{OLS slope on } \iota_{i,t-1} \text{ in a panel regression of } \iota_{it},\\
h_{13}(y) &\equiv b_\varepsilon(y) && \text{OLS slope on } \varepsilon_{it} \text{ in the same regression,}
\end{aligned}
$$

where $(b_\iota(y), b_\varepsilon(y))$ are the coefficients from the pooled OLS regression

$$\iota_{it} = a(y) + b_\iota(y)\,\iota_{i,t-1} + b_\varepsilon(y)\,\varepsilon_{it} + u_{it},$$

estimated over all (i,t) for which $\iota_{i,t-1}$ and ε_{it} are observed.

The first six moments $(h_1, \ldots, h_6)$ correspond to our original investment–capital specification: they jointly target the unconditional level and volatility of capital, the dynamics of the investment rate, and the contemporaneous and lead–lag response of investment to productivity. The additional seven moments $(h_7, \ldots, h_{13})$ are specifically designed to sharpen identification of the adjustment-cost parameter ϕ, by exploiting higher-order variability of the investment rate, its first differences, its comovement with productivity innovations, and regression coefficients from a simple panel regression.

In the SMM estimator we compute the empirical moment vector

$$\hat{h}^{\mathrm{data}} = h\big(y^{\mathrm{data}}\big), \tag{46}$$

using these definitions.

For any candidate parameter vector $\vartheta = (\theta, \phi)$, we simulate B artificial panels $y^{\mathrm{sim},b}(\vartheta)$ from the structural model (each with the same dimensions as y^{data}), using the policy function $\varphi(\cdot; \vartheta)$:

$$y^{\mathrm{sim},b}(\vartheta) \sim \mathrm{Model}(\vartheta), \qquad b = 1, \ldots, B. \tag{47}$$

For each simulated panel we compute

$$\hat{h}^{\mathrm{sim},b}(\vartheta) = h\big(y^{\mathrm{sim},b}(\vartheta)\big), \qquad \bar{h}^{\mathrm{sim}}(\vartheta) = \frac{1}{B}\sum_{b=1}^{B}\hat{h}^{\mathrm{sim},b}(\vartheta).$$

Let W^{SMM} be a $K \times K$ positive definite weight matrix. In our two-step implementation we first use $W^{\text{SMM}} = I_K$ and then set W^{SMM} equal to an estimate of the inverse covariance matrix of the moment conditions, $\widehat{\Sigma}_g^{-1} \approx \mathbb{V}\text{ar}\big(h^{\text{sim}} - h^{\text{data}}\big)^{-1}$, where $\widehat{\Sigma}_g$ is constructed from a firm-level bootstrap of the data moments and Monte Carlo variation of the simulated moments.

Define the discrepancy

$$d(\vartheta) \equiv \bar{h}^{\text{sim}}(\vartheta) - \hat{h}^{\text{data}} \in \mathbb{R}^K$$

and the SMM objective

$$Q(\vartheta) \equiv d(\vartheta)^\top W^{\text{SMM}} d(\vartheta). \tag{48}$$

The SMM estimator is

$$\hat{\vartheta}_N^{\text{SMM}} \equiv \arg\min_{\vartheta \in \Theta} Q(\vartheta).$$

Identification requires that the mapping $\vartheta \mapsto \mathbb{E}[h(y^{\text{sim}}(\vartheta))]$ be injective (locally) around ϑ_0. Under regularity conditions, $\hat{\vartheta}_N^{\text{SMM}}$ is consistent and asymptotically normal [e.g. 48, 49]. Because $\bar{h}^{\text{sim}}(\vartheta)$ is estimated using a finite number B of simulated panels, the asymptotic variance includes an additional simulation-error component that vanishes as $B \to \infty$ (often appearing as a factor $1 + 1/B$ when simulated and empirical sample sizes match).

Overidentification and J-tests (GMM and SMM)

Both GMM and SMM are method-of-moments estimators that typically match $q \geq p$ moments with p structural parameters. When $q > p$, the overidentifying restrictions implied by the model can be tested using a Hansen-type J-test that has the same asymptotic χ^2 form for both estimators.

Let $r_N(\vartheta)$ denote the vector of sample moment conditions used in the criterion, normalized so that $\sqrt{N}\, r_N(\vartheta_0)$ has a non-degenerate limit at the true parameter ϑ_0. In our setting:

- for GMM, $r_N(\vartheta) = g_N(\vartheta)$ defined in (38);
- for SMM, $r_N(\vartheta) = \bar{h}_{\text{sim}}(\vartheta) - \hat{h}_{\text{data}}$ as in (49), where $\hat{h}_{\text{data}}$ is defined in (47) and $\bar{h}_{\text{sim}}(\vartheta)$ is the Monte Carlo average of the simulated moment vectors.

Let Σ be the asymptotic covariance matrix of $\sqrt{N}\, r_N(\vartheta_0)$ and let $\hat{\Sigma}_N$ be a consistent estimator of Σ. In practice, $\hat{\Sigma}_N$ is:

- for GMM, the firm-cluster estimator $\hat{S}_N$ in (42);
- for SMM, the sum of the sampling and simulation components, e.g.

$$\hat{\Sigma}_N = \hat{\Sigma}_{\text{data}} + \frac{1}{B}\,\hat{\Sigma}_{\text{sim}},$$

  where $\hat{\Sigma}_{\text{data}}$ is obtained from a firm-level bootstrap of the empirical moments and $\hat{\Sigma}_{\text{sim}}$ is the covariance of $\hat{h}_{\text{sim},b}(\hat{\vartheta})$ across the B simulation replications at the estimated parameter $\hat{\vartheta}$.

Using the efficient weight matrix $W_N = \hat{\Sigma}_N^{-1}$, the J-statistic is defined as

$$J_N(\hat{\vartheta}) \equiv \big(\sqrt{N}\, r_N(\hat{\vartheta})\big)^\top W_N \big(\sqrt{N}\, r_N(\hat{\vartheta})\big) = N\, r_N(\hat{\vartheta})^\top W_N r_N(\hat{\vartheta}).$$

Under the null hypothesis that the model is correctly specified and that there exists ϑ_0 such that the population moment conditions are exactly satisfied, and under standard regularity conditions

(including that the number of simulations B is large enough and/or $\hat{\Sigma}_N$ properly accounts for simulation noise), we have

$$J_N(\hat{\vartheta}) \overset{d}{\Rightarrow} \chi^2_{q-p}.$$

Thus, large values of $J_N(\hat{\vartheta})$ relative to the χ^2_{q-p} distribution lead to rejection of the joint validity of the overidentifying restrictions, both for GMM and for SMM.

2.2.2 Unified implementation (GMM and SMM)

In this subsection we describe how we implement both GMM and SMM in TensorFlow

Parameter reparameterization and initialization. For numerical stability and unconstrained optimization, we work with an unconstrained parameter vector $\omega \in \mathbb{R}^p$ and map it to the structural parameter vector $\vartheta = \Psi(\omega)$ via smooth transformations that enforce bounds:

- Production curvature $\theta \in (0, 1)$:
$$\theta = \sigma(\omega_\theta),$$
where $\sigma(\cdot)$ is the logistic function.
- Adjustment-cost parameter $\phi > 0$:
$$\phi = \exp(\omega_\phi).$$
- Other parameters with natural bounds are treated analogously. Calibrated parameters such as r, δ, and parameters of the z-process are kept fixed.

In TensorFlow, we treat ω as the trainable variable and implement $\Psi(\omega)$ as a mapping returning ϑ. Both the GMM and SMM objectives are then defined as functions of ω.

Common building blocks: simulation and Euler residual evaluation. We implement a simulation routine

$$\texttt{sim\_panel}(\vartheta, \texttt{seed}) \mapsto \{k_{it}, z_{it}, I_{it}\}_{i,t}$$

that generates a panel using the policy function $\varphi(\cdot; \vartheta)$. To evaluate Euler residuals $u_{i,t+1}(\vartheta)$ for $t = 0, \ldots, T-1$, the simulated panel must contain all quantities at times t and $t+1$. In our Monte Carlo design we therefore store one extra period so that $(k_{i,T+1}, I_{i,T}, z_{i,T+1})$ are available if needed. (Equivalently, one can drop the last residual observation.)

We also implement a helper that, given a panel $\{k_{it}, z_{it}, I_{it}\}$ and ϑ, computes Euler residuals $u_{i,t+1}(\vartheta)$ using (32) and, if requested, constructs instruments Z_{it} and moments $m(X_{it}, \vartheta) = Z_{it} u_{i,t+1}(\vartheta)$ as in (35).

GMM objective and optimization. Given a (possibly synthetic) data panel y^{data}, we:

1. Fix an instrument set Z_{it}, e.g. the baseline choice (**??**).
2. For each trial ω (and $\vartheta = \Psi(\omega)$), compute moments $m(X_{it}, \vartheta)$ and firm averages $\bar{m}_i(\vartheta) = T^{-1} \sum_{t=0}^{T-1} m(X_{it}, \vartheta)$, then form $g_N(\vartheta) = N^{-1} \sum_{i=1}^{N} \bar{m}_i(\vartheta)$ as in (37).

3. First step: minimize
$$\mathcal{L}_1^{\mathrm{GMM}}(\omega) = J_N^{(1)}(\vartheta) = g_N(\vartheta)^\top g_N(\vartheta)$$
using a gradient-based optimizer (e.g. Adam or L–BFGS), with gradients obtained via automatic differentiation.

4. At the first-step solution $\hat{\omega}^{(1)}$, compute $\hat{S}_N$ via the firm-cluster estimator (41) and set $W_N^{(2)} = \hat{S}_N^{-1}$.

5. Second step: minimize
$$\mathcal{L}_2^{\mathrm{GMM}}(\omega) = J_N^{(2)}(\vartheta) = g_N(\vartheta)^\top W_N^{(2)} g_N(\vartheta),$$
starting from $\hat{\omega}^{(1)}$, to obtain $\hat{\omega}^{(2)}$ and $\hat{\vartheta}_N^{\mathrm{GMM}} = \Psi(\hat{\omega}^{(2)})$.

6. Inference: compute the Jacobian $\hat{D}_N = \partial g_N(\vartheta)/\partial \vartheta^\top$ via automatic differentiation and assemble the sandwich covariance estimator $\hat{V}_N$ using (42) with $\hat{S}_N$ and $W_N^{(2)}$. For specification testing, compute the Hansen J-statistic $N\, J_N(\hat{\vartheta}_N^{\mathrm{GMM}})$ and compare it to χ^2_{q-p} as in (43).

SMM objective and optimization. For SMM we reuse the same parameterization $\Psi(\omega)$ and simulation routine `sim_panel`, but we replace the GMM loss by the SMM distance-matching loss:

1. Compute empirical evaluation moments once: $\hat{h}^{\mathrm{data}} = h(y^{\mathrm{data}})$ as in (46).

2. Fix a Monte Carlo design: number of simulated panels B, length T, number of firms N, and a set of pre-drawn shocks or random seeds. For variance reduction we use *common random numbers*: for each ω we reuse the same underlying draws of the shock innovations (implemented via a reparameterization in TensorFlow).

3. For each trial ω (and $\vartheta = \Psi(\omega)$), simulate B panels $y^{\mathrm{sim},b}(\vartheta)$ and compute $\hat{h}^{\mathrm{sim},b}(\vartheta) = h(y^{\mathrm{sim},b}(\vartheta))$. Average over b to obtain $\bar{h}^{\mathrm{sim}}(\vartheta)$.

4. Define the SMM loss
$$\mathcal{L}^{\mathrm{SMM}}(\omega) = \big(\bar{h}^{\mathrm{sim}}(\vartheta) - \hat{h}^{\mathrm{data}}\big)^\top W^{\mathrm{SMM}} \big(\bar{h}^{\mathrm{sim}}(\vartheta) - \hat{h}^{\mathrm{data}}\big),$$
where W^{SMM} is a fixed positive definite weight matrix (e.g. identity or diagonal inverse-variance weights for the moments).

5. Minimize $\mathcal{L}^{\mathrm{SMM}}(\omega)$ over ω using a gradient-based optimizer. Because simulation is implemented with common random numbers, gradients of $\mathcal{L}^{\mathrm{SMM}}$ with respect to ω are available via automatic differentiation.

In Monte Carlo experiments (where the "true" ϑ_0 is known) we compare GMM and SMM along standard performance metrics (bias, variance, coverage, and moment-fit distances), using the same simulation, policy, and parameterization modules. This unified implementation makes it straightforward to switch between GMM and SMM simply by changing the loss function from $\mathcal{L}_1^{\mathrm{GMM}}/\mathcal{L}_2^{\mathrm{GMM}}$ to $\mathcal{L}^{\mathrm{SMM}}$, while keeping all other components of the code identical.

2.3 Effectiveness metric

In this section, we discuss the effectiveness metrics that we use to compare GMM/SMM implemented for the basic structural model in Section 2. Our goal is to assess how well each method

1. recovers the true parameters in controlled Monte Carlo experiments, and
2. reproduces key features of the data and delivers reliable predictions in settings where the true parameters are unknown.

To this end, we employ three complementary metrics. The first two rely on knowledge of the true parameter vector and are therefore used only in Monte Carlo simulations. The last one does *not* require knowledge of the true parameters and can be applied both in simulation studies and with real data.

Notation. Let $\vartheta_0 \in \mathbb{R}^p$ denote the true parameter vector used to generate data in the Monte Carlo experiments. For each estimator $j \in \{\text{GMM}, \text{SMM}\}$ and each Monte Carlo replication $m = 1, \ldots, M$, let $\hat{\vartheta}_j^{(m)}$ be the corresponding estimate. We use $\hat{\vartheta}_j$ (without superscript) to denote the estimate obtained from a single dataset.

Metric 1: Bias, standard deviation, and RMSE of estimates (requires true parameters)

The first metric evaluates the finite-sample accuracy of each estimator by comparing its estimates to the known true parameter vector ϑ_0 across Monte Carlo replications. For each component $k = 1, \ldots, p$ of the parameter vector, define the Monte Carlo bias, standard deviation (SD), and root mean squared error (RMSE) for estimator j as

$$\text{Bias}_j(\vartheta_k) = \frac{1}{M} \sum_{m=1}^{M} \left(\hat{\vartheta}_{j,k}^{(m)} - \vartheta_{0,k} \right), \tag{49}$$

$$\text{SD}_j(\vartheta_k) = \sqrt{\frac{1}{M-1} \sum_{m=1}^{M} \left(\hat{\vartheta}_{j,k}^{(m)} - \overline{\hat{\vartheta}}_{j,k} \right)^2}, \tag{50}$$

$$\text{RMSE}_j(\vartheta_k) = \sqrt{\frac{1}{M} \sum_{m=1}^{M} \left(\hat{\vartheta}_{j,k}^{(m)} - \vartheta_{0,k} \right)^2}, \tag{51}$$

where

$$\overline{\hat{\vartheta}}_{j,k} = \frac{1}{M} \sum_{m=1}^{M} \hat{\vartheta}_{j,k}^{(m)}$$

is the Monte Carlo mean of the k-th parameter estimate under method j.

These quantities allow us to compare the bias–variance trade-off of the estimators considered in Section 3.2 (e.g. GMM and SMM) in a controlled setting where ϑ_0 is known:

- Lower absolute bias indicates that an estimator is more accurate on average.
- Lower SD indicates that an estimator is more stable across samples.
- Lower RMSE combines these two aspects into a single measure of overall finite-sample accuracy.

This metric is only meaningful in Monte Carlo experiments, where the true parameter vector ϑ_0 is known by construction.

Metric 2: Coverage of confidence intervals

The second metric evaluates how well each method quantifies sampling uncertainty about the parameters. For a given nominal confidence level $1-\alpha$ (e.g. 95%), each estimator j delivers, in each Monte Carlo replication m, a confidence interval for each parameter component ϑ_k:

$$\big[\underline{\vartheta}_{j,k}^{(m)},\, \overline{\vartheta}_{j,k}^{(m)}\big].$$

The Monte Carlo coverage probability for parameter ϑ_k under estimator j is defined as

$$\mathrm{Cov}_j(\vartheta_k) \;=\; \frac{1}{M}\sum_{m=1}^{M} \mathbf{1}\Big\{\vartheta_{0,k} \in \big[\underline{\vartheta}_{j,k}^{(m)},\, \overline{\vartheta}_{j,k}^{(m)}\big]\Big\}, \tag{52}$$

where $\mathbf{1}\{\cdot\}$ denotes the indicator function.

If an estimator produces well-calibrated intervals, we expect $\mathrm{Cov}_j(\vartheta_k)$ to be close to the nominal level $1-\alpha$ for all k. Deviations from the nominal coverage provide information about:

- *Undercoverage* (coverage below $1-\alpha$): intervals are too narrow or centered incorrectly, leading to overconfident inference.
- *Overcoverage* (coverage above $1-\alpha$): intervals are too wide or overly conservative.

This metric is again only defined when ϑ_0 is known, and is therefore used in Monte Carlo experiments to compare the reliability of the uncertainty quantification across the estimators considered in previous section.

Metric 3: Fit of non-targeted (auxiliary) moments

In addition to the $q = 13$ targeted moments used in the GMM/SMM objective in Section 3.2, we assess the broader empirical fit of the model using a set of *auxiliary* (non-targeted) moments. These are economically relevant features of the data (e.g., autocorrelations, cross-moments, distributional statistics) that are not explicitly targeted in estimation. Metric 3 compares how well different estimators reproduce such non-targeted features of the *real* data (synthetic in our case).

More generally, let $\{h_\ell(\cdot)\}_{\ell=1}^{L}$ denote a collection of auxiliary moment functions that are not included in the GMM/SMM objective, and define for a generic panel y the auxiliary moment vector

$$a(y) \;=\; \big(h_1(y), \ldots, h_L(y)\big)' \in \mathbb{R}^L.$$

In our baseline application we take $L = 6$ and specify $a(y)$ as

$$a(y) \;=\; \begin{pmatrix} \mathbb{E}[\iota_t] \\ \sqrt{\mathrm{Var}[\iota_t]} \\ \mathbb{E}[z_t] \\ \sqrt{\mathrm{Var}[\ln z_t]} \\ \mathrm{corr}(\ln k_t, z_t) \\ \mathrm{corr}(\ln k_t, \ln k_{t-1}) \end{pmatrix},$$

For each Monte Carlo replication r we observe a data panel y_r^{data} (the "real" data, synthetic in our case). For estimator $j \in \{\mathrm{GMM}, \mathrm{SMM}\}$ let $\hat{\vartheta}_{j,r}$ denote the parameter estimate obtained from y_r^{data}. Using the observed data, we compute the auxiliary moments

$$\mu_r^{\mathrm{data}} \;=\; a\big(y_r^{\mathrm{data}}\big),$$

and, holding $\hat{\vartheta}_{j,r}$ fixed, we simulate a long artificial panel $y^{\text{sim}}_{j,r} = \text{sim}(\hat{\vartheta}_{j,r})$ from the structural model and compute the corresponding model-implied auxiliary moments

$$\mu^{\text{model}}_{j,r} \;=\; a\big(y^{\text{sim}}_{j,r}\big).$$

Let $\mu^{\text{data}}_{r,\ell}$ and $\mu^{\text{model}}_{j,r,\ell}$ denote the ℓ-th components of these vectors, $\ell = 1, \ldots, L$.

For each auxiliary moment ℓ and estimator j we define the (scaled) moment discrepancy in replication r as

$$\Delta_{j,r,\ell} \;=\; \mu^{\text{model}}_{j,r,\ell} - \mu^{\text{data}}_{r,\ell}, \qquad \widetilde{\Delta}_{j,r,\ell} \;=\; \frac{\Delta_{j,r,\ell}}{s_{r,\ell}},$$

where $s_{r,\ell}$ is a scaling factor used to make discrepancies across heterogeneous moments comparable. In our implementation we set

$$s_{r,\ell} \;=\; \max\{|\mu^{\text{data}}_{r,\ell}|,\, 10^{-6}\},$$

so that

$$\widetilde{\Delta}_{j,r,\ell} \;=\; \frac{\mu^{\text{model}}_{j,r,\ell} - \mu^{\text{data}}_{r,\ell}}{\max\{|\mu^{\text{data}}_{r,\ell}|,\, 10^{-6}\}}.$$

An overall scalar measure of auxiliary-moment fit for estimator j in replication r is then

$$D^{\text{aux}}_{j,r} \;=\; \frac{1}{L}\sum_{\ell=1}^{L} \big|\widetilde{\Delta}_{j,r,\ell}\big| \quad \text{or} \quad D^{\text{aux}}_{j,r} \;=\; \frac{1}{L}\sum_{\ell=1}^{L} \big(\widetilde{\Delta}_{j,r,\ell}\big)^2. \tag{53}$$

Smaller values of $D^{\text{aux}}_{j,r}$ indicate that estimator j yields parameter estimates $\hat{\vartheta}_{j,r}$ under which the model reproduces the auxiliary features of the data reasonably well. We report the distribution of $D^{\text{aux}}_{j,r}$ across replications (e.g. its mean and quantiles) for each estimator j as Metric 3, providing an out-of-sample check of the model's ability to match empirically relevant moments that were not directly used in the GMM/SMM estimation.

2.4 Extension: Bayesian Estimation via Hamiltonian Monte Carlo(Bonus Q1)

In previous section we developed *frequentist* estimators for the basic investment model of Strebulaev Section 3.1 using GMM and SMM. Roughly speaking, those methods try to pick parameter values so that certain sample moments (e.g. means, variances, autocorrelations) implied by the model match the corresponding moments in the data.

In this section we instead take a *Bayesian* point of view. In a Bayesian approach:

- we treat the model parameters ϑ as unknown random variables;
- we encode our beliefs about plausible parameter values in a *prior distribution*;
- we combine the prior with the likelihood of the observed data to obtain a *posterior* distribution for ϑ;
- we then summarize or use this full posterior (e.g. posterior means, credible intervals, posterior predictive simulations).

A key feature of the Bayesian estimator developed here is its close alignment with the *full structural model*. Specifically, both productivity z_t and the true capital stock k_t are treated as

latent (unobserved) states. The model imposes the endogenous capital dynamics implied by the optimal policy, where capital evolves according to the structural law of motion:

$$k_{t+1} = \varphi(k_t, z_t; \vartheta).$$

This formulation is expressed in **state-space** form, and a **nonlinear filtering** method is employed to compute an approximation to the likelihood based on the synthetic data and a candidate parameter vector ϑ. The likelihood is then combined with a **prior** to construct the posterior distribution of ϑ. To sample from this **posterior**, we use **Hamiltonian Monte Carlo** (HMC), a Markov chain Monte Carlo (MCMC) algorithm that leverages gradient information to efficiently explore moderate- to high-dimensional parameter spaces (see, e.g., Neal, 2011[50]).

State-Space Formulation and Likelihood

Our next step is to rewrite the basic investment model in a way that is convenient for filtering and likelihood evaluation. The natural framework for this is a state-space model.

- The *state equation* describes how the unobserved (latent) variables evolve over time.
- The *measurement equation* describes how the observed data are generated from the latent states, plus measurement noise.

Productivity dynamics

We begin with productivity. Recall that productivity $z_t > 0$ follows an AR(1) process in logs:

$$\ln z_{t+1} = \rho \ln z_t + \sigma_\varepsilon \varepsilon_{t+1}, \qquad \varepsilon_{t+1} \sim \mathcal{N}(0,1), \tag{54}$$

with $|\rho| < 1$ and $\sigma_\varepsilon > 0$.

It is often convenient to work directly with log-productivity:

$$x_t := \ln z_t.$$

Then the productivity dynamics can be written as

$$x_{t+1} = \rho x_t + \sigma_\varepsilon \varepsilon_{t+1}, \tag{55}$$

which is a standard linear Gaussian state equation.

Observed variables and measurement error

For each firm i and time period t we assume:

- Productivity z_{it} (or equivalently $x_{it} = \ln z_{it}$) is *not* directly observed.
- The firm's capital stock is measured with noise: the data report some k_{it}^{obs}, which is only an imperfect measure of the true capital stock k_{it}.

The observables are

$$\{k_{it}^{\text{obs}},\ y_{it}\},$$

where:

- k_{it}^{obs} is the reported beginning-of-period capital;
- y_{it} is the log of observed profits.

The model's production function is

$$\pi(k_t, z_t) = z_t k_t^{\theta}.$$

Hence, ignoring measurement noise for a moment, the *true* log profits would be

$$\ln \pi(k_t, z_t) = \ln z_t + \theta \ln k_t = x_t + \theta \ln k_t.$$

To allow for noise in observed profits, we write:

$$y_{it} = \ln \pi_{it}^{\text{obs}} = x_{it} + \theta \ln k_{it} + \sigma_y \eta_{it}, \qquad \eta_{it} \sim \mathcal{N}(0,1), \tag{56}$$

where:

- k_{it} is the *latent true* capital stock;
- $\sigma_y > 0$ is the standard deviation of the profit measurement error.

Next we connect latent capital to the observed capital data. In logs we assume:

$$\ln k_{it}^{\text{obs}} = \ln k_{it} + \sigma_k^{\text{obs}} \xi_{it}, \qquad \xi_{it} \sim \mathcal{N}(0,1), \tag{57}$$

with $\sigma_k^{\text{obs}} > 0$. This specification means that observed capital is a noisy *multiplicative* version of true capital, which is often realistic in practice (e.g. reporting errors or book value vs. true economic value).

Including measurement error serves two purposes:

- It is empirically plausible that capital data are noisy.
- It avoids degeneracies in the likelihood: if capital were assumed to be observed exactly, then combining this with a deterministic capital transition could lead to a nearly singular likelihood.

Structural capital dynamics

The main structural restriction is that capital evolves *endogenously* through the firm's optimal investment decision $k_{t+1} = \varphi(k_t, z_t; \vartheta)$. To use this in a probabilistic state-space model, we need a transition *density*, that is, a probability distribution for k_{t+1} given (k_t, z_t). If we used a purely deterministic policy $k_{t+1} = \varphi(\cdot)$ with no noise, and combined it with measurement error in k_t, the likelihood might become numerically unstable.

To deal with this, we add a small Gaussian "decision shock" in logs:

$$\ln k_{t+1} = \ln \varphi\big(k_t, e^{x_t}; \vartheta\big) + \sigma_k \nu_{t+1}, \qquad \nu_{t+1} \sim \mathcal{N}(0,1), \tag{58}$$

with $\sigma_k > 0$. Interpretation:

- The main part of the decision k_{t+1} comes from the optimal policy $\varphi(k_t, z_t; \vartheta)$.
- The small shock $\sigma_k \nu_{t+1}$ allows for small deviations around this policy. These deviations can represent unmodeled frictions, approximation error in $\widehat{\varphi}$, or small optimization mistakes by firms.
- Mathematically, adding this noise ensures that k_{t+1} has a non-degenerate density, which improves numerical stability when computing the likelihood.

Single-firm nonlinear state-space system

For notational simplicity, consider a single firm and drop the index i. Define the two-dimensional latent state vector:

$$s_t := \begin{pmatrix} x_t \\ \ln k_t \end{pmatrix}.$$

Using the previous equations, our nonlinear state-space model becomes:

$$x_{t+1} = \rho x_t + \sigma_\varepsilon \varepsilon_{t+1}, \qquad \varepsilon_{t+1} \sim \mathcal{N}(0,1), \tag{59}$$

$$\ln k_{t+1} = \ln \varphi\big(k_t, e^{x_t}; \vartheta\big) + \sigma_k \nu_{t+1}, \qquad \nu_{t+1} \sim \mathcal{N}(0,1), \tag{60}$$

$$y_t = x_t + \theta \ln k_t + \sigma_y \eta_t, \qquad \eta_t \sim \mathcal{N}(0,1), \tag{61}$$

$$\ln k_t^{\text{obs}} = \ln k_t + \sigma_k^{\text{obs}} \xi_t, \qquad \xi_t \sim \mathcal{N}(0,1), \tag{62}$$

for $t = 1, \ldots, T$. All shocks $(\varepsilon_t, \nu_t, \eta_t, \xi_t)$ are assumed independent over time and across equations, conditional on past states.

Equations (59)–(62) define a *nonlinear* but conditionally Gaussian state-space model:

- Conditional on current states, the next state and current observations are Gaussian, because all shocks are Gaussian.
- The nonlinearity comes from the policy function $\varphi(\cdot)$ inside (60).

Because of this nonlinearity, the exact Kalman filter (which assumes that both the state equation and the measurement equation are linear in the state) is no longer valid without approximation. We therefore need a *nonlinear filtering* method.

Likelihood via nonlinear filtering

Let

$$o_t := \big(y_t, \ln k_t^{\text{obs}}\big)$$

denote the vector of observables at time t for a single firm. The (approximate) likelihood of the data for that firm, given parameters ϑ, can be written as

$$p(o_{1:T} \mid \vartheta) = \prod_{t=1}^{T} p(o_t \mid o_{1:t-1}, \vartheta),$$

where $p(o_t \mid o_{1:t-1}, \vartheta)$ is the *predictive density* of o_t given past observations and the parameter vector ϑ.

Computing these predictive densities exactly is not feasible here because of the nonlinear capital transition in (60). Instead, we approximate them using a nonlinear filtering algorithm. In what follows we focus on the *unscented Kalman filter* (UKF), which approximates the distribution of s_t by a Gaussian and propagates it through the nonlinear transition using a small deterministic set of "sigma points". We discuss the UKF in more detail in Subsection 2.4 and in the algorithmic steps below.

Panel extension

Now consider a panel of firms $i = 1, \ldots, N$. We assume that productivity innovations are independent across firms, conditional on parameters. Under this assumption:

- the likelihood for the panel is the product of firm-specific likelihoods;
- equivalently, the log-likelihood is the sum of firm-specific log-likelihoods.

Hence,

$$\log p\big(\{o_{it}\}_{i,t} \mid \vartheta\big) \;=\; \sum_{i=1}^{N} \log p\big(o_{i,1:T} \mid \vartheta\big), \tag{63}$$

where $o_{it} = (y_{it}, \ln k_{it}^{\text{obs}})$.

This (approximated) panel likelihood is what we use in the Bayesian estimator.

Prior Specification

In the Bayesian framework we must specify a *prior distribution* $p(\vartheta)$ for the parameters, reflecting our beliefs about plausible values before seeing the data. The data will then update this prior to a posterior distribution.

We choose *weakly informative* priors: these priors restrict parameters to plausible regions (e.g.uniform).

For simplicity, we assume the prior factorizes across parameters (i.e. they are independent under the prior). As in the main text, we treat the discount rate r and depreciation rate δ as fixed at calibrated values.

Posterior Distribution and HMC Sampling

Given the prior $p(\vartheta)$ and the likelihood $p(\text{data} \mid \vartheta)$ (computed via nonlinear filtering as in (63)), Bayes' rule yields the posterior:

$$p(\vartheta \mid \text{data}) \;\propto\; p(\text{data} \mid \vartheta)\, p(\vartheta), \tag{64}$$

where "data" denotes the panel $\{k_{it}^{\text{obs}}, y_{it}\}_{i=1,\ldots,N;\, t=1,\ldots,T}$.

In most realistic models (including ours), the posterior density does not have a simple closed form. We therefore approximate it by drawing samples using Markov chain Monte Carlo (MCMC).

Hamiltonian Monte Carlo (HMC). HMC is an MCMC method that uses gradient information to propose efficient moves in parameter space. Very roughly:

- Standard random-walk Metropolis algorithms propose small random steps in parameter space and often mix slowly, especially when the posterior is high-dimensional or highly correlated.
- HMC instead introduces an auxiliary "momentum" variable and simulates a *Hamiltonian dynamical system* that approximately follows the gradient of the log-posterior. This allows the chain to make larger, well-informed moves while maintaining a high acceptance rate.
- Adaptive variants such as NUTS automatically choose the number of steps taken in each HMC trajectory, reducing manual tuning.

To apply HMC we need:

- a log-posterior function

$$\ell(\vartheta) := \log p(\vartheta \mid \text{data}) = \log p(\text{data} \mid \vartheta) + \log p(\vartheta)$$

  that can be evaluated deterministically for each ϑ;

- the gradient of $\ell(\vartheta)$ with respect to ϑ, which we typically obtain using automatic differentiation tools.

The UKF-based likelihood satisfies these requirements as long as our policy approximation $\widehat{\varphi}(\cdot;\vartheta)$ is differentiable in both the states and parameters.

Filtering Choice and Alternatives

We now briefly discuss the choice of nonlinear filter and why the UKF is well suited to our Bayesian HMC approach.

Chosen filter: Unscented Kalman Filter (UKF). We use the UKF to approximate the likelihood in (63). The UKF is designed for models where:

- the state-space dynamics and/or measurement equations are nonlinear;
- the process and measurement noise are Gaussian.

The main idea of the UKF is:

- Approximate the filtering distribution of the state s_t by a Gaussian with mean m_t and covariance P_t.
- Represent this Gaussian not by random samples, but by a small deterministic set of points ("sigma points") chosen to match the mean and covariance exactly.
- Propagate these sigma points through the nonlinear transition and measurement functions to approximate the mean and covariance of the resulting distributions.

Advantages of the UKF.

- *Compatibility with the structural model.* The UKF can handle the nonlinear policy function $k_{t+1} = \varphi(k_t, z_t; \vartheta)$ directly, without requiring a first-order Taylor approximation (unlike the Extended Kalman Filter).
- *Deterministic and differentiable.* Given parameters and data, the UKF outputs a deterministic approximate log-likelihood. The operations involved (matrix square roots, linear algebra, evaluation of $\widehat{\varphi}$) are differentiable, so we can compute gradients of the log-likelihood with respect to ϑ using automatic differentiation. This is essential for HMC.
- *Computationally feasible.* Our state vector is low-dimensional ($(x_t, \ln k_t)$), so the number of sigma points is small, and the UKF is computationally cheap per time step.

Limitations of the UKF.

- *Approximation error.* Because the UKF approximates the filtering distribution by a Gaussian, it may introduce bias if the true filtering distribution is very non-Gaussian (e.g. strongly skewed or multi-modal).
- *Problems near boundaries or kinks.* In models with occasionally binding constraints (e.g. irreversibility, borrowing constraints) or other nonlinearities that create kinks, the Gaussian approximation can perform poorly.

Alternative filters.

- *Extended Kalman Filter (EKF).* The EKF linearizes the nonlinear functions using first-order Taylor expansions (Jacobian matrices). It can be faster but may be less accurate and less stable for highly nonlinear dynamics.
- *Particle filters (sequential Monte Carlo).* Particle filters can in principle provide a very accurate (even asymptotically exact) approximation to the likelihood in nonlinear, non-Gaussian models. However:
  - they produce a *stochastic* estimate of the likelihood;
  - standard HMC requires a *deterministic* log-posterior; using noisy likelihood estimates breaks its theoretical guarantees.

  In those cases one usually turns to particle MCMC methods (like particle marginal Metropolis–Hastings or particle Gibbs) rather than HMC.

Given that we want a deterministic, differentiable likelihood that respects the nonlinear structural law of motion for capital, the UKF is a natural choice to combine with HMC.

Implementation: UKF likelihood and HMC

Step 1: Differentiable policy approximation

Solve the dynamic programming problem of Section 3.1 to obtain a differentiable, parametrized (amortized) policy

$$\widehat{\varphi}(k, z; \vartheta).$$

Step 2: Unconstrained parameterization

Work on $\mathbb{R}^7$ via

$$(\theta^*, \rho^*, \log \sigma_\varepsilon, \log \sigma_y, \log \sigma_k, \log \sigma_k^{\text{obs}}, \log \phi) \in \mathbb{R}^7,$$

and set

$$\begin{aligned} \theta &= \sigma(\theta^*) \in (0,1), \quad \sigma(u) = \frac{1}{1+e^{-u}}, \\ \rho &= \tanh(\rho^*) \in (-1,1), \\ \sigma_\varepsilon &= e^{\log \sigma_\varepsilon}, \quad \sigma_y = e^{\log \sigma_y}, \quad \sigma_k = e^{\log \sigma_k}, \\ \sigma_k^{\text{obs}} &= e^{\log \sigma_k^{\text{obs}}}, \quad \phi = e^{\log \phi}. \end{aligned}$$

Rewrite the priors (**??**)–(**??**) in terms of the unconstrained variables, adding the usual log-Jacobian terms for logistic and exponential transformations.

Step 3: UKF-based likelihood for a single firm

Given ϑ:

1. *Observed series:*
$$o_t = \big(y_t, \ln k_t^{\text{obs}}\big), \qquad t = 1, \dots, T.$$

2. *Initial state:* let $s_1 = (x_1, \ln k_1)$ with mean m_1 and covariance P_1, e.g.
$$\mathbb{E}[x_1] = 0, \qquad \mathbb{E}[\ln k_1] = \ln k_1^{\text{obs}},$$
and take P_1 with large diagonal elements.

3. *Noise covariances:*
$$Q = \begin{pmatrix} \sigma_\varepsilon^2 & 0 \\ 0 & \sigma_k^2 \end{pmatrix}, \qquad R = \begin{pmatrix} \sigma_y^2 & 0 \\ 0 & (\sigma_k^{\text{obs}})^2 \end{pmatrix}.$$

4. *For $t = 1, \dots, T$ (standard UKF recursion):*

   (a) From (m_t, P_t), construct sigma points $\{s_t^{(j)}\}$ using UKF parameters (α, β, κ).

   (b) *Measurement update:*
   - Propagate sigma points through the measurement mapping $h(\cdot)$ (equations (61)–(62)) with zero noise to obtain $h(s_t^{(j)})$.
   - Compute predicted mean $\hat{o}_t$ and covariance S_t (plus R), and the state–measurement cross-covariance.
   - Form Kalman gain K_t, update
   $$m_t^{\text{post}} = m_t + K_t(o_t - \hat{o}_t), \quad P_t^{\text{post}} = P_t - K_t S_t K_t^\top.$$
   - Add $\log \mathcal{N}(o_t \mid \hat{o}_t, S_t)$ to the running log-likelihood.

   (c) *State prediction:*
   - Generate sigma points from $(m_t^{\text{post}}, P_t^{\text{post}})$.
   - Propagate them through the nonlinear transition (59)–(60) with policy $\widehat{\varphi}$ and zero process noise.
   - Take their mean and covariance as (m_{t+1}, P_{t+1}) and add Q.

5. Summing the log Gaussian predictive densities yields $\log p(o_{1:T} \mid \vartheta)$ for the firm.

Step 4: Panel likelihood

For firms $i = 1, \dots, N$ with series $\{o_{it}\}_{t=1}^T$,

$$\log p\big(\{o_{it}\}_{i,t} \mid \vartheta\big) = \sum_{i=1}^N \log p\big(o_{i,1:T} \mid \vartheta\big),$$

where each $\log p(o_{i,1:T} \mid \vartheta)$ is computed by the UKF in Step 3.

Step 5: Log-posterior and gradient

1. Map the unconstrained parameter vector to ϑ (Step 2).
2. Compute $\log p(\vartheta)$ with Jacobian terms.
3. Compute $\log p(\text{data} \mid \vartheta)$ via Steps 3–4.
4. Form
$$\ell(\vartheta) = \log p(\vartheta \mid \text{data}) = \log p(\text{data} \mid \vartheta) + \log p(\vartheta).$$
5. Use automatic differentiation (e.g. Stan, JAX, TFP) to obtain $\nabla_{\text{unconstr}}\ell$.

Step 6: Hamiltonian Monte Carlo

1. *Initialization:* choose starting values (e.g. from GMM/SMM or calibration); optionally run multiple chains.
2. *HMC dynamics:* at each iteration,
   - draw a momentum vector (e.g. $\mathcal{N}(0, I)$);
   - simulate leapfrog steps using $\nabla\ell$;
   - accept/reject via the Metropolis rule based on the Hamiltonian change.
3. *Adaptation (warm-up):* tune step size (target acceptance ≈ 0.65–0.8) and optionally a mass matrix; NUTS can also adapt the number of leapfrog steps.
4. *Sampling:* fix tuned hyperparameters and run the chain to obtain posterior draws; summarize with posterior moments, intervals, and posterior predictive quantities.

Step 7: Practical considerations

- *Vectorization/parallelism:* UKF evaluations are independent across firms and can be vectorized or parallelized.
- *Numerical stability:* use stable linear algebra (e.g. Cholesky), add small diagonal jitter to near-singular covariance matrices.
- *Runtime:* each HMC iteration runs the UKF over T for all N firms; this is feasible for moderate panels and low-dimensional states with efficient code and modern hardware.

Comparison with GMM and SMM

- *Structural discipline:* the likelihood enforces the endogenous capital law of motion implied by the optimal policy, using more of the model's structure than moment-based estimators.
- *Uncertainty quantification:* the posterior delivers probability statements about parameters and functions of them, and naturally propagates parameter uncertainty into counterfactuals.
- *Efficiency vs. sensitivity:* when the model and UKF approximation are adequate, Bayesian likelihood-based methods can be more efficient than GMM/SMM, but they are more sensitive to distributional assumptions and filtering/policy approximations.

- *Computational cost:* filtered likelihood evaluation within HMC is typically more expensive per iteration than moment evaluation in GMM/SMM, but remains tractable for moderate panels. According to our experiments, the run time for the three methods under our design is rank as: GMM<SMM<HMC+UKF.

2.5 Results and Discussion: Comparison of GMM, SMM, and HMC–UKF Bayesian Estimators

This section summarizes and compares the finite-sample performance of three estimators for the structural parameters (θ, ϕ) in the Monte Carlo design described previously: (i) a two-step Euler-equation GMM estimator, (ii) a two-step simulation-based SMM estimator, and (iii) a Bayesian Hamiltonian Monte Carlo estimator with an unscented Kalman filter (HMC–UKF) approximation to the likelihood. All results are based on $R = 100$ independent replications. For each replication we estimate (θ, ϕ) by all three methods and compute three performance metrics: point estimation (Metric 1), interval coverage (Metric 2), and auxiliary-moment fit (Metric 3). We also report standard overidentification diagnostics for the two classical (GMM and SMM) estimators. The remaining tuning and prior parameters, including those specific to the HMC–UKF implementation, are documented in the configuration file.

Point estimation (Metric 1)

Tables 3 and 4 report the Monte Carlo distributions of the three estimators.

Table 3: Metric 1: Monte Carlo performance of GMM and SMM estimators

	True value	Mean	Bias	RMSE
		GMM		
θ	0.7000	0.6998	-2.0×10^{-4}	5.0×10^{-4}
ϕ	2.0000	1.9847	-1.53×10^{-2}	4.84×10^{-2}
		SMM		
θ	0.7000	0.7000	≈ 0	2.0×10^{-4}
ϕ	2.0000	2.0001	1.0×10^{-4}	4.6×10^{-3}

Table 4: Metric 1: Monte Carlo performance of the HMC–UKF estimator

	True value	Mean	Bias	RMSE
θ	0.7000	0.7000	≈ 0	≈ 0
ϕ	2.0000	2.0060	6.0×10^{-3}	1.42×10^{-2}

For θ, all three estimators exhibit negligible bias. The GMM bias is -0.0002, the SMM bias is numerically zero, and the HMC posterior mean is numerically indistinguishable from the true value across replications. SMM is about 2.5 times more precise than GMM, with an RMSE of 0.0002 compared to 0.0005 for GMM. For HMC, the posterior standard deviation and RMSE for θ are below the reporting precision, indicating an extremely tight posterior and very high precision, comparable to or better than SMM in this design.

For ϕ, the contrast among estimators is sharper. GMM displays a small but non-negligible negative bias (-0.0153) and an RMSE of 0.0484. SMM is essentially unbiased with an RMSE of 0.0046, reducing the RMSE for ϕ by roughly an order of magnitude relative to GMM. The HMC–UKF estimator is also highly accurate: the average posterior mean is 2.006, implying a small upward bias of 0.006 and an RMSE of about 0.014. Thus, for ϕ the GMM estimator is least precise, SMM attains the lowest RMSE, and HMC–UKF lies in between but still delivers quite precise estimates.

Taken together, these results indicate that all three procedures are well-centered in finite samples, with SMM and HMC–UKF substantially more efficient than Euler-equation GMM, especially for ϕ. This is not surprising: the amortized policy is an approximation to the true solution of the model, and in SMM we use this policy both to generate synthetic data and to simulate the model, thereby exploiting a rich set of carefully chosen panel moments. By contrast, the Euler-equation GMM estimator relies on a more parsimonious set of moment conditions. A plausible interpretation is that SMM makes more efficient use of the information in the data, while the HMC–UKF estimator achieves high precision by combining the amortized policy with an approximate (UKF-based) likelihood.

Interval coverage (Metric 2)

Metric 2 evaluates the frequentist coverage of nominal 95% intervals. For GMM we construct 95% confidence intervals using cluster-robust sandwich standard errors with optimal weighting. For SMM we use a delta-method approximation based on the two-step optimal-weighting objective, with bootstrap and simulation-based estimates of the moment covariance matrix. For HMC–UKF we construct nominal 95% Bayesian credible intervals from the posterior draws.

The empirical coverage rates are:

$$\Pr\big(\theta_0 \in CI_{0.95}^{\mathrm{GMM}}\big) \approx 0.89, \qquad \Pr\big(\phi_0 \in CI_{0.95}^{\mathrm{GMM}}\big) \approx 0.89,$$

$$\Pr\big(\theta_0 \in CI_{0.95}^{\mathrm{SMM}}\big) \approx 0.93, \qquad \Pr\big(\phi_0 \in CI_{0.95}^{\mathrm{SMM}}\big) \approx 0.94,$$

$$\Pr\big(\theta_0 \in \mathrm{CrI}_{0.95}^{\mathrm{HMC}}\big) \approx 1.00, \qquad \Pr\big(\phi_0 \in \mathrm{CrI}_{0.95}^{\mathrm{HMC}}\big) \approx 0.88.$$

For GMM, both parameters exhibit under-coverage relative to the 95% nominal level, by about 6 percentage points. This suggests that the asymptotic normal approximation—combined with the finite-sample behavior of the clustered sandwich estimator and the chosen instruments—leads to confidence intervals that are somewhat too narrow in this design.

For SMM, the reported coverage rates are much closer to the nominal 95% level, with only mild under-coverage. Given that the SMM variance estimates combine a nonparametric bootstrap over firms with a Monte Carlo estimate of the simulation variance, this pattern suggests that the delta-method approximation is performing well and that the optimal weighting matrix is accurately estimated. Remaining discrepancies from exact 95% coverage are plausibly due to the moderate number of replications ($R = 100$), simulation noise in the SMM objective, and the linearization inherent in the delta method.

For HMC–UKF, the 95% credible intervals for θ are slightly conservative in the frequentist sense, covering the true value in essentially all replications. This is consistent with the extremely tight posterior for θ: small changes in the likelihood are enough to keep the true value within the high-probability region. For ϕ, the empirical coverage of about 88% falls modestly below the nominal 95% level, indicating some under-coverage: the posterior credible intervals for ϕ are a bit too narrow in finite samples. Possible contributors include the UKF approximation to the latent-state likelihood, the relatively sharp prior bounds implemented via the sigmoid reparametrization, and the fact

that ϕ is less tightly identified than θ in the simulated design. Nevertheless, the under-coverage is moderate, and the credible intervals still contain the true ϕ in the vast majority of replications.

Comparing across methods, SMM delivers the most accurate frequentist calibration overall, with coverage rates closest to 95% for both parameters. GMM and HMC–UKF both display some under-coverage for ϕ, while GMM also under-covers for θ and HMC–UKF slightly over-covers for θ.

Auxiliary-moment fit (Metric 3)

Metric 3 assesses how well the estimated models replicate a set of auxiliary moments that are not directly targeted in the GMM, SMM, or HMC likelihood/objective functions: the mean of z_t, the standard deviation of $\ln z_t$, the correlation between $\log k_t$ and z_t, and the autoregressive coefficient of $\log k_t$. For each replication and each estimator, we compute a scaled distance D_{aux} between the auxiliary moments in the data and those implied by simulating the model at the point estimates (for GMM and SMM) or at the HMC posterior mean, and then average D_{aux} across replications.

The mean auxiliary-moment distances are

$$\overline{D}_{\text{aux}}^{\text{GMM}} = 0.0094, \qquad \overline{D}_{\text{aux}}^{\text{SMM}} = 0.0089, \qquad \overline{D}_{\text{aux}}^{\text{HMC}} = 0.0080.$$

All three numbers are very small, indicating that the estimated models reproduce the low-dimensional distributional properties of (k_t, z_t) quite closely, including both first and second moments and dynamic correlations. Typical scaled deviations for individual moments (the components $d_1, \ldots, d_4$) are on the order of 0.5–3% in absolute value. The SMM estimator attains slightly lower average discrepancy than GMM, which is consistent with its smaller parameter RMSEs, while the HMC–UKF estimator achieves the smallest average auxiliary-moment distance among the three. Overall, however, the differences across methods are minor, and all three approaches match the auxiliary moments quite well.

Overidentification tests and Q–Q diagnostics

We next consider tests of the overidentifying restrictions and the associated Q–Q diagnostics for the GMM and SMM estimators. (Overidentification tests are not directly applicable to the HMC–UKF Bayesian estimator, which is based on an approximate likelihood rather than a system of sample moment conditions.)

GMM. The Hansen J-test for the Euler-equation GMM estimator uses $q = 7$ moment conditions and $p = 2$ parameters, so that the test statistic J_N should be approximately χ^2_5 under the null of correct specification. Across replications the rejection rate at the 5% level is 3%, somewhat below the nominal size. Consistently, the mean of $N \times J_N$ is 3.288, compared to the theoretical χ^2_5 mean of 5.0, and the empirical standard deviation (2.871) is also slightly below the χ^2_5 standard deviation of about 3.16. The reported quantiles confirm that the distribution of $N \times J_N$ is stochastically dominated by χ^2_5.

Figure 9 plots the empirical quantiles of $N \times J_N$ against Monte Carlo draws from the χ^2_5 distribution. The points lie close to the 45-degree line at the lower end but fall systematically below it for moderate and large quantiles, reflecting the conservative behavior of the J-test in this design. This pattern could arise from several sources—for example, weak or highly collinear instruments, small effective sample size at the firm level, or finite-sample bias in the clustered sandwich variance estimator—but without additional diagnostics we cannot distinguish among these explanations.

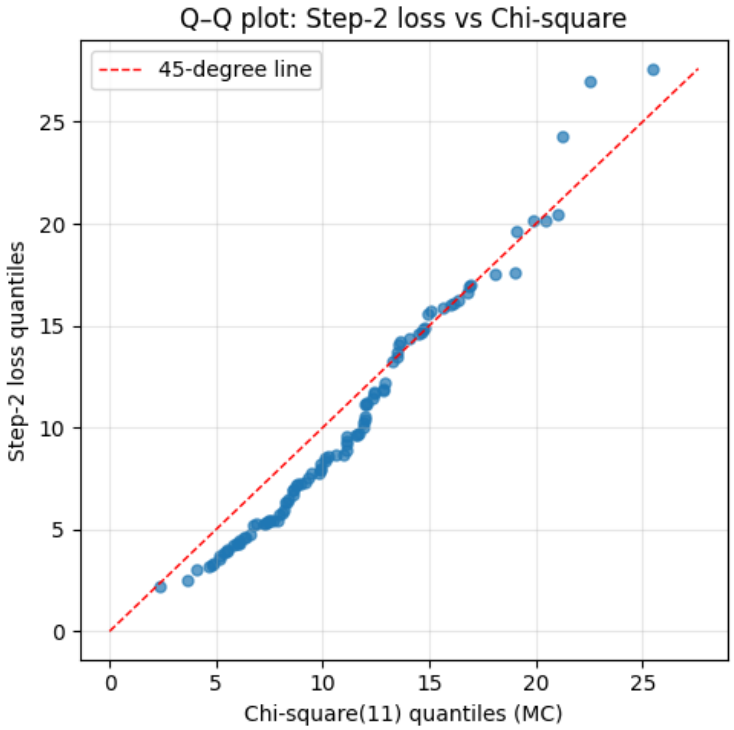

Figure 8: Q–Q plot of the SMM step-2 loss against the χ^2_{11} reference distribution. Each point corresponds to a replication, plotting the empirical quantile of the minimized SMM objective against the corresponding Monte Carlo quantile from χ^2_{11}. The dashed red line is the 45-degree line.

SMM. For SMM, the two-step objective function at the optimum should be approximately χ^2_{q-p} with $q = 13$ moments and $p = 2$ parameters, so that the step-2 loss is asymptotically χ^2_{11}. The reported rejection rate of the SMM J-test at the 5% level is 9%, modestly above nominal. The mean of the step-2 loss (9.901) is slightly below the theoretical χ^2_{11} mean of 11, while the empirical standard deviation (5.513) is somewhat larger than the theoretical value $\sqrt{2 \times 11} \approx 4.69$. The quantiles show small deviations but broadly track the χ^2_{11} reference distribution.

Figure 8 displays the Q–Q plot of the SMM step-2 loss against a simulated χ^2_{11} distribution. The points lie close to the 45-degree line over most of the support, with mild deviations in the extreme right tail. This visual impression accords with the numerical diagnostics: the SMM loss behaves very much like a chi-square random variable with the appropriate degrees of freedom, but there is some evidence of slightly heavier upper tails, which helps explain the modest over-rejection of the SMM J-test.

Comparison. Overall, both classical estimators pass the overidentification diagnostics comfortably. The GMM J-test is mildly conservative, whereas the SMM J-test slightly over-rejects. In light of the very small parameter biases and the good auxiliary-moment fit for both estimators, these deviations from nominal size are plausibly attributable to finite-sample effects rather than gross misspecification of the structural model.

Overall assessment. Across the three metrics, the estimators paint a coherent picture. SMM and HMC–UKF deliver essentially unbiased and highly precise point estimates, particularly for

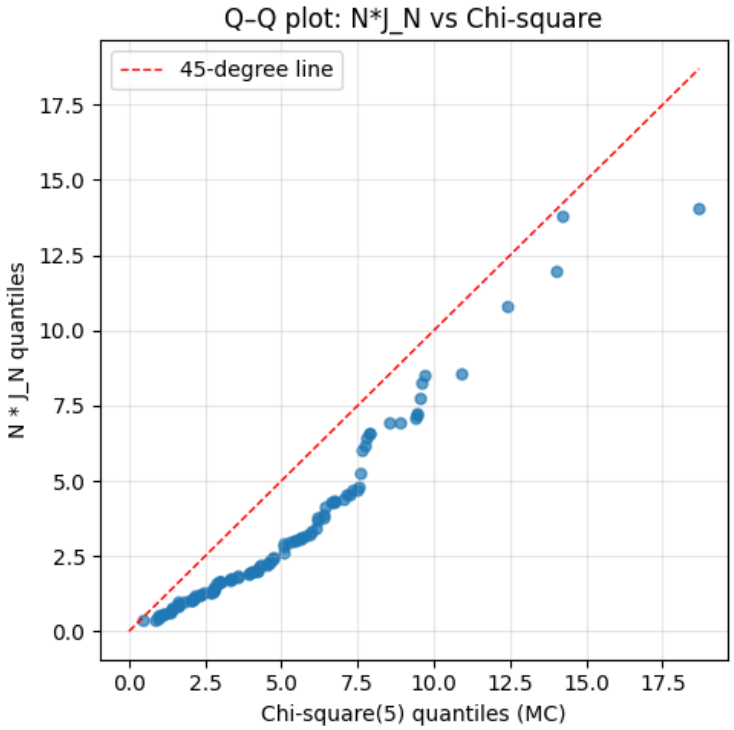

Figure 9: Q–Q plot of $N \times J_N$ for the Euler-equation GMM estimator against the χ^2_5 reference distribution. Each point corresponds to a replication, plotting the empirical quantile of $N \times J_N$ against the Monte Carlo quantile from χ^2_5. The dashed red line is the 45-degree line.

θ, whereas GMM is less efficient, especially for ϕ. SMM attains the most accurate frequentist coverage, while HMC–UKF produces well-calibrated (if slightly conservative) credible intervals for θ and only moderate under-coverage for ϕ. All three estimators match untargeted auxiliary moments very closely, with the HMC–UKF estimator performing best in terms of the average auxiliary-moment distance but with only small differences relative to GMM and SMM. Taken together, these findings suggest that, in this setting, Bayesian estimation with an amortized policy and a UKF-based likelihood is a reliable and informative complement to the classical GMM and SMM procedures.

3 Future Work

1. **Model Complexity**
   In our current work, we focus on solving and estimating relatively simple structural corporate finance models. These implementations primarily serve illustrative purposes. Future research could delve into more complex models that incorporate additional terms in the cash flow identity and exhibit more intricate structures etc. Furthermore, the methodological framework could be expanded by exploring alternative Markov Chain Monte Carlo (MCMC) techniques and filtering methods beyond Hamiltonian Monte Carlo (HMC) and the Unscented Kalman Filter (UKF).

2. **Machine Learning Perspectives**
   The deep learning approach adopted in this work shares similarities with methods used in deep reinforcement learning (DRL), particularly in the use of policy functions and value functions.

It would be interesting to explore the applicability of DRL algorithms to the models considered in this study. For example, replacing the AIO loss formalism with an actor-critic framework could be some direction to try out.

3. **Real-World Applications**
   The current implementation is primarily theoretical and based on synthetic data. An important future direction would be to apply the model to real-world firm-level data. This would allow us to evaluate the model's practical relevance in realistic settings and assess how effectively it can support real-life decision-making in corporate finance contexts.

Acknowledgements

I would like to thank JPMorgan MLCOE team for designing such challenging and engaging projects. It has been a truly rewarding experience, and I've learned a great deal throughout the process.

References

[1] Jr. Lucas, Robert E. Asset prices in an exchange economy. *Econometrica*, 46(6):1429–1446, 1978.

[2] Jr. Lucas, Robert E. and Edward C. Prescott. Investment under uncertainty. *Econometrica*, 39(5):659–681, September 1971.

[3] Ilya A Strebulaev, Toni M Whited, et al. Dynamic models and structural estimation in corporate finance. *Foundations and Trends in Finance*, 6(1–2):1–163, 2012.

[4] George Tauchen. Finite state markov-chain approximations to univariate and vector autoregressions. *Economics Letters*, 20(2):177–181, 1986.

[5] George Tauchen and Robert Hussey. Quadrature-based methods for obtaining approximate solutions to nonlinear asset pricing models. *Econometrica*, 59(2):371–396, 1991.

[6] Ronald A. Howard. *Dynamic Programming and Markov Processes*. MIT Press, Cambridge, MA, 1960.

[7] John Rust. Numerical dynamic programming in economics. In H. M. Amman, David A. Kendrick, and John Rust, editors, *Handbook of Computational Economics*, volume 1, pages 619–729. Elsevier, Amsterdam, 1996.

[8] Mario J. Miranda and Paul L. Fackler. *Applied Computational Economics and Finance*. MIT Press, Cambridge, MA, 2002.

[9] Nancy L. Stokey, Jr. Lucas, Robert E., and Edward C. Prescott. *Recursive Methods in Economic Dynamics*. Harvard University Press, Cambridge, MA, 1989.

[10] II Coleman, Wilbur John. Solving the stochastic growth model by policy-function iteration. *Journal of Business Economic Statistics*, 8(1):27–29, January 1990.

[11] II Coleman, Wilbur John. Equilibrium in a production economy with an income tax. *Econometrica*, 59(4):1091–1104, July 1991.

[12] Christopher D. Carroll. The method of endogenous gridpoints for solving dynamic stochastic optimization problems. *Economics Letters*, 91(3):312–320, June 2006.

[13] Francisco Barillas and Jesus Fernandez-Villaverde. A generalization of the endogenous grid method. *Journal of Economic Dynamics and Control*, 31(8):2698–2712, August 2007.

[14] Lawrence J. Christiano and Jonas D. M. Fisher. Algorithms for solving dynamic models with occasionally binding constraints. *Journal of Economic Dynamics and Control*, 24(8):1179–1232, July 2000.

[15] Giulio Fella. A generalized endogenous grid method for non-smooth and non-concave problems. *Review of Economic Dynamics*, 17(2):329–344, April 2014.

[16] Kenneth L. Judd. Projection methods for solving aggregate growth models. *Journal of Economic Theory*, 58(2):410–452, December 1992.

[17] Kenneth L. Judd. *Numerical Methods in Economics*. MIT Press, Cambridge, MA, 1998.

[18] S. Borağan Aruoba, Jesús Fernández-Villaverde, and Juan F. Rubio-Ramírez. Comparing solution methods for dynamic equilibrium economies. *Journal of Economic Dynamics and Control*, 30(12):2477–2508, December 2006.

[19] Kenneth L. Judd, Lilia Maliar, Serguei Maliar, and Rafael Valero. Smolyak method for solving dynamic economic models: Lagrange interpolation, anisotropic grid and adaptive domain. *Journal of Economic Dynamics and Control*, 44:92–123, 2014.

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

[21] Nathalie Moyen. Investment–cash flow sensitivities: Constrained versus unconstrained firms. *Journal of Finance*, 59(5):2061–2092, 2004.

[22] Christopher A. Hennessy and Toni M. Whited. How costly is external financing? evidence from a structural estimation. *The Journal of Finance*, 62(4):1705–1745, August 2007.

[23] Sheridan Titman and Sergey Tsyplakov. A dynamic model of optimal capital structure. *Review of Finance*, 11(3):401–451, 2007.

[24] Hayne E. Leland and Klaus Bjerre Toft. Optimal capital structure, endogenous bankruptcy, and the term structure of credit spreads. *Journal of Finance*, 51(3):987–1019, 1996.

[25] Andrea Gamba and Alexander Triantis. The value of financial flexibility. *Journal of Finance*, 63(5):2263–2296, 2008.

[26] Ronald W. Anderson and Suresh Sundaresan. Design and valuation of debt contracts. *Review of Financial Studies*, 9(1):37–68, 1996.

[27] Hua Fan and Suresh M. Sundaresan. Debt valuation, renegotiation, and optimal dividend policy. *Review of Financial Studies*, 13(4):1057–1099, 2000.

[28] Lars Peter Hansen. Large sample properties of generalized method of moments estimators. *Econometrica*, 50(4):1029–1054, July 1982.

[29] Whitney K. Newey and Daniel McFadden. Large sample estimation and hypothesis testing. In Robert F. Engle and Daniel McFadden, editors, *Handbook of Econometrics*, volume 4, chapter 36, pages 2111–2245. Elsevier, Amsterdam, 1994.

[30] Lars Peter Hansen and Kenneth J. Singleton. Generalized instrumental variables estimation of nonlinear rational expectations models. *Econometrica*, 50(5):1269–1286, September 1982.

[31] Toni M. Whited. Debt, liquidity constraints, and corporate investment: Evidence from panel data. *The Journal of Finance*, 47(4):1425–1460, September 1992.

[32] Stephen R. Bond and Costas Meghir. Dynamic investment models and the firm's financial policy. *Review of Economic Studies*, 61(2):197–222, April 1994.

[33] Inessa Love. Financial development and financing constraints: International evidence from the structural investment model. *Review of Financial Studies*, 16(3):765–791, July 2003.

[34] Laura Xiaolei Liu, Toni M. Whited, and Lu Zhang. Investment-based expected stock returns. *Journal of Political Economy*, 117(6):1105–1139, 2009.

[35] Qiang Kang, Qiao Liu, and Rong Qi. The sarbanes-oxley act and corporate investment: A structural assessment. *Journal of Financial Economics*, 96(2):291–305, 2010.

[36] Ariel Pakes and David Pollard. Simulation and the asymptotics of optimization estimators. *Econometrica*, 57(5):1027–1057, September 1989.

[37] Darrell Duffie and Kenneth J. Singleton. Simulated moments estimation of markov models of asset prices. *Econometrica*, 61(4):929–952, July 1993.

[38] Christian Gourieroux, Alain Monfort, and Eric Renault. Indirect inference. *Journal of Applied Econometrics*, 8(S1):S85–S118, 1993.

[39] V. Joseph Hotz and Robert A. Miller. Conditional choice probabilities and the estimation of dynamic models. *Review of Economic Studies*, 60(3):497–529, July 1993.

[40] V. Joseph Hotz, Robert A. Miller, Seth Sanders, and Jeffrey Smith. A simulation estimator for dynamic models of discrete choice. *Econometrica*, 62(5):827–862, September 1994.

[41] Christopher A. Hennessy and Toni M. Whited. Debt dynamics. *The Journal of Finance*, 60(3):1129–1165, June 2005.

[42] Harry DeAngelo, Linda DeAngelo, and Toni M. Whited. Capital structure dynamics and transitory debt. *Journal of Financial Economics*, 99(2):235–261, 2011.

[43] Rui Albuquerque and Enrique Schroth. Quantifying private benefits of control from a structural model of block trades. *Journal of Financial Economics*, 96(1):33–55, 2010.

[44] Shaojin Li, Toni M. Whited, and Yufeng Wu. Collateral, taxes, and leverage. *Journal of Financial Economics*, 121(1):1–22, 2016.

[45] Toni M. Whited. Debt, liquidity constraints, and corporate investment: Evidence from panel data. *The Journal of Finance*, 47(4):1425–1460, September 1992.

[46] Stephen Bond and Costas Meghir. Dynamic investment models and the firm's financial policy. *The Review of Economic Studies*, 61(2):197–222, April 1994.

[47] Lars Peter Hansen. Large sample properties of generalized method of moments estimators. *Econometrica*, 50(4):1029–1054, July 1982.

[48] Ariel Pakes and David Pollard. Simulation and the asymptotics of optimization estimators. *Econometrica*, 57(5):1027–1057, September 1989.

[49] Darrell Duffie and Kenneth J. Singleton. Simulated moments estimation of markov models of asset prices. *Econometrica*, 61(4):929–952, July 1993.

[50] Radford M. Neal. Mcmc using hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo*, pages 113–162. Chapman and Hall/CRC, 2011.