Discrete-Continuous Dynamic Choice: A Soft-to-Hard Reinforcement Learning Framework with an Application to Sovereign Default*

Bo Li[a], Serguei Maliar[b,c], Peiyuan Zhang[a]

[a] *School of Economics, Peking University, Beijing, China*
[b] *Department of Economics, Santa Clara University, USA*
[c] *Hoover Institution, Stanford University, USA*

Abstract

Structural smoothing techniques like auxiliary income or taste shocks are widely used to resolve non-differentiable boundaries in discrete-continuous macroeconomic models, but they can distort economic predictions and introduce numerical instability. We propose a soft-to-hard (StH) reinforcement learning framework that solves the exact, unsmoothed economy. The algorithm initializes on a smoothed approximation but systematically anneals this smoothing to zero during training, enforcing exact hard-switch Bellman operators at convergence. We provide two complementary StH implementations: tabular Q-learning (StH-Q), which delivers high numerical precision in low-dimensional settings, and neural actor-critic (StH-AC), which ensures scalability in high-dimensional environments. Applied to a multi-period sovereign default model, StH recovers the exact equilibrium where permanent smoothing falters.

Keywords: Discrete-continuous choice, Sovereign default, Long-term debt, Reinforcement learning, Actor–critic, Computational economics
JEL classification: C61, C63, C68, F34

*This research was supported by NSF grant SES-1949430, National Natural Science Foundation of China (No. 72450002), and Beijing Philosophy and Social Science Foundation (No. 24DTR024).

Email addresses: libo.econ@pku.edu.cn (Bo Li), maliars@stanford.edu (Serguei Maliar), zpyuan@pku.edu.cn (Peiyuan Zhang)

Working paper, October 24, 2026

1 Introduction

Some of the most consequential decisions in economics are discrete leaps: a worker entering unemployment, a household buying a home, a firm exiting a market, or a sovereign nation defaulting on its debt. When these qualitative leaps are intertwined with continuous quantitative decisions—such as consumption, investment, or borrowing—they form the discrete-continuous dynamic choice (DCDC) models that serve as the workhorse of modern quantitative economics. However, embedding these dual decision margins introduces profound computational friction. Discrete choice thresholds fracture value functions, spawning non-differentiable kinks that shatter global concavity. As these kinks propagate exponentially during backward induction, they paralyze standard calculus-based optimization, trigger severe numerical pathologies, and push traditional computational algorithms to the brink of intractability.

To solve DCDC models, the literature largely pursues two grid-based methodological tracks. The first handles non-convexities exactly using geometric filtering (Iskhakov et al., 2017; Druedahl, 2021) and constrained optimization (Su and Judd, 2012). While these approaches solve the unsmoothed economy, tracking complex non-differentiable boundaries across dense grids is computationally demanding. The second track achieves numerical tractability by permanently smoothing the environment via auxiliary income shocks (Chatterjee and Eyigungor, 2012) or extreme-value taste shocks (Rust, 1987; Gordon, 2019; Dvorkin et al., 2021; Mihalache, 2024). While this smoothing ensures differentiability and faster convergence, it fundamentally alters the target economic environment and risks distorting quantitative predictions. Furthermore, because both tracks traditionally rely on state-space discretization, they remain highly vulnerable to the curse of dimensionality.

We propose a scalable soft-to-hard (StH) reinforcement learning (RL) framework that solves the exact, unsmoothed economy. Methodologically, our approach bridges classical structural microeconometrics (McFadden, 1974) with maximum-entropy reinforcement learning (Haarnoja et al., 2018). While the econometric literature traditionally uses the log-sum-exp operator to model permanent Type-I extreme value preference shocks, StH repurposes this mathematical structure—which directly mirrors soft actor-critic value functions—as a purely temporary computational regularization device. By systematically annealing the smoothing parameter to zero during training, StH uses soft choice probabilities to preserve gradient flow across non-differentiable decision boundaries before converging precisely on the exact, hard-switch Bellman operators.

We deliver two complementary StH implementations that fundamentally advance reinforcement learning for discrete-continuous economic environments. First, StH-Q extends classical model-based tabular RL (e.g., Dyna-Q; Sutton, 1991) by integrating a novel dual-temperature annealing mechanism directly with co-evolving general equilibrium operators, eliminating persistent exploratory noise to achieve exact grid-level precision. Second, StH-AC adapts deep neural actor-critic architectures (Sutton and Barto, 2018) to overcome the vanishing gradient bottleneck inherent in hard decision boundaries. By embedding a continuous gradient penetration mechanism—rather than relying on decoupled auxiliary classifiers—StH-AC preserves exact structural payoffs and guarantees scalable convergence in high-dimensional spaces without permanent smoothing biases.

We evaluate the framework across two demanding economic environments. First, in the canonical long-term sovereign default model of Chatterjee and Eyigungor (2012)—a laboratory combining non-convex default choices, continuous borrowing, and recursive bond pricing schedules—we compare four distinct computational approaches: two traditional permanent smoothing benchmarks (auxiliary income shocks and extreme-value taste shocks) alongside our two soft-to-hard implementations (StH-Q and StH-AC). Second, to evaluate scalability, we apply the neural StH-AC method to the endogenous-capital economy of Gordon and Guerrón-Quintana (2018). Across these environments, the numerical results highlight four main findings:

1. *The limits of baseline consistency:* While all methods produce similar quantitative predictions under standard calibrations, heightened economic volatility severely amplifies the structural distortions introduced by traditional smoothing.

2. *Algorithmic breakdown of limit recovery:* Shrinking traditional smoothing parameters toward zero fails to recover the exact hard-switch economy. Instead, it triggers fixed-point non-convergence, numerical instability, and severe prediction swings (e.g., mean debt ratios fluctuating between 16.9% and 22.7%).

3. *Exact grid fidelity (StH-Q):* Tabular StH-Q delivers the most precise solution of all methods considered, solving the exact unsmoothed economy in just three minutes with zero default-rule inconsistency and Bellman residuals on the order of 10^{-6}.

4. *High-dimensional scalability (StH-AC):* Neural StH-AC successfully eliminates permanent structural bias in our baseline model, maintaining an exceptionally

low decision inconsistency of 0.065% without relying on grid-based discretization. Crucially, it efficiently handles the more challenging endogenous-capital economy, closely matching the dense-grid benchmark in just 2% of the computing time.

The remainder of the paper is organized as follows. Section 2 formulates the general DCDC framework and establishes the theoretical foundation of the StH solver. Section 3 presents the sovereign default application and reviews benchmark smoothing methods. Section 4 details the StH-Q and StH-AC implementations. Section 5 compares the numerical performance of the smoothing and StH methods across multi-period debt environments. Section 6 presents the endogenous-capital application, and finally, Section 7 concludes.

2 A Soft-to-Hard (StH) solver for DCDC environments

In this section, we formally define a generic DCDC environment, construct the soft-to-hard theoretical framework, and establish its key analytical properties. We then present two complementary operational algorithms: StH-Q, which leverages tabular action values and model-based planning for grid-level accuracy in lower-dimensional settings, and StH-AC, which uses deep neural function approximation to resolve non-differentiable bottlenecks in high-dimensional state spaces.

2.1 Generic DCDC environment and the StH principle

Consider an infinite-horizon, discrete-time Markov decision process indexed by period $t = 0, 1, 2, \ldots, \infty$. Let Γ represent endogenous equilibrium objects (such as pricing functions, market-clearing conditions, or wage distributions) that are updated iteratively until equilibrium consistency is achieved.

State and action spaces. At the beginning of period t, an agent observes a state vector $s \in S$, containing all payoff-relevant endogenous state variables and exogenous stochastic drivers. Conditional on s, the agent selects a discrete choice d and a continuous action x:
$\star$ *Discrete action* (d): A qualitative selection from a finite set $D = \{0, 1, \ldots, K-1\}$. To keep notation concise, we detail the standard binary choice case where $d \in \{0, 1\}$.
$\star$ Continuous action (x): A quantitative choice drawn from a compact, continuous nonempty feasible set $X(s, d) \subseteq \mathbb{R}^n$, which may depend on both the current state and the chosen discrete action. We assume that the branch maxima are finite and attained.

Preferences and dynamics. The agent earns immediate payoff $U(s, d, x)$ and discounts future rewards by $\beta \in (0, 1)$. The state transitions according to a Markov probability kernel $P(s' \mid s, d, x)$, governing the probability density of next period's state s'.

The non-convexity bottleneck. The overall value function $V(s)$ represents the upper envelope of the choice-specific value functions (branches) $V^d(s)$:

$$V(s) = \max_{d \in \{0,1\}} \{V^d(s)\}, \tag{1}$$

where each branch value solves the continuous-choice problem:

$$V^d(s) = \max_{x \in X(s,d)} \left\{ U(s, d, x) + \beta \mathbb{E}_{s'|s,d,x}[V(s')] \right\}. \tag{2}$$

Even when individual branch value functions $V^0(s)$ and $V^1(s)$ are strictly concave and well-behaved, the hard upper envelope operator in Equation (1) introduces non-differentiable kinks across state boundaries where $V^0(s) = V^1(s)$. In multi-period settings, evaluating expectations over future kinked value functions destroys global concavity and halts standard gradient-based optimization algorithms.

The smooth continuation operator. This is a critical step of StH analysis: we replace the hard envelope in Equation (1) with a smooth, temperature-controlled log-sum-exp continuation operator $C_\tau(s)$:

$$C_\tau(s) = \tau \log \left[\exp\left(\frac{V^1(s)}{\tau} \right) + \exp\left(\frac{V^0(s)}{\tau} \right) \right], \tag{3}$$

where $\tau > 0$ denotes a temperature parameter controlling the sharpness of the transition. Equation (3) induces a soft choice probability for selecting discrete branch $d = 1$:

$$p^1_\tau(s) = \frac{1}{1 + \exp\left(-\frac{V^1(s) - V^0(s)}{\tau} \right)}. \tag{4}$$

From an algorithmic standpoint, this formulation aligns the discrete choice probabilities of structural econometrics (McFadden, 1974; Rust, 1987; Berry et al., 1995) with the soft Bellman updates of continuous control (Haarnoja et al., 2018). The key distinction lies in the conceptual role of the temperature parameter τ: rather than treating smoothing as a fixed structural primitive of preferences or market friction, StH treats τ as a transient continuation parameter. This allows the solver to leverage soft optimization dynamics during early iterations before restoring the exact, unsmoothed Bellman system at convergence.

Soft computation, hard refinement, and evaluation. The StH workflow proceeds in two distinct stages followed by exact evaluation:

1. *Soft Regularization Stage:* Value, policy, and equilibrium functions are updated using $C_\tau(s)$ with $\tau > 0$. Temperature annealing $\tau \to \tau_{\text{final}}$ smooths non-differentiable kinks, allowing value information and gradients to flow freely across choice boundaries.

2. *Hard Refinement Stage:* The temperature parameter is set strictly to zero ($\tau = 0$), collapsing $C_\tau(s)$ back to the exact hard upper envelope:

$$C_{\text{hard}}(s) = \max\{V^1(s), V^0(s)\}, \quad p^1_{\text{hard}}(s) = \mathbf{1}\{V^1(s) \geq V^0(s)\}. \tag{5}$$

   Updates continue until value functions, continuous policies, and general equilibrium objects Γ converge under the exact hard-switch operator.

This sequence guarantees that first-stage softness serves strictly as an algorithmic warm-start mechanism, while final outputs remain entirely free from structural smoothing bias.

Mathematical properties. The convergence and gradient transmission properties of $C_\tau(s)$ are governed by two formal propositions (proofs provided in Appendix A):

Proposition 1. (Gradient Penetration) Given the soft continuation operator $C_\tau(s)$ with temperature $\tau > 0$, the partial derivatives with respect to the underlying branch values satisfy:

$$\frac{\partial C_\tau(s)}{\partial V^1(s)} = p^1_\tau(s), \quad \frac{\partial C_\tau(s)}{\partial V^0(s)} = 1 - p^1_\tau(s). \tag{6}$$

Implication: While the hard maximum operator places full weight (1) on the winning branch and zero weight (0) on the losing branch, Proposition 1 confirms that for any $\tau > 0$, the losing branch retains a strictly positive analytical derivative ($\min\{p^1_\tau(s), 1 - p^1_\tau(s)\} > 0$). This allows numerical solvers to update policies in non-dominant branches based on local state information.

Proposition 2. (Hard-Operator Recovery Bound) For any state s, finite branch values $(V^0(s), V^1(s))$, and temperature $\tau > 0$, $C_\tau(s)$ is bounded by:

$$\max\{V^1(s), V^0(s)\} \leq C_\tau(s) \leq \max\{V^1(s), V^0(s)\} + \tau \log 2. \tag{7}$$

Furthermore, as $\tau \downarrow 0$, $C_\tau(s)$ converges uniformly to $\max\{V^1(s), V^0(s)\}$, and $p^1_\tau(s)$ converges pointwise to $\mathbf{1}\{V^1(s) \geq V^0(s)\}$ everywhere away from exact ties.

Implication: The structural approximation error induced by log-sum-exp smoothing is strictly bounded by $\tau \log 2$ and vanishes at rate $\mathcal{O}(\tau)$, ensuring exact recovery of the original hard Bellman equation during hard refinement.

2.2 StH-Q: tabular action-value learning

StH-Q unites classical grid-based dynamic programming with model-based tabular reinforcement learning (e.g., Dyna-Q; Sutton, 1991; Sutton and Barto, 2018). Traditional quantitative macroeconomics solves discrete-continuous models via grid-based value iteration, whereas the RL literature leverages experience replay and background planning ("offline dreaming") to accelerate value propagation. Relative to standard Dyna-style algorithms—which assume exogenous Markov environments and rely on heuristic ε-greedy exploration—StH-Q introduces a novel dual-temperature annealing mechanism integrated directly with co-evolving general equilibrium pricing operators. In this implementation, temporary soft choice rules generate robust initial estimates of the coupled value and pricing functions; experience replay and model-based planning then iteratively update tabulated action values. By systematically extinguishing both action-level and branch-level smoothing, StH-Q achieves exact hard-switch operator recovery on discrete state-action grids, providing an ultra-precise numerical reference for lower-dimensional environments.

Algorithmic formalism. StH-Q operationalizes the StH principle on finite discrete grids $S_h \subset S$ and $X_h \subset X$. It stores action values $Q^1(s,x)$ for branch 1 and branch values $V^0(s)$ for branch 0, assuming continuous action choices are active in branch 1.

At iteration n, given sampled state–action pairs (s,x), the target and temporal-difference (TD) updates are:

$$Y^n(s,x) = U(s,1,x) + \beta \mathbb{E}_{s'|s,1,x}\left[C^n_{\tau_n}(s')\right], \tag{8}$$

$$Q^{1,n+1}(s,x) = Q^{1,n}(s,x) + \alpha_Q\left[Y^n(s,x) - Q^{1,n}(s,x)\right], \tag{9}$$

$$V^{1,n}(s) = \max_{x \in X_h(s,1)} Q^{1,n}(s,x), \quad x^n(s) \in \arg\max_{x \in X_h(s,1)} Q^{1,n}(s,x), \tag{10}$$

where $\alpha_Q \in (0,1]$ is the learning rate and $C^n_{\tau_n}(s')$ evaluates Equation (3) during soft training ($\tau_n > 0$) and Equation (5) during hard refinement ($\tau_n = 0$).

Integration with model-based planning and experience replay. To execute offline dreaming, StH-Q queries known state-transition structures and stochastic shock distributions during background planning steps. The algorithm conducts hypothetical off-policy sweeps across previously visited and simulated state-action

trajectories, propagating continuation value estimates efficiently across the state-action lattice without requiring additional online environment evaluations.

Algorithmic innovations over standard tabular RL. While building on foundational Dyna-Q principles, StH-Q introduces three structural features necessary for dynamic economic environments:

1. *Dual-temperature annealing:* Standard RL algorithms rely on static exploratory rules (ε-greedy or fixed softmax distributions) that leave persistent noise in learned policy rules. StH-Q implements a dual-temperature schedule: an action-level temperature $\eta > 0$ smooths continuous grid selection via $V_\eta^1(s) = \eta \log \sum_{x \in X_h} \exp\left(Q^1(s,x)/\eta\right)$, while a branch-level temperature $\tau > 0$ smooths the discrete envelope. Annealing both parameters $(\eta, \tau) \to 0$ eliminates exploratory noise and recovers exact policy maximizers.

2. *Co-evolution of general equilibrium operators:* Unlike standard Dyna frameworks designed for stationary MDPs with fixed reward rules, StH-Q integrates endogenous general equilibrium operators Γ directly into the offline planning loop. As value tables update, equilibrium objects (such as asset pricing schedules or wage rules) are re-estimated concurrently, maintaining general equilibrium consistency.

3. *Discrete-continuous hybrid envelope:* Instead of treating actions as purely discrete or purely continuous, StH-Q applies dense tabular row updates over discrete choice branches alongside exact grid maximization over continuous action sets $X_h(s,d)$.

2.3 StH-AC: continuous deep neural actor-critic

StH-AC implements the soft-to-hard principle within a deep neural actor-critic architecture (Sutton and Barto, 2018). Optimizing neural policies end-to-end in these environments presents a fundamental bottleneck: continuous decisions dictate state transitions, but hard upper-envelope operators create non-differentiable switching thresholds. StH-AC eliminates this bottleneck by substituting the hard upper envelope with the differentiable continuation operator $C_\tau(s)$ during neural training, guaranteeing smooth gradient transmission across decision boundaries. As the temperature vanishes, StH-AC refines the coupled policy, value, and pricing networks strictly under exact hard-switch operators, providing a scalable, high-dimensional complement to StH-Q.

Network architecture and bounded parameterization. StH-AC separates the functional components into four neural networks:

$$x_\omega(s), \quad V_\theta^1(s), \quad V_\psi^0(s), \quad \Gamma_\varphi(s,x), \tag{11}$$

where x_ω is the continuous actor conditional on $d=1$, V_θ^1 and V_ψ^0 represent the choice-specific critic networks, and Γ_φ approximates general equilibrium expectations. We parameterize $x_\omega(s)$ using bounded activation functions (such as a scaled sigmoid σ):

$$x_\omega(s) = X_{\max}\sigma(f_\omega(s)), \tag{12}$$

ensuring proposed actions lie strictly within valid domain bounds $[0, X_{\max}]$.

Critic losses and equilibrium learning. The branch critics are optimized by minimizing mean-squared Bellman errors over sampled state batches:

$$V^1(s) = U^1(s, x_\omega(s)) + \beta\mathbb{E}_{s'|s,d=1}[C_\tau(s')], \tag{13}$$
$$V^0(s) = U^0(s) + \beta\mathbb{E}_{s'|s,d=0}[C_\tau(s')], \tag{14}$$

with critic regression losses given by:

$$\mathcal{L}_{V^1} = \mathbb{E}\left[\left(V_\theta^1(s) - V^1(s)\right)^2\right], \quad \mathcal{L}_{V^0} = \mathbb{E}\left[\left(V_\psi^0(s) - V^0(s)\right)^2\right]. \tag{15}$$

Equilibrium network Γ_φ is trained concurrently by replacing hard discrete indicators in target calculations with soft choice probabilities $p_\tau^1(s')$. Target network parameters are held fixed (detached) during regression updates to stabilize neural training.

Actor optimization via gradient penetration. Define the actor's objective function for branch 1 at state s as:

$$V_\omega^1(s) = U^1(s, x_\omega(s)) + \beta\mathbb{E}_\varepsilon[C_\tau(s')], \tag{16}$$

where $s' = F(s, 1, x_\omega(s), \varepsilon)$ represents the state transition function.

The operational distinction between hard training and soft training lies in the continuation derivative. Under exact hard execution, the continuation gradient with respect to policy parameters ω exhibits a step discontinuity:

$$\nabla_\omega C_{\text{hard}}(s'_\omega) = \begin{cases} \nabla_\omega V^1(s'_\omega), & V^1(s'_\omega) \geq V^0(s'_\omega), \\ \nabla_\omega V^0(s'_\omega), & V^0(s'_\omega) > V^1(s'_\omega). \end{cases} \tag{17}$$

Equation (17) shows that if branch 1 is locally non-dominant ($V^0 > V^1$), the gradient of V^1 with respect to the continuous action is completely severed ($\nabla_\omega C_{\text{hard}} = \nabla_\omega V^0$).

During soft training ($\tau > 0$), Proposition 1 ensures that derivative transmission varies smoothly:

$$\nabla_\omega C_\tau(s'_\omega) = p^1_\tau(s'_\omega)\nabla_\omega V^1(s'_\omega) + \left[1 - p^1_\tau(s'_\omega)\right] \nabla_\omega V^0(s'_\omega). \tag{18}$$

Differentiating the actor objective in Equation (16) yields:

$$\nabla_\omega V^1_\omega(s) = \nabla_\omega U^1(s, x_\omega(s)) + \beta \mathbb{E}_\varepsilon \left[\nabla_\omega C_\tau(s'_\omega)\right]. \tag{19}$$

Because $p^1_\tau(s'_\omega) \in (0, 1)$, Equation (19) guarantees non-zero gradient flow across decision boundaries, enabling the continuous actor to improve branch policies even in regions where the branch is currently out-of-the-money.

Feasibility penalties and loss function. To handle economic constraints $g(s, x) \leq 0$ (such as budget balance or non-negative consumption) during early training, we introduce a quadratic feasibility penalty:

$$\text{pen}_\omega(s) = \max\{g(s, x_\omega(s)), 0\}^2. \tag{20}$$

The total actor loss function combines structural payoffs with feasibility enforcement:

$$\mathcal{L}_{\text{actor}} = -\mathbb{E}_s \left[U^1_{\text{safe}}(s, x_\omega(s)) + \beta \mathbb{E}_\varepsilon[C_\tau(F(s, 1, x_\omega(s), \varepsilon))]\right] + \lambda_{\text{feas}} \mathbb{E}_s \left[\text{pen}_\omega(s)\right], \tag{21}$$

where U^1_{safe} represents a floored utility evaluation that prevents numerical overflow on infeasible trial actions, and $\lambda_{\text{feas}} > 0$ controls penalty weight.

Annealing schedule and hard refinement. Soft training uses a geometric temperature schedule across training epochs $t \in [0, T_{\text{ann}}]$:

$$\tau(t) = \tau_{\text{init}} \left(\frac{\tau_{\text{final}}}{\tau_{\text{init}}}\right)^{\min\{1, t/T_{\text{ann}}\}}, \tag{22}$$

with $0 < \tau_{\text{final}} \leq \tau_{\text{init}}$. The temperature remains at τ_{final} until soft training ends. The solver then enters the hard refinement stage ($\tau = 0$), re-coupling the actor x_ω, critics V^1_θ, V^0_ψ, and equilibrium network Γ_φ strictly under the exact hard operators in Equation (5). The hard-refinement block reconciles the actor, critics, and equilibrium network with this operator before evaluation and reporting. Evaluation uses the exact hard rule in Equation (5).

Algorithmic innovations over existing deep RL methods. StH-AC embeds the soft-to-hard principle within a deep neural actor-critic architecture (Sutton and Barto, 2018), parameterizing policies, values, and prices as continuous functions. Because adding a state variable simply enlarges the network's input layer rather than exponentially expanding a tensor-product grid, StH-AC naturally bypasses the curse of dimensionality. Complementing StH-Q's role in smaller benchmarks, StH-AC introduces two structural innovations for discrete-continuous models:

1. *Extending standard deep RL to discrete choice:* Most deep learning applications in economics focus exclusively on continuous choice environments (Maliar et al., 2021; Han et al., 2026; Azinovic et al., 2022; Fernández-Villaverde et al., 2023); see Fernández-Villaverde (2026) for a recent review. Standard continuous RL algorithms fail in discrete-continuous settings because hard upper-envelope operators create non-differentiable thresholds. If a discrete branch evaluates as suboptimal during early training, it receives zero gradient. This severed derivative flow traps the algorithm, preventing the network from learning valid policies in non-dominant branches. StH-AC's continuation operator eliminates this bottleneck by preserving continuous gradient flow across all decision boundaries, forcing the network to learn globally valid policies.

2. *Unified structural payoffs vs. auxiliary classification:* To bypass non-differentiability, Maliar and Maliar (2022) frame discrete decisions as a classification task using an auxiliary cross-entropy head. While stabilizing, this decouples choice probabilities from structural value functions, risking classification bias. StH-AC eliminates auxiliary classifiers entirely. By embedding the annealing temperature τ directly into the continuation values, discrete and continuous choices remain tightly coupled through the Bellman equations, ensuring the final hard-refined solution exactly matches the unsmoothed economic environment.

3 The baseline sovereign default environment

While the StH methodology is broadly applicable to any dynamic environment featuring joint discrete and continuous choices, we evaluate its performance in the rigorous testing ground of sovereign default. Pioneered in short-term debt settings by Eaton and Gersovitz (1981) and Arellano (2008), and extended to long-term debt frameworks by Hatchondo and Martinez (2009), Sánchez et al. (2014), and Chatterjee and Eyigungor (2012) (hereafter CE2012), these models pair a hard binary decision to repay or default with continuous debt issuance and non-linear, recursive bond pricing. This intersection of non-differentiable decision thresholds and endogenous asset pricing makes sovereign default an ideal benchmark for the StH framework. Indeed, these severe computational frictions have historically driven the literature to bypass dense grids altogether by adopting continuous-time formulations (Aguiar and Amador, 2020) or random maturity structures (Arellano and Ramanarayanan, 2012).

We deploy our framework across two distinct applications within this literature. Our primary benchmark is the canonical CE2012 long-term debt model.

To rigorously evaluate the exact StH solution in this baseline setting, we compare our results against two traditional smoothing paradigms—auxiliary income shocks and extreme-value taste shocks—that the field has long relied upon for numerical tractability. Moving beyond the baseline, Section 6 introduces our second application: the endogenous-capital economy of Gordon and Guerrón-Quintana (2018). This more demanding environment challenges the neural StH-AC solver to handle simultaneous continuous policies—both sovereign borrowing and physical capital accumulation—demonstrating the framework's scalability in environments where traditional grid-based methods face severe computational bottlenecks.

3.1 Mapping the core environment

The sovereign enters each period with state $s = (z, B)$, where z is exogenous endowment income and $B \geq 0$ is the inherited stock of long-term debt. The sovereign chooses a discrete action $d \in \{R, D\}$ denoting repayment ($d = 1$) or default ($d = 0$, with indicator $\delta(z, B) = 1$). Conditional on repayment, it selects next-period debt $x = B' \in [0, B_{\max}]$ subject to strictly positive consumption:

$$c^R(z, B, B') = z - \kappa B + q(z, B')[B' - (1 - \lambda)B] > 0, \tag{23}$$

where $\kappa \equiv \lambda + (1 - \lambda)\text{coup}$ is the per-unit servicing cost for debt maturing with probability $\lambda \in (0, 1)$ and paying coupon $\text{coup} > 0$. The price schedule $q(z, B')$ represents the equilibrium expectation network Γ from our general framework, and $q(z, B')[B' - (1 - \lambda)B]$ denotes net revenue from new issuance.

3.2 The original unsmoothed economy

In the original structural economy, no unobserved shocks smooth the sovereign's decision. Upon repayment, the sovereign remains in credit markets and solves:

$$V^R(z, B) = \max_{B' \in [0, B_{\max}]} \left\{ u(c^R(z, B, B')) + \beta \sum_{z'} P_z(z, z') V(z', B') \right\}. \tag{24}$$

If the sovereign defaults, it is excluded from credit markets, regains access with probability $\chi \in (0, 1)$ with zero debt, and has default output $h(z) = z - \max\{0, d_0 z + d_1 z^2\}$ (capped at $z^* \in \arg\max_z h(z)$). Its default value is:

$$V^D(z) = u(h(z)) + \beta \sum_{z'} P_z(z, z')[\chi V(z', 0) + (1 - \chi) V^D(z')]. \tag{25}$$

The total value is the non-differentiable upper envelope $V(z,B) = \max\{V^R(z,B), V^D(z)\}$, and default follows the strict indicator $\delta(z,B) = \mathbf{1}\{V^D(z) > V^R(z,B)\}$. International lenders price debt according to the default probability next period:

$$q(z,B') = \frac{1}{1+r_f} \sum_{z'} P_z(z,z')\big(1-\delta(z',B')\big)\left[\lambda + (1-\lambda)(\text{coup} + q(z',A(z',B')))\right], \tag{26}$$

where $A(z',B')$ denotes the optimal continuous debt policy conditional on repayment. Equations (23)–(26) define the hard-switch baseline economy.

For the quantitative calibration, one period is one quarter, utility is $u(c) = -1/c$ ($\gamma = 2$), and $\beta = 0.954$. Log income follows an AR(1) process $\ell_t = \rho_z \ell_{t-1} + \varepsilon_t$ with $\rho_z = 0.9485$ and baseline innovation volatility $\sigma_z = 0.0271$ ($z_t = \exp(\ell_t)$). The risk-free rate is $r_f = 0.01$, maturity probability $\lambda = 0.05$, coupon coup $= 0.03$, and reentry probability $\chi = 0.0385$. The default loss parameters are $d_0 = -0.1882$ and $d_1 = 0.2456$.

3.3 Modified economies: artificial smoothing paradigms

To contextualize the StH solver, we compare it against two standard literature formulations that alter the structural model to achieve computational tractability. To reduce clutter, we outline only the main idea of these smoothing approaches. Full recursive equations are supplied in Appendix B.

Auxiliary income shocks Chatterjee and Eyigungor (2012) add a small i.i.d. transitory shock to repayment consumption. Entry into default resets the shock to its lower support point, $-\bar{m}$. Realized choices remain hard, while lenders integrate over the shock to form a smooth default probability. The baseline scale is $\sigma_{\text{aux}} = 0.003$, with $\bar{m} = 0.006$; Appendix B gives the complete recursion.

Taste-shock smoothing Alternatively, nested extreme-value taste shocks (Gordon, 2019; Mihalache, 2024) smooth decisions at two temperature scales (ρ_D, ρ_B). Continuous debt choices on grid G_B use scale ρ_B to form a log-sum-exp repayment value $V^R_{\rho_B}(z,B) = \rho_B \log \sum_{B' \in G_B} \exp(Q^R(z,B,B')/\rho_B)$, while discrete default risk uses scale ρ_D to yield smooth default probabilities:

$$P_{\rho_D}(\text{default} \mid z,B) = \frac{\exp\left(V^D(z)/\rho_D\right)}{\exp\left(V^R_{\rho_B}(z,B)/\rho_D\right) + \exp\left(V^D(z)/\rho_D\right)}. \tag{27}$$

Following Mihalache (2024), we evaluate this benchmark at $(\rho_D, \rho_B) = (5 \times 10^{-4}, 10^{-5})$.

4 Implementing StH for the sovereign default model

We now apply the general StH framework to the sovereign-default model. The state, continuous decision, and two value branches become income and debt, next-period borrowing, and repayment and default values. The bond-price schedule closes the borrower–lender equilibrium. We give the model-specific equations and implementation choices below; Algorithms 1 and 2 summarize the procedures, and Appendix D records the full numerical configurations.

4.1 StH-Q in the sovereign-default model

In the generic Q formulation, $s = (z, B)$, $x = B'$, $V^1 = V^R$, and $V^0 = V^D$. The feasible action grid is determined by positive repayment consumption. StH-Q maintains a table of repayment action values, $Q^R(z, B, B')$, together with the default value $V^D(z)$ and bond price $q(z, B')$. The upper-case Q^R denotes an action value; the lower-case q denotes a bond price.

At a sampled feasible state–action triple, the expected temporal-difference target and update equations are:

$$\begin{aligned} Y^n(z, B, B') &= u\big(z - \kappa B + q^n(z, B')[B' - (1-\lambda)B]\big) \\ &\quad + \beta \sum_{z'} P_z(z, z') V_\tau^n(z', B'), \qquad (28) \\ Q^{R,n+1}(z, B, B') &= Q^{R,n}(z, B, B') + \alpha_Q[Y^n(z, B, B') - Q^{R,n}(z, B, B')], \qquad (29) \\ V^{R,n}(z, B) &= \max_{B' \in \mathcal{B}(z,B;q^n)} Q^{R,n}(z, B, B'), \qquad (30) \\ g^n(z, B) &= \arg\max_{B' \in \mathcal{B}(z,B;q^n)} Q^{R,n}(z, B, B'). \qquad (31) \end{aligned}$$

Here κ is the debt-service coefficient, and $\mathcal{B}$ excludes choices with nonpositive consumption. The envelope V_τ smooths the repayment/default maximum during initial training and becomes the exact maximum when $\tau = 0$. Default values and lender prices are updated from the corresponding model equations.

The implementation combines replay and planning with damped price updates and averaging of late iterations. A final hard TD block holds the averaged price fixed and refines borrower values. The hard phase uses the feasible stored-Q maximizer and the same repayment/default rule in borrower and lender updates.

Algorithm 1 StH-Q: temporary smoothing and hard tabular refinement

Require: Model primitives, state and action grids, update and exploration schedules

Initialize Q^R, V^D, q using temporary soft model-based iterations
for the main learning budget **do**
    Simulate experience and form experience/coverage batches
    Update sampled action values with the expected TD target
    Perform model-based planning at selected states; update V^D and q
    Reduce the continuation temperature initially, then continue with hard targets
    Collect the designated late-iteration snapshots
end for
Average the snapshots; hold the averaged price fixed for hard TD refinement
Export feasible stored-Q greedy actions and hard repayment/default choices

4.2 StH-AC in the sovereign-default model

We now describe an implementation of the StH solver for the CE2012 sovereign default environment. The StH-AC implementation has two training stages followed by separate evaluation and reporting. The first stage performs 1,000,000 soft-training updates while annealing the smoothing operator to a low positive terminal temperature. The second stage performs 100,000 hard-refinement updates of the actor, repayment and default values, and bond-price network using the exact hard operators, corresponding operationally to zero temperature. After training, policies, prices, residuals, and simulations are evaluated and reported under the hard repay/default rule.

Network architecture and economic bounds All four networks $B'_\omega(z, B)$, $V^R_\theta(z, B)$, $V^D_\psi(z)$, and $q_\phi(z, B')$, corresponding to the generic objects in Equation (11), use feedforward multilayer perceptrons with two hidden layers of width 256 and ReLU activations. To ensure the solver respects the economic bounds of the CE2012 model from the first iteration, we apply terminal sigmoid activations $\sigma(\cdot)$ scaled by strict economic limits to the actor and price network:

- *Continuous Actor (B'_ω)*: Maps state $(z, B) \mapsto B' \in [0, B_{\max}]$. Parameterized as $B'_\omega(z, B) = B_{\max}\sigma(f_\omega(z, B))$.
- *Repayment Critic (V^R_θ)*: Maps state $(z, B) \mapsto V^R \in \mathbb{R}$. Linear output, unbounded.
- *Default Critic (V^D_ψ)*: Maps state $z \mapsto V^D \in \mathbb{R}$. Linear output, unbounded.

- *Price Network (q_ϕ)*: Maps future state (z, B') to the admissible price interval $[0, q^{\text{rf}}]$. Parameterized as $q_\phi(z, B') = q^{\text{rf}}\sigma(f_\phi(z, B'))$.

The risk-neutral upper bound on the bond price is strictly enforced using $q^{\text{rf}} = \frac{\lambda+(1-\lambda)\text{coup}}{r_{\text{f}}+\lambda}$.

The networks represent the continuous debt policy function, branch values, and bond prices. StH-AC obtains the repayment/default decision by comparing the two value branches, while the actor learns the repayment-contingent debt choice through gradients of its economic objective.

Model targets and actor updates Following the StH methodology, we replace the hard continuation value and hard default indicator with the soft operators $C_{\tau_D}(z, B)$ and $p^R_{\tau_D}(z, B)$, controlled by temperature τ_D. The actor's current proposal implies repayment consumption:

$$c_\omega(z, B) = z - \kappa B + q_\phi(z, B'_\omega(z, B))\left[B'_\omega(z, B) - (1-\lambda)B\right]. \tag{32}$$

For all training targets, we use the positive consumption floor $\underline{c} > 0$ and define $c^{\text{safe}}_\omega(z, B) = \max\{c_\omega(z, B), \underline{c}\}$ and $\text{pen}_\omega(z, B) = \max\{\underline{c} - c_\omega(z, B), 0\}^2$. The Bellman regression targets for the critics are adapted to the CE2012 transitions:

$$V^R(z, B) = u(c^{\text{safe}}_\omega(z, B)) + \beta \sum_{z'} P_z(z, z')C_{\tau_D}(z', B'_\omega(z, B)), \tag{33}$$

$$V^D(z) = u(h(z)) + \beta \sum_{z'} P_z(z, z')\left[\chi C_{\tau_D}(z', 0) + (1-\chi)V^D_\psi(z')\right]. \tag{34}$$

Similarly, the pricing equation target utilizes the soft repayment probability to make the detached lender-pricing target vary smoothly with the actor's future choices:

$$q(z, B') = \frac{1}{1+r_{\text{f}}} \sum_{z'} P_z(z, z')p^R_{\tau_D}(z', B')\left[\lambda + (1-\lambda)\left(\text{coup} + q_\phi(z', B''_\omega(z', B'))\right)\right], \tag{35}$$

where $B''_\omega(z', B')$ denotes the borrowing choice of the target actor in the future state. Equations (33), (34), and (35) are the CE2012 counterparts of the generic soft Bellman targets. The critics and the price network are trained using standard mean-squared-error losses against these targets.

The actor optimizes the structural economic objective directly using the repayment consumption and feasibility definitions above. The unified actor loss is therefore:

$$L_{\text{actor}} = -\mathbb{E}\left[u(c^{\text{safe}}_\omega(z, B)) + \beta\mathbb{E}_{z'|z}C_{\tau_D}(z', B'_\omega(z, B))\right] + \lambda_{\text{feas}}\mathbb{E}\left[\text{pen}_\omega(z, B)\right]. \tag{36}$$

Equation (36) applies the generic actor objective to the CE2012 budget constraint in Equation (32). During an actor update, the price and value functions in this objective are the online networks.

Since $V^D(z')$ does not depend on the debt choice B', gradient penetration has the model-specific form:

$$\frac{\partial C_{\tau_D}(z', B')}{\partial B'} = p^R_{\tau_D}(z', B') \frac{\partial V^R_\theta(z', B')}{\partial B'}, \tag{37}$$

$$\frac{\partial C_H(z', B')}{\partial B'} = \mathbf{1}_{\{V^R_\theta(z', B') \geq V^D_\psi(z')\}} \frac{\partial V^R_\theta(z', B')}{\partial B'}, \tag{38}$$

away from ties. These expressions concern the next-period continuation gradient; the actor loss is not multiplied by a current-period repayment probability.

In the sovereign default application, the income transition is independent of the debt choice, and the actor sets next-period debt. Under the hard operator, selecting the default branch shuts off the continuation derivative with respect to debt. Temporary smoothing assigns a positive analytical weight to the repayment branch and transmits its debt derivative to the actor whenever that derivative is nonzero. This gradient penetration overcomes the loss of continuation guidance caused by hard switching and helps the actor improve its repayment policy near the default boundary.

During hard refinement, the actor remains free and the soft continuation in Equation (36) is replaced by the hard upper envelope:

$$\begin{aligned} L^H_{\text{actor}} = & -\mathbb{E}\left[u(c^{\text{safe}}_\omega(z, B)) + \beta \mathbb{E}_{z'|z} \max\left\{V^R_\theta(z', B'_\omega(z, B)), V^D_\psi(z')\right\}\right] \\ & + \lambda_{\text{feas}} \mathbb{E}\left[\text{pen}_\omega(z, B)\right]. \end{aligned} \tag{39}$$

In Equation (39), the actor differentiates through the online price and value networks, following the selected next-period continuation branch of the hard maximum except at ties. The actor is trained as the repayment-contingent debt choice at every sampled state; there is no current-period hard default indicator in its loss. By contrast, the hard critic and price-network regression targets are detached and use target-network copies (denoted by a superscript minus); the hard price target uses the next-period indicator $\mathbf{1}_{\{V^{R,-} \geq V^{D,-}\}}$. Thus, the hard block continues to update the actor, critics, price network, and target networks; it is not merely an evaluation pass.

Algorithm 2 StH-AC: soft neural training followed by hard refinement

Require: Model primitives, networks, feasible parameterization, training schedules

Initialize online/target networks and pretrain under a fixed policy
for the soft-training budget **do**
    Draw state batches and anneal the temporary continuation temperature
    Update the actor through online values/prices and fit detached critic/price targets
    Alternate actor and regression updates; synchronize target networks
end for
for the hard-refinement budget **do**
    Continue actor, critic, and price updates with the hard continuation and decision rules
end for
Output policy, value, and price functions under the hard decision rule

4.3 Validation in the one-period sovereign default model

To validate the StH framework, we first evaluate it in a canonical one-period sovereign default model (Arellano, 2008). Both implementations exactly recover the high-density value function iteration (VFI) benchmark for the original, unsmoothed economy (Appendix C), with StH-Q delivering near-machine precision. Crucially, this environment validates StH-AC's gradient penetration mechanism. In a robustness check (Appendix C.3), we deliberately initialized the networks with severe distortions, generating a flat, unresponsive asset policy. While such poor initializations trap traditional hard-switch networks in zero-gradient dead zones, StH-AC's soft probabilities continuously transmit non-zero gradients to the actor network. This enables the solver to escape stagnation and systematically converge to the true equilibrium.

5 Numerical results I: the baseline sovereign default model of Chatterjee and Eyigungor (2012)

This section benchmarks the StH framework against traditional auxiliary-shock and taste-shock solvers in our primary application—the canonical sovereign debt model of Chatterjee and Eyigungor (2012)—and highlights four main results. First, under the baseline calibration, all four methods deliver accurate and reliable solutions. Second, StH-Q achieves superior precision by directly solving the exact hard-switch economy without ad-hoc parameters. Third, auxiliary-shock and taste-shock solvers face non-

convergence and structural policy distortions when attempting to recover the exact boundary by shrinking parameters toward zero. Finally, StH-AC is subject to sampling variation; however, its effect is modest, zero-mean, and strictly controllable via sample averaging.

5.1 A comparison analysis of accuracy and execution time

To evaluate numerical performance, we map all four solvers (Aux, Taste, StH-Q, and StH-AC) to a common 70,000-state discrete grid (200 income and 350 debt points). Each solution then undergoes a "hard-operator audit" to assess its exact fidelity against the unsmoothed target economy. We compute four diagnostic statistics: borrower Bellman error (R_m^V), repayment policy error (R_m^π), lender pricing error (R_m^q), and default-rule inconsistency (D_m), which measures how often the exported binary decision conflicts with an exact hard Bellman update. Formal definitions and mapping protocols are detailed in Appendix E.

5.1.1 Accuracy metrics and default-rule consistency

Table 1 summarizes runtime and common-grid accuracy metrics across the four methods.

Table 1: Hard-operator accuracy and computation times under the baseline calibration.

Method	Bellman RMSE	Repayment-policy optimality RMS	Pricing RMSE	Default-rule inconsistency (%)	Time (seconds)
Auxiliary shocks	2.544×10^{-4}	1.534×10^{-5}	1.800×10^{-3}	0.35571	151.4
Taste shocks	1.831×10^{-5}	7.093×10^{-6}	1.677×10^{-3}	0.00143	263.3
StH-Q	2.254×10^{-6}	2.539×10^{-6}	1.466×10^{-3}	0.00000	200.0
StH-AC	1.999×10^{-3}	9.017×10^{-5}	2.290×10^{-3}	0.06571	1728.7
StH-AC (5×)	1.407×10^{-3}	2.262×10^{-5}	1.941×10^{-3}	0.05286	8322.5

Note: Bellman and pricing RMSEs use all states of the common 200×350 grid; policy optimality RMS uses recomputed repayment states. Default-rule inconsistency is the percentage of states where the exported rule differs from the rule implied by one hard Bellman update. StH-AC (5×) denotes the run with 5 million soft updates and 500,000 hard updates. Audit definitions and timing conventions are in Appendix E and Appendix D.

StH-Q achieves the lowest error across all metrics: a borrower Bellman error of 2.25×10^{-6}, a policy error of 2.54×10^{-6}, a pricing error of 1.47×10^{-3}, and absolute zero default-rule inconsistency ($D_m = 0.00\%$). Achieving grid-level fidelity on the exact hard economy, StH-Q serves as the definitive numerical benchmark for low-dimensional models.

The remaining solvers highlight a critical contrast between numerical approximation error and structural distortion. StH-AC yields larger Bellman residuals—typical for neural networks—yet its policy and pricing errors remain competitive, and its default inconsistency is just 0.06571%. The permanently smoothed Aux and Taste benchmarks yield Bellman residuals one to two orders of magnitude larger than StH-Q, with default inconsistencies of 0.36% and 0.0014%, respectively.

Crucially, the origin of these inconsistencies differs. StH-AC's discrepancies stem purely from numerical approximation, as it optimizes against the exact hard target. For Aux and Taste, however, the discrepancies arise from a combination of numerical approximation errors and structural distortions: permanently altering the objective function with exogenous shocks forces systematic deviations from the true target that persist regardless of computational grid density.

5.1.2 Execution time: reconciling precision with scalability

Assessing computational efficiency requires contrasting tabular grid contraction with neural stochastic optimization. Grid-based solvers execute via fixed-point iterations on a discretized state space: once resolution is fixed, convergence reaches a deterministic accuracy floor that cannot be lowered without spatial refinement. By contrast, StH-AC optimizes neural function approximators via stochastic gradient descent across soft-training and hard-refinement phases, enabling researchers to trade additional compute time directly for numerical precision.

As reported in Table 1, grid-based methods execute in 2–4 minutes, whereas baseline StH-AC requires roughly 30 minutes. However, extending training—as in StH-AC (5×)—reduces Bellman, policy, and pricing errors by 29.6%, 74.9%, and 15.2%, respectively, with continued training reducing error even further. While tabular deterministic grid methods are superior in low-dimensional settings, the next section demonstrates the decisive computational advantages of StH-AC's gridless architecture as state-space dimensionality expands.

5.2 Macroeconomic implications

We examine the economic properties of the four solvers (Aux, Taste, StH-Q, and StH-AC) across business cycle statistics, equilibrium functions, and simulated dynamics. Full simulation protocols and conventions are detailed in Appendix D.4.

5.2.1 Business cycle statistics

Table 2 summarizes the business cycle statistics.

Under the baseline calibration, all four solvers yield tightly aligned core business cycle moments: mean annual spreads range from 8.15% to 8.57%, spread standard deviations from 4.45% to 4.75%, and mean debt-to-output ratios from 69.18% to

Table 2: Business cycle moments under the baseline calibration.

Indicator	Auxiliary shocks	Taste shocks	StH-Q	StH-AC
Calibration targets				
Mean spread	0.08151	0.08336	0.08356	0.08573
Std. spread	0.04452	0.04600	0.04611	0.04750
Mean debt/output	0.69988	0.69327	0.69177	0.69484
Other moments				
Default frequency	0.06888	0.07049	0.06903	0.07329
Average debt service	0.05484	0.05431	0.05419	0.05440
$\sigma(\log C)/\sigma(\log Y)$	1.10676	1.10622	1.10529	1.10305
$\sigma(NX/Y)/\sigma(\log Y)$	0.20345	0.20208	0.20082	0.20329
$\mathrm{Corr}(\log C, \log Y)$	0.98563	0.98584	0.98598	0.98527
$\mathrm{Corr}(\mathrm{Spread}, \log Y)$	-0.65002	-0.65066	-0.64251	-0.61200
$\mathrm{Corr}(NX/Y, \log Y)$	-0.43674	-0.43833	-0.43731	-0.41791

Note: Moments are averaged across 256 paths of 5,000 quarters each, with the first 500 quarters discarded.

69.99%. This quantitative agreement between the smoothed and hard-switch solvers demonstrates that structural smoothing does not alter core macroeconomic mechanisms in the standard CE2012 environment.

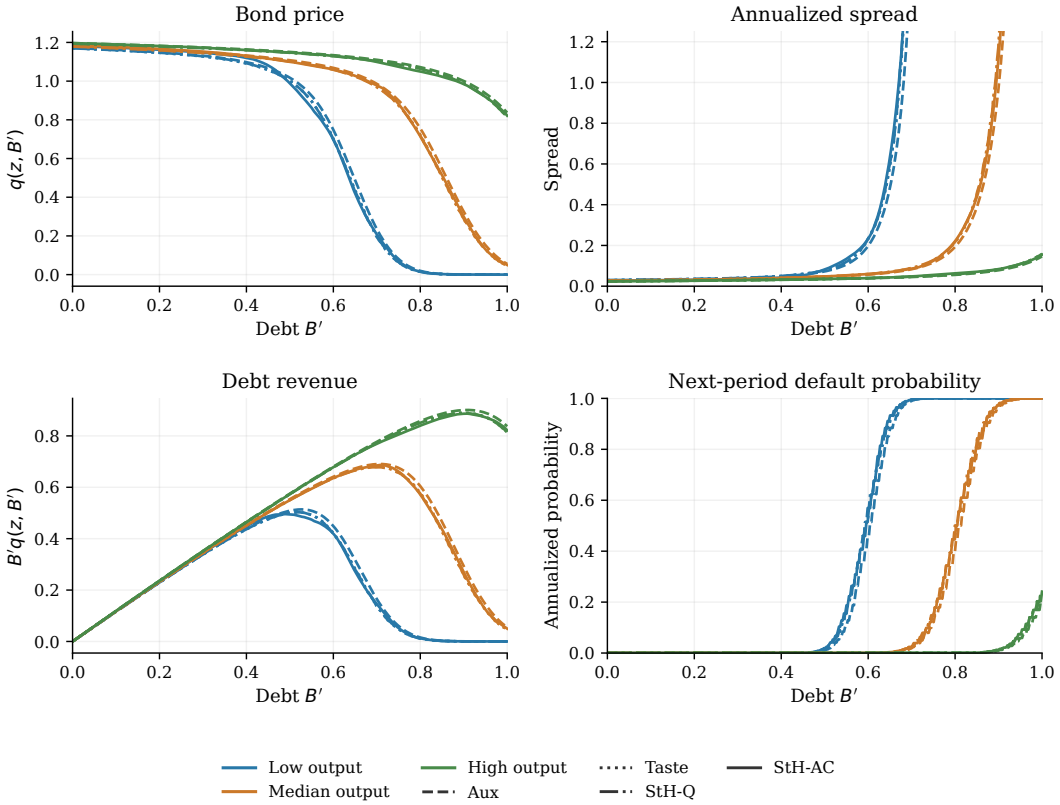
5.2.2 Equilibrium functions

Figure 1 presents the key equilibrium objects for the four methods. All four methods produce closely aligned equilibrium functions over the vast majority of the state space.

5.2.3 Simulated paths

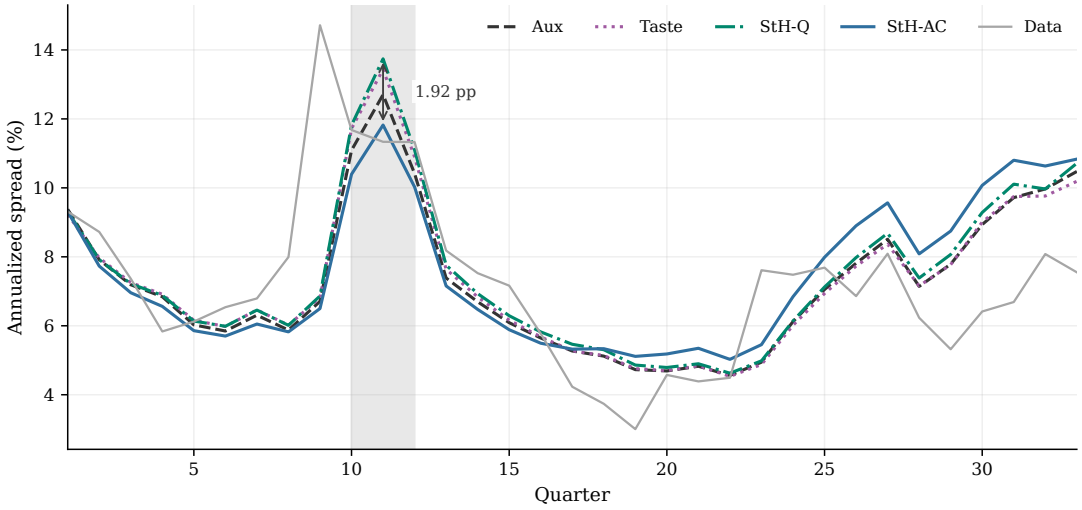
Figure 2 plots the simulated baseline spread trajectories over time. Across all four solvers, the paths exhibit identical cyclical dynamics and remain closely aligned throughout the simulation horizon.

Fig. 1: Equilibrium functions under the baseline calibration.



Note: Low and high states are approximately one stationary standard deviation below and above median log output.

Fig. 2: Simulated spreads under the baseline calibration.



Note: Methods share identical initial conditions and exogenous output histories. The light-gray band marks the largest cross-method gap in quarter 11. Empirical data are from Chatterjee and Eyigungor (2012).

5.3 The role of algorithm parameters in (non)convergence and numerical (in)stability

While the StH framework directly solves the exact hard-switch economy, Aux and Taste rely on artificial smoothing parameters to achieve numerical stability. This dependence raises two fundamental questions: how sensitive are macroeconomic outcomes to parameter selection, and does driving these parameters toward zero recover the unsmoothed limit? To sharpen these dynamics, we focus our analysis on a high income volatility stress test ($\sigma_z = 0.054184$), where the numerical distortions and instabilities of smoothing parameters become particularly pronounced.

5.3.1 Convergence diagnostics and algorithmic breakdown

This smoothing is governed by the standard deviation (σ_{aux}) in the auxiliary-shock model, and by default (ρ_D) and borrowing (ρ_B) temperatures in the taste-shock model; reducing these parameters toward zero sharpens the decision boundaries toward the exact hard-switch limit. Table 3 evaluates stability using iterations to convergence (K, capped at 3,000 for Aux and 1,000 for Taste following their original implementations) and terminal iteration-to-iteration sup-norm updates (err_V, err_q).

Table 3: Convergence diagnostics under the high-volatility calibration.

Method	σ_{aux}	ρ_D	ρ_B	K	terminal err_V	terminal err_q
Auxiliary shocks	0.012000	–	–	702	2.577×10^{-6}	9.974×10^{-6}
Auxiliary shocks	0.007500	–	–	3000	6.201×10^{-6}	1.258×10^{-4}
Auxiliary shocks	0.003000	–	–	3000	4.827×10^{-4}	8.771×10^{-3}
Auxiliary shocks	0.001500	–	–	3000	7.063×10^{-4}	1.491×10^{-2}
Auxiliary shocks	0.000750	–	–	3000	5.404×10^{-4}	9.569×10^{-3}
Taste shocks	–	0.002	4×10^{-5}	1000	2.156×10^{-5}	5.144×10^{-4}
Taste shocks	–	0.00125	2.5×10^{-5}	1000	2.413×10^{-3}	7.284×10^{-2}
Taste shocks	–	5×10^{-4}	10^{-5}	1000	3.720×10^{-3}	7.348×10^{-2}
Taste shocks	–	2.5×10^{-4}	5×10^{-6}	1000	2.926×10^{-3}	9.415×10^{-2}
Taste shocks	–	1.25×10^{-4}	2.5×10^{-6}	1000	1.277×10^{-3}	4.336×10^{-2}
StH-Q	–	–	–	-	2.714×10^{-4}	1.373×10^{-5}
StH-AC	–	–	–	-	2.708×10^{-4}	4.950×10^{-5}

Note: Entries report sup-norm changes. Auxiliary and Taste compare successive iterations; $K = 3,000$ and $K = 1,000$ indicate that the respective iteration caps were reached. StH-Q uses the final main-training update on the full grid. StH-AC uses one hard update at 8,192 fixed states with the actor held fixed.

Tightening smoothing parameters severely destabilizes fixed-point iteration. As σ_{aux} shrinks from 0.012 to 0.00075, four of five auxiliary-shock specifications hit the $K = 3{,}000$ iteration cap, yielding terminal pricing updates (err_q) up to 1.49×10^{-2}.

The taste-shock formulation proves even more fragile: as (ρ_D, ρ_B) shrink to $(1.25 \times 10^{-4}, 2.5 \times 10^{-6})$, all tested specifications hit the $K = 1{,}000$ cap, leaving terminal pricing updates as large as 9.42×10^{-2}—a nearly 10% iteration-to-iteration gap in bond-price units. These breakdowns demonstrate that parameter tightening induces algorithmic failure rather than smooth convergence toward the exact hard-switch boundary.

In contrast, StH methods operate directly on the exact hard-switch environment without artificial smoothing parameters. For tabular StH-Q, the final training update shifts $V^H = \max\{V^R, V^D\}$ by 2.71×10^{-4} and q by 1.37×10^{-5} in sup-norm. For StH-AC, applying a single hard optimizer update across 8,192 frozen post-solution states shifts V^H by 2.71×10^{-4} and q by 4.95×10^{-5}. Both StH variants achieve tight numerical convergence, sharply contrasting with the non-convergence of smoothed benchmarks.

5.3.2 Deformation of policy and pricing functions

Crucially, these convergence failures generate severe structural deformations in the computed policy and pricing functions. Figure 3 illustrates this fragility by plotting bond price and spread schedules under the high-volatility calibration as smoothing parameters shrink for the auxiliary-shock (top panel) and taste-shock (bottom panel) solvers.

The economic distortions are stark. In the auxiliary-shock model, reducing σ_{aux} causes price and spread schedules to diverge, disrupting local parameter monotonicity near the default threshold. The taste-shock formulation exhibits even greater numerical fragility: as smoothing temperatures shrink, the computed schedules develop spurious oscillations and non-monotone kinks.

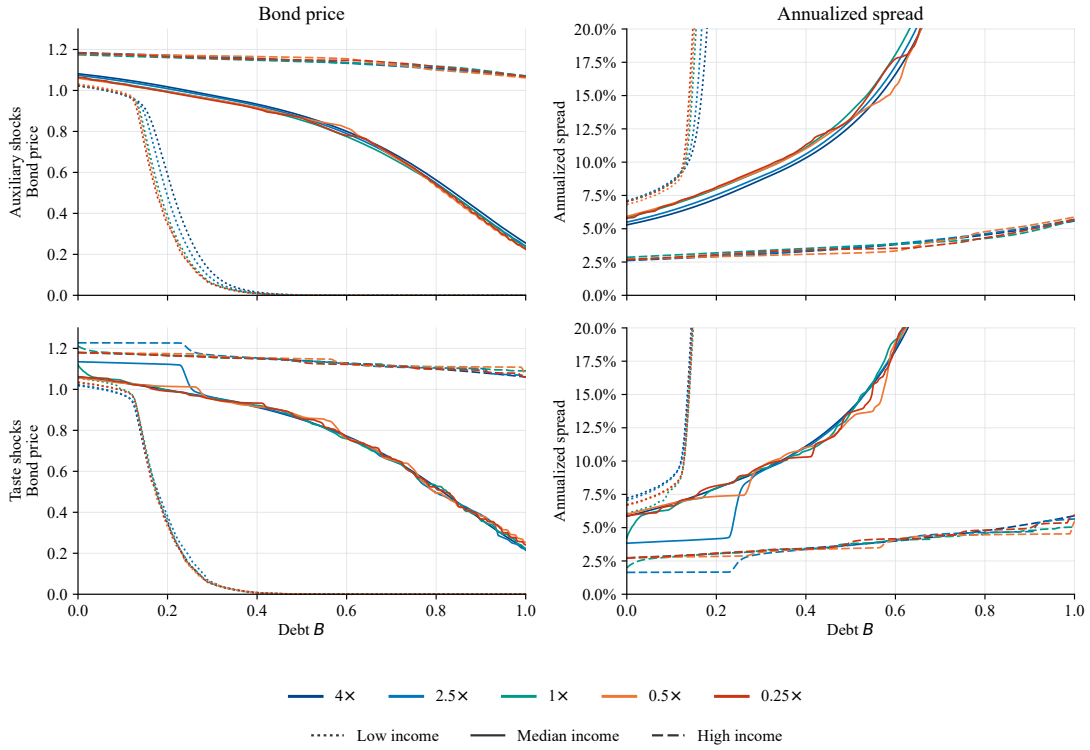
In turn, the StH framework circumvents these smoothing pathologies entirely by directly solving the exact hard-switch economy without auxiliary parameters. While StH-Q is completely free of artificial parameters, initialization variance can potentially affect StH-AC performance. Because the neural solver relies on stochastic gradient optimization and function approximation, its output naturally exhibits variation across random initializations. To evaluate training consistency under high volatility, Figure 4 assesses 16 independently trained StH-AC runs.

Crucially, this seed sensitivity is modest, structurally benign, and never induces non-convergence or numerical instability.

5.3.3 Structural distortions versus controllable simulation bias

Arbitrary parameter smoothing severely distorts macroeconomic outcomes. Table 4 tracks core moments across progressively tightened specifications.

Fig. 3: Sensitivity of equilibrium functions for the auxiliary-shock and taste-shock methods under the high-volatility calibration.



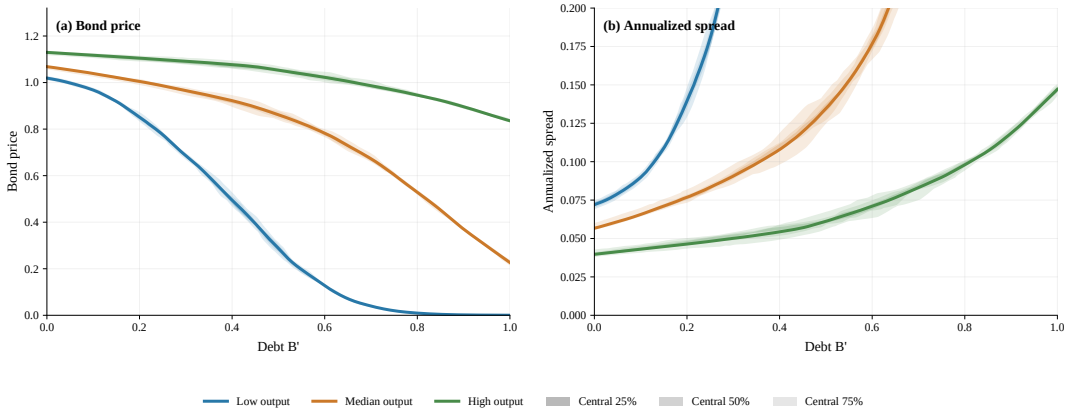
Note: The multipliers (4, 2.5, 1, 0.5, and 0.25) scale $\sigma_{\text{aux}} = 0.003$ for the auxiliary-shock method and the pair $(\rho_D, \rho_B) = (5 \times 10^{-4}, 10^{-5})$ for the taste-shock method.

Simulated outcomes are highly unstable and non-monotone. Varying σ_{aux} shifts the mean debt-to-output ratio between 0.52901 and 0.61866—a 16.9% within-method range. The taste-shock formulation exhibits even greater sensitivity: adjusting (ρ_D, ρ_B) moves the ratio between 0.50258 and 0.61677 (a 22.7% spread). Spreads and default frequencies show similarly erratic responses. Because the tightest specifications hit iteration caps, these reported moments reflect non-converged endpoints rather than true equilibria.

These findings carry significant economic implications. Ad-hoc smoothing introduced purely for computational tractability creates severe distortions: a 17% to 23% prediction variance driven entirely by numerical artifacts undermines quantitative policy evaluation.

In contrast, StH-AC solution variance across random seeds is modest. As reported in Table 4, across 16 independent seeds, the mean debt-to-output ratio is 0.56985 with

Fig. 4: Sensitivity of StH-AC equilibrium functions to training seeds under the high-volatility calibration.



Note: Solid lines are means across 16 independently trained solutions; shaded bands show the central 25%, 50%, and 75% of the cross-seed distribution. Low and high states are approximately one stationary standard deviation below and above median log output.

a standard deviation of 0.02677—a coefficient of variation of just 4.7%. Crucially, StH-AC's cross-seed variation is a zero-mean sampling artifact that decreases at rate $1/N$ when averaging across N independent runs. StH-AC thus replaces the large, irreducible structural distortions of smoothing parameters with a modest, strictly controllable simulation variance.

Table 4: Business cycle moment sensitivity under the high-volatility calibration.

Method	σ_{aux}	ρ_D	ρ_B	Mean spread	Mean debt/output	Default frequency
Auxiliary shocks	0.012000	–	–	0.16385	0.59647	0.11430
Auxiliary shocks	0.007500	–	–	0.16586	0.59253	0.11570
Auxiliary shocks	0.003000	–	–	0.17519	0.61866	0.12791
Auxiliary shocks	0.001500	–	–	0.13560	0.52901	0.09774
Auxiliary shocks	0.000750	–	–	0.14821	0.56233	0.10849
Taste shocks	–	0.002	4×10^{-5}	0.16936	0.58553	0.11773
Taste shocks	–	0.00125	2.5×10^{-5}	0.17191	0.58066	0.12479
Taste shocks	–	5×10^{-4}	10^{-5}	0.18198	0.61677	0.12912
Taste shocks	–	2.5×10^{-4}	5×10^{-6}	0.13146	0.50258	0.08931
Taste shocks	–	1.25×10^{-4}	2.5×10^{-6}	0.16181	0.56902	0.11082
StH-Q	–	–	–	0.16976	0.60539	0.12572
StH-AC	–	–	–	0.15916	0.56985	0.11104
				(0.01575)	(0.02677)	(0.01010)

Note: Moments are averaged across 256 paths of 5,000 quarters each, with the first 500 quarters discarded. StH-AC reports the mean across 16 independently trained solutions, with sample standard deviations in parentheses.

6 Numerical results II: the endogenous-capital economy of Gordon and Guerrón-Quintana (2018)

Grid-based and neural methods strike fundamentally different trade-offs between local numerical precision and dimensional scalability. As Judd (1992) established, traditional computational methods evaluated on dense spatial grids achieve exceptional local fidelity in small-scale applications but quickly succumb to the curse of dimensionality. Conversely, deep learning frameworks relying on stochastic simulation offer highly scalable representations of equilibrium functions. In particular, as Maliar and Maliar (2022) show for DCDC models, these neural approaches trade some degree of local precision for a computational complexity that scales gracefully with the state dimension.

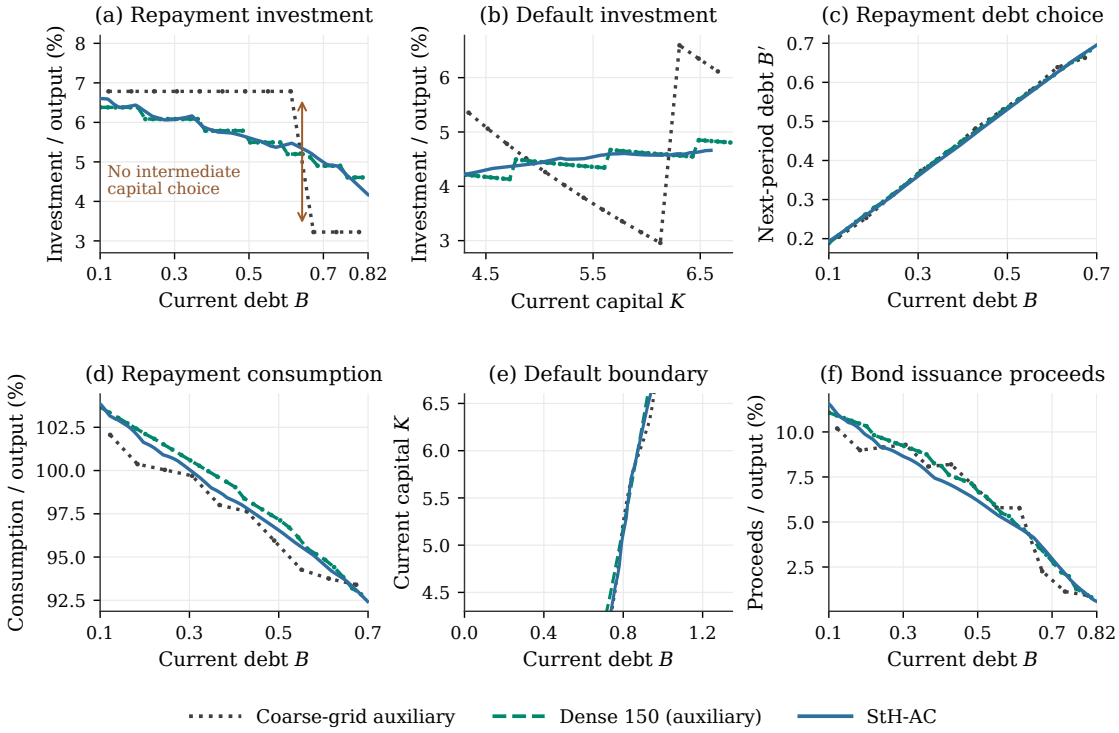
To evaluate the scalability of StH-AC in more complex environments, we apply it to the model of Gordon and Guerrón-Quintana (2018), which combines sovereign default, long-term debt, and capital accumulation. Even this modest expansion of the state space—from two to three continuous variables relative to Chatterjee and Eyigungor (2012)—fundamentally alters the computational landscape. Whereas StH-AC requires more compute time than deterministic grid-based methods in the two-state baseline, here it matches Dense 150 policy curves and default dynamics in approximately 2% of the reference solver's computation time. Below, we provide a summary of the results; the complete model description, calibration, and implementation details are deferred to Appendix F.

6.1 Discrete grids and continuous policies

This section illustrates how grid resolution affects investment policies and simulated economic dynamics. Dense 100 and Dense 150 denote grids with 100×100 and 150×150 debt–capital state nodes, respectively; the figures use Dense 150.

A joint grid search faces a trade-off between computational cost and choice resolution. In Figure 5(a), the coarse-grid policy jumps between adjacent capital choices because intermediate actions are unavailable. Refining the grid provides finer investment steps; only the debt and capital state and choice grids change, with all other settings held fixed. StH-AC directly optimizes continuous debt and capital choices using a grid-free representation of endogenous states and controls. Its investment policies in panels (a) and (b), and the default boundary in panel (e), closely track the Dense 150 results at approximately 2% of the computing time (Table 5). Panels (c), (d), and (f) compare the associated debt choices, repayment consumption, and bond issuance proceeds.

Fig. 5: Policies and the default boundary in the endogenous-capital model.



Note: All panels use $A = 0.5407$. Repayment policies are shown only where repayment is chosen. Cross-sections and the $m = 0$ default boundary are described in Appendix F.3.

6.2 Economic implications of grid resolution

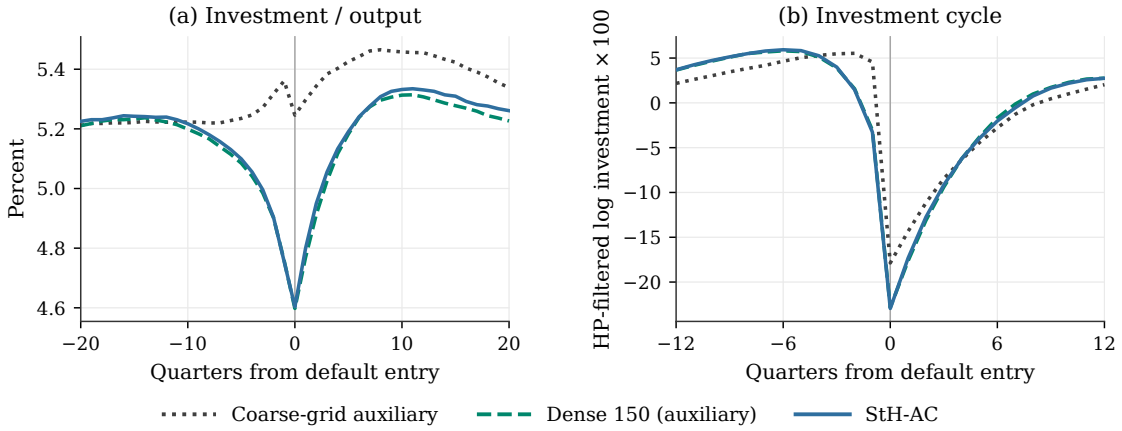
Figure 6 illustrates how grid resolution affects simulated investment dynamics.

In panel (a), the baseline investment-to-output ratio rises locally prior to default, whereas the Dense 150 profile declines. We use this investment-to-output profile to highlight the quantitative impact of grid resolution: the coarse-grid solution diverges significantly, whereas the StH-AC solution closely tracks the high-precision Dense 150 grid. Panel (b) complements this analysis by reporting the HP-filtered investment-cycle measure of Gordon and Guerrón-Quintana (2018).

Table 5 presents the computational details alongside the complete set of simulated moments.

Computing time rises sharply as the state and choice grids are refined. This illustrates the burden of joint grid searches, for which the number of grid combinations grows exponentially with the number of dimensions.

Fig. 6: Investment around sovereign default in the endogenous-capital model.



Note: Events are centered on default entry (quarter zero). Panel (a) reports the investment-to-output ratio in levels. Panel (b) reports HP1600-filtered log investment (×100), following Gordon and Guerrón-Quintana (2018).

Table 5: Grid resolution versus continuous choice: Simulated moments and computational efficiency.

	Coarse grid	Dense 100	Dense 150	StH-AC
Algorithm details and performance				
Debt-capital state grid	50×50	100×100	150×150	Continuous
Debt-capital choice grid	250×250	500×2500	1000×3000	Continuous
Auxiliary income shock s.d., σ_m	0.003	0.003	0.003	None
Computing time (seconds)	243.2	27,479.0	155,709.7	3,379.8
Simulated moments				
Debt/output	0.7004	0.6928	0.6940	0.6990
Mean annualized spread (%)	8.2372	7.7917	7.6688	7.6895
Spread standard deviation (pp)	4.4280	3.9212	3.6632	3.4948
Quarterly default frequency (%)	1.2841	1.2339	1.2237	1.2321
Mean investment/output (%)	5.2073	5.1670	5.1644	5.1762
Output volatility (%)	4.8910	4.8618	4.8574	4.8701
Consumption volatility (%)	5.9749	5.7205	5.7094	5.7639
Investment volatility (%)	12.9658	9.7693	9.8481	9.9748
Relative consumption volatility	1.2216	1.1766	1.1754	1.1835
Relative investment volatility	2.6510	2.0094	2.0274	2.0482

Note: Moment definitions and computing details follow Appendix F.3.

StH-AC requires 3,379.8 seconds, compared to 155,709.7 seconds for the Dense 150 solution (approximately 2.17% of its runtime). As the table demonstrates, re-

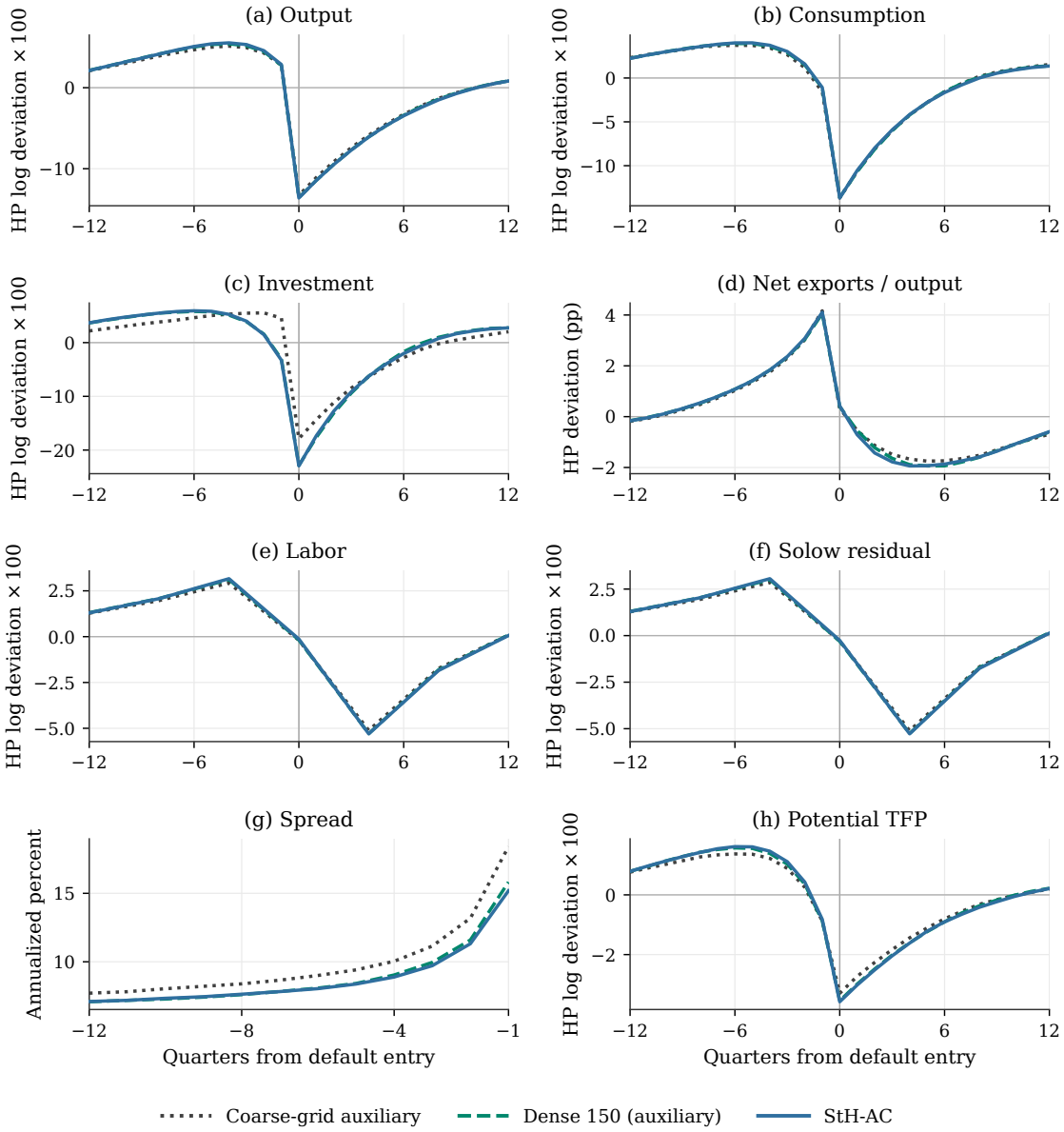
fining the debt and capital grids reduces investment volatility from 12.9658% to 9.8481% and the spread standard deviation from 4.4280 to 3.6632 percentage points. StH-AC solves for continuous debt and capital states and choices with the auxiliary income shock set to zero, successfully replicating the improved numerical precision of the Dense 150 grid at a fraction of the computational cost.

6.3 StH-AC and the dense numerical solution

Figure 7 focuses on a central finding of Gordon and Guerrón-Quintana (2018), namely that output, consumption, and investment expand, contract, and recover around default events. Notably, StH-AC produces event profiles that closely match those of the high-precision auxiliary Dense 150 solution.

In summary, our analysis demonstrates that StH-AC successfully overcomes the computational bottlenecks inherent in joint grid searches. By operating over continuous state and choice spaces, StH-AC eliminates the resolution constraints that distort investment and default dynamics in coarse-grid approximations. The method delivers the high numerical accuracy of a dense-grid approximation while maintaining computational tractability, establishing its potential for solving high-dimensional discrete-continuous choice models.

Fig. 7: Default-event profiles in the endogenous-capital model.



Note: Events are centered on default entry (quarter zero). Filtering, averaging, and simulation conventions follow Appendix F.3.

7 Conclusion

In this paper, we introduce the Soft-to-Hard (StH) reinforcement learning framework to solve dynamic macroeconomic models featuring discrete-continuous choices. We deliver two complementary implementations tailored to distinct computational demands: tabular StH-Q provides an ultra-precise solver for small- and medium-scale applications, while neural StH-AC provides a highly scalable architecture for high-dimensional environments. By isolating numerical regularization strictly to the learning phase, the StH framework avoids both the permanent structural distortions inherent in traditional smoothing methods and the severe numerical instability that arises when attempting to eliminate them. Consequently, both StH implementations solve the exact, unsmoothed economy with high precision and robust convergence.

The ability to evaluate exact decision rules carries immediate implications for structural estimation. When solution algorithms embed permanent smoothing distortions, structural estimation procedures such as the simulated method of moments inevitably absorb these computational artifacts. For instance, parameters governing risk aversion or time preference may systematically shift to compensate for artificially dampened tail risk. By removing permanent smoothing, StH eliminates this source of identification bias, ensuring that estimated structural parameters reflect true economic fundamentals rather than numerical workarounds.

Although applied here exclusively to sovereign default, the StH architecture operates natively on general discrete-continuous choice operators without requiring problem-specific derivations. A natural frontier is scaling this framework to heterogeneous-agent general equilibrium models. In these environments, discrete non-convexities—such as lumpy investment, firm entry and exit, or household bankruptcy—have historically forced researchers to choose between rigid, low-dimensional grids and heavy behavioral smoothing. By providing a domain-agnostic, scalable methodology, StH enables the exploration of high-dimensional macroeconomic environments without sacrificing structural fidelity.

References

Aguiar, M., Amador, M., 2020. Self-fulfilling debt dilution: Maturity and multiplicity in debt models. American Economic Review 110, 2783–2818. doi:10.1257/aer.20180831.

Arellano, C., 2008. Default risk and income fluctuations in emerging economies. American Economic Review 98, 690–712. doi:10.1257/aer.98.3.690.

Arellano, C., Ramanarayanan, A., 2012. Default and the maturity structure in sovereign bonds. Journal of Political Economy 120, 187–232. doi:10.1086/666589.

Azinovic, M., Gaegauf, L., Scheidegger, S., 2022. Deep equilibrium nets. International Economic Review 63, 1471–1525. doi:10.1111/iere.12575.

Berry, S., Levinsohn, J., Pakes, A., 1995. Automobile prices in market equilibrium. Econometrica 63, 841–890. doi:10.2307/2171802.

Chatterjee, S., Eyigungor, B., 2012. Maturity, indebtedness, and default risk. American Economic Review 102, 2674–2699. doi:10.1257/aer.102.6.2674.

Druedahl, J., 2021. A guide on solving non-convex consumption-saving models. Computational Economics 58, 747–775. doi:10.1007/s10614-020-10045-x.

Dvorkin, M., Sánchez, J.M., Sapriza, H., Yurdagul, E., 2021. Sovereign debt restructurings. American Economic Journal: Macroeconomics 13, 26–77. doi:10.1257/mac.20190220.

Eaton, J., Gersovitz, M., 1981. Debt with potential repudiation: Theoretical and empirical analysis. The Review of Economic Studies 48, 289–309. doi:10.2307/2296886.

Fernández-Villaverde, J., 2026. Deep learning for solving economic models. Journal of Economic Literature 64, 829–875. URL: https://www.aeaweb.org/articles?id=10.1257/jel.20261794, doi:10.1257/jel.20261794.

Fernández-Villaverde, J., Hurtado, S., Nuño, G., 2023. Financial frictions and the wealth distribution. Econometrica 91, 869–901. doi:10.3982/ECTA18180.

Gordon, G., 2019. Efficient Computation with Taste Shocks. Working Paper 19-15. Federal Reserve Bank of Richmond. doi:10.21144/wp19-15.

Gordon, G., Guerrón-Quintana, P.A., 2018. Dynamics of investment, debt, and default. Review of Economic Dynamics 28, 71–95. URL: https://www.sciencedirect.com/science/article/pii/S1094202517300649, doi:10.1016/j.red.2017.07.007.

Haarnoja, T., Zhou, A., Abbeel, P., Levine, S., 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, in: Proceedings of the 35th International Conference on Machine Learning, pp. 1861–1870.

Han, J., Yang, Y., E, W., 2026. DeepHAM: A global solution method for heterogeneous agent models with aggregate shocks. Quantitative Economics 17, 297–341. doi:10.3982/QE2190.

Hatchondo, J.C., Martinez, L., 2009. Long-duration bonds and sovereign defaults. Journal of International Economics 79, 117–125. doi:10.1016/j.jinteco.2009.07.002.

Iskhakov, F., Jørgensen, T.H., Rust, J., Schjerning, B., 2017. The endogenous grid method for discrete-continuous dynamic choice models with (or without) taste shocks. Quantitative Economics 8, 317–365. doi:10.3982/QE643.

Judd, K.L., 1992. Projection methods for solving aggregate growth models. Journal of Economic Theory 58, 410–452. doi:10.1016/0022-0531(92)90061-L.

Maliar, L., Maliar, S., 2022. Deep learning classification: Modeling discrete labor choice. Journal of Economic Dynamics and Control 135, 104295.

Maliar, L., Maliar, S., Winant, P., 2021. Deep learning for solving dynamic economic models. Journal of Monetary Economics 122, 76–101. doi:10.1016/j.jmoneco.2021.07.004.

McFadden, D., 1974. Conditional logit analysis of qualitative choice behavior, in: Zarembka, P. (Ed.), Frontiers in Econometrics. Academic Press, New York, pp. 105–142.

Mihalache, G., 2024. Solving default models. In preparation for the Oxford Research Encyclopedia of Economics and Finance.

Rust, J., 1987. Optimal replacement of GMC bus engines: An empirical model of Harold Zurcher. Econometrica 55, 999–1033. doi:10.2307/1911259.

Sánchez, J.M., Sapriza, H., Yurdagul, E., 2014. Sovereign Default and the Choice of Maturity. Working Paper 2014-031. Federal Reserve Bank of St. Louis. doi:10.20955/wp.2014.031.

Su, C.L., Judd, K.L., 2012. Constrained optimization approaches to estimation of structural models. Econometrica 80, 2213–2230. doi:10.3982/ECTA7925.

Sutton, R.S., 1991. Dyna, an integrated architecture for learning, planning, and reacting. SIGART Bulletin 2, 160–163. doi:10.1145/122344.122377.

Sutton, R.S., Barto, A.G., 2018. Reinforcement Learning: An Introduction. 2 ed., MIT Press, Cambridge, MA.

Appendix A Proofs of Propositions 1 and 2

Appendix A.1 Proof of proposition 1 (gradient penetration)

We begin with the definition of the soft continuation value:

$$\mathcal{C}_\tau(s) = \tau \log \left[\exp\left(\frac{V^1(s)}{\tau}\right) + \exp\left(\frac{V^0(s)}{\tau}\right) \right]$$

To find the gradient with respect to $V^1(s)$, we apply the chain rule. First, taking the derivative of the outer logarithmic function $\log(u)$, where $u = \exp(V^1(s)/\tau) + \exp(V^0(s)/\tau)$:

$$\frac{\partial \mathcal{C}_\tau(s)}{\partial V^1(s)} = \tau \cdot \left(\frac{1}{\exp\left(\frac{V^1(s)}{\tau}\right) + \exp\left(\frac{V^0(s)}{\tau}\right)} \right) \cdot \frac{\partial}{\partial V^1(s)} \left[\exp\left(\frac{V^1(s)}{\tau}\right) + \exp\left(\frac{V^0(s)}{\tau}\right) \right]$$

Next, we evaluate the inner derivative. Because $V^0(s)$ is treated as a constant with respect to $V^1(s)$, its derivative is zero:

$$\frac{\partial \mathcal{C}_\tau(s)}{\partial V^1(s)} = \tau \cdot \left(\frac{1}{\exp\left(\frac{V^1(s)}{\tau}\right) + \exp\left(\frac{V^0(s)}{\tau}\right)} \right) \cdot \left(\frac{1}{\tau} \exp\left(\frac{V^1(s)}{\tau}\right) \right)$$

The temperature parameter τ cancels out:

$$\frac{\partial \mathcal{C}_\tau(s)}{\partial V^1(s)} = \frac{\exp\left(\frac{V^1(s)}{\tau}\right)}{\exp\left(\frac{V^1(s)}{\tau}\right) + \exp\left(\frac{V^0(s)}{\tau}\right)}$$

To map this to a standard sigmoid function, we divide the numerator and the denominator by $\exp(V^1(s)/\tau)$:

$$\frac{\partial \mathcal{C}_\tau(s)}{\partial V^1(s)} = \frac{1}{1 + \frac{\exp\left(\frac{V^0(s)}{\tau}\right)}{\exp\left(\frac{V^1(s)}{\tau}\right)}} = \frac{1}{1 + \exp\left(-\frac{V^1(s) - V^0(s)}{\tau}\right)} = p^1_\tau(s)$$

By symmetric application of the chain rule with respect to $V^0(s)$, we obtain:

$$\frac{\partial \mathcal{C}_\tau(s)}{\partial V^0(s)} = \frac{\exp\left(\frac{V^0(s)}{\tau}\right)}{\exp\left(\frac{V^1(s)}{\tau}\right) + \exp\left(\frac{V^0(s)}{\tau}\right)} = 1 - p_\tau^1(s)$$

Implication: Because the exponential function is strictly positive, for any finite values of $V^1(s)$, $V^0(s)$, and strictly positive temperature $\tau > 0$, the probabilities satisfy $0 < p_\tau^1(s) < 1$. The derivative weight for the losing branch is positive but can be arbitrarily close to zero when the value gap is large. The size of the optimizer step and convergence of the coupled neural system depend on additional features of the training procedure. ■

Appendix A.2 Proof of proposition 2 (hard-operator limit)

First, we establish the bounds on the soft continuation value. Let us define the maximum of the two branches as $M \equiv \max\{V^1(s), V^0(s)\}$. We can rewrite the argument of the log-sum-exp function by factoring out the dominant exponential $\exp(M/\tau)$:

$$\mathcal{C}_\tau(s) = \tau \log\left[\exp\left(\frac{M}{\tau}\right)\left(\exp\left(\frac{V^1(s) - M}{\tau}\right) + \exp\left(\frac{V^0(s) - M}{\tau}\right)\right)\right]$$

Using the property of logarithms $\log(a \cdot b) = \log(a) + \log(b)$:

$$\mathcal{C}_\tau(s) = \tau \log\left(\exp\left(\frac{M}{\tau}\right)\right) + \tau \log\left[\exp\left(\frac{V^1(s) - M}{\tau}\right) + \exp\left(\frac{V^0(s) - M}{\tau}\right)\right]$$

$$\mathcal{C}_\tau(s) = M + \tau \log\left[\exp\left(\frac{V^1(s) - M}{\tau}\right) + \exp\left(\frac{V^0(s) - M}{\tau}\right)\right]$$

By definition, one of the branches equals M, making its corresponding exponent zero (so $\exp(0) = 1$). The other branch must be less than or equal to M. Let $\Delta \leq 0$ denote the difference between the smaller branch and M. The equation simplifies to:

$$\mathcal{C}_\tau(s) = M + \tau \log\left(1 + \exp\left(\frac{\Delta}{\tau}\right)\right)$$

Because $\Delta \leq 0$ and $\tau > 0$, the exponential term satisfies $0 < \exp(\Delta/\tau) \leq 1$. Adding 1 to all sides gives:

$$1 < 1 + \exp\left(\frac{\Delta}{\tau}\right) \leq 2$$

Taking the logarithm (which is strictly increasing) and multiplying by τ preserves the inequalities:

$$0 < \tau \log\left(1 + \exp\left(\frac{\Delta}{\tau}\right)\right) \leq \tau \log 2$$

Substituting this back into our expression for $\mathcal{C}_\tau(s)$ yields the bounded envelope:

$$0 \leq \mathcal{C}_\tau(s) - \max\{V^1(s), V^0(s)\} \leq \tau \log 2$$

By the Squeeze Theorem, as the temperature $\tau \downarrow 0$, the term $\tau \log 2 \to 0$, forcing the soft continuation $\mathcal{C}_\tau(s)$ to converge uniformly to the hard maximum $\max\{V^1(s), V^0(s)\}$.

Second, we analyze the soft choice probability $p^1_\tau(s)$ as $\tau \downarrow 0$. Let $D \equiv V^1(s) - V^0(s)$ denote the advantage of branch 1. The probability is given by:

$$p^1_\tau(s) = \frac{1}{1 + \exp\left(-\frac{D}{\tau}\right)}$$

We evaluate the limit in three distinct cases:

- Strict Preference for Branch 1 ($D > 0$): As $\tau \downarrow 0$, the exponent $-D/\tau \to -\infty$. Consequently, $\exp(-\infty) \to 0$, and $p^1_\tau(s) \to \frac{1}{1+0} = 1$.
- Strict Preference for Branch 0 ($D < 0$): As $\tau \downarrow 0$, the exponent $-D/\tau \to +\infty$. Consequently, $\exp(+\infty) \to +\infty$, and $p^1_\tau(s) \to \frac{1}{1+\infty} = 0$.
- Indifference ($D = 0$): For any $\tau > 0$, $-D/\tau = 0$, meaning $\exp(0) = 1$. The probability remains exactly $p^1_\tau(s) = \frac{1}{1+1} = 0.5$.

Thus, $p^1_\tau(s)$ converges to $\mathbf{1}\{V^1(s) \geq V^0(s)\}$ at every state with unequal branch values. At a tie, $p^1_\tau(s) = 1/2$ for every positive temperature. The hard rule assigns equality to branch 1 by convention, separately from this probability limit. ■

Appendix B Modified economies and artificial smoothing paradigms

This appendix gives the recursive formulations of the two smoothing benchmarks used in Section 3.

Appendix B.1 Auxiliary income shocks

CE2012 adds a zero-mean i.i.d. transitory shock m with distribution F_m and density f_m, obtained by truncating and renormalizing a normal distribution with scale σ_{aux} on $[-\bar{m}, \bar{m}]$, where $\bar{m} = 2\sigma_{\text{aux}}$. The baseline smoothing scale is $\sigma_{\text{aux}} = 0.003$, with $\bar{m} = 0.006$. The shock enters repayment consumption; entry into default resets its current realization to the lower support point:

$$c_m^R(z, B, B') = z + m - \kappa B + q(z, B')[B' - (1-\lambda)B], \tag{B.1}$$

$$c_{-\bar{m}}^D(z) = h(z) - \bar{m}. \tag{B.2}$$

Let $\bar{V}(z, B) = \int V(z, B, m)\, dF_m(m)$ and $\bar{X}(z) = \int X(z, m)\, dF_m(m)$ denote values before the future transitory shock is realized. The repayment value and exclusion value satisfy

$$V^R(z, B, m) = \max_{\substack{B' \in [0, B_{\max}] \\ c_m^R(z,B,B') > 0}} \{u(c_m^R(z, B, B')) + \beta E_{z'|z} \bar{V}(z', B')\}, \tag{B.3}$$

$$X(z, m) = u(h(z) + m) + \beta E_{z'|z}[\chi \bar{V}(z', 0) + (1-\chi)\bar{X}(z')]. \tag{B.4}$$

Exclusion continuation integrates future shocks, whereas the value of entering default is $X(z, -\bar{m})$, independent of the current repayment shock. Thus

$$V(z, B, m) = \max\{V^R(z, B, m), X(z, -\bar{m})\}, \tag{B.5}$$

$$\delta(z, B, m) = \mathbf{1}\{X(z, -\bar{m}) > V^R(z, B, m)\}, \tag{B.6}$$

$$\bar{\delta}(z, B) = \int \delta(z, B, m)\, dF_m(m). \tag{B.7}$$

Lenders integrate the realized hard decision and the repayment-conditional debt policy $A(z, B, m)$ over the auxiliary shock:

$$\begin{aligned} q(z, B') = \frac{1}{1 + r_f} \sum_{z'} P_z(z, z') \int [1 - \delta(z', B', m')] \\ \times [\lambda + (1-\lambda)\{\text{coup} + q(z', A(z', B', m'))\}]\, dF_m(m'). \end{aligned} \tag{B.8}$$

Equality of branch values is assigned to repayment. The sensitivity exercise scales σ_{aux} and the support bound together.

The benchmark evaluates these integrals using 11 equal-width shock intervals. Each interval receives its normalized normal-CDF probability mass; within an interval, regions assigned to default or a borrowing choice receive weights proportional to their lengths. Repayment and continuing-exclusion utilities use interval midpoints; default entry retains the lower-support reset defined above. This discretization is held fixed across the sensitivity exercise.

Appendix B.2 Taste-shock smoothing

Nested extreme-value taste shocks smooth the borrowing and default decisions at scales ρ_B and ρ_D, respectively. For feasible choices $B' \in G_B$, the repayment action value is

$$Q^R(z, B, B') = u(c^R(z, B, B')) + \beta \sum_{z'} P_z(z, z')V(z', B'), \tag{B.9}$$

where $c^R(z, B, B') = z - \kappa B + q(z, B')[B' - (1 - \lambda)B]$. Infeasible choices receive zero choice probability and are excluded from the sums below.

$$V^R_{\rho_B}(z, B) = \rho_B \log \sum_{B' \in G_B^{\text{feas}}} \exp\{Q^R(z, B, B')/\rho_B\}, \tag{B.10}$$

$$\pi_{\rho_B}(B'|z, B) = \frac{\exp\{Q^R(z, B, B')/\rho_B\}}{\sum_{\widetilde{B} \in G_B^{\text{feas}}} \exp\{Q^R(z, B, \widetilde{B})/\rho_B\}}. \tag{B.11}$$

The exclusion recursion is

$$V^D(z) = u(h(z)) + \beta \sum_{z'} P_z(z, z')[\chi V(z', 0) + (1 - \chi)V^D(z')]. \tag{B.12}$$

At the upper decision layer, integrating over default taste shocks gives

$$V(z, B) = \rho_D \log[\exp\{V^R_{\rho_B}(z, B)/\rho_D\} + \exp\{V^D(z)/\rho_D\}], \tag{B.13}$$

$$P_{\rho_D}(\text{default}|z, B) = \frac{\exp\{V^D(z)/\rho_D\}}{\exp\{V^R_{\rho_B}(z, B)/\rho_D\} + \exp\{V^D(z)/\rho_D\}}. \tag{B.14}$$

Lender pricing averages over both decision layers:

$$\begin{aligned} q(z, B') = \frac{1}{1 + r_f} \sum_{z'} P_z(z, z')[1 - P_{\rho_D}(\text{default}|z', B')] \\ \times \left[\lambda + (1 - \lambda) \left\{ \text{coup} + \sum_{B'' \in G_B^{\text{feas}}} \pi_{\rho_B}(B''|z', B')q(z', B'') \right\}\right]. \end{aligned} \tag{B.15}$$

The common additive extreme-value constant is normalized to zero. The sensitivity exercise scales both temperatures together from $(\rho_D, \rho_B) = (5 \times 10^{-4}, 10^{-5})$, preserving $\rho_D/\rho_B = 50$.

Appendix C Solving the one-period sovereign default model

As an initial test exercise, we evaluate the StH framework in a canonical one-period sovereign default model (Arellano, 2008). This model features a hard default boundary without long-term bond resale dynamics or debt dilution, thereby isolating boundary optimization from multi-period pricing feedback.

Appendix C.1 The one-period sovereign default model

The one-period sovereign default model is formulated over a net-asset domain $a \in [-0.20, 0.40]$, where negative values denote debt. The endowment process y is i.i.d., and continuation expectations are evaluated using an 11-node Gauss–Hermite quadrature rule. The repayment budget constraint is given by

$$c^R(y, a, a') = y + a - q(a')a', \tag{C.1}$$

where a' denotes next-period net assets and $q(a')$ is the bond price schedule.

Let β_b be the borrower discount factor and β_c the lender discount factor. The repayment value $V^R(y, a)$ and total value $V(y, a)$ satisfy

$$V^R(y, a) = \max_{a' \in [-0.20, 0.40]} \left\{ u(c^R(y, a, a')) + \beta_b E_{y'} V(y', a') \right\}, \tag{C.2}$$

$$V(y, a) = \max\{V^R(y, a), V^D(y)\}. \tag{C.3}$$

The repayment maximization is restricted to choices with $c^R(y, a, a') > 0$.

Default resets assets to zero and imposes a one-period output cost governed by $h(y) = y_{min} + \lambda_D(y - y_{min})$, with $\lambda_D = 0.8$ and no market reentry. Normal autarky output resumes in subsequent periods:

$$W(y) = u(y) + \beta_b E_{y'} W(y'), \tag{C.4}$$

$$V^D(y) = u(h(y)) + \beta_b E_{y'} W(y'). \tag{C.5}$$

The one-period bond price schedule is determined by risk-neutral lenders:

$$q(a') = \beta_c \Pr\left\{ V^R(y', a') \geq V^D(y') \right\}. \tag{C.6}$$

Following Arellano (2008), we calibrate the model using a borrower discount factor of $\beta_b = 0.90$, a lender discount factor of $\beta_c = 0.96$, a CRRA utility coefficient of 2, and a log-endowment standard deviation of 0.05.

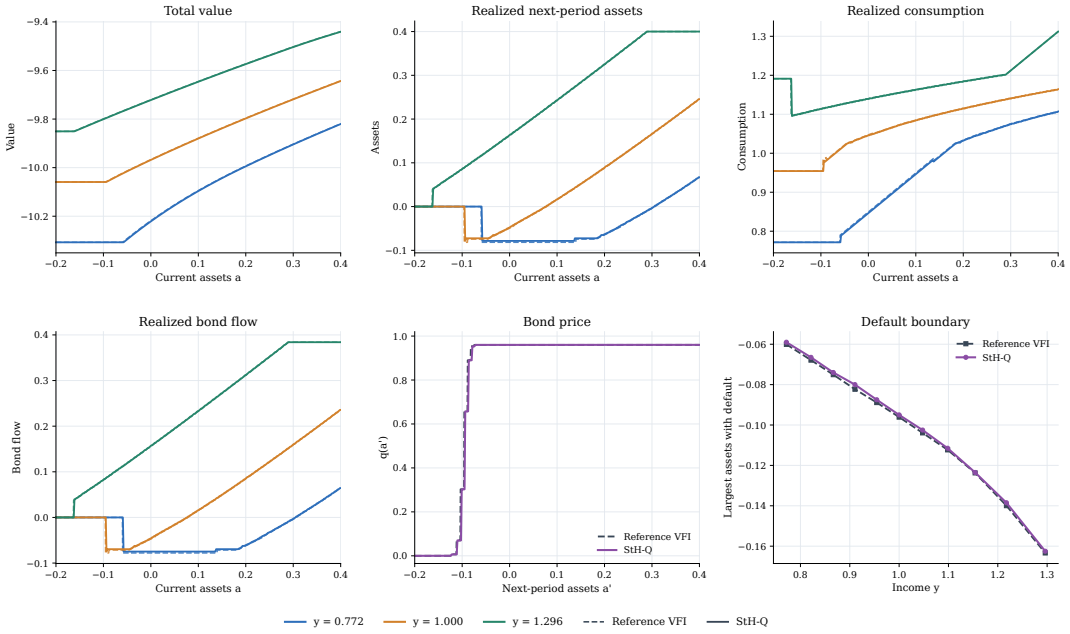
To ensure a rigorous evaluation, the Value Function Iteration (VFI) reference is tailored to the specific StH implementation being tested. For the tabular StH-Q comparison (Figure C.1), the VFI reference computes lender repayment probabilities via a discrete weighted summation of hard repayment indicators across the 11 income nodes. Conversely, for the neural StH-AC comparison (Figure C.2), the VFI reference calculates these probabilities continuously using the normal cumulative distribution function evaluated at the interpolated default cutoff. Across all evaluations, the StH framework consistently employs an 11-node Gauss–Hermite quadrature.

Appendix C.2 StH-Q in the one-period sovereign default model

StH-Q uses the repayment budget, permanent exclusion after default, and primitive parameters specified above. The table has 11 income states and 401 points for both current and next-period assets. Temporary soft model-based iterations initialize the table; the main block combines simulated experience, expected TD updates, and planning, then continues under hard targets. Averaged late snapshots initialize a fixed-price hard TD finish. VFI supplies the comparison solution.

The run uses 12,000 main updates and 4,000 finishing updates. On the Q grid, the maximum price-equation residual is 7.77×10^{-15}, the maximum repayment Bellman residual is 6.39×10^{-14}, and the maximum feasible policy gap is zero. The VFI reference uses 1,000 asset points. Figure C.1 compares the resulting functions on their respective grids.

Fig. C.1: StH-Q solution for the one-period sovereign default model.



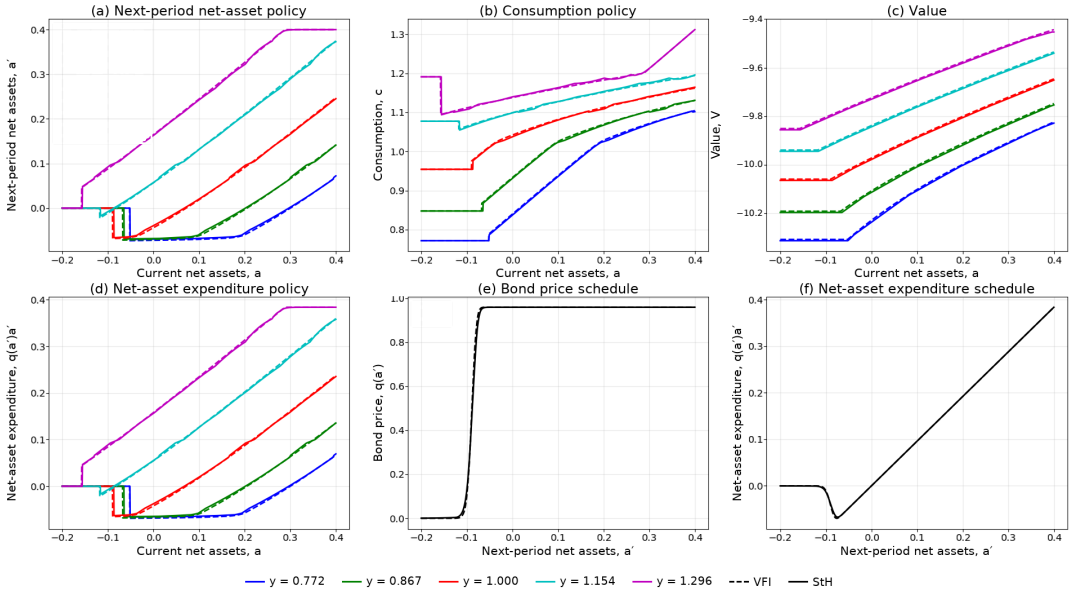
Note: For comparison, an accurate VFI reference solution is also shown.

Appendix C.3 StH-AC in the one-period sovereign default model

To rigorously evaluate the neural StH-AC solver in the one-period sovereign default model, we conduct two experiments: verifying the exact converged solution and testing the algorithm's robustness to severe initialization errors.

The converged solution Figure C.2 presents the solution of the one-period sovereign default model, expressed in net-asset notation. Solid curves represent the StH-AC hard-evaluation outcomes, while dashed curves denote the high-precision grid-based VFI reference. The exact visual overlap provides a direct numerical validation of the StH-AC solver's accuracy at convergence.

Fig. C.2: StH-AC solution for the one-period sovereign default model.



Note: For comparison, an accurate VFI reference solution is also shown.

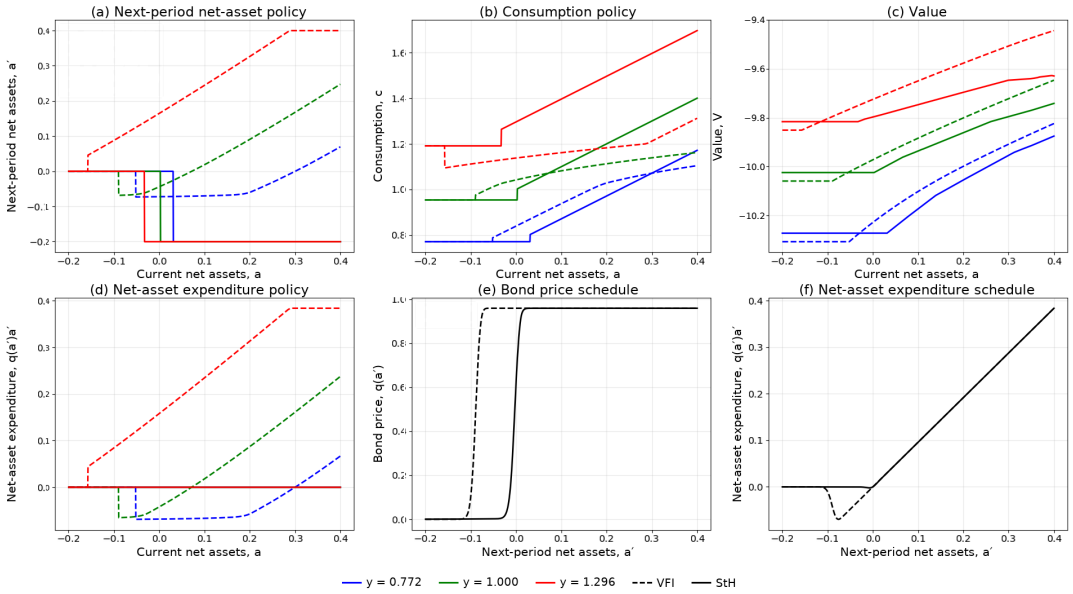
Robustness to poor initialization A critical test of any deep reinforcement learning solver is its resilience to poor initialization. We therefore investigate whether StH-AC's gradient penetration mechanism can recover the true equilibrium when neural network parameters are initialized far from the VFI target region.

Under these distorted initial conditions, the solver first enters a distinct stagnation phase. As illustrated in Figure C.3, the asset policy degenerates into a flat, state-unresponsive schedule, causing the value and consumption functions to deviate severely from the VFI reference.

Under a traditional hard-switch formulation, the network would remain permanently trapped in this zero-gradient dead zone. In StH-AC, however, the soft continuation values continuously transmit weighted future-value derivatives to the actor network, while soft repayment probabilities smooth the detached pricing targets. Driven by these persistent, non-zero gradient updates, the network successfully escapes stagnation. Figure C.4 illustrates this recovery stage, demonstrating how the gradient penetration mechanism systematically guides the neural approximations back to the exact VFI benchmark.

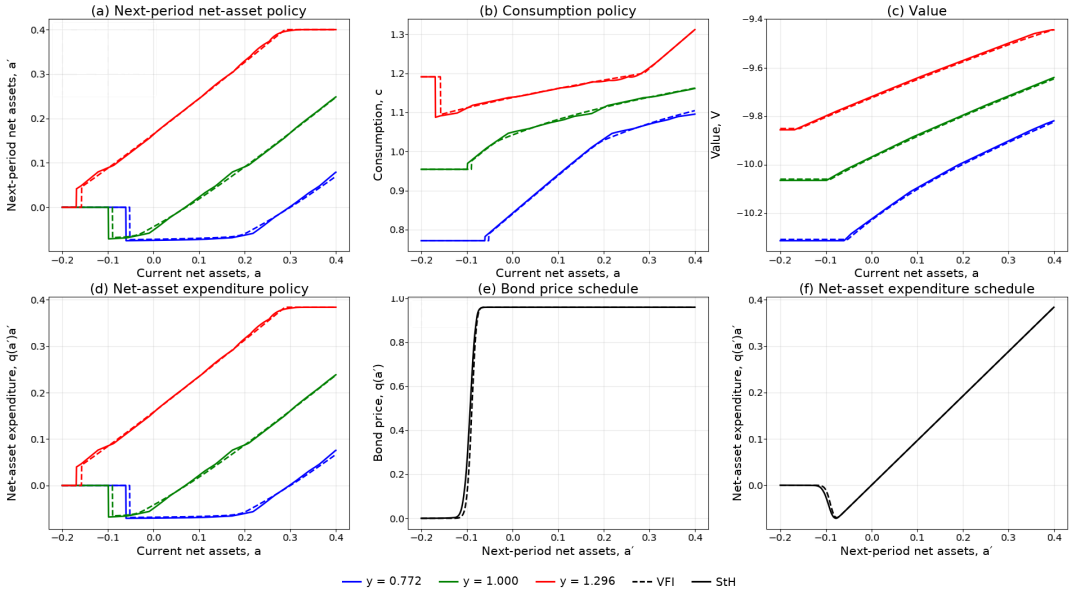
We conclude not only that StH-AC isolates the exact hard default boundary under standard conditions, but also that its soft-to-hard architecture provides critical numerical robustness against severe parameter distortions.

Fig. C.3: StH-AC stagnation under poor initialization.



Note: For comparison, an accurate VFI reference solution is also shown.

Fig. C.4: StH-AC recovery from poor initialization.



Note: For comparison, an accurate VFI reference solution is also shown.

Appendix D Implementation details and benchmark training diagnostics

Appendix D.1 Hardware and timing conventions

The execution times reported in Table 1 reflect implementation-specific hardware setups and timing conventions. The auxiliary-shock baseline runs on CPU using 8 MPI ranks, while the taste-shock baseline executes across 16 CPU threads. The StH solvers leverage GPU acceleration: StH-Q is implemented in PyTorch on an NVIDIA RTX 3080 GPU, and StH-AC is implemented in JAX on an NVIDIA RTX 5090 GPU. For StH-Q, the reported runtime encompasses soft initialization, the primary learning phase, and the fixed-price temporal difference (TD) finish. For StH-AC, total training time sums the pretraining, soft-learning, and hard-refinement stages; this total excludes non-training overheads such as JIT compilation, checkpointing, data export, and evaluation. Timings for the auxiliary- and taste-shock benchmarks reflect their native solver runtimes.

The StH-AC (5×) run in Table 1 uses 5 million soft updates and 500,000 hard updates; pretraining and the annealing schedule are unchanged.

Appendix D.2 StH-Q configuration and full update sequence

The persistent repayment Q table contains $200 \times 350 \times 350$ entries in float64 arithmetic. Initialization starts from model primitives and a fixed rollover policy, followed by four full-action soft stages with $(\tau_D, \tau_B) = (0.03, 0.003)$, $(0.02, 0.002)$, $(0.01, 0.001)$, $(0.003, 0.0003)$. These stages initialize borrower values and prices for main learning. Unselected Q entries retain their values; after a row update, its feasible maximum and policy are refreshed. These initialization costs are included in the reported solution time.

Main training uses 1,024 simulated environments and a TD batch of 16,384 observations, half from valid repayment experience and half from randomly drawn states with exploratory actions. The learning rates are $\alpha_Q = 1$ and $\alpha_D = 0.5$. The price update has rate 0.002 and right-multiplies its residual by $(I + 16 D_B^\top D_B)^{-1}$, where D_B is the first-difference matrix along the stored debt indices. The preconditioner scales the pricing updates; residuals are evaluated using the original pricing equation. Each iteration also updates all actions at up to 32 randomly selected states. The default temperature falls from 0.003 to 0.0003 over 1,000 iterations and is then set to zero.

The 225 snapshots collected every 40 iterations over the final 9,000 iterations of the 26,000-iteration main run are averaged. Main exploration declines from 0.3 to 0.1 over the first 14,400 iterations and then remains at 0.1. The final 1,000 TD

iterations freeze price, disable planning, and use exploration probability 0.4, with most exploratory actions near the current greedy action. They update the actual Q table and default values. The final export takes the feasible argmax of stored Q entries. The finishing stage refines borrower values conditional on the averaged price.

Algorithm D.1 Tabular StH-Q with simulated experience

1: Initialize Q^R, V^D, q from the model's temporary soft solution.
2: **for** $n = 1, \ldots, 26{,}000$ **do**
3: Simulate repayment, default, exclusion, and reentry with ϵ-greedy debt choices.
4: Draw a minibatch combining valid repayment experience and random states.
5: Update sampled Q entries using Equation (29); refresh their greedy policies.
6: Update default values and lender prices; back up all actions at 32 sampled states.
7: Use $\tau > 0$ for the first 1,000 iterations and $\tau = 0$ thereafter.
8: Accumulate tables every 40 iterations after iteration 17,000.
9: **end for**
10: Average the accumulated tables and hold the averaged price fixed.
11: Perform 1,000 hard TD iterations without planning; export the stored-Q greedy policy.
12: Evaluate borrower and lender equations and simulate economic outcomes separately.

Appendix D.3 StH-AC configuration and full update sequence

The following settings describe the StH-AC configuration.

Training dynamics. We use Adam with batch size 256. Soft-phase learning rates are 3×10^{-4} for both critics and the price network and 3×10^{-5} for the actor. The hard phase multiplies these rates by 0.02. Gradients are clipped to maximum norm 10. Target networks use Polyak coefficients $\alpha = 0.0005$ in the soft phase and $\alpha = 0.001$ in the hard phase, under the convention $\vartheta^- \leftarrow (1-\alpha)\vartheta^- + \alpha\vartheta$. The feasibility penalty is $\lambda_{feas} = 10^4$ with consumption floor $\underline{c} = 10^{-6}$. When the floor is inactive and the penalty is zero, the actor uses the return derivative in Equation (19); otherwise, the update also differentiates the safe payoff and penalty. General state batches allocate 25% of draws to small repay/default value gaps; 20% of price batches target states with next-period default probability between 0.05 and 0.95.

Annealing schedule and hard refinement Temperature decay follows the geometric schedule in Equation (22) over $T_{\text{ann}} = 300{,}000$ iterations, decaying from $\tau_{D,\text{init}} = 0.05$ to $\tau_{D,\text{final}} = 0.002$. The temperature remains fixed at 0.002 for the remainder of the 1,000,000-update soft-training phase.

Following soft training, the solver enters a 100,000-update hard refinement stage (steps 1,000,001 through 1,100,000) with $\tau = 0$. This stage strictly enforces the exact hard operators from Equation (5), fully reconciling the actor x_ω, critics (V_θ^1, V_ψ^0), and equilibrium network Γ_ϕ under the unsmoothed economic environment. The actor remains fully flexible throughout this block, with no initial parameter freezing or anchor penalties. Final model evaluation and diagnostic reporting are performed at the step 1,100,000 endpoint.

Calibration, solution domain, and expectations The income process specified in Section 3 is discretized into $N_z = 200$ points spanning ± 3 stationary standard deviations of log income. All expectations over future z' are computed exactly via matrix-vector multiplication with the transition matrix P_z. Debt states B are sampled continuously from $[0, B_{max}]$, with $B_{max} = 1.05$, during both the soft-training and final hard-refinement stages.

Algorithm D.2 summarizes the full solution procedure. The pseudocode distinguishes the two training stages—soft initial training and hard refinement—from the separate final hard evaluation.

Algorithm D.2 Soft-to-hard actor–critic solver for sovereign default

Require: Model primitives, networks $\{B'_\omega, V^R_\theta, V^D_\psi, q_\phi\}$, soft budget $T_S = 10^6$, hard budget $T_H = 10^5$

Ensure: Hard-evaluation policies, prices, residuals, and simulated moments

Initialize online networks and target copies.
Fit the actor to $B' = (1-\lambda)B$ for 10,000 initialization updates at $\tau_D = 0.05$.
Hold the actor fixed and pretrain V^R, V^D, and q for 40,000 further updates.
Synchronize target networks, clear Adam states, and reset the main training key.
for $t = 1, \ldots, T_S$ **do**
    Draw a state minibatch with boundary-focused observations
    Set $\tau_D \leftarrow \tau_D(t)$, annealed over the first 3×10^5 updates
    if $(t-1) \bmod 5000 < 1000$ **then**
        Update B'_ω through online q_ϕ and the online soft next-period continuation, with the feasibility penalty
    else
        Form detached target-network soft Bellman and lender-pricing targets
        Update V^R_θ, V^D_ψ, and q_ϕ
    end if
    Clip active gradients and Polyak-update target networks
end for
Replace the soft operators used in the actor and regression targets by their exact hard counterparts
Multiply all learning rates by 0.02
for $k = 1, \ldots, T_H$ **do**
    Draw a state minibatch with boundary-focused observations
    if $(k-1) \bmod 5000 < 1000$ **then**
        Update the free actor through online q_ϕ and the online hard next-period upper envelope
    else
        Form detached target-network hard Bellman and indicator-based pricing targets
        Update V^R_θ, V^D_ψ, and q_ϕ
    end if
    Clip active gradients and Polyak-update target networks
end for
Evaluate once under the exact repay/default rule at step $T_S + T_H$

Appendix D.4 Simulation protocol

For Tables 2 and 4, we simulate 256 paths of 5,000 quarters each and discard the first 500 quarters of each path. Market-access moments exclude default and exclusion quarters and require more than 20 consecutive quarters of market access. Default frequencies use observations in which default can be chosen, subject to the same 20-quarter filter. Path-level moments are averaged across paths; the two relative-volatility measures in Table 2 are ratios of the averaged standard deviations.

Appendix E Common hard-operator audit

This appendix defines the common hard-operator audit introduced in Section 5.1.

Appendix E.1 Audit protocols and operators

All solutions are mapped to the common grid $G = G_z \times G_B$, with 200 income nodes and 350 debt nodes. For each solver m, let V_m, q_m, A_m, and $\delta_m \in \{0, 1\}$ denote the mapped continuation value, price schedule, repayment-conditional continuous debt policy, and exported binary default rule (1 for default, 0 for repayment). The continuation value retains the inclusive value for the Taste benchmark; it is not replaced by the hard maximum prior to the audit. Feasible audit choices must satisfy positive repayment consumption:

$$c^R(s, B'; q_m) = z - \kappa B + q_m(z, B')[B' - (1 - \lambda)B] \tag{E.1}$$

$$G_B^{feas}(s; q_m) = \{B' \in G_B : c^R(s, B'; q_m) > 0\} \tag{E.2}$$

To assess solution fidelity against the exact hard target economy, we first apply the unsmoothed repayment and default operators to the mapped solutions:

$$Q_m^H(s, B') = u(c^R(s, B'; q_m)) + \beta \sum_{z'} P_z(z, z') V_m(z', B'), \qquad s = (z, B). \tag{E.3}$$

$$(T_R^H V_m)(s) = \max_{B' \in G_B^{feas}(s; q_m)} Q_m^H(s, B') \tag{E.4}$$

$$(T_D^H V_m)(z) = u(h(z)) + \beta \sum_{z'} P_z(z, z')[\chi V_m(z', 0) + (1 - \chi) V_m^D(z')] \tag{E.5}$$

The lender pricing operator uses each method's exported decision and continuous policy:

$$\begin{aligned}(\mathcal{T}_q^H q_m)(z, B') = \frac{1}{1 + r_f} \sum_{z'} P_z(z, z')[1 - \delta_m(z', B')] \\ \times [\lambda + (1 - \lambda)\{\text{coup} + q_m(z', A_m(z', B'))\}].\end{aligned} \tag{E.6}$$

Appendix E.2 Residuals and error diagnostics

We define the recomputed hard choice and repayment set, assigning ties to repayment:

$$\tilde{\delta}_m(s) = \mathbf{1}\{(T_D^H V_m)(s) > (T_R^H V_m)(s)\}, \quad \tilde{G}_m^R = \{s \in G : \tilde{\delta}_m(s) = 0\} \tag{E.7}$$

Using these exact operators, we compute four diagnostic statistics for each method:

1. *Borrower Bellman RMSE (R_m^V):* Measures the root-mean-square residual of the hard upper-envelope Bellman equation across all states in G:

$$R_m^V = \left[\frac{1}{|G|}\sum_{s\in G}\left(V_m(s) - \max\{(T_R^H V_m)(s), (T_D^H V_m)(s)\}\right)^2\right]^{1/2} \tag{E.8}$$

2. *Repayment policy optimality RMS (R_m^π):* Quantifies policy inefficiency on the recomputed repayment set $\tilde{G}_m^R$. It evaluates the non-negative value shortfall relative to the optimal feasible grid choice:

$$\rho_m^{pol}(s) = \left[\max_{B' \in G_B^{feas}(s;q_m)} Q_m^H(s, B') - Q_m^H(s, A_m(s))\right]_+, \quad [x]_+ \equiv \max\{x, 0\} \tag{E.9}$$

$$R_m^\pi = \left[\frac{1}{|\tilde{G}_m^R|}\sum_{s\in\tilde{G}_m^R}(\rho_m^{pol}(s))^2\right]^{1/2} \tag{E.10}$$

   Because a continuous exported action can lie between audit-grid choices, a negative untruncated difference is recorded as zero.

3. *Pricing RMSE (R_m^q)*: Evaluates the root-mean-square error of the lender risk-neutral pricing schedule under the exported policy and default rule:

$$R_m^q = \left[\frac{1}{|G|}\sum_{s\in G}(q_m(s) - (\mathcal{T}_q^H q_m)(s))^2\right]^{1/2} \tag{E.11}$$

4. *Default-rule inconsistency (D_m):* Measures the percentage of states where the exported binary rule $\delta_m(s)$ differs from the updated implied decision $\tilde{\delta}_m(s)$ obtained after one exact hard Bellman update (counting both directions of disagreement):

$$D_m = \frac{100}{|G|}\sum_{s\in G}\mathbf{1}\{\delta_m(s) \neq \tilde{\delta}_m(s)\} \tag{E.12}$$

Equal weights apply within each audit set. These quantities retain the units of their statewise residuals, except D_m, which is a percentage.

Appendix E.3 Mapping conventions across smoothed models

To audit the smoothed benchmarks against the exact hard target economy, each solver's output is mapped to the common audit grid using method-specific rules:

- *Auxiliary Income Shocks (Aux):* The hard reference update is evaluated at zero current auxiliary shock ($m = 0$). The continuous default probability exported by the Aux solver is discretized into a binary decision rule by thresholding at 0.5 (i.e., $\delta_{\text{Aux}} = 1$ if $P_{\text{default}} \geq 0.5$).
- *Taste-Shock Smoothing (Taste):* The mapped value function retains the model's inclusive value (the expected value integrated over taste shocks). Its binary default rule is constructed by directly comparing the underlying repayment and default branch values ($\delta_{\text{Taste}} = 1$ if $V^D > V^R$).
- *Off-Grid Policy Evaluation:* $A_m(s)$ represents the continuous debt choice exported on repayment states. Whenever an off-grid continuous choice $A_m(s)$ falls between discrete grid points, continuation returns Q_m^H and bond prices q_m are evaluated using standard linear interpolation.

The common audit evaluates solution fidelity by applying the exact hard Bellman and lender operators in a single pass to these mapped outputs. This isolates structural approximation errors relative to the true unsmoothed economy, whereas each method's native stopping diagnostics (which measure convergence to its own smoothed problem) are evaluated separately.

Appendix F Implementation details of the endogenous capital model

We analyze the long-term debt benchmark model of Gordon and Guerrón-Quintana (2018), omitting the sudden-stop extension.

Appendix F.1 Model and Calibration

Let A denote productivity, $B \geq 0$ inherited debt, and K inherited capital. Period utility is

$$u(c, L) = \frac{(c - \eta L^{\omega}/\omega)^{1-\gamma}}{1-\gamma}, \quad c - \eta L^{\omega}/\omega > 0. \tag{F.1}$$

Production in repayment and default is $Y_j = A_j K^\alpha L_j^{1-\alpha}$, where $A_R = A$ and $A_D = [1-\kappa(A)]A$. Labor is determined statically:

$$L_j = \left[\frac{(1-\alpha)A_j K^\alpha}{\eta}\right]^{\frac{1}{\alpha+\omega-1}}, \quad j \in \{R, D\}. \tag{F.2}$$

Capital adjustment costs are $\Psi(K', K) = \frac{\Theta}{2}(K' - K)^2$, and investment is $I = K' - (1-\delta_K)K$. We write capital depreciation as δ_K to distinguish it from the default indicator used elsewhere in the paper.

With χ denoting the re-entry probability, the value functions satisfy:

$$V(A, B, K) = \max\left\{V^R(A, B, K), V^D(A, K)\right\}, \tag{F.3}$$

$$V^R(A, B, K) = \max_{B', K'} \left\{u(c_R, L_R) + \beta \mathbb{E}\left[V(A', B', K') \mid A\right]\right\}, \tag{F.4}$$

$$V^D(A, K) = \max_{K'_D} \left\{u(c_D, L_D) + \beta \mathbb{E}\left[(1-\chi)V^D(A', K'_D) + \chi V(A', 0, K'_D) \mid A\right]\right\}. \tag{F.5}$$

Default eliminates debt and excludes the sovereign from credit markets, but does not eliminate its capital stock. The default branch therefore retains an investment decision, and market re-entry occurs with zero debt. The budget constraints are:

$$c_R = Y_R + (1-\delta_K)K - K' - \Psi(K', K) - [\lambda + (1-\lambda)\text{coup}]B + q(A, B', K')[B' - (1-\lambda)B], \tag{F.6}$$

$$c_D = Y_D + (1-\delta_K)K - K'_D - \Psi(K'_D, K). \tag{F.7}$$

Choices satisfy $B' \geq 0$, $K', K'_D > 0$, and the effective consumption restriction in (F.1).

Risk-neutral foreign lenders discount future cash flows at rate r_f. Let $d(A, B, K) = \mathbf{1}\{V^D(A, K) > V^R(A, B, K)\}$, where ties break in favor of repayment. With (B'', K'') denoting next period's repayment choices, the bond price function is:

$$q(A, B', K') = \frac{\mathbb{E}\left[(1 - d(A', B', K'))\left\{\lambda + (1-\lambda)\left[\text{coup} + q(A', B'', K'')\right]\right\} \mid A\right]}{1 + r_f}. \tag{F.8}$$

Capital thus enters both future repayment incentives and the continuation value of outstanding sovereign debt.

We retain the effective calibration and normalization of the public replication code available at https://ideas.repec.org/c/red/ccodes/14-216.html, summarized in Table F.1. The default output-loss function in that code is:

$$\kappa(A) = \text{clip}\left(0.094720943 + 0.35596918\left(\frac{A}{\bar{A}} - 1\right), 0, 0.96\right), \quad \bar{A} = 0.5420840681. \tag{F.9}$$

The coefficients apply to the normalized productivity deviation $A/\bar{A}-1$. The code's steady-state investment-to-output normalization is 0.049, which implies the capital depreciation rate reported below. Expectations use the original 88-state productivity grid and transition probability matrix.

Table F.1: Calibration of the endogenous-capital application.

Parameter	Value	Parameter	Value
Discount factor β	0.94628699	Risk aversion γ	2
Capital share α	0.36	Labor weight η	0.64
Labor exponent ω	1.6360566	Adjustment cost Θ	21.159856
Depreciation δ_K	0.008943187	Maturity λ	0.05
Coupon coup	0.03	Risk-free rate r_f	0.01
Re-entry probability χ	0.10	Persistence ρ_A	0.95
Innovation s.d. σ_A	0.016913608	Debt upper bound	3
Capital lower bound	2.191612452	Capital upper bound	10.958062261

Note: One period is one quarter. Values follow the effective replication code calibration. The productivity process is retained through its original 88-state transition matrix.

Equations (F.1)–(F.8) describe the $m = 0$ economy used by StH-AC. The numerical reference grid solutions retain the authors' auxiliary income shock $m \sim \mathcal{N}(0, 0.003^2)$, truncated to $[-0.006, 0.006]$. This shock is added to the repayment budget, and future realizations are integrated in continuation values and bond pricing; the default budget contains no such shock.

Appendix F.2 StH-AC configuration and training

We follow the StH-AC framework established in Section 4.2 and Appendix D.3. The endogenous-capital implementation utilizes repayment and default actors, repayment and default critics, and a bond-price network. All neural networks feature two hidden layers of 256 units per layer. All five networks use ReLU activations.

Optimization is performed using Adam with a batch size of 512. The training process comprises 50,000 pretraining updates based on a fixed policy, followed by 1.3 million soft updates and 50,000 hard updates. Within each 5,000-update cycle, 1,000 updates are allocated to the actors and 4,000 to the critics and price network. The exploration temperature decays exponentially from 0.2 to 0.002 over the first 500,000 soft updates, remains at 0.002 for the remainder of the soft stage, and is strictly zero during hard refinement. Initial learning rates for the actors and the value/price networks are set to 10^{-5} and 10^{-4}, respectively, in the soft stage, and 1.6×10^{-6} and 1.2×10^{-5} in the hard stage. Both decay exponentially within their respective stages.

Appendix F.3 Numerical benchmarks, simulation details, and timing

The baseline reference model employs a grid with 50 nodes per endogenous state and 250 per control. Dense 100 uses 100 nodes per endogenous state, 500 debt choices, and 2,500 capital choices; Dense 150 uses 150 nodes per endogenous state, 1,000 debt choices, and 3,000 capital choices, maintaining the same calibration and productivity process. Both dense-grid runs reach the original authors' stopping criteria after 188 equilibrium iterations. Repayment policy cross-sections fix capital at $K = 5.41194$ for all methods, with Dense 150 linearly interpolated between adjacent capital nodes. Figure 5 uses $A = 0.5407$ throughout. Its $m = 0$ default boundaries are interpolated from each solution's evaluation grid.

All reported moments are computed from 1,000 simulated paths of 3,000 quarters, discarding the first 1,500 quarters as a burn-in period. The baseline model uses the common-draw sample, whereas the Dense 100 and Dense 150 simulations use independent random draws. Output, consumption, and investment volatilities are computed by applying the HP filter (smoothing parameter 1600) to each path's log series, calculating the sample standard deviation, and averaging across paths. Relative consumption and investment volatilities are expressed as the ratio of their respective mean standard deviations to output volatility. Debt-to-output and spread moments exclude periods of default, market exclusion, and the 20 quarters immediately following market re-entry. Quarterly yields are compounded to calculate annualized spreads:

$$\text{Spread}(q) = 100 \left[\left(1 - \lambda + \frac{\lambda + (1-\lambda)\text{coup}}{q} \right)^4 - (1 + r_f)^4 \right]. \tag{F.10}$$

Event profiles are conditional averages centered on each solution's default entry. Except for spreads, the plotted series are HP1600-filtered deviations multiplied by 100: net exports/output is filtered in levels, and other series in logs. Labor and Solow-residual profiles use trailing averages of up to four quarters, using available observations at the left endpoint and then sampling every fourth quarter. Spreads are annualized and shown before default.

For 50 CPU threads, the coarse grid, Dense 100, and Dense 150 require 243.2, 27,479.0, and 155,709.7 seconds, respectively. StH-AC training takes 3,379.8 seconds on one NVIDIA RTX 5090 GPU.